

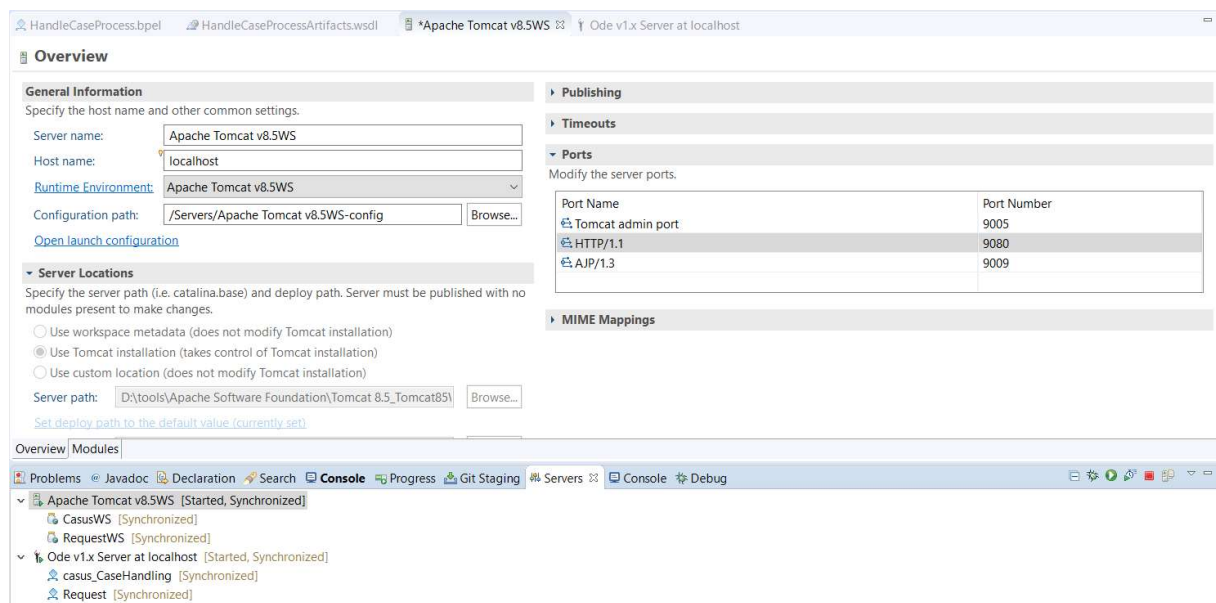
Develop, Deploy and run BPEL cases

Prerequisites

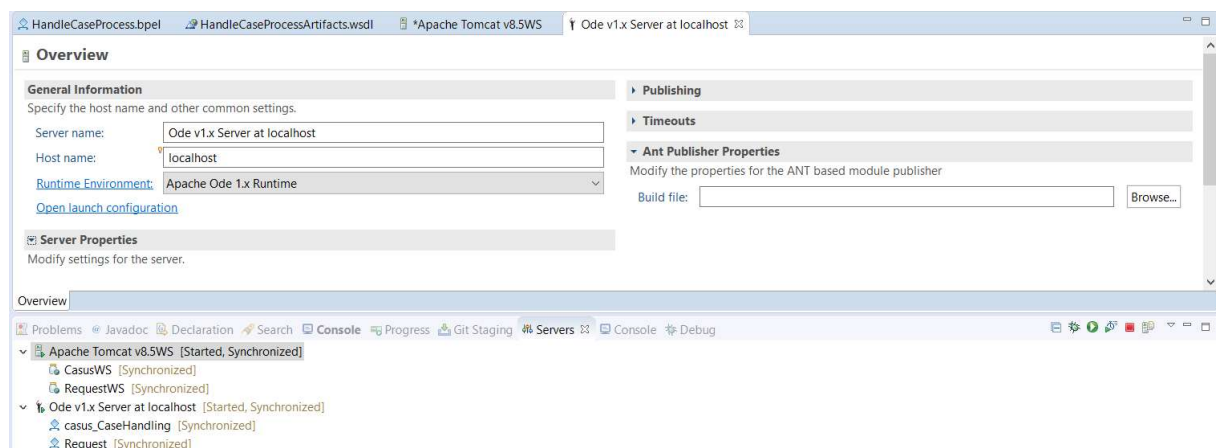
- Eclipse Oxygen, Version: Oxygen.1 Release (4.7.1) (<https://projects.eclipse.org/releases/oxygen>)
- Eclipse BPEL Designer 1.1.1 (<https://projects.eclipse.org/projects/soa.bpel>)
- Apache ODE 1.3.7: Apache ODE (<http://ode.apache.org/>)
- Apache Tomcat Version 8.5.23 (<http://tomcat.apache.org/>)
- SoapUI 5.5.0: a tool for testing web services. (<https://www.soapui.org/>)

Steps

1. Import projects ODE/casus_CaseHandling, ODE/request, WS/CasusWS and WS/RequestWS
2. Create server in Eclipse/Oxygen/server tab for deploying web services, configure HTTP port 9080



3. Create server in Eclipse/Oxygen/server tab for deploying BPEL processes



4. Deploy projects and start servers

5. Test servers

The screenshot shows two web interfaces. The top interface is the Tomcat Web Application Manager, displaying a table of applications. The bottom interface is the Apache ODE Console, showing a dashboard with statistics and a list of process packages.

Tomcat Web Application Manager

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle x 30 minutes
CasusWS	None specified	CasusWS	true	0	Start Stop Reload Undeploy Expire sessions with idle x 30 minutes
RandomHelloService	None specified	RandomHelloService	true	0	Start Stop Reload Undeploy Expire sessions with idle x 30 minutes
RequestWS	None specified	RequestWS	true	0	Start Stop Reload Undeploy Expire sessions with idle x 30 minutes

Apache ODE Console

Dashboard

- 2 Process Packages
- 2 Process Models
- 0 Process Instances

0 Active Instances

0 Suspended Instances

0 Completed Instances

0 Failed Instances

0 Terminated Instances

6. Run BPEL processes

- Create SoapUI project using WSDL
<http://localhost:8080/ode/processes/HandleCaseProcess?wsdl>
- Create SoapUI project using WSDL
<http://localhost:8080/ode/processes/HandleRequest?wsdl>
- Run for example

The screenshot shows the SoapUI interface with a SOAP request and response. The request is a `han:Request` with a `han:payload` of "Hello world". The response is a `HandleRequestResponse` with a `tns:result` of "Ok".

Raw XML

```
<?xml version='1.0' encoding='UTF-8'?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Header/><soapenv:Body><han:Request><han:payload>Hello world</han:payload></han:Request></soapenv:Body></soapenv:Envelope>
```

Raw XML

```
<?xml version='1.0' encoding='UTF-8'?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Body><HandleRequestResponse xmlns="http://handerequest.ws"><tns:result>Ok</tns:result></HandleRequestResponse></soapenv:Body></soapenv:Envelope>
```

- Check web service logging in console

