Information & Egovernment authority

CIO-ESB Handover

[Enterprise Service Bus Handover Document]

[12May 2018]

Amendment History

| Version Control | | | | |
| --- | --- | --- | --- | --- |
| Ver. | Date | Name | Role | Summary of Changes |
| 0.1 | 12 May 2019 | Mohamed Hafez | Sr.System Analyst | Draft Version |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

| Approvals | | | | |
| --- | --- | --- | --- | --- |
| Ver. | Date | Name | Role | Signature |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents

[INTRODUCTION 4](#_Toc8640439)

[SETTING WORKSPACE UP 4](#_Toc8640440)

[PRE-REQUESITS 4](#_Toc8640441)

[Workspace Configuration 4](#_Toc8640442)

[Services Development 18](#_Toc8640443)

[New Service 18](#_Toc8640444)

[Existing Services 20](#_Toc8640445)

[**Packaging And Deployment** 20](#_Toc8640446)

[Producing a staging package 20](#_Toc8640447)

[Producing a production package 23](#_Toc8640448)

# INTRODUCTION

This document describe in details, gradually how to setup the CIO-ESB component workspace, development procedure and packaging

# SETTING WORKSPACE UP

### PRE-REQUESITS

###### 1- Download Java Development Kit 8

From <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> OR get it locally from \\smsgho\public\ESB

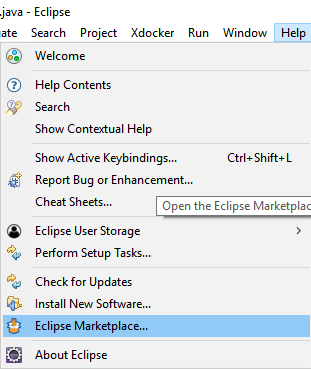
###### 2- Download eclipse oxygen

From <https://www.eclipse.org/downloads/> OR get it locally from \\smsgho\public\ESB

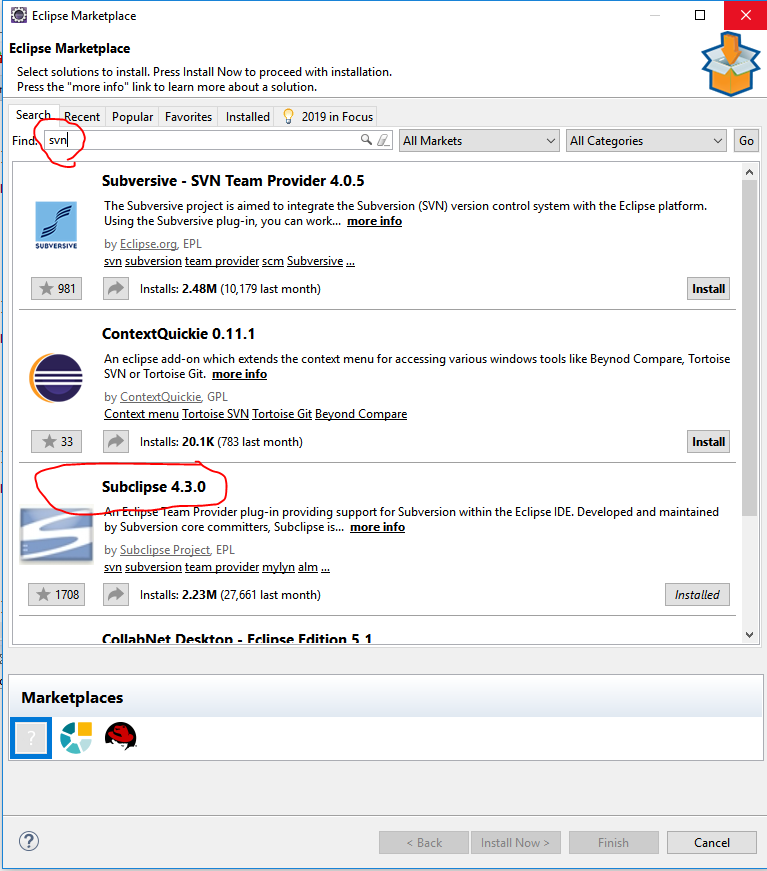
### Workspace Configuration

###### Source Control Plugin Installation (SVN)

After Installing the eclipse, open the eclipse market

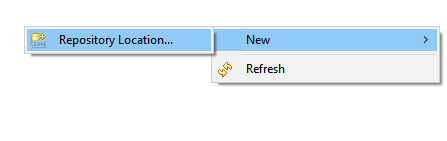


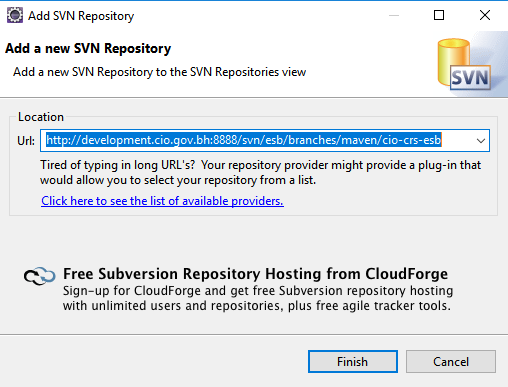
Search and Install SVN plugin



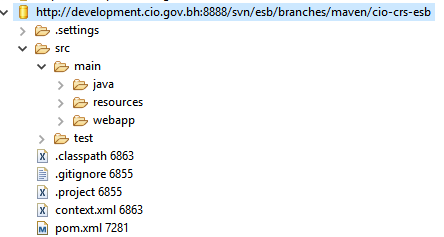
After restarting the workspace, choose SVN perspective and add the following source URL:

<http://development.cio.gov.bh:8888/svn/esb/branches/maven/cio-crs-esb>

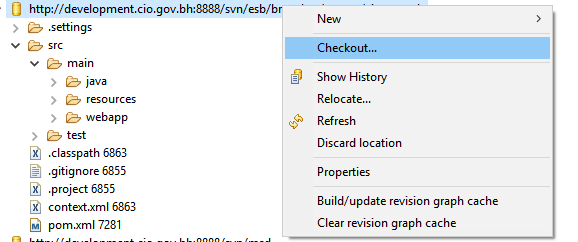


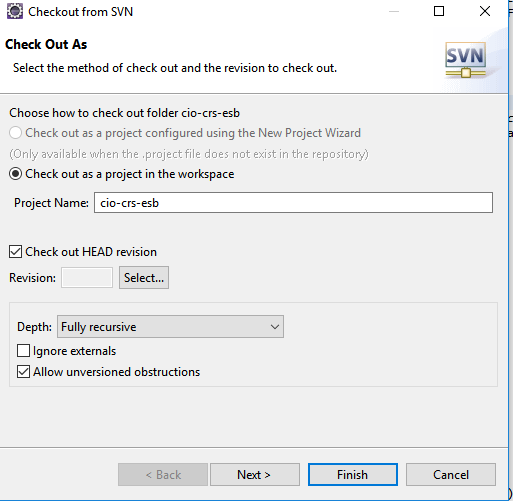


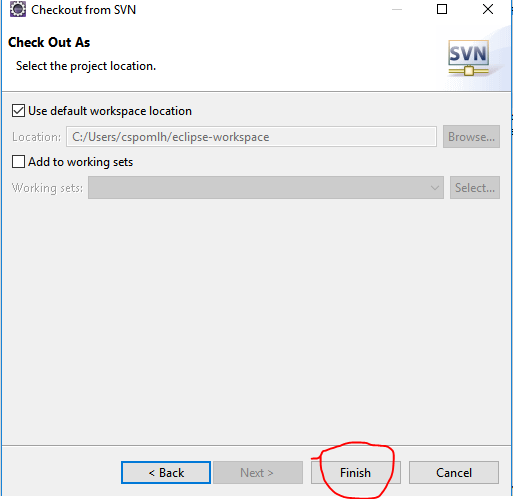
You may asked for your user/password for SVN that will sent to you in email.



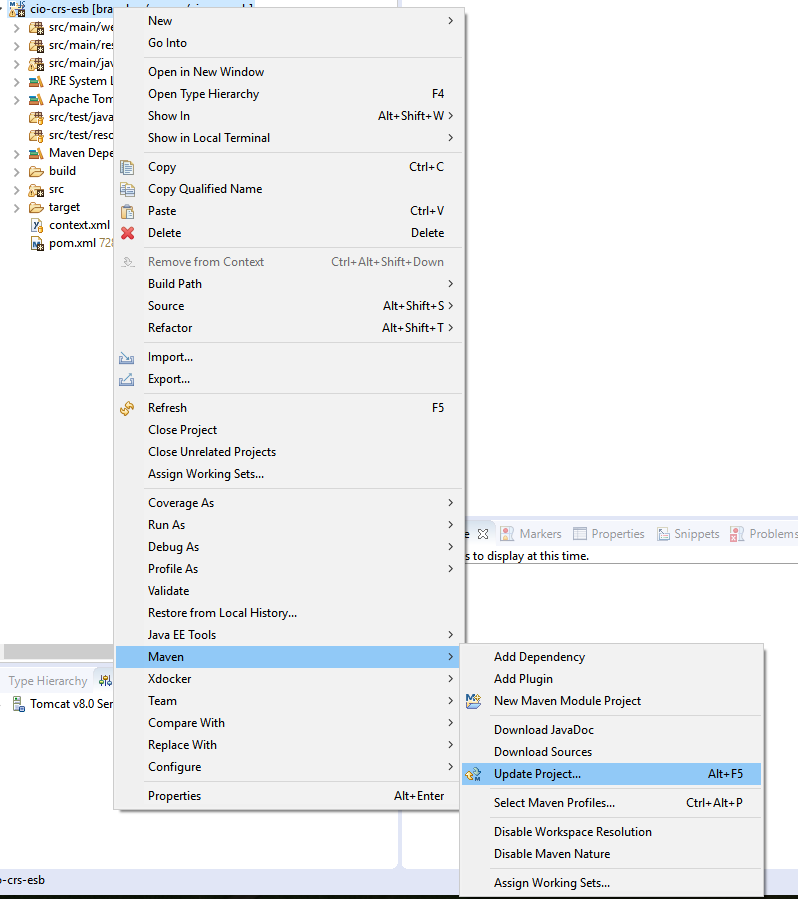
Now checkout the project as maven project







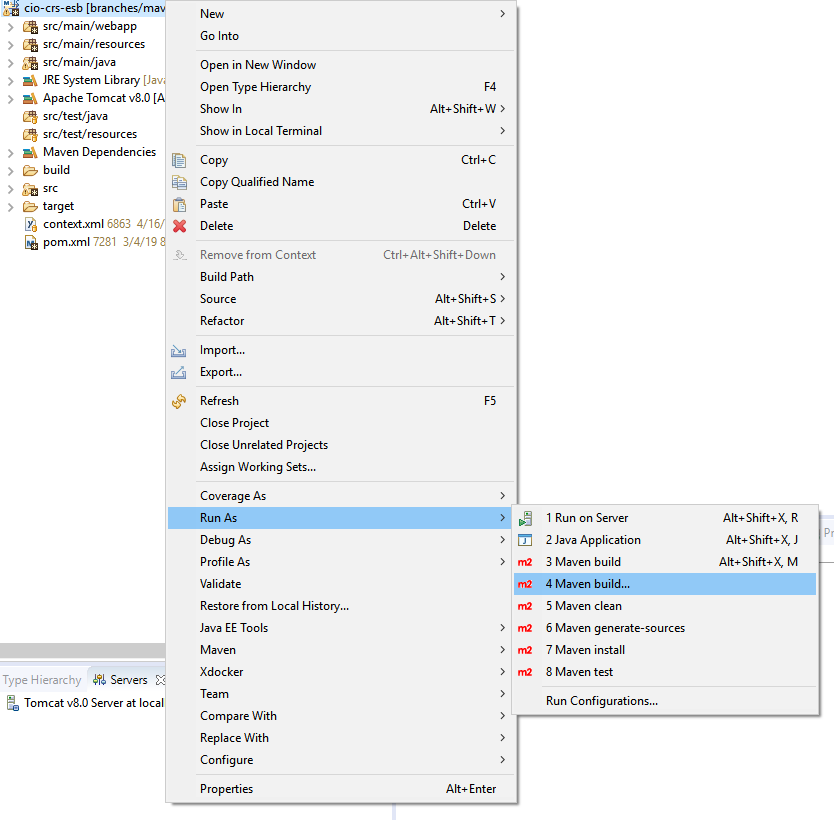
Return to the J2EE prespective and choose maven -> update Project



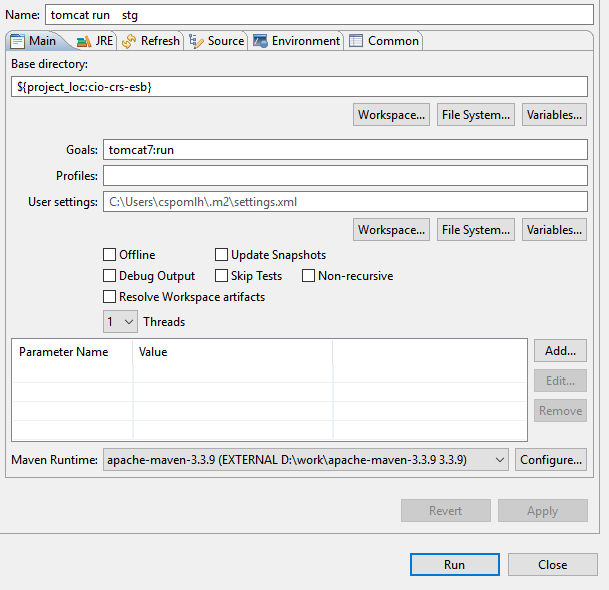
Running Local Server Accessing CRS Staging Services

To run tomcat 7 on java 8, we use tomcat7: run maven tomcat plugin, please follow the following steps:

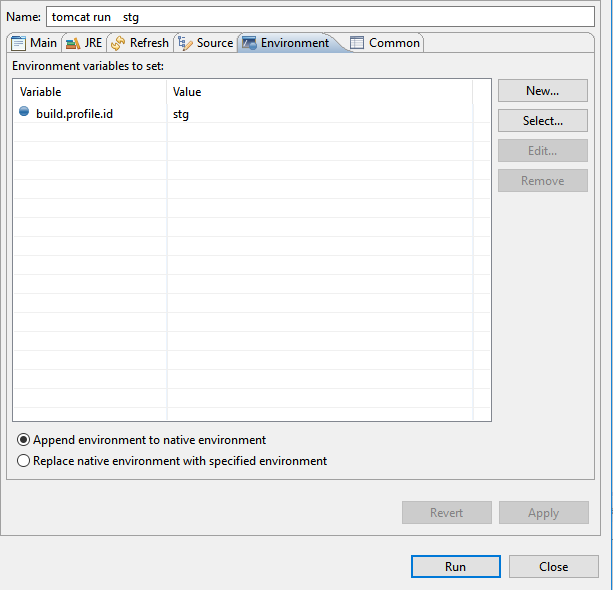
1. Choose run->maven build



1. Fill in the below Base Directory, GOAL field as in the picture



1. In the environment tab , add the variable build.profile.id with value “stg”

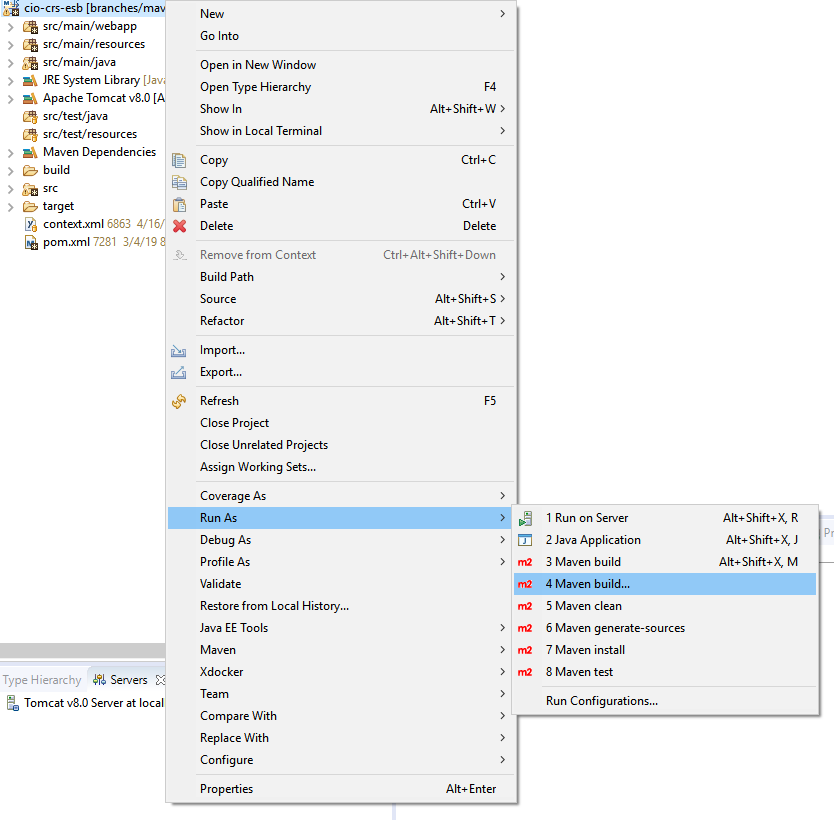


1. Press run button

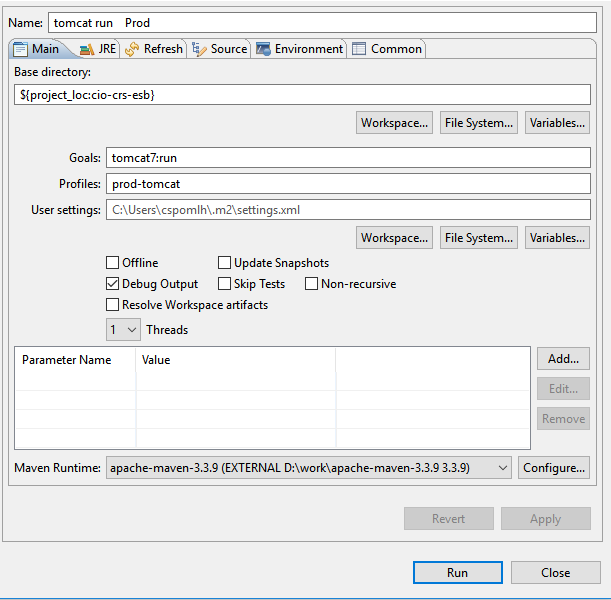
Running Local Server Accessing CRS Production Services

To run tomcat 7 on java 8, we use tomcat7: run maven tomcat plugin, please follow the following steps:

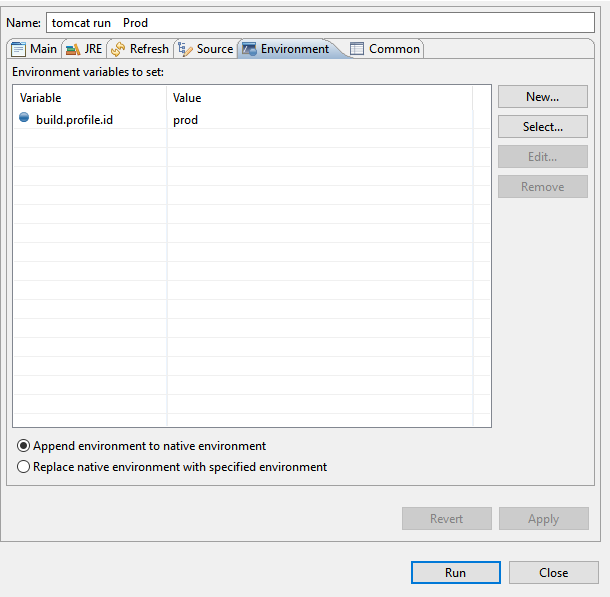
1. Choose run->maven build



1. Fill in the below Base Directory, GOAL,PROFILES field as in the picture

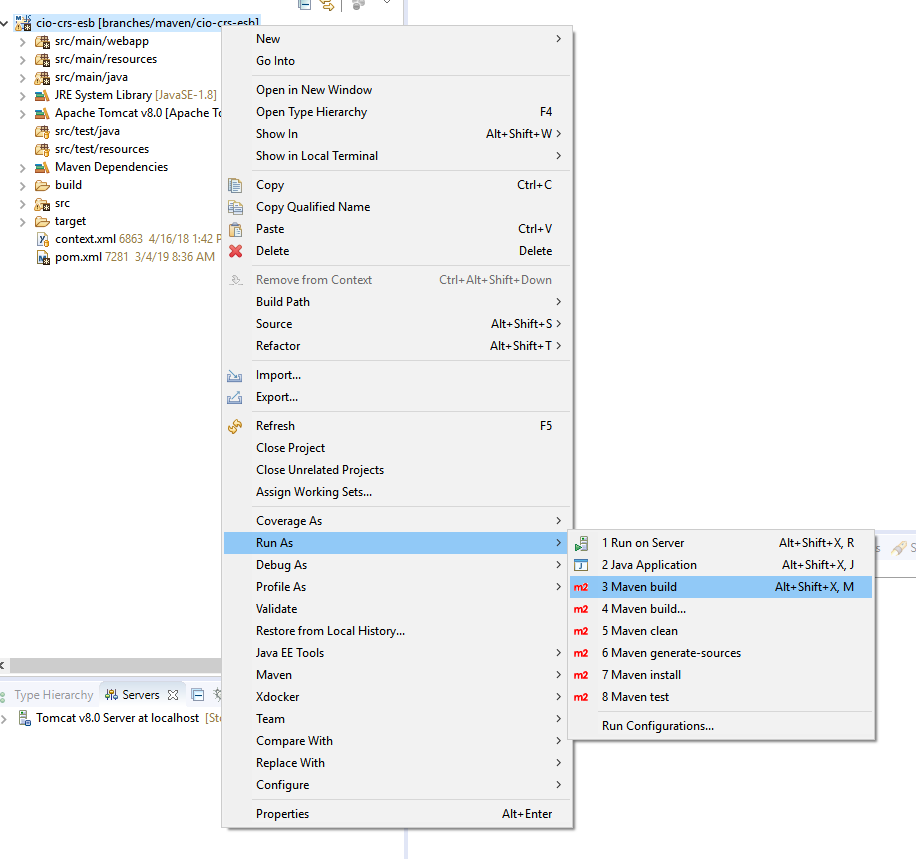


1. In the environment tab , add the variable build.profile.id with value “prod”

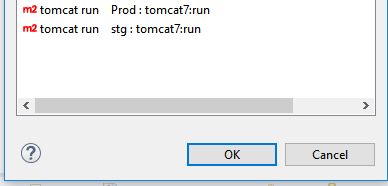


1. Press run button

NOTE: this is for running the server for the first time, in the second time you may choos the first build option as follows:



Then choose which one of preconfigured to run:



# Services Development

### New Service

Creating a new Service

1. **Create three packages for (implantation, interface and DTO)**
   1. Create package for service

Ex: bh.gov.cio.integration.crs.retrieve.person.biometric.service

* 1. Create interface

EX: PersonBiometricPhotoServiceInterface

* 1. Expose interface to be web service by using @WebService annotation and targetNamespace used in the generated WSDL @WebService(name="PersonBiometricPhotoService",targetNamespace= "http://service.biometric.person.retrieve.crs.integration.cio.gov.bh/")
  2. Specify the name of the result of the operation in the generated WSDL by using @WebResult annotation

@WebResult(name = "PhotoDetails")

* 1. Exposes the method to web service clients by using @ WebMethod

@WebMethod(operationName = "getPersonPhoto")

* 1. Specifies that input parameter name in the WSDL file by using @WebParam

PersonBiometricPhotoDTO getPersonPhoto(@WebParam(mode = WebParam.Mode.IN, name = "Security",header = true) SecurityTagObject security,

@WebParam(name = "cprNumber") @XmlElement(required = true) Integer cprNumber,

@WebParam(name = "blockNumber") @XmlElement(required = true) Integer blockNumber,

@WebParam(name = "cardExpiryDate") @XmlElement(required = true) Date cardExpiryDate) throws ApplicationExceptionInfo;

1. **Create package for implantation class**

This Class contains the implementation logic for the webservice

EX: bh.gov.cio.integration.crs.retrieve.person.biometric

* 1. Create class PersonBiometricPhotoServiceImp Implments interface PersonBiometricPhotoServiceInterface.
  2. Specify the Role and name of the service by using @Secured & @WebMethod

@Override

@Secured(

{ "ROLE\_getPersonPhoto" })

@WebMethod(operationName = "getPersonPhoto")

public PersonBiometricPhotoDTO getPersonPhoto(SecurityTagObject security, Integer cprNumber, Integer blockNumber, Date cardExpiryDate)

throws ApplicationExceptionInfo

* 1. Use the reference to CRS service

To get all the desired information according to the provided logic you start developing the service referencing CRS to return the webservice load

1. **Create package for DTO (Data Transfer Object )**

This class contains all the required return field for the webservice

EX: bh.gov.cio.integration.crs.retrieve.person.biometric.dto

* 1. Create class

Ex: PersonBiometricPhotoServiceDTO

declare variables for DTO and generate setter and getter.

Define the order in which the fields are written in the XML file

@XmlType(name = "PersonBiometricPhoto", propOrder =

{ "cprNumber", "personPhoto", "photoDate" })

1. **Add declaration of the service at exposed-soap-services-config.xml**

Ex

<!-- Biometric Photo Service -->

<jaxws:endpoint id="PersonBiometricPhotoService"implementor="bh.gov.cio.integration.crs.retrieve.person.biometric.PersonBiometricPhotoServiceImpl" endpointName="ns:PersonBiometricPhotoServiceWS" address="/PersonBiometricPhotoService.svc"

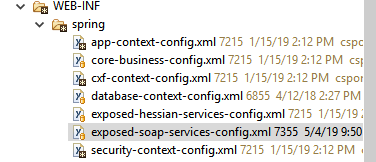
wsdlLocation="/WEB-INF/wsdl/PersonBiometricPhoto.wsdl"

xmlns:ns="http://service.biometric.person.retrieve.crs.integration.cio.gov.bh/" />

### Existing Services

###### Existing Services List

You can find the full list of existing exposed services if you open the file exposed-soap-services-config.xml

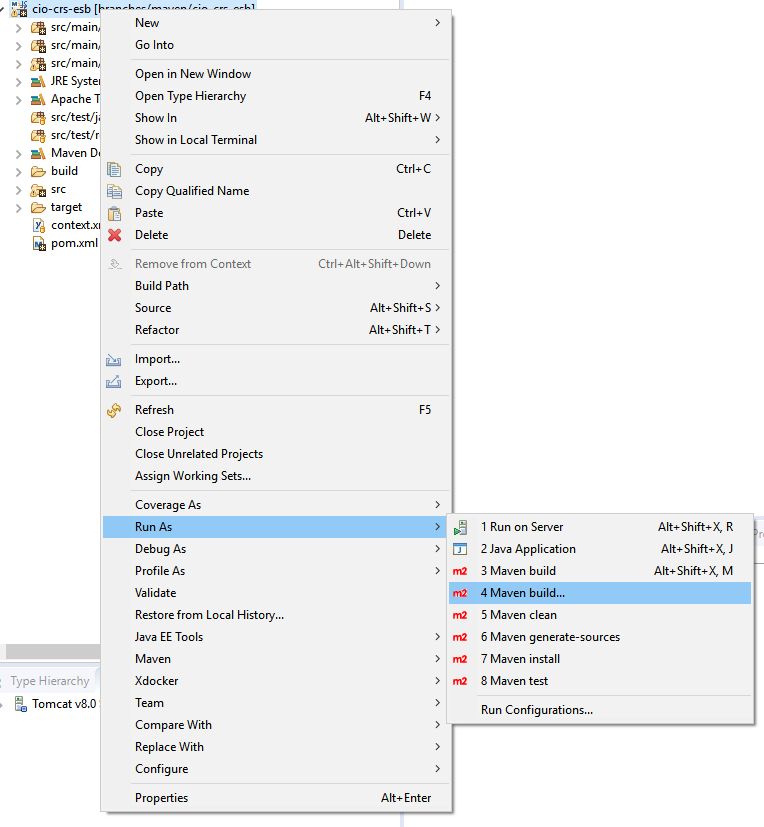


**Packaging And Deployment**

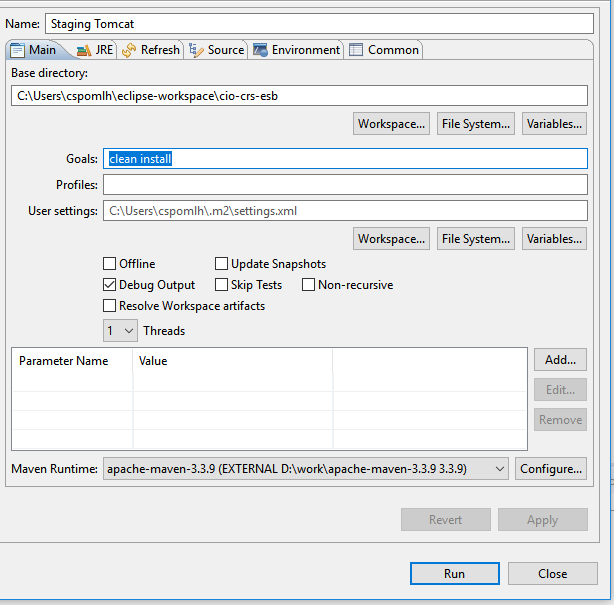
### Producing a staging package

Producing a staging package is done by running a maven command to generate a staging package as follows:

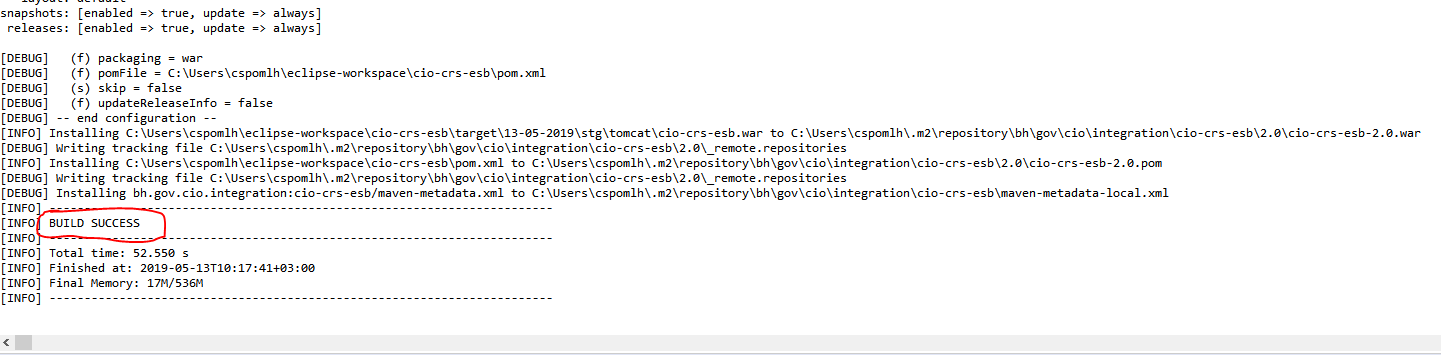
1. **Right click on the project and choose run -> build**



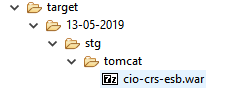
1. **In the next step set the GOAL Field value to “clean install”**



After run complete you should get Build Success Message in the console

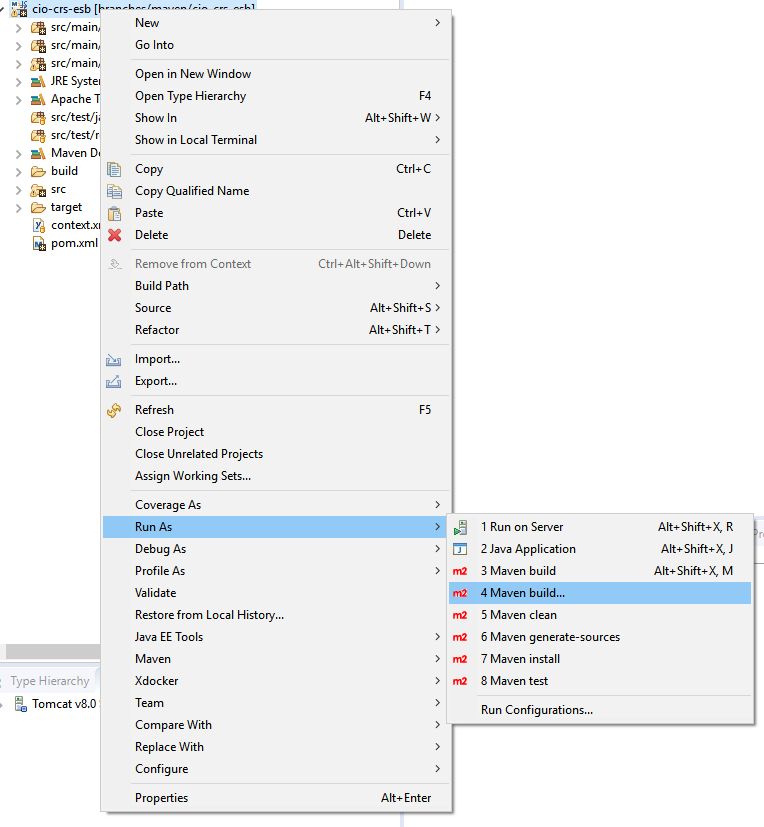


You will find the war package inside the target folder /date-of-todat/stg/tomcat/cio-crs-esb.war

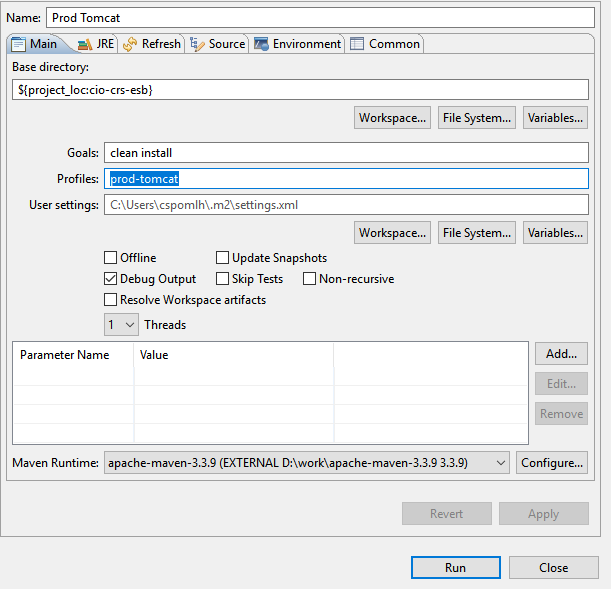


* 1. Producing a production package

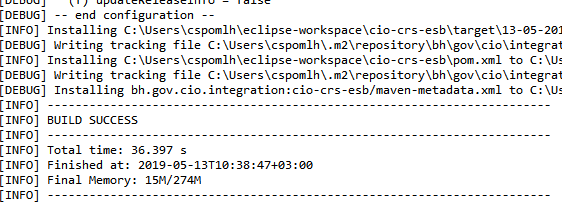
1. **Right click on the project and choose run -> build**



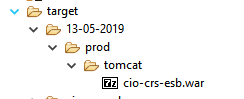
1. **In the next step set the GOAL Field value to “clean install” And put “prod-tomcat” in the profiles field**



After run complete you should get Build Success Message in the console



You will find the war package inside the target folder /date-of-todat/prod/tomcat/cio-crs-esb.war



1. **To enable debug mode :** <http://localhost:8080/cio-crs-esb/logbackAdmin.jsp?operation=changeLogLevel&logger=bh&newLogLevel=debug&logNameFilter=&logNameFilterType=>
2. **To place war file in shared folder for deployment**

* [**\\10.10.1.213\requests\**](file:///\\10.10.1.213\requests\)
* **User name : IGA**
* **Password : IGA1234**

1. **For production deployment raise remody request**
2. **For staging mail to “**iGA Middleware Support
3. **tomcat7:run – it will in the staging or production**
4. **clean install – clear the “target folder and ”build war” file**
5. 36888167 – hafiz
6. <http://stg-crsesb.gov.bh/cio-crs-esb/svc/EmploymentBasicDetailsService.svc?wsdl>
7. <http://crsesb.gov.bh/cio-crs-esb/svc/EmploymentBasicDetailsService.svc?wsdl>
8. <http://localhost:8080/cio-crs-esb/svc/IsSpouseService.svc?wsdl>
9. <http://localhost:8080/cio-crs-esb/logbackAdmin.jsp>