# Summary

This document details the process of installing the NREL mediator service onto an OpenESB server. The service, as currently written, supports two methods: create mainenance order, and get maintenance orders. The service performs the following functions. It receives a request and splits it, sending it to two remote SOAP servers. It also outputs the requests to logfiles.

# Installation of OpenESB

## Downloading OpenESB:

1. Download OpenESB (<http://www.open-esb.net/>). My version 3.0.5
2. Unzip the file
3. Move the directory to /usr/local

## Starting the OpeneSB Server:

1. Start the server :   
   sh> /usr/local/OpenESB-SE-3.0.5/OE-Instance/bin/openesb.sh start &
2. Server Console :  
   <http://localhost:4848/plugin/webui>
3. Login : admin/admin

## Component Installation

OpenESB supports plugins. The mediator requires installation of the following components :

* BPEL SE – scalable orchestrator (Component) bpelse.jar
* FILE BC – provides interacting with files (Component) filebc.jar
* HTTP BC FULL – provides messaging over HTTP (Component) httpbc-full.jar
* Wsdlextlib.jar (Shared lib)
* Encoderlib.jar (Shared lib)

### Installing Shared Libraries

* Access web console : <http://localhost:4848/plugin/webui>
* Click on Shared Libraries (on left menu)
* Click Install
* Select file : /usr/local/OpenESB-SE-3.0.5/OE-Components/**encoderlib.jar**
* Click ‘Start upload’

Repeat process for **wsdlextlib.jar**

### Installing the 3 components

* Access web console : <http://localhost:4848/plugin/webui>
* Click on Components (on left menu)
* Click Install
* Click ‘Choose File’
* Select file : /usr/local/OpenESB-SE-3.0.5/OE-Components/**bpelse.jar**
* Click Install

Repeat process for **filebc.jar** and **httpbcfull.jar**

### JNDI Configuration

1. Copy /usr/local/git/nrel/config/context.xml to /usr/local/OpenESB-SE-3.0.5/OE-Instance/config (overwriting original context file). This file points to the local mysql instance for openesb.

**Congratulations.**

# Installation of Mediator Application

* Access web console : <http://localhost:4848/plugin/webui>
* Click ‘Service Assemblies’ (left side menu)
* Click on ‘+ Deploy’ button
* Choose file :   
  git/ExecuteDERGroupDispatchesApp/dist/ExecuteDERGroupDispatchesApp.zip Click ‘Start upload’
* Click ‘Start’ button on the service assembly page
* Repeat process for file :  
  git/ExecuteDERGroupsApp/dist/ExecuteDERGroupsApp.zip

Note: the start will partially fail at this point, but it will also create the endpoint variables (initially set to null) for the next step.

# Endpoint Configuration

The mediator endpoint are configurable. There are six endpoints, three for Get, and three for Create:

* CreateDERGroup\_PORT – This is the port that the mediator SOAP service (openesb) listens on for create der requests. It logs the requests and sends them to the two remote services for execution.
* CREATEDERGROUP\_SVR1 –SOAP service for create der group (client’s server) order remote server 1. Note : The actual destination is set by looking up vendor name in database.
* CREATEDERGROUP\_SVR1 – This is the port that the mediator SOAP service (openesb) listens on for create der requests. It logs the requests and sends them to the two remote services for execution
* CreateDERGroupDispatches\_PORT – SOAP service for DER Groups dispatch (openesb server)
* CreateDERGroupDispatches\_SVR1 – SOAP server for der dispatches (client’s server). Note : The actual destination is set by looking up vendor name in database.
* CreateDERGroupDispatches\_LOG – SOAP service runing tomcat server that logs messages to nrel database

The urls for these three addresses must be configured within the OpenESB web console.

Do the following in order for each of the six variables :

* Access web console : <http://localhost:4848/plugin/webui>
* Click on Components (left menu)
* Click on sun-http-binding item
* Click on ‘Application Variables’
* Do the following steps for the each variable (example below):
  + Click ‘+ Add’
  + The vars should already be created, but null. Fill in with your own values which should resemble those below
  + Application Configuration Name : NREL\_CREATE\_SERVICE <http://localhost:8081/epriConnect/MaintOrderServiceCreate>

|  |  |
| --- | --- |
| **Name** | **Value** |
| CreateDERGroup\_SVR1 | http://localhost:8085/epriConnect/executeDERGroups |
| CreateDERGroup\_PORT | http://localhost:8081/epriConnect/createDERGroups |
| CreateDERGroup\_LOG | http://localhost:8080/epriConnect/createDERGroups |
| CreateDERGroupDispatches\_SVR1 | http://localhost:8085/epriConnect/executeDERGroupDispatches |
| CreateDERGroupDispatches\_PORT | http://localhost:8081/epriConnect/executeDERGroupDispatches |
| CreateDERGroupDispatches\_LOG | http://localhost:8080/epriConnect/executeDERGroupDispatches |

Note : If you are lazy (or don’t have two epriConnect servers set up), you can set R1 and R2 to point to the same service, ie: port 8080/epriConnect/MaintOrderGet|Create

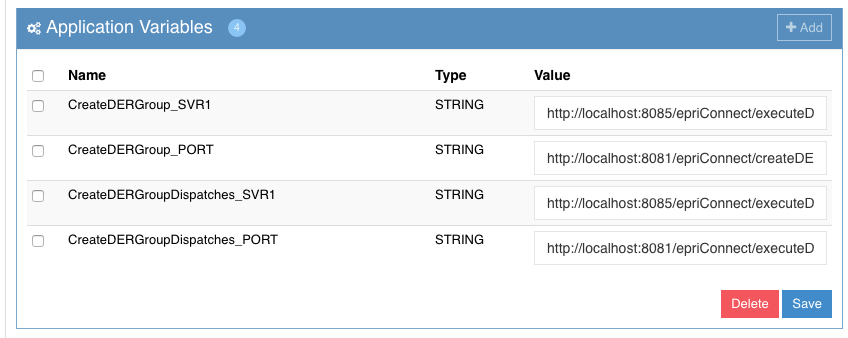


Figure : Configuration Variables

# Start Mediator Service

* Click on ‘Service Assemblies’ on left menu
* Click on ExecuteDERGroupsApp
* Click on Shutdown
* Click on ‘Start’
* Repeat process for ExecuteDERGroupDispatchesApp

At this point the service should be up and ready to receive connections.

# Testing Mediator Application

### Set up (open) SOAP UI Project to send create DER group

* Open Soapui – git/nrel/61968/soap-ui/ExecuteDERGroups-nrel-soapui-project.xml
* Select CreateDerGroups testCase, Steps/CreateDERGroups
* Set url : http://localhost:8081/epriConnect/createDERGroups
* (This should be the url set for CreateDERGroup\_PORT – See EndPointConfiguration)

### Set up (open) SOAP UI Project to send execute DER group dispatches

* Open Soapui – git/nrel/61968/soap-ui/ExecuteDERGroupdispatches-nrel-soapui-project.xml
* Select CreateDerGroupDispatches testCase, Steps/CreateDERGroupDispatch
* Set url : http://localhost:8081/epriConnect/executeDERGroupDispatches
* (This should be the url set for CreateDERGroupDispatches\_PORT – See EndPointConfiguration)

### Start both epricConnect services

* Start two copies of tomcat (instructions for running two versions of tomcat on the same server are included in Appendix A)
* Drop the epriConnect.war file to your tomcat/webapps directory
* Verify that the your SOAP services are by doing the following :
  + Browse to the url you entered as CreateDERGroup\_LOG, appending ‘?wsdl’ to it, eg: http://localhost:8080/epriConnect/createDERGroups?wsdl
  + Browse to the url for CreateDERGroupDispatches\_LOG, appending ‘?wsdl’ to it, eg: http://localhost:8080/epriConnect/executeDERGroupDispatches

### Start Mediator Service

* Access web console : <http://localhost:4848/plugin/webui>
* Click Service Assemblies
* Start ExecuteDERGroupsApp
* Browse to the url you set during configuration, appending ‘?wsdl’ to the url, eg: http://localhost:8081/epriConnect/createDERGroups?wsdl (you should see the wsdl for the get service)
* Start ExecuteDERGroupDispatchesApp
* Browse to the url you set during configuration, appending ‘?wsdl’ to the url, eg: http://localhost:8081/epriConnect/executeDERGroupDispatches?wsdl (you should see the wsdl for the create service)

### Executing Basic Test

#### Create Test

Instructions are provided for testing ExecuteDERGroups. The instructions for testing ExecuteDERGroupDispatches differ only in files opened.

This test will do the following :

* Send a create DER Group create request from SOAPUI to the mediator service
* The mediator will then
  + Send the SOAP request to R1 (client server)
  + Send the SOAP request to R2 (tomcat logging server)

Execute the Test :

* Open soapui file : **git/nrel/61968/soap-ui/ExecuteDERGroups-nrel-soapui-project.xml**
* Open test : **DERGroups\_Binding testSuite/CreateDERGroups TestCase/Test Steps/CreateDERGroups**
* Set the endpoint to point to point to the mediator : http://localhost:8081/epriConnect/createDERGroups

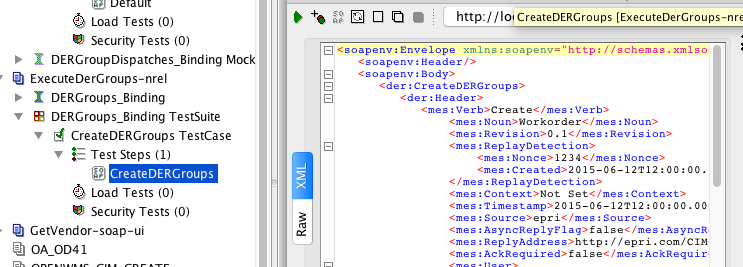


Figure : Soap UI Test Case

* Execute!

Verify results :

* Response window in SOAP UI
* Database contains new log message, sent and received

## Mediator Flow

(Out of date, new explanation and pictures to follow)

The following represents the flow of the mediator.

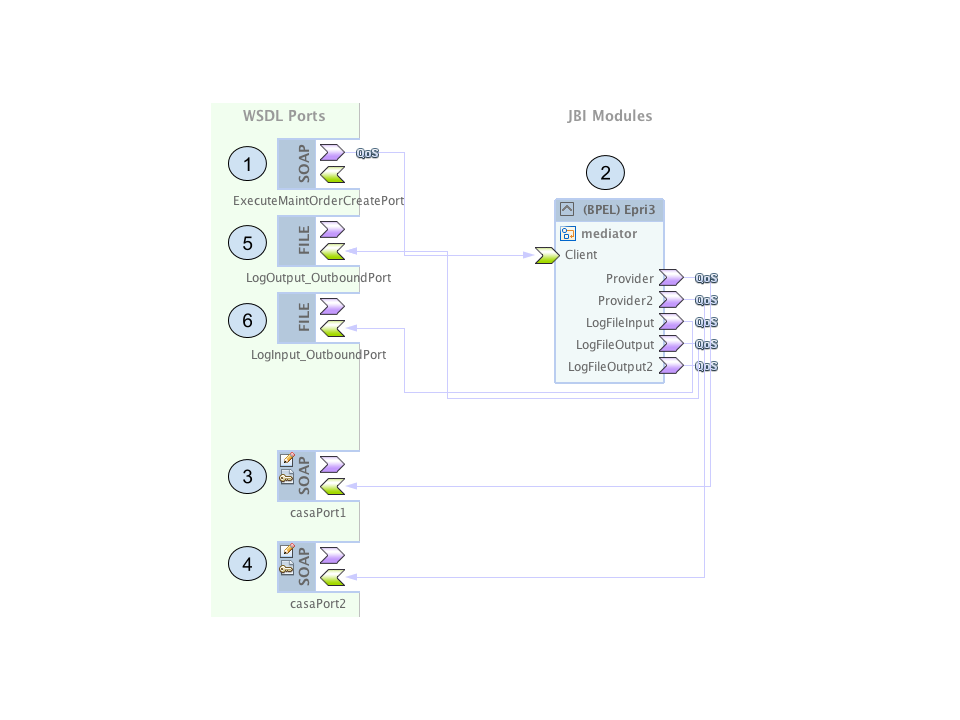


Figure : Mediator Flow

1. Mediator Service Port – Port defined by SERVICE\_ENDPOINT application variable
2. Mediator Service – Routes soap messages
3. EpriConnect service endpoint – REMOTE\_1
4. EpriConnect service endpoint 2 – REMOTE\_2
5. Log output file for SOAP Message sent, and message received from REMOTE\_1
6. Log output file for SOAP Message received back from REMOTE\_2

# Appendix A – Creating a Second Tomcat Instance on a Server

Process for creating a second tomcat instance on a unix box

* sh> cp –rf /usr/local/tomcat /usr/local/tomcat2
* sh> vi /usr/local/tomcat2/conf/server.xml

<Server port="8006" shutdown="SHUTDOWN"> <!— bump it up 1 from first instance —>

<Connector port="8085" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8444" /> <!— again, bump ports up by one —>

<Connector port="8010" protocol="AJP/1.3" redirectPort="8444" />