PYMMA

OpenESB SE Enterprise Edition V3.0
Installation guide

PYMMA

Document identifier:

Pymma document: 770-001

Location:

www.pymma.com

Editor:

Pymma Services: contact@pymma.com

Abstract:

This document provides a short guide to install the OpenESB Standalone EditionV3.0.

Status:

This Document is in a beta state.

ABOUT PYMMA CONSULTING

Pymma Services is a technical architect bureau founded in 1999 and headquartered in London, United Kingdom . It provides expertise in service oriented integration systems design and implementation. Leader of OpenESB project, Pymma is recognised as one of the main actors in the integration landscape. It deeply invests in open source projects such as Drools rules engine. Pymma is a European company based in London with regional offices in France, Belgium and Canada. (contact@pymma.com or visit our website on www.pymma.com)

Copyright

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Pymma Services LTD. This manual is provided "as is" and Pymma is not responsible for disclaims all warranties of any kind with respect to third-party content, products, and services. Pymma will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Trademark Notice

Pymma is a registered of Pymma Engineering LTD. Java is registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Table of Contents

1 Introduction

OpenESB Standalone Edition is the latest release of OpenESB. Previous legacy versions relied on Glassfish Application Server and its architecture to run OpenESB in a JEE container. Efficient and reliable, this architecture is not accurate anymore with the new agile and scalable architectures. OpenESB legacy edition memory size and start up time are not really compatible with virtualisation and cloud. Something new, something faster had to be designed to highlight OpenESB incredible capabilities to design and implement service oriented projects within these architectures.
OpenESB Standalone Edition (OE SE) offers the lightest and the most efficient integration tools on the market. Powerful, prompt, scalable with a very low memory footprint, OE SE is ready for virtualisation and cloud deployment.

1.1 Compatibility with OpenESB legacy Edition

OpenESB Standalone Edition is compatible with OpenESB V2.3.x but the features linked with the application server such as JEE Service Engine and JNDI configurations are not supported anymore. An OpenESB 2.3 project without JEE Service engine features can run on OE V3.0 SE. In return, there is a backward compatibility between OE V 3.0 SE and OE Glassfish Edition 2.3.x as well.

1.2 OpenESB V3.0 Standalone Edition content

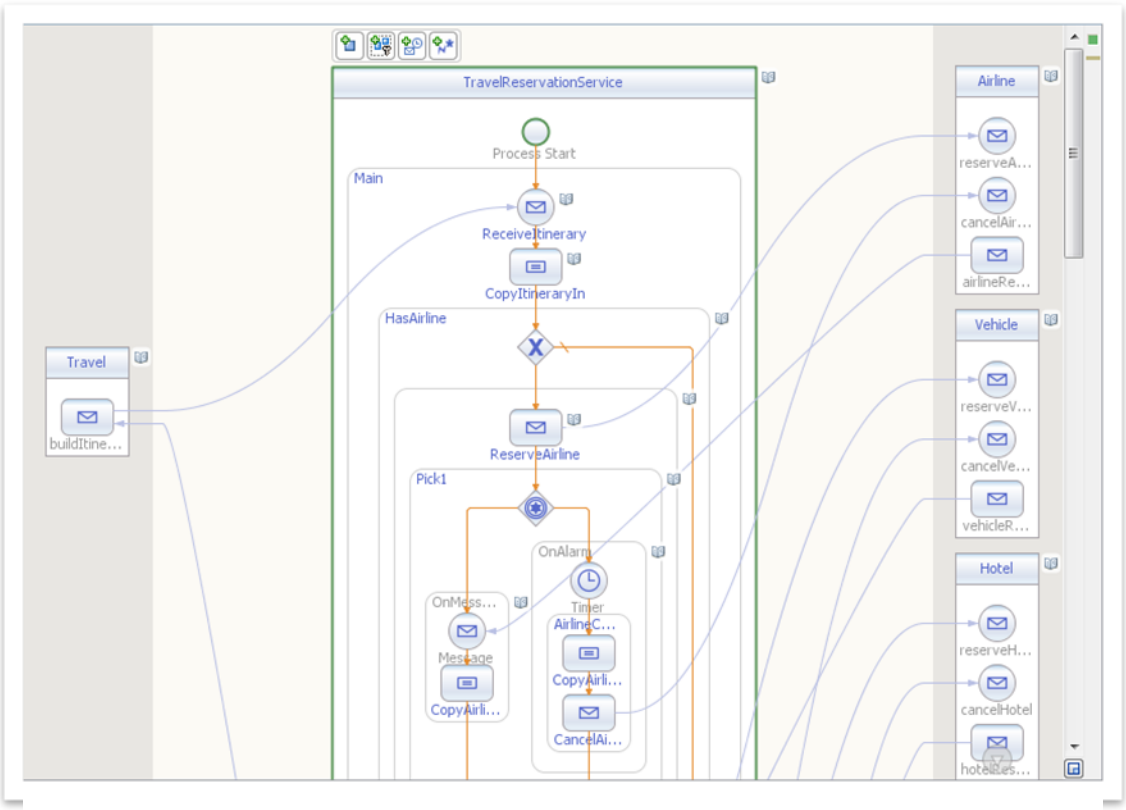
OE SE has been divided into 3 main parts.

- OpenESB studio
- OpenESB instance & Web Admin Console
- OpenESB components

1.2.1 OpenESB Studio

OpenESB studio V3.0 is an Integrated Development Environment based on Netbeans. It contains Netbeans regular Java and Java Enterprise tools used to develop application Java projects, as well as many plugins dedicated to OpenESB, designed to develop service oriented integration projects. The plugins deal with:

- XML document
- Schema XML documents
- WSDL document
- BPEL document
- Composite application
- Intelligent Event processes
- Human workflow
- And many more components.

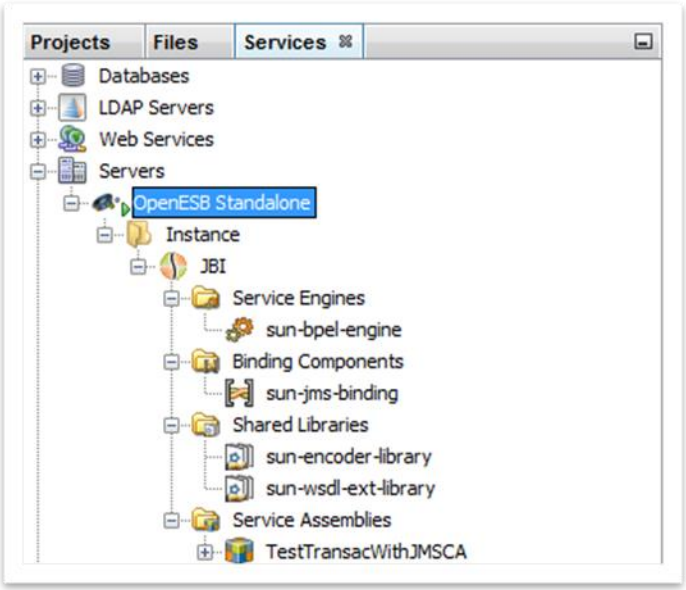


Example of OE plugin: OpenESB BPEL Editor

OpenESB is a powerful IDE which contains all the features required by development teams

- Powerful editors
- Mapper
- Debuggers
- Profilers
- Refactoring
- …

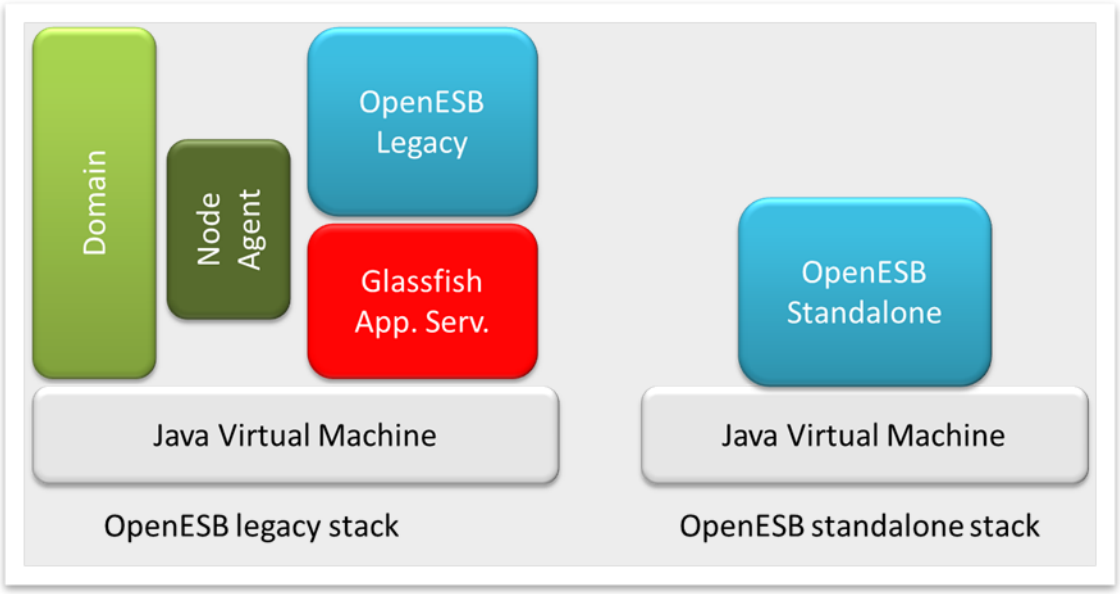
OE Studio could be connected directly with one or more instances of OpenESB Standalone Edition and allows the user to deploy, control and test her/his OE applications without any additional tools.



OE Studio Connection to OE SE instance

1.2.2 OpenESB Instance

OpenESB instance is the core part of OpenESB. It is the place where applications and components are deployed and run. Unlike OpenESB legacy edition, OpenESB instance does not require any additional softwares or containers to start and run in a simple JVM.

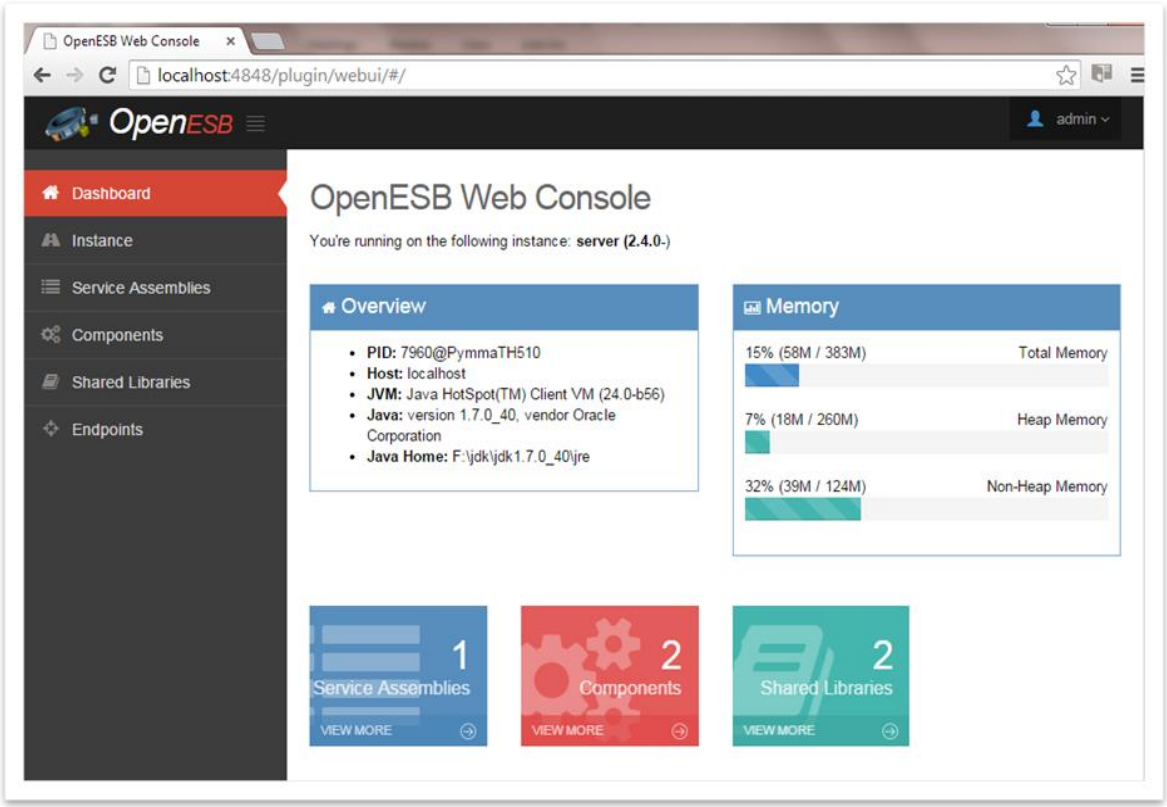


OpenESB legacy and standalone stacks

Administrative tools such as Domain and Node Agent are not required anymore. Even if the word "instance" is used to define an instance of OpenESB standalone, it shouldn't be mixed up with Glassfish instance used in the previous edition of OpenESB.
A huge effort has been done to reduce OE Core footprint (less than 100 Mo) and to improve its reliability and scalability. We implement a new monitoring API and console to monitor the instances) level as well.

1.2.3 OpenESB Web admin console

Until today, OpenESB administration was embedded in Glassfish admin console. Since Glassfish does not belong to OpenESB stack, a new console has been designed, developed and embedded with OpenESB code. Light and very smart, OpenESB web console offers the same administrative features that the legacy one. Today, you can easily install libraries, components, deploy services assemblies with your browser chrome, Firefox.



OpenESB Web admin console

More information on the admin console in our document **770-003: OpenESB Web Admin Console.**

1.2.4 Components

More than 35 components or libraries are provided with OpenESB SE. Many of them have been deeply modified and improved. New and more efficient libraries have been used to improve performance and reliability.
Unlike OpenESB legacy editions, OpenESB team decided not to pre-install the components and the libraries in the instance in order to improve memory footprint and start-up time. For more details on "How to deploy a component or a library" please have a look at our documents **770-002 OE SE Hello World, 770-004 OE Admin Guide**

1.3 Supported Operating systems

The table below lists the operating systems on which OE SE tests ran successfully. OpenESB community provides a community support for the OS below. Nevertheless "Enterprise Operating System" users are expecting a high level of reliability and a professional support for their development and production. We recommend "**Enterprise Users**" to choose "OpenESB Enterprise Edition" from Pymma for better reliability and professional support.

Operating System	Recommended
Ubuntu 14.4 Desktop 64	Community and Enterprise Edition
Ubuntu 14.4 LTS 64	Community and Enterprise Edition
Centos 6.x	Community and Enterprise Edition
Centos 7.x	Community and Enterprise Edition
Redhat Enterprise 6	Pymma Enterprise Edition
Redhat Enterprise 7	Pymma Enterprise Edition
Solaris 10-11-12	Pymma Enterprise Edition

Aix 6.x to 7.x	Pymma Enterprise Edition
Mac OS	Community and Enterprise Edition
Windows 7 32 and 64 bits	Community and Enterprise Edition
Windows 8 32 and 64 bits	Community and Enterprise Edition
Windows 8.1 32 and 64 bits	Community and Enterprise Edition
Windows server 2003	Pymma Enterprise Edition
Windows server 2008	Pymma Enterprise Edition
Windows server 2012	Pymma Enterprise Edition

1.4 JDK Supported version

OpenESB Standalone Edition runs on Java 1.7 32 or 64 bits. Small 32 bits configurations could be enough to run hundreds of thousands to millions of simple processes per day. Nevertheless, we strongly recommend you to run OE SE in a 64 bits mode for scalability purposes.
We recommend you to use the latest 1.7 Oracle JDK to run OE SE. OpenJDK 1.7 can be used but during the tests, we face some issue with Netbeans (15Q3). Note that Aix was tested successfully OpenESB SE with IBM JDK.

1.5 External software and applications supported

Through its connectors OpenESB communicates with many external applications and products. Please find below a short list of database and message queuing systems compatible with OpenESB. The list is not exhaustive; if you don't see your product on the list, it doesn't mean it's incompatible with OpenESB but rather that we haven't test it yet. Please contact us for further information on OpenESB compatibility.

1.5.1 SQL Databases

Editor	Version
Oracle	Oracle 9i
Oracle	Oracle 10
Oracle	Oracle 11
Oracle	MySQL 5.x
PostgreSQL	PostgreSQL 8.x
PostgreSQL	PostgreSQL 9.x
Apache	Derby
Oracle	Java DB
Sysbase	SQL Server
IBM	DB2 8.x, 9.x and 10.x

You can add to this list databases supporting JDBC specifications

1.5.2 Message Queuing Systems

Editor	Version
IBM	Websphere MQ
Oracle	OpenMQ
Apache	Active MQ

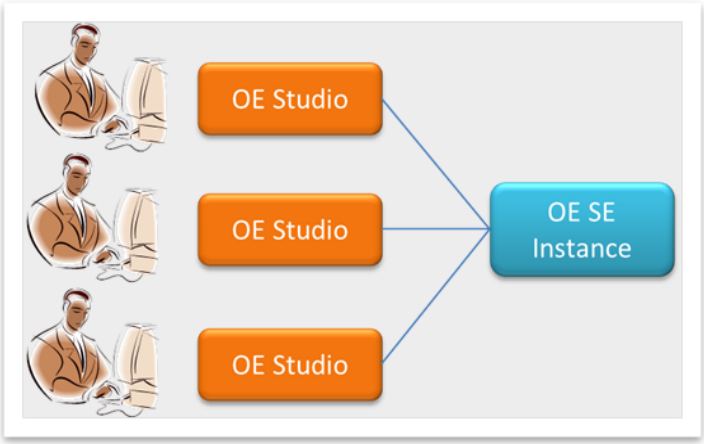
You can add to this list the products supporting JMS specifications

2 Hardware requirements

2.1 Hardware requirements at the design time

When designing, developers tend to run their IDE and OpenESB instances on the same machine. Sophisticated configurations allow them to share the same instances of OE SE and work together on larger units of development.

One OE Instance per developer



Share OE SE instance(s)

In both cases, machines used for the development must be set up with enough power and memory to work comfortably. OE Studio runs on any decent machine with 4 GB. However if you would like to improve your productivity we recommend you to use a 8 GB machine with 17 inches screen.
OpenESB requires at least 2 GB free space disk to be installed: 60% for the studio, 35% for the components and 5% for the instance. We recommend you to dedicate 10 GB to OE SE.

2.2 Hardware requirements at the runtime

OpenESB Standalone is a very light java process, which can run with limited resources; it can run on a simple Raspberry PI system designed for embedded applications for cars, robots and industrial machines. Nevertheless, OE SE has been designed to process billions of messages every day and the performance you will obtain is proportional to the hardware resource you dedicate to your projects. The feedback received from our customers who are using

OpenESB as infrastructures for their SOA projects, shows that OpenESB consumes very few CPU but requires enough memory to run properly. At the start, OpenESB takes up all the memory available on the machine and then manages it to process messages in the most efficient way. So if you dedicate X GB amount of memory to your OpenESB project, X GB will be used whatever the size of your project.
OE configurations with 2 GB are able to process millions of messages per day. We recommend you to set up your JVM memory between 2-4 GB for your first tests and QA. For the benchmark and production configuration, set up your JVM between 4 and 8 GB. With more memory, you can face issues due to lack of GC optimisation and long garbage collecting impacts. If you require more power from OpenESB or improvements in OE high availability, we advise you to create additional OpenESB instances and run your project in multi-instances configurations.
For further information, see our documents: **770-010 OpenESB optimisation** and **770-008 OpenESB Multi-instance mode.**

3 OpenESB Standalone Edition Installation

As the application server doesn't need to be installed anymore, OE SE installation is easier and faster than the previous legacy versions of OpenESB.

3.1 Install Java JDK

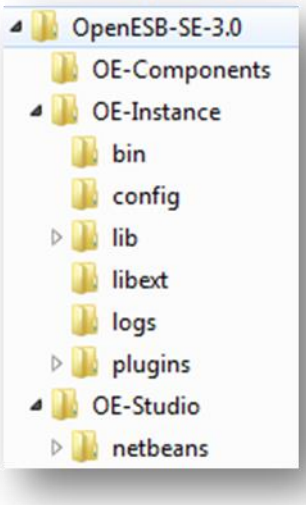
Install on your machine a JDK 1.7 or more (The most recent will be the best) then set up the environment variable JAVA_HOME.

3.2 Download OpenESB

OpenESB can be downloaded on OpenESB (www.open-esb.net) or Pymma (www.pymma.com) website.
OpenESB can be downloaded in two ways.
The first way is to download a simple and large zip file named **OpenESB-QuickStart-Standalone-x.x.zip** where x.x is the version number. This large file contains the three parts of OpenESB (studio, instance and components).
The second way is to download OpenESB parts one at a time. This way allows you to just change the part of your OpenESB that is required and upgrade the part you want. The zip files are **OpenESB-Components-Standalone-x.x.zip, OpenESB-Instance-Standalone-x.x.zip, OpenESB-Studio-Standalone-x.x.zip**
We recommend you to download first the quick start zip, then when you are more familiar with OpenESB to select the part you want to upgrade.

3.3 Install OpenESB

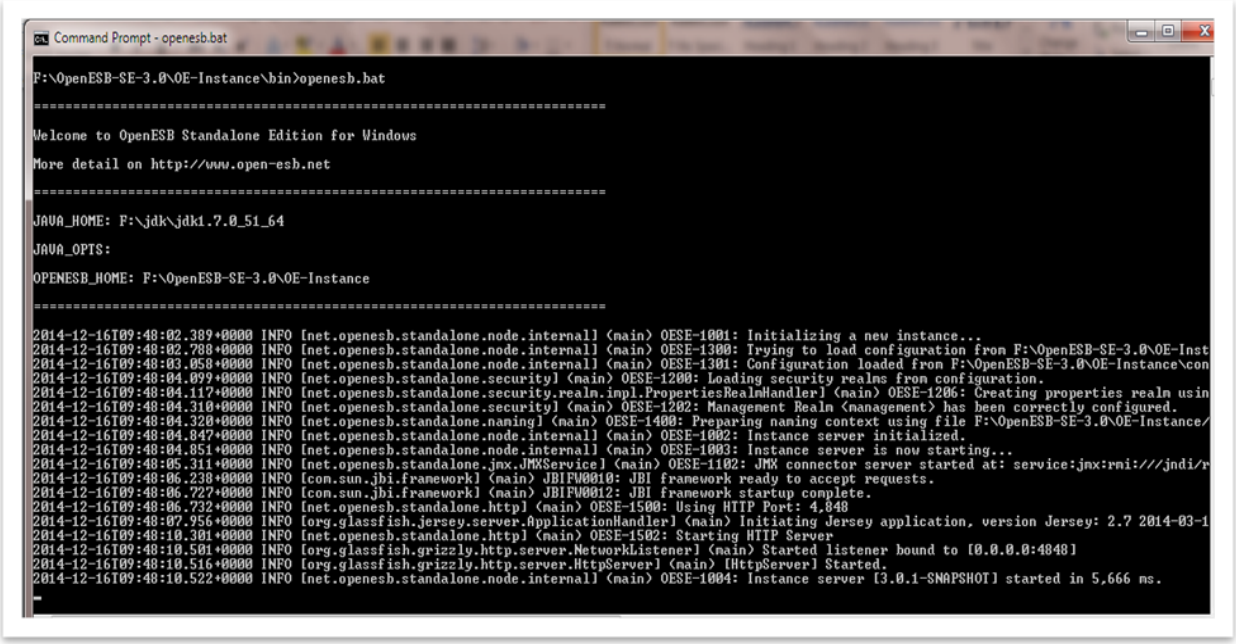
Unzip the OpenESB-QuickStart-Standalone-x.x.zip in a convenient place on your disk (in this document we choose "F:\" on a windows system).After unzipping the file, it contains 3 main subdirectories OE-Components, OE-Instance and OE-Studio. Congratulation! OpenESB is now installed on your machine.
Please note some noticeable directories:



…/OE-Instance/bin	Batch or shell files to start OE SE
…/OE-Studio/NetBeans	Complete Netbeans installation
…/OE-Studio/NetBeans/bin	File to start OE-studio
…/OE-Instance/libext	Copy your external libraries in this file and start OE SE. You libraries will be taken into account in OE SE class path
…/OE-Instance/logs	OE SE Logs can be found here

3.4 Test your installation

Open a console and go to the directory …/OE-Studio/bin and start OpenESB.bat on Windows or OpenESB.sh on Linux/Unix systems.



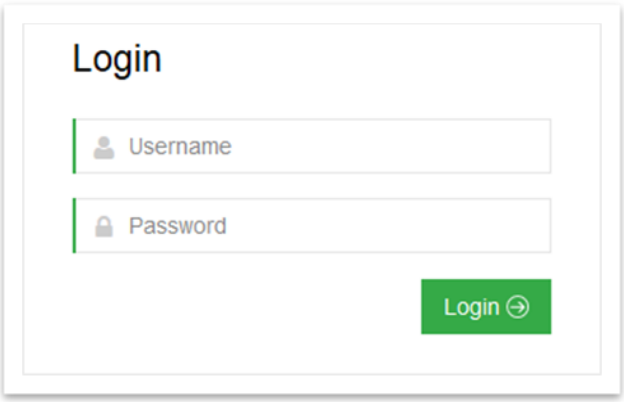
OpenESB SE starts in few seconds.

From now, OpenESB Standalone edition is ready to run.

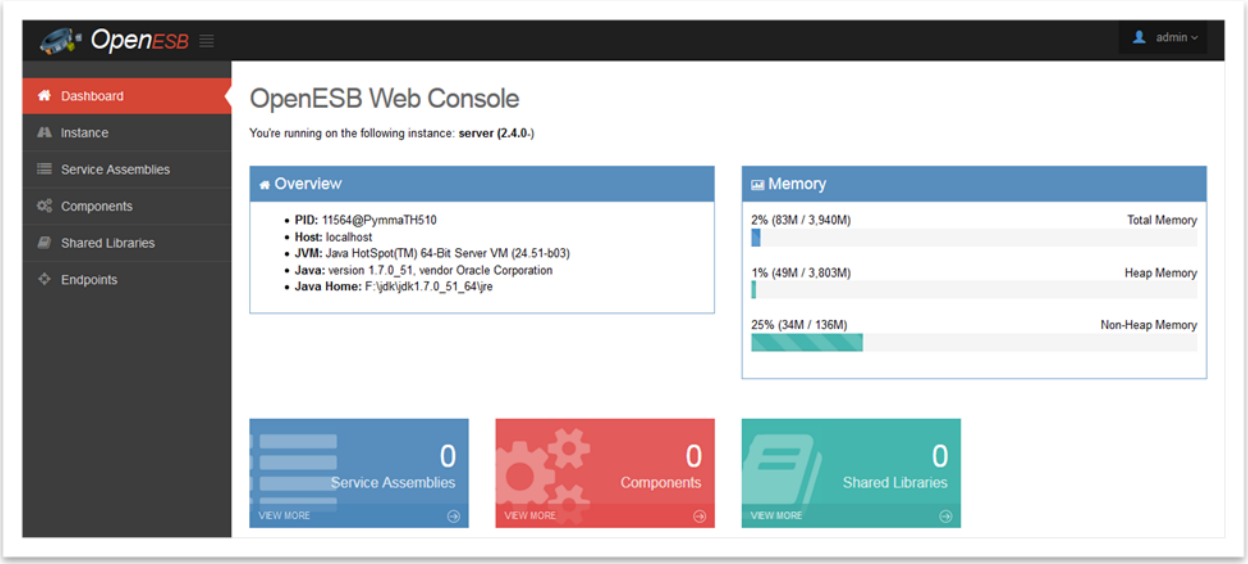
4 Test OpenESB Web admin console

The next step is to test OE Web admin console. As explained above, the web console replaces the Glassfish admin console and supports all the admin command to manage components, libraries and service assemblies.
OE SE has been tested with Chrome, Firefox and Safari browsers. Some trials have been successfully made with recent Internet Explorer versions, but we don't guarantee the same reliability with this browser.
In your browser type: **http://localhost:4848/webui/**

After a few seconds, the login screen appears:



By default the login and password are "admin" and "admin".

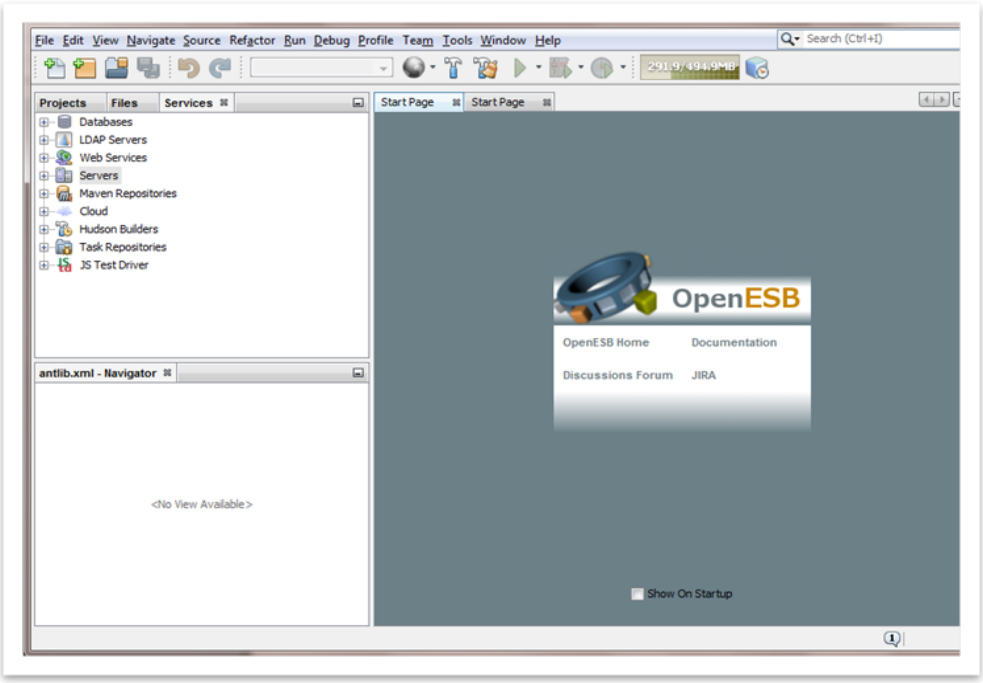


OE Web admin console

OE web admin console is ready and can be used to install the components and deploy your projects.
For more information on the console, please have a look at our document: **770-003 OE Web admin console.**

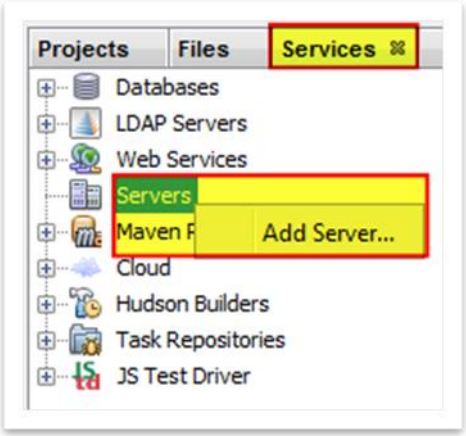
5 OE Studio

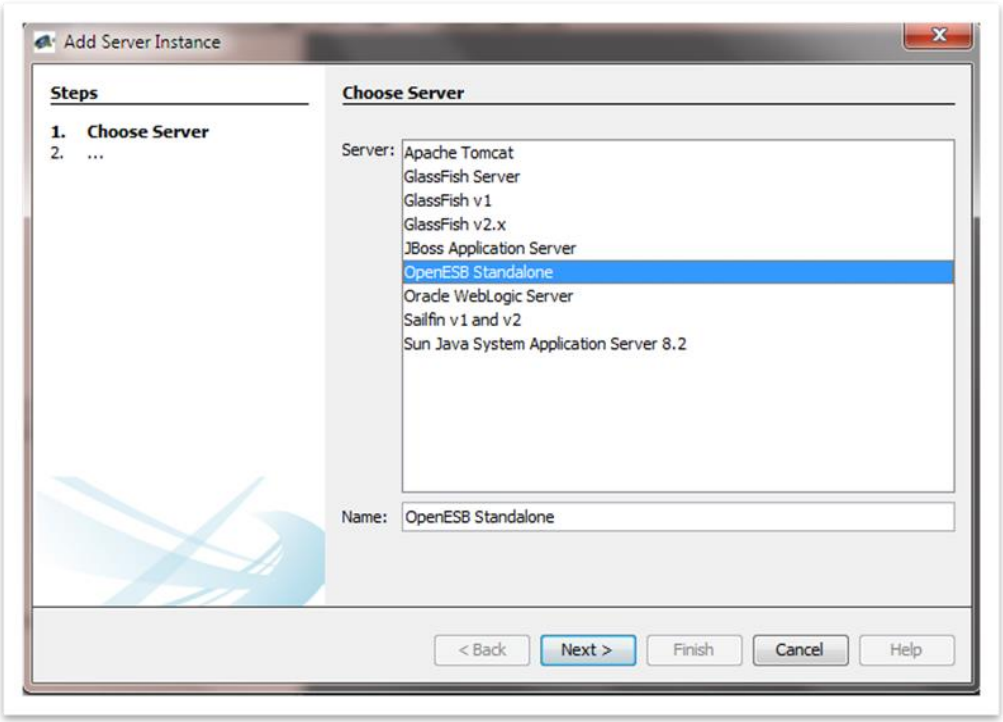
To start OE studio, run the executable file OpenESB.exe (for windows) or OpenESB (for Linux/Unix) found in the directory …\OpenESB-SE-3.0\OE-Studio\Netbeans\bin.
After a few seconds, the splash OpenESB 3.0 appears.
Regarding the number and the size of the projects you manage with OpenESB Studio, it needs from a few seconds to a few minutes to open.



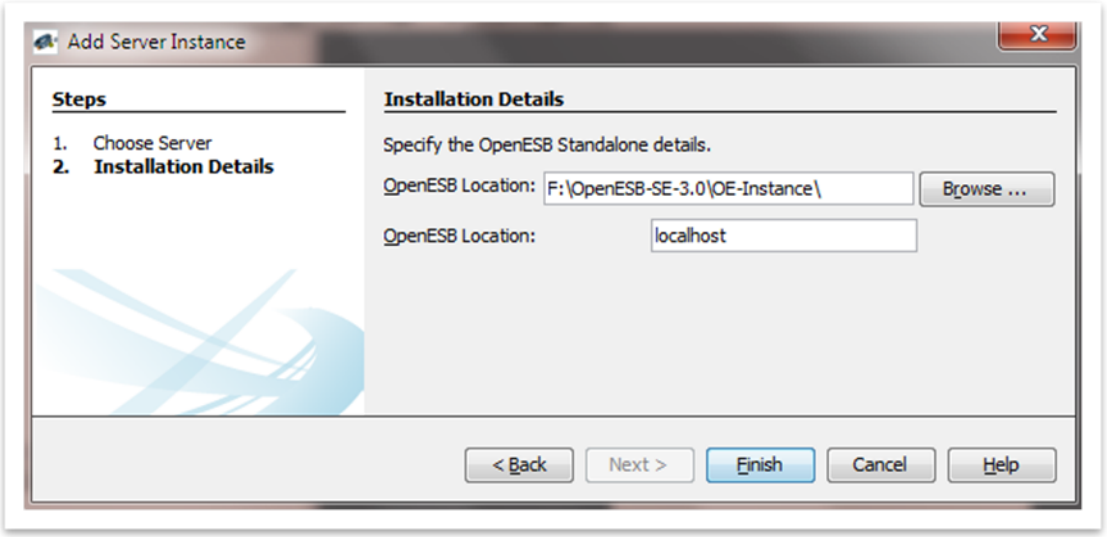
OE Studio home page

OE-Studio is ready to run. The first task to do is to connect the IDE with OE-Instance that was just started beforehand.
On the left side of the screen, select the tab services then Servers. Click right on Servers node and select Add Server.



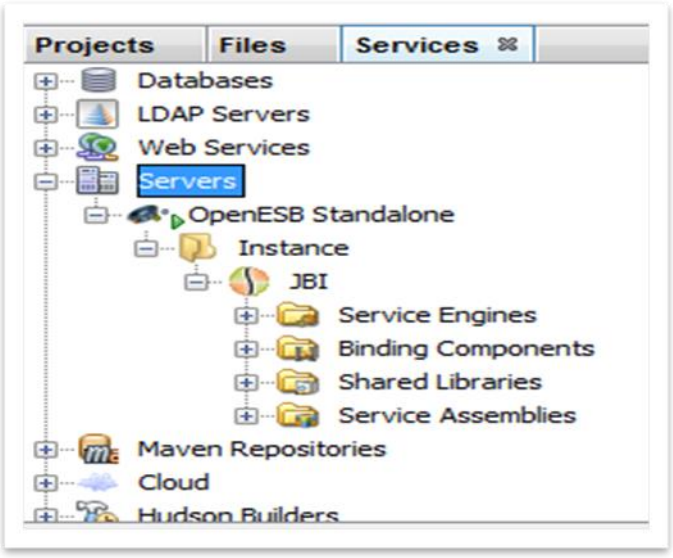


In the server list, select OpenESB Standalone then click on Next.



Installation Details

Select in the first OpenESB location the directory where you installed OE-Instance and leave "**localhost"** as value for the second location. Then click on Finish.



OE Instance hierarchy from OE Studio

OE Instance hierarchy is now available from OE-Studio. From there, you can install any components, deploy your service assemblies, start and stop them and debug Java and BPEL applications.
We recommend you to install the components from the web console since it offersmore administration capability than OE-Studio.

6 Next steps

The next step will be to install OpenESB components and Libraries on your OpenESB instance then create and deploy your project with OpenESB Studio.
We advise you to read the following document: **770-002 OE Hello World** and the administrative guide: **770-002 OE Administrative Guide.**

7 Help and support

7.1 From the community

You can find all our OpenESB documentations on the OpenESB official web site:
www.open-esb.net.
If you have any questions or would like to share your feedback, use the OpenESB forum at:
http://**openesb**-community-forum.794670.n2.**nabble**.com
Feel free to notify us with a bug or suggest how to improve our services on :
https://openesb.atlassian.net/secure/Dashboard.jspa

7.2 From Pymma

Pymma is deeply involved in the community and offers services and consulting on OpenESB. Pymma has professional services that can assist you from the development of your SOA design, implementation and ongoing management. All of our skills and background are based on our extensive first-hand experience and industry-leading methods.

Pymma releases an OpenESB Enterprise Edition with many additional enterprise features and a professional support.

In addition to OpenESB development, Pymma designed a new Service-Oriented development process named Rebecca to help business, architect and development team during the design and the implementation of their service oriented projects with OpenESB or any other service oriented development tool.

Feel free to contact us by email at contact@pymma.com for any further information on our OpenESB Services.