README

In the zip archive you'll find:

- README.pdf file.
- UK_Nightlife-report.pdf file containing the report of the developed project;
- UK_Nightlife-BPMN.bpm file containing the modeled BPMN of the project; It can be opened with Bizagi Modeler;
- UK_Nightlife-BPMN-workflow_net.pnml file containing the modeled workflow net of the BPMN representation of the project. It can be opened with WoPeD;
- UK_Nightlife-project folder containing the source code of the developed project by using OpenESB 2.3.1;
  In detail, the folder is structured as described below:
  - UkNightlife folder containing the BPEL implementation of the project and UkNightlifeCA folder that contains the relative composite application to be deployed;
  - GetCityByPostCodeProxy folder containing the BPEL implementation of the proxy service and GetCityByPostCodeCA folder that contains the associated composite application to be deployed;
  - GetCityByPostCodeLocal folder containing the BPEL implementation of the SOAP local dummy service and GetCityByPostCodeLocalCA folder that contains the relative composite application to be deployed;
- UK_Nightlife-workflow_net.pnml file containing the modeled workflow net of the project. It can be opened with WoPeD.

In order to run the project you have to follow the steps below:

1. Lunch OpenESB 2.3.x.
2. From the menu, File > Open Project.
3. Search the unzipped UK_Nightlife-project folder and select all the folders that it contains and click Open Project.
4. If a popup window appears informing about an error with Glassfish, ignore it and click yes.
5. In the Project tab right click and select Clean and Build for every projects starting from those without *CA in the name and subsequently to those with *CA in the name.
6. Right click and select Deploy for all the projects with *CA in the name.
7. Run the test cases located in the UkNightlifeCA > Test.
8. If the output window reports BUILD FAILED ignore it and check the test result.

Note:

- In order to run the timeout test the GetCityByPostCodeProxy has to be modified by introducing an artificial delay in the BPEL implementation that simulates the missing reply from proxy.
- The RESTful web service invoked require a free app_key that has been embedded in the BPEL implementation allowing to test the developed project in a transparent way.