

FIRMS-Explanation of Target Commands Used in build.xml

Community Grids Lab, Indiana University

1. Introduction:

This document provides a description of each of the commands used in the build.xml file. Also provided is a build.properties file where users can initialize all the arguments for the various ANT target commands. Users can, if they so desire, also override the modified default parameters (contained in the build.properties) while running the individual target commands.

Listed below is the list of variables in the build.properties file. For each such variable we have include a description of the information encapsulated within these variables and their current default values. It is expected that the users will modify the values associated with these properties to suit their execution environment.

omii.catalina.home = Holds the Tomcat home directory. In this case it is Tomcat within the OMII container.

omii.catalina.home=/usr/local/OMII/jakarta-tomcat-5.0.25

tomcat.home = Holds the Tomcat home directory. In this case it is Tomcat within the OMII container. This variable is same as Catalina home variable but defined exclusively for Tomcat to execute Start and Stop commands from terminal.

tomcat.home =/usr/local/OMII/jakarta-tomcat-5.0.25/

firms.home = Holds the FIRMS root directory. In this case it is tomcat within the OMII container
firms.home=.

tomcat.port = Holds the Tomcat port number within OMII Container.

tomcat.port = 18080

#WSR

wsr.source.hostname = Holds the value of Source host name. You have to change the value to your local machine. This variable is used while running WSR on the Filter Pipeline.

wsr.source.hostname= localhost

wsr.source.port = Holds the TCP port number for the Source running on Filter Pipeline.

wsr.source.port= 8000

wsr.source.cofig = Holds the Source Configuration file name.

wsr.source.config = config/wsr/WsrSourceConfig.properties

wsr.sink.hostname = Holds the value of Sink host name. You have to change the value to your local machine. This variable is used while running WSR on the Filter Pipeline.

wsr.sink.hostname=localhost

wsr.sink.port = Holds the Sink TCP port number running on the Filter Pipeline.

wsr.sink.port=9000

wsr.sink.config =Holds the Sink configuration file name.

wsr.sink.config = config/wsr/WsrSinkConfig.properties

#WSRM

wsrc.source.hostname – Holds the value of Source host name. You have to change the value to your local machine. This variable is used while running WSRM on the Filter Pipeline.

wsrc.source.hostname= localhost

wsrc.source.port – Holds the TCP port number for the Source running on Filter Pipeline.

wsrc.source.port= 8000

wsrc.source.config – Holds the Source Configuration file name.

wsrc.source.config = config/wsrc/wsrc1.txt

wsrc.sink.hostname –Holds the value of Sink host name. You have to change the value to your local machine. This variable is used while running WSRM on the Filter Pipeline.

wsrc.sink.hostname=localhost

wsrc.sink.port –Holds the Sink TCP port number running on the FilterPipeline.

wsrc.sink.port=9000

wsrc.sink.config –Holds the Sink configuration file name.

wsrc.sink.config = config/wsrc/wsrc2.txt

#WSR

wsr.serviceA.axisadmin =Holds the admin service URL for the Service A running within OMII Container.

wsr.serviceA.axisadmin = -lhttp://localhost:18080/axis/services/AdminService

wsr.serviceA.deploy.wsddfile = Holds the Service A Deployment Descriptor file name and location.

wsr.serviceA.deploy.wsddfile=config/wsr/ServiceADeploy.wsdd

wsr.serviceA.undeploy.wsddfile = Holds the Service A Undeployment Descriptor file name and location.

wsr.serviceA.undeploy.wsddfile=config/wsr/ServiceAUndeploy.wsdd

wsr.serviceB.axisadmin =Holds the admin service URL for the Service B running within OMII Container.

wsr.serviceB.axisadmin = -lhttp://localhost:18080/axis/services/AdminService

wsr.serviceB.deploy.wsddfile = Holds the Service B Deployment Descriptor file name and location.

wsr.serviceB.deploy.wsddfile=config/wsr/ServiceBDeploy.wsdd

wsr.serviceB.undeploy.wsddfile = Holds the Service B Undeployment Descriptor file name and location.

wsr.serviceB.undeploy.wsddfile=config/wsr/ServiceBUndeploy.wsdd

wsr.ts.axisadmin = Holds the admin service URL for the Target running wihtin OMII Container.

wsr.ts.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

wsr.ts.deploy.wsddfile = Holds the target Deployment Descriptor file name and location.
wsr.ts.deploy.wsddfile=config/wsr/reliabilityTS.wsdd

wsr.ts.undeploy.wsddfile = Holds the target Undeployment Descriptor file name and location.
wsr.ts.undeploy.wsddfile=config/wsr/undeployTS.wsdd

wsr.proxy.axisadmin = Holds the admin service URL for the Proxy Service running within OMII Container.
wsr.proxy.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

wsr.proxy.deploy.wsddfile = Holds the Proxy Service Deployment Descriptor file name and location.
wsr.proxy.deploy.wsddfile=config/wsr/DeployProxy.wsdd

wsr.proxy.undeploy.wsddfile = Holds the Proxy Service Undeployment Descriptor file name and location.
wsr.proxy.undeploy.wsddfile=config/wsr/UndeployProxy.wsdd

wsr.tsproxy.axisadmin = Holds the admin service URL for the Target Proxy Service running within OMII Container.
wsr.tsproxy.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

wsr.tsproxy.deploy.wsddfile = Holds the Target Proxy Service Deployment Descriptor file name and location.
wsr.tsproxy.deploy.wsddfile=config/wsr/reliabilityTSProxy.wsdd

wsr.tsproxy.undeploy.wsddfile = Holds the Target Proxy Service Undeployment Descriptor file name and location.
wsr.tsproxy.undeploy.wsddfile=config/wsr/undeployTSProxy.wsdd

wsr.servlet.url — Holds the Axis Servlet URL.
wsr.servlet.url=-http://localhost:18080/axis/servlet/AxisServlet

wsr.webservice.serviceA = Holds the Service A Web Service name.
wsr.webservice.serviceA=WsrServiceA

wsr.webservice.serviceB = Holds the Service B Web Service name.
wsr.webservice.serviceB =WsrServiceB

wsr.webservice.target = Holds the Target Web Service name.
wsr.webservice.target=WsrTS

wsr.webservice.proxy = Holds the Proxy Web Service name.
wsr.webservice.proxy=WsrProxyService

wsr.webservice.targetproxy = Holds the Target proxy Web Service name.
wsr.webservice.targetproxy=WsrTSProxy

#WSRM

wsrn.serviceA.axisadmin – Holds the admin service URL for the Sink running within OMII Container.

wsrn.serviceA.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

wsrn.serviceA.deploy.wsdd – Holds the Sink Deployment Descriptor file name and location.

wsrn.serviceA.deploy.wsdd=config/wsrn/wsdd/handler/ServiceADeploy.wsdd

wsrn.serviceA.undeploy.wsdd – Holds the Sink Undeployment Descriptor file name and location.

wsrn.serviceA.undeploy.wsdd=config/wsrn/wsdd/handler/ServiceAUndeploy.wsdd

wsrn.serviceB.axisadmin – Holds the admin service URL for the Source running within OMII Container.

wsrn.serviceB.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

wsrn.serviceB.deploy.wsdd – Holds the Source Deployment Descriptor file name and location.

wsrn.serviceB.deploy.wsdd=config/wsrn/wsdd/handler/ServiceBDeploy.wsdd

wsrn.serviceB.undeploy.wsdd – Holds the Source Undeployment Descriptor file name and location.

wsrn.serviceB.undeploy.wsdd=config/wsrn/wsdd/handler/ServiceBUndeploy.wsdd

wsrn.targetservice.axisadmin – Holds the admin service URL for the Target Service running within OMII Container.

wsrn.targetservice.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

wsrn.targetservice.deploy.wsdd – Holds the Target Service Deployment Descriptor file name and location.

wsrn.targetservice.deploy.wsdd=config/wsrn/wsdd/proxy/TargetServiceDeploy.wsdd

wsrn.targetservice.undeploy.wsdd – Holds the Target Service Undeployment Descriptor file name and location.

wsrn.targetservice.undeploy.wsdd=config/wsrn/wsdd/proxy/TargetServiceUndeploy.wsdd

wsrn.proxyserviceB.axisadmin – Holds the admin service URL for the Sink Proxy Service running within OMII Container.

Wsrn.proxyserviceB.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

wsrn.proxyserviceB.deploy.wsdd – Holds the Sink Proxy Service Deployment Descriptor file name and location.

wsrn.proxyserviceB.deploy.wsdd=config/wsrn/wsdd/proxy/ProxyServiceDeploy.wsdd

wsrn.proxyserviceB.undeploy.wsdd – Holds the Sink Proxy Service Undeployment Descriptor file name and location.

wsrn.proxyserviceB.undeploy.wsdd=config/wsrn/wsdd/proxy/ProxyServiceUndeploy.wsdd

wrm.clientservice.axisadmin – Holds the admin service URL for the Client Service running within OMII Container.

wrm.clientservice.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

wrm.clientservice.deploy.wsdd – Holds the Client Service Deployment Descriptor file name and location.

wrm.clientservice.deploy.wsdd=config/wrm/wsdd/proxy/ClientServiceDeploy.wsdd

wrm.clientservice.undeploy.wsdd – Holds the Client Service Undeployment Descriptor file name and location.

wrm.clientservice.undeploy.wsdd=config/wrm/wsdd/proxy/ClientServiceUndeploy.wsdd

wrm.proxyserviceA.axisadmin – Holds the admin service URL for the Source Proxy service running within OMII Container

wrm.proxyserviceA.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

wrm.proxyserviceA.deploy.wsdd – Holds the Source Proxy service Deployment Descriptor file name and location.

wrm.proxyserviceA.deploy.wsdd=config/wrm/wsdd/proxy/ProxyServiceDeploy.wsdd

wrm.proxyserviceA.undeploy.wsdd – Holds the Source Proxy service Undeployment Descriptor file name and location.

wrm.proxyserviceA.undeploy.wsdd=config/wrm/wsdd/proxy/ProxyServiceUndeploy.wsdd

wrm.servlet.url – Holds the Axis Servlet URL.

wrm.servlet.url=-lhttp://localhost:18080/axis/servlet/AxisServlet

wrm.webservice.serviceB – Holds the Sink Web Service name.

wrm.webservice.serviceB=WsrmserviceB

wrm.webservice.serviceA – Holds the Source Web Service name.

wrm.webservice.serviceA=WsrmserviceB

wrm.webservice.targetservice – Holds the Target Web Service name.

wrm.webservice.targetservice=WsrmserviceTargetService

wrm.webservice.proxyservice – Holds the Sink & Source Proxy Web Service names on their machines.

wrm.webservice.proxyservice=WsrmserviceProxyService

wrm.webservice.clientservice – Holds the Client Web Service name.

wrm.webservice.clientservice=WsrmserviceClientService

#COMMON

soapmonitor.axisadmin = Holds the admin service URL for SOAPMonitor service running within OMII Container.

soapmonitor.axisadmin=-lhttp://localhost:18080/axis/services/AdminService

soapmonitor.deploy.wsddfile = Holds the SOAPMonitor service Deployment Descriptor file name and location.

soapmonitor.deploy.wsddfile=config/deploySoapMonitor.wsdd

soapmonitor.undeploy.wsddfile = Holds the SOAPMonitor service Undeployment Descriptor file name and location.

soapmonitor.undeploy.wsddfile=config/undeploySoapMonitor.wsdd

servlet.url = Holds the Axis Servlet URL.

servlet.url=-http://localhost:18080/axis/servlet/AxisServlet

webservice.soapmonitor = Holds the SOAPMonitor Web Service name.

webservice.soapmonitor=SOAPMonitorService

WSR DATABASE

Postgreddatabase_JDBC_Driver= Holds the driver name for PostgreSQL database.

Postgreddatabase_JDBC_Driver= org.postgresql.Driver

Postgreddatabase_WSR_Source_url = Holds the URL of WSR source database for PostgreSQL.

Postgreddatabase_WSR_Source_url=jdbc:postgresql://localhost:5432/wsrone

Postgreddatabase_WSR_username = Holds the username of PostgreSQL database.

Postgreddatabase_WSR_username = test

Postgreddatabase_WSR_password- Holds the password of PostgreSQL database.

Postgreddatabase_WSR_password =test

Postgreddatabase_WSR_Sink_url = Holds the URL of WSR Sink database for PostgreSQL.

Postgreddatabase_WSR_Sink_url= jdbc:postgresql://localhost:5432/wsrtwo

Mysqldatabase_JDBC_Driver = Holds the driver name for MySQL database

Mysqldatabase_JDBC_Driver= org.gjt.mm.mysql.Driver

Mysqldatabase_WSR_Source_url = Holds the URL of WSR Source Database for MySQL

Mysqldatabase_WSR_Source_url =jdbc:mysql://localhost:3306/wsrone

Mysqldatabase_WSR_username = Holds the username of MySQL database

Mysqldatabase_WSR_username= root

Mysqldatabase_WSR_password = Holds the password for MySQL database

Mysqldatabase_WSR_password = root

Mysqldatabase_WSR_Sink_url = Holds the URL of WSR Sink database for MySQL

Mysqldatabase_WSR_Sink_url= jdbc:mysql://localhost:3306/wsrtwo

#WSRM DATABASE

pg_database_JDBC_Driver- Holds the driver name for PostgreSQL database.

pg_database_JDBC_Driver= org.postgresql.Driver

`pg_database_WSRM_Source_url`- Holds the URL of WSRM source database for Postgre SQL.

`pg_database_WSRM_Source_url =jdbc:postgresql://localhost:5432/wsrnone`

`pg_database_WSRM_username`- Holds the username of Postgre SQL database.

`Pg_database_WSRM_username= test`

`pg_database_WSRM_password`- Holds the password of Postgre SQL database.

`pg_database_WSRM_password =test`

`pg_database_WSRM_Sink_url`- Holds the URL of WSRM Sink database for Postgre SQL.

`pg_database_WSRM_Sink_url= jdbc:postgresql://localhost:5432/wsrmtwo`

`Database_JDBC_Driver`- Holds the driver name for MySQL database

`Database_JDBC_Driver= org.gjt.mm.mysql.Driver`

`Database_WSRM_Source_url`- Holds the URL of WSRM Source Database for MySQL

`Database_WSRM_Source_url =jdbc:mysql://localhost:3306/wsrnone`

`Database_WSRM_username`-Holds the username of MySQL database

`Database_WSRM_username= root`

`Database_WSRM_password`- Holds the password for MySQL database

`Database_WSRM_password = root`

`Database_WSRM_Sink_url`= Holds the URL of WSRM Sink database for MySQL

`Database_WSRM_Sink_url= jdbc:mysql://localhost:3306/wsrmtwo`

prepare: Creates classes directory under current directory structure

Usage: `$ant prepare`

compile: Compiles all files under the src folder and copies all the class files to classes folder by preserving the same directory structure. It also creates NbWebInfra.jar file and puts under lib directory.

Usage: `$ant compile`

client-dir: Creates client directory from base directory structure.

Usage: `$ant client-dir`

server-dir: Creates server directory from base directory structure.

Usage: `$ant server-dir`

ServerWar: Builds firms.war server distribution

Usage: `$ant ServerWar`

build-server: Creates server distribution.

Usage: `$ant build-server`

build-client: Creates client distribution

Usage: `$ant build-client`

build-all: Creates client and server distribution

Usage: \$ant build-all

start: Starts tomcat server.

Usage: \$ant start

stop: Stops tomcat server.

Usage: \$ant stop

Release = Echoes a message "WS-ReliableMessaging Release: <http://msdn.microsoft.com/ws/2005/02/ws-reliablemessaging/> and WS-Reliability Release: Aug-2004 <http://docs.oasis-open.org/wsrn/2004/06/WS-Reliability-CD1.086.pdf> "to the standard output

Usage: \$ant release

jar: Creates a jar file called NbWebInfra.jar from all the files of classes' folder and put it into lib folder

Usage: \$ant jar

build-doc: Generates java doc

Usage: \$ant build-doc

clean: Cleans all generated class files and documentation.

Usage: \$ant clean

test: Run JUnit tests for cgl.narada.wsinfra.util, cgl.narada.wsinfra.wsa.impl, cgl.narada.wsinfra.wsr.impl and cgl.narada.wsinfra.wsrn.impl package.

Usage: \$ant test

2. Targets required for OMII deployment and testing

deploy-firms: Copies all jar, classes, configuration Files required for deployment of FIRMS web services within OMII Tomcat Instance.

Usage: \$ant deploy-firms

undeploy-firms: Deletes all jar, classes and configuration files previously copied for deployment of FIRMS web services within OMII container.

Usage: \$ant undeploy-firms

#WSR

deploy-wsrserviceA: Deploys service A within OMII Container running on port 18080

Usage: \$ ant deploy-wsrserviceA -Dwsr.serviceA.axisadmin=-lhttp://localhost:18080/axis/services/AdminService -Dwsr.serviceA.deploy.wsddfile=config/wsr/ServiceADeploy.wsdd

test-wsrserviceA: Test the deployed service A within OMII Container running on port 18080

Usage: \$ ant test-wsrserviceA -Dwsr.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -Dwsr.webservice.serviceA =WsrServiceA

undeploy-wsrserviceA: Undeploys service A from OMII Container running on port 18080

Usage: \$ ant undeploy-wsrserviceA -Dwsr.serviceA.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsr.serviceA.undeploy.wsddfile=config/wsr/ServiceAUndeploy.wsdd

deploy-wsrserviceB: Deploys service B within OMII Container running on port 18080
Usage: \$ ant deploy-wsrserviceB -Dwsr.serviceB.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsr.serviceB.deploy.wsddfile=config/wsr/ServiceBDeploy.wsdd

test-wsrserviceB: Test the deployed service B within OMII Container running on port 18080
Usage: \$ ant test-wsrserviceB -
 Dwsr.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsr.webservice.serviceB =WsrServiceB

undeploy-wsrserviceB: Undeploys service B from OMII Container running on port 18080
Usage: \$ ant undeploy-wsrserviceB -Dwsr.serviceB.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsr.serviceB.undeploy.wsddfile=config/wsr/ServiceBUndeploy.wsdd

deploy-wsrts: Deploys Target service within OMII Container running on port 18080
Usage: \$ ant deploy-wsrts -Dwsr.ts.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsr.ts.deploy.wsddfile=config/wsr/reliabilityTS.wsdd

test-target: Test the deployed Target web service within OMII Container running on port 18080
Usage: \$ ant test-target -
 Dwsr.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsr.webservice.target=WsrTS

undeploy-wsrts: Undeploys Taget service from OMII Container running on port 18080
Usage: \$ ant undeploy-wsrts -Dwsr.ts.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsr.ts.undeploy.wsddfile=config/wsr/undeployTS.wsdd

deploy-wsrproxy: Deploys proxy service within OMII Container running on port 18080
Usage: \$ ant deploy-wsrproxy -Dwsr.proxy.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -Dwsr.
 proxy.deploy.wsddfile=config/wsr/DeployProxy.wsdd

test-proxy: Test the deployed Proxy web service within OMII Container running on port 18080
Usage: \$ ant test-proxy -
 Dwsr.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsr.webservice.proxy=WsrProxyService

undeploy-wsrproxy: Undeploys proxy service from OMII Container running on port 18080
Usage: \$ ant undeploy-wsrproxy -Dwsr.proxy.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsr.proxy.undeploy.wsddfile=config/wsr/UndeployProxy.wsdd

deploy-wsrtsproxy: Deploys Target proxy service within OMII Container running on port 18080

Usage: \$ ant deploy-wsrtsproxy -Dwsr.tsproxy.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsr.tsproxy.deploy.wsddfile=config/wsr/reliabilityTSProxy.wsdd

test-targetproxy: Test the deployed Target Proxy web service within OMII Container running on port 18080

Usage: \$ ant test-targetproxy -
 Dwsr.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsr.webservice.targetproxy=WsrTSProxy

undeploy-wsrtsproxy: Undeploys Target proxy service from OMII Container running on port 18080

Usage: \$ ant undeploy-wsrtsproxy -Dwsr.tsproxy.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsr.tsproxy.undeploy.wsddfile=config/wsr/undeployTSProxy.wsdd

run-sourceclient: Runs source service client endpoint

Usage: \$ant run-sourceclient

run-sourceclientframe: Runs source service client endpoint as a Frame

Usage: \$ant run-sourceclientframe

run-sourceproxyclient: Runs source proxy service client

Usage: \$ant run-sourceproxyclient

#WSRM

deploy-serviceA: Deploys serviceA within OMII Container running on port 18080

Usage: \$ ant deploy-serviceA -Dwsrm.serviceA.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.serviceA.deploy.wsdd=config/wsr/wsdd/handler/ServiceADeploy.wsdd

test-serviceA: Test the deployed Service A web service within OMII Container running on port 18080

Usage: \$ ant test-serviceA -
 Dwsrm.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsrm.webservice.serviceA=WsrServiceA

undeploy-serviceA: Undeploys ServiceA (Source) service from OMII Container running on port 18080

Usage: \$ ant undeploy-serviceA -Dwsrm.serviceA.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.serviceA.undeploy.wsdd=config/wsr/wsdd/handler/ServiceAUndeploy.wsdd

deploy-serviceB: Deploys service B(Sink) service within OMII Container running on port 18080

Usage: \$ ant deploy-serviceB -Dwsrm.serviceB.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.serviceB.deploy.wsdd=config/wsr/wsdd/handler/ServiceBDeploy.wsdd

test-serviceB: Test the deployed Service B (Sink) web service within OMII Container running on port 18080

Usage: \$ ant test-serviceB -
 Dwsrm.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsrm.webservice.serviceB=WsrmserviceB

undeploy-serviceB: Undeploys service B (Sink) service from OMII Container running on port 18080

Usage: \$ ant undeploy-serviceB -Dwsrm.serviceB.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.serviceB.undeploy.wsdd=config/wsrmserviceB/wsdd/handler/ServiceBUndeploy.wsdd

deploy-clientservice: Deploys Client service running with Source proxy within OMII Container running on port 18080

Usage: \$ ant deploy-clientservice -Dwsrm.clientservice.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.clientservice.deploy.wsdd=config/wsrmserviceB/wsdd/proxy/ClientServiceDeploy.wsdd

test-clientservice: Test the deployed Client web service within OMII Container running on port 18080

Usage: \$ ant test-clientservice -
 Dwsrm.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsrm.webservice.clientservice=WsrmserviceBClientService

undeploy-clientservice: Undeploys Client service from OMII Container running on port 18080

Usage: \$ ant undeploy-clientservice -Dwsrm.clientservice.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.clientservice.deploy.wsdd=config/wsrmserviceB/wsdd/proxy/ClientServiceUndeploy.wsdd

deploy-proxy-serviceA: Deploys Proxy Service (Source) service within OMII Container running on port 18080

Usage: \$ ant deploy-proxy-serviceA -Dwsrm.proxy-serviceA.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.proxy-serviceA.deploy.wsdd=config/wsrmserviceB/wsdd/proxy/ProxyServiceDeploy.wsdd

test-proxy-serviceA: Test the deployed Proxy Service (source) web service within OMII Container running on port 18080

Usage: \$ ant test-proxy-serviceA -
 Dwsrm.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsrm.webservice.proxy-serviceA=WsrmserviceBProxyService

undeploy-proxy-serviceA: Undeploys proxy service A (Source) from OMII Container running on port 18080

Usage: \$ ant undeploy-proxy-serviceA -Dwsrm.proxy-serviceA.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.proxy-serviceA.undeploy.wsdd=config/wsrmserviceB/wsdd/proxy/ProxyServiceUndeploy.wsdd

deploy-proxy-serviceB: Deploys Proxy Service B (Sink) service within OMII Container running on port 18080

Usage: \$ ant deploy-proxyServiceB -Dwsrm.proxyServiceA.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.proxyServiceB.deploy.wsdd=config/wsrM/wsdd/proxy/ProxyServiceDeploy.ws
 dd

test-proxyServiceB: Test the deployed Proxy Service (Sink) web service within OMII Container running on port 18080

Usage: \$ ant test-proxyServiceB -
 Dwsrm.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsrm.webservice.proxyService=WsrMProxyService

undeploy-proxyServiceB: Undeploys proxy service B (Sink) from OMII Container running on port 18080

Usage: \$ ant undeploy-proxyServiceB -Dwsrm.proxyServiceB.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.proxyServiceB.undeploy.wsdd=config/wsrM/wsdd/proxy/ProxyServiceUndepl
 oy.wsdd

deploy-targetService: Deploys Target service with Sink Proxy within OMII Container running on port 18080

Usage: \$ ant deploy-targetService -Dwsrm.targetService.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.targetService.deploy.wsdd=config/wsrM/wsdd/proxy/TargetServiceDeploy.ws
 dd

test-targetService: Test the deployed Target web service within OMII Container running on port 18080

Usage: \$ ant test-targetService -
 Dwsrm.servlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwsrm.webservice.targetService=WsrMTargetService

undeploy-targetService: Undeploys Target service from OMII Container running on port 18080

Usage: \$ ant undeploy-targetService -Dwsrm.targetService.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 Dwsrm.targetService.deploy.wsdd=config/wsrM/wsdd/proxy/TargetServiceUndeploy.
 wsdd

run-sampleClient: Runs source service client endpoint

Usage: \$ ant run-sampleClient

#COMMON

deploy-soapMonitor: Deploys soapMonitor service within OMII Container

Usage: \$ ant deploy-soapMonitor -DsoapMonitor.axisadmin=-
 lhttp://localhost:18080/axis/services/AdminService -
 DsoapMonitor.deploy.wsddfile=config/deploySoapMonitor.wsdd

test-soapMonitor: Test the deployed SOAP Monitor service within OMII Container running on port 18080

Usage: \$ ant test-soapMonitor -
 Dservlet.url=http://localhost:18080/axis/servlet/AxisServlet -
 Dwebservice.soapMonitor=SOAPMonitorService

undeploy-soapmonitor: Undeploys soapmonitor service from OMII Container running on port 18080

Usage: \$ ant undeploy-soapmonitor -Dsoapmonitor.axisadmin=-
lhttp://localhost:18080/axis/services/AdminService -
Dsoapmonitor.undeploy.wsddfile=config/undeploySoapMonitor.wsdd

3. Targets required to test using Filter Pipeline

#WSR

wsr-sourcefilter = Starts source service end point.

Usage: \$ ant wsr-sourcefilter -Dwsr.source.hostname=localhost -
Dwsr.source.port=8000 -Dwsr.source.config=config/wsr/WsrSourceConfig.properties

wsr-sinkfilter = Starts first sink service end point.

Usage: \$ ant wsr-sinkfilter -Dwsr.sink.hostname=localhost
-Dwsr.sink.port=9000 -Dwsr.sink.config=config/wsr/WsrSinkConfig.properties

#WSRM

wsr-node1: Starts source service end point.

Usage: \$ ant wsrm-node1 -Dwsrm.source.hostname=localhost -
Dwsrm.source.port=8000 -Dwsrm.source.config=config/wsr/wsr1.txt

wsr-node2: Starts first sink service end point.

Usage: \$ ant wsrm--node2 -Dwsrm.sink.hostname=localhost -
Dwsrm.sink.port=9000 -Dwsrm.sink.config=config/wsr/wsr2.txt

4. Targets required to create and drop PostgreSQL and MySQL tables

#WSR

dropdb-postgresource= Drops all the tables from the source database of PostgreSQL

createdb-postgresource= creates all the tables for source database of PostgreSQL

dropdb-postgresink= Drops all the tables from the sink database of PostgreSQL

createdb-postgresink= creates all the tables for sink database of PostgreSQL.

dropdb-mysqlsource= Drops all the tables from the source database of MySQL

createdb-mysqlsource= creates all the tables for source database of MySQL

dropdb-mysqlsink= Drops all the tables from the sink database of MySQL

createdb-mysqlsink= creates all the tables for sink database of MySQL

#WSRM

pgsql-dropdb-source: Drops all the tables from the WSRM source database of PostgreSQL

pgsql-createdb-source: creates all the tables for WSRM source database of PostgreSQL

pgsql-dropdb-sink: Drops all the tables from the WSRM sink database of PostgreSQL

pgsql-createdb-sink- creates all the tables for WSRM sink database of PostgreSQL.

dropdb-source: Drops all the tables from the WSRM source database of MySQL

createdb-source: Creates all the tables for WSRM source database of MySQL

dropdb-sink: Drops all the tables from the WSRM sink database of MySQL

createdb-sink- creates all the tables for WSRM sink database of MySQL