

# JBoss ESB 4.0 Beta 1

---

Trailblazer

JBESB-TB-9/22/06



## Legal Notices

The information contained in this documentation is subject to change without notice.

JBoss Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. JBoss Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Java™ and J2EE is a U.S. trademark of Sun Microsystems, Inc. Microsoft® and Windows NT® are registered trademarks of Microsoft Corporation. Oracle® is a registered U.S. trademark and Oracle9™, Oracle9 Server™ Oracle9 Enterprise Edition™ are trademarks of Oracle Corporation. Unix is used here as a generic term covering all versions of the UNIX® operating system. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

## Copyright

JBoss, Home of Professional Open Source Copyright 2006, JBoss Inc., and individual contributors as indicated by the @authors tag. All rights reserved.

See the copyright.txt in the distribution for a full listing of individual contributors. This copyrighted material is made available to anyone wishing to use, modify, copy, or redistribute it subject to the terms and conditions of the GNU General Public License, v. 2.0. This program is distributed in the hope that it will be useful, but WITHOUT A WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details. You should have received a copy of the GNU General Public License, v. 2.0 along with this distribution; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

## Software Version

JBoss ESB 4.0 Beta 1

## Restricted Rights Legend

Use, duplication, or disclosure is subject to restrictions as set forth in contract subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause 52.227-FAR14.

© Copyright 2006 JBoss Inc.

---

# Contents

---

# About This Guide

## What This Guide Contains

---

The **Error! Reference source not found.** contains information describing the Loan Broker Scenario deployed with the JBoss ESB 4.0 Beta 1. The information in this document is meant to help understand how to use the ESB in a “real-world” scenario not unlike what many developers and architects might face in their day to day project work.

**Note:**For the beta release, we recommend that you use this manual in conjunction with the [Programmers Guide](#), the [user forum](#) (<http://www.jboss.com/index.html?module=bb&op=viewforum&f=246>) and the javadocs associated with the code.

## Audience

---

This guide is most relevant to engineers who are responsible for using JBoss ESB 4.0 Beta 1 installations and want to know how it relates to SOA and ESB principles.

## Prerequisites

---

None.

## Organization

---

This guide contains the following chapters:

**Chapter 1, Scenario Overview:** What is the “problem” we are applying the JBoss ESB against.

**Chapter 2, How to Deploy:** A step-by-step how-to deploy the trailblazer and run it.

## Documentation Conventions

---

The following conventions are used in this guide:

|                            |   |
|----------------------------|---|
| <i>Italic</i>              | In paragraph text, italic identifies the titles of documents that are being referenced. When used in conjunction with the Code text described below, italics identify a variable that should be replaced by the user with an actual value.                                      |
| <b>Bold</b>                | Emphasizes items of particular importance.  |
| Code                       | Text that represents programming code.  |
| <b>Function   Function</b> | A path to a function or dialog box within an interface. For example, "Select File   Open." indicates that you should select the Open function from the File menu.   |
| ( ) and                    | <p>Parentheses enclose optional items in command syntax. The vertical bar separates syntax items in a list of choices. For example, any of the following three items can be entered in this syntax:</p> <pre>persistPolicy (Never   OnTimer   OnUpdate   NoMoreOftenThan)</pre> |
| <b>Note:</b>               | A note highlights important supplemental information.   |
| <b>Caution:</b>            | A caution highlights procedures or information that is necessary to avoid damage to equipment, damage to software, loss of data, or invalid test results.   |

n  
g Conventions

## Additional Documentation

---

In addition to this guide, the following guides are available in the JBoss ESB 4.0 Beta 1 documentation set:

1. JBoss ESB 4.0 Beta 1 *Programmer's Guide*: Provides guidance for writing applications.
2. JBoss ESB 4.0 Beta 1 *Getting Started Guide*: Provides a quick start reference to configuring and using the ESB.
3. JBoss ESB 4.0 Beta 1 *Configuring Hypersonic Guide*: This is necessary for setting up the Hypersonic database if you want to use it within the trailblazer.

## Contacting Us

---

Questions or comments about JBoss ESB 4.0 Beta 1 should be directed to our support team.

# Scenario Overview

## Overview

---

The purpose of an ESB, as part of an SOA strategy and architecture, is to be the central command and control through which disparate applications communicate, publish, and ultimately consume data to and from the enterprise systems in an organization. It is also ideally suited to allow your in-house systems to communicate with your partners systems.

Now that we created the JBoss ESB 4.0 Beta 1 release, we needed a scenario, or a problem domain to address. A tool is only as good as the problem it solves the saying goes. With that in mind, we needed some sort of scenario which would allow you to easily see the features and capabilities of the JBoss ESB, and how it could aid you in your enterprise with your particular needs. One such scenario has already been documented and prototypes by other ESB vendors, because it is such an obvious scenario that anyone can relate to. The scenario is the Loan Broker, as described in the EAI Patterns website. Why come up with a different scenario? If it's good enough for the EAI Patterns guys, it's good enough for us as well.

The scenario in summary is the following:

A Loan Broker system is responsible for collecting Loan Requests from Customers and then forwarding those Requests to Banks for Processing. Once the Banks have processed those Requests, Quote Responses, if any from each of the banks is sent back to the Customer for their review, and hopefully for the bank's sake, acceptance from the Customer.

The Loan Broker needs to collect the data required by each of the banks, log it, and then forward it to each of the banks using the communication protocol and data format that each bank expects to receive such data. Once the bank has decided to issue a Quote, the Loan Broker must receive the data, again in a unique format and communication protocol specified by each of the banks.

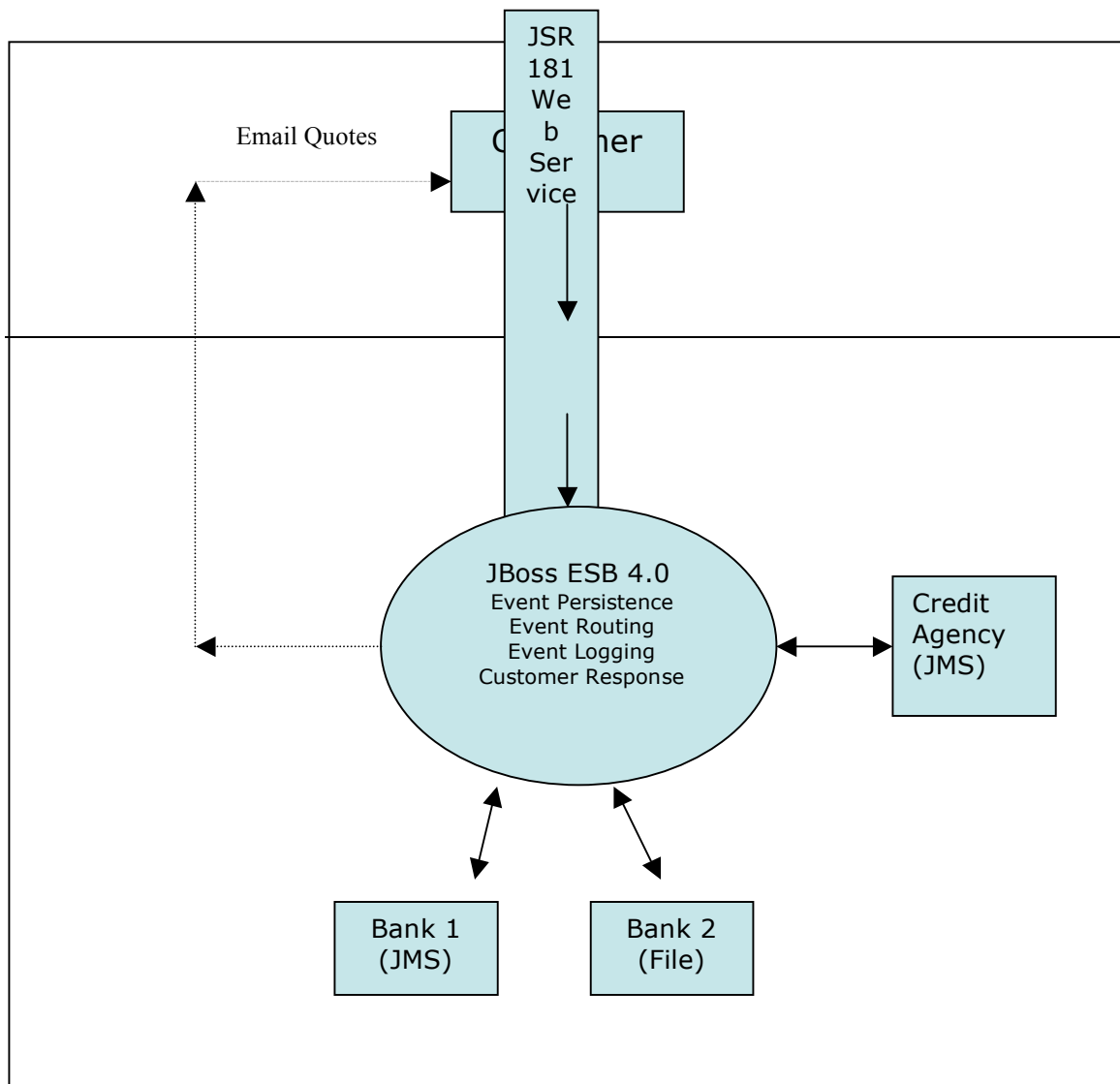
In a nutshell, the Loan Broker is required to: collect, transform, and ship data around to various systems; an ideal candidate for an ESB solution.

The JBoss ESB Beta 1 Trailblazer will have the following components:

- A JSr-181 Web Service endpoint – this is the “entry point” into the ESB
- a simulated set of Banks (2) which will receive and transmit data in 2 distinct format and communication channels.
- Communication protocols: JMS and File

- Data Formats: XML and flat file delimited
- For more detailed information on how the simulated Banks work, please look at the source code found in the source packages through SVN access. Remember please that the Banks are a simulation.

The diagram below shows the scenario provided in the Trailblazer:



# Scenario Overview

## Overview

---

The purpose of an ESB, as part of an SOA strategy and architecture, is to be the central command and control through which disparate applications communicate, publish, and ultimately consume data to and from the enterprise systems in an organization. It is also ideally suited to allow your in-house systems to communicate with your partners systems.

Now that we created the JBoss ESB 4.0 Beta 1 release, we needed a scenario, or a problem domain to address. A tool is only as good as the problem it solves the saying goes. With that in mind, we needed some sort of scenario which would allow you to easily see the features and capabilities of the JBoss ESB, and how it could aid you in your enterprise with your particular needs. One such scenario has already been documented and prototypes by other ESB vendors, because it is such an obvious scenario that anyone can relate to. The scenario is the Loan Broker, as described in the EAI Patterns website. Why come up with a different scenario? If it's good enough for the EAI Patterns guys, it's good enough for us as well.

The scenario in summary is the following:

A Loan Broker system is responsible for collecting Loan Requests from Customers and then forwarding those Requests to Banks for Processing. Once the Banks have processed those Requests, Quote Responses, if any from each of the banks is sent back to the Customer for their review, and hopefully for the bank's sake, acceptance from the Customer.

The Loan Broker needs to collect the data required by each of the banks, log it, and then forward it to each of the banks using the communication protocol and data format that each bank expects to receive such data. Once the bank has decided to issue a Quote, the Loan Broker must receive the data, again in a unique format and communication protocol specified by each of the banks.

In a nutshell, the Loan Broker is required to: collect, transform, and ship data around to various systems; an ideal candidate for an ESB solution.

The JBoss ESB Beta 1 Trailblazer will have the following components:

- A JSr-181 Web Service endpoint – this is the “entry point” into the ESB
- a simulated set of Banks (2) which will receive and transmit data in 2 distinct format and communication channels.
- Communication protocols: JMS and File



- Data Formats: XML and flat file delimited
- For more detailed information on how the simulated Banks work, please look at the source code found in the source packages through SVN access. Remember please that the Banks are a simulation.

The diagram below shows the scenario provided in the Trailblazer:

# How to Deploy

## Overview

---

Deploying and running the Trailblazer consists of the following steps:

1. Deploy the JBoss ESB 4.0 Beta 1 as per the Getting Started Guide
2. Configure the Loan Broker ESB Listeners and other property settings related to launching the Loan Broker process in your environment
3. Configure and deploy the simulated Bank application in your environment
4. Deploy the sample JSP page into your JBoss AS installation.

### *The configuration script*

Run ant with the configure target (`ant configure` in the `bankloanbrokerdemo` directory). Using the answers supplied, the script automatically configures the following files:

- `loanbroker.properties`
- `conf/LoanBrokerConfig.xml` (this is the biggest win)
- `banks/bank.properties`

Sample output is shown below.

```
Buildfile: build.xml

compile-antx:

configure:

    [input] This script will help you configure your local project for
    running the JBoss ESB LoanBroker Trailblazer. You can reconfigure the
    Trailblazer at any time by re-running this script.

    [input] Press enter to start...

    [input] Enter the host address of the JBoss Application Server on which
    your deployed JBoss ESB is running (for "localhost" simply press return):

    [input] Enter the path to the root of your local JBoss Application Server
    Installation (required for access to JBoss libraries):

    /tfennelly/home/jboss-4.0.4.GA

    [input] The following questions will configure the email notification
    parameters required by the Trailblazer.

    [input] Press enter to continue...

    [input] Enter the SMTP Host Name (for "localhost" simply press return):
```

smtp.gmail.com

[input] Enter the SMTP Host Port (for "25" simply press return):

[input] Enter the SMTP User Name of your user account on "smtp.gmail.com":

[tom.fennelly@gmail.com](mailto:tom.fennelly@gmail.com)

[input] Enter the password associated with your SMTP User account "[tom.fennelly@gmail.com](mailto:tom.fennelly@gmail.com)":

xxxxxxxxxx

[input] Is authentication of the SMTP User account "[tom.fennelly@gmail.com](mailto:tom.fennelly@gmail.com)" required by your SMTP host "smtp.gmail.com": (true,false)

true

[input] Enter the "From" address for email notifications from your user account "[tom.fennelly@gmail.com](mailto:tom.fennelly@gmail.com)" on "smtp.gmail.com":

[tom.fennelly@gmail.com](mailto:tom.fennelly@gmail.com)

filter-file:

[copy] Copying 1 file to /home/TFennelly/JBoss/ESB/product/docs/samples/trailblazer/bankloanbrokerdemo

filter-file:

[copy] Copying 1 file to /home/TFennelly/JBoss/ESB/product/docs/samples/trailblazer/bankloanbrokerdemo/conf

filter-file:

[copy] Copying 1 file to /home/TFennelly/JBoss/ESB/product/docs/samples/trailblazer/bankloanbrokerdemo/banks

[input] That's it, the Trailblazer is now configured! You can reconfigure the Trailblazer at any time by re-running this script.

[input] Press enter to finish...

BUILD SUCCESSFUL

Total time: 1 minute 56 seconds

## ***Deploying the JBoss ESB***

Please see the Getting Started Guide for more information on how to do this.

## **Loan Broker ESB-TrailBlazer Configuration**

Some general notes on settings in the cong/LoanBrokerConfig.xml file:

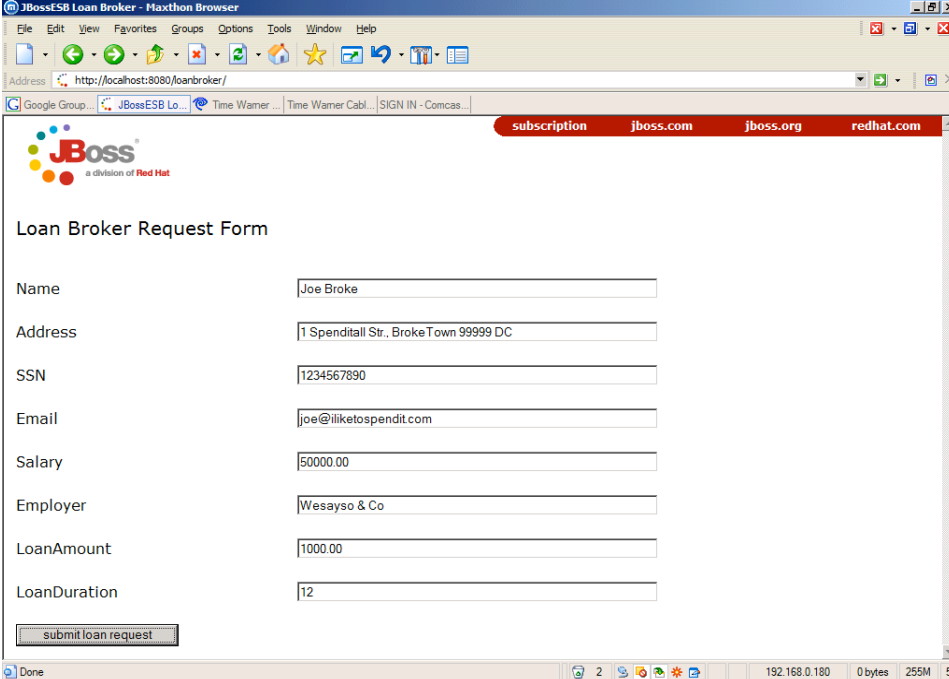
- Uses a JMS “message selector” property for each of the various listeners. Listeners will only accept JMS messages which have the correct message selector specified in its *listenMsgSelector* property
- Uses Queues A/B/C/D provided out of the box by the default JBoss AS installation
- *Notifications* take place at the App. Server, so the URI properties set must be visible and are relative to your JBoss AS server instance
- As per the Getting Started Guide, JBoss ESB 4.0 Beta 1 requires EJB3

## **Deploying the Trailblazer**

1. Deploy the sample web page and web service client end-point to your JBoss AS installation:
  - `<Trailblazer_HOME> ant deploy`
2. Run the LoanBroker Listener process:
  - `<Trailblazer_HOME> ant runListener`
3. Run the simulation Banks:
  - `<Trailblazer_HOME>/banks run “ant runFileBank”` and `“ant runJMSBank”`

## Execute the Loan Broker Web Application

Open a browser and navigate to <http://localhost:8080/loanbroker>.

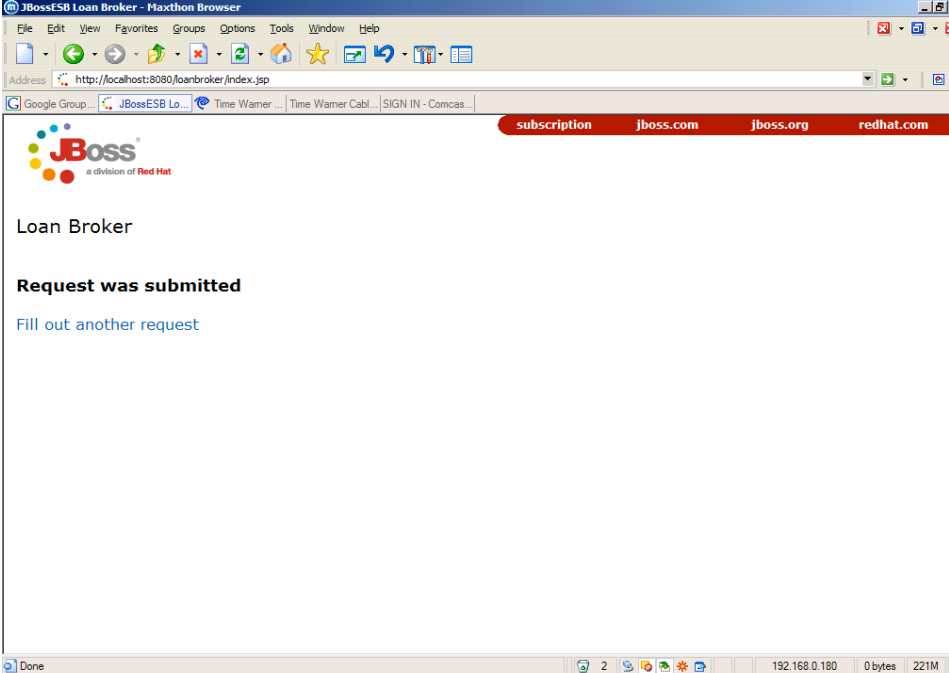


The screenshot shows a web browser window titled "JBossESB Loan Broker - Maxthon Browser". The address bar displays "http://localhost:8080/loanbroker/". The page features the JBoss logo (a division of Red Hat) and a navigation bar with links for "subscription", "jboss.com", "jboss.org", and "redhat.com". The main content is titled "Loan Broker Request Form" and contains several input fields:

|              |  |
|--------------|--|
| Name         | <input type="text" value="Joe Broke"/>                             |
| Address      | <input type="text" value="1 Spendital Str., Broke Town 99999 DC"/> |
| SSN          | <input type="text" value="1234567890"/>                            |
| Email        | <input type="text" value="joe@iliketospendit.com"/>                |
| Salary       | <input type="text" value="50000.00"/>                              |
| Employer     | <input type="text" value="Wesayso &amp; Co"/>                      |
| LoanAmount   | <input type="text" value="1000.00"/>                               |
| LoanDuration | <input type="text" value="12"/>                                    |

At the bottom of the form is a "submit loan request" button. The browser's status bar at the bottom shows "Done", "2" tabs, and the IP address "192.168.0.180".

Click "submit loan request". The browser will display



The screenshot shows the same web browser window after the form submission. The address bar now displays "http://localhost:8080/loanbroker/index.jsp". The page content is titled "Loan Broker" and displays a confirmation message:

**Request was submitted**  
[Fill out another request](#)

The browser's status bar at the bottom shows "Done", "2" tabs, and the IP address "192.168.0.180".

General notes on deploying and running the Trailblazer:

- Assumes you have JBoss AS deployed on the default port 8080 – if different, alter the url in step 4 above
- Assumes you have ANT/bin directory in your PATH