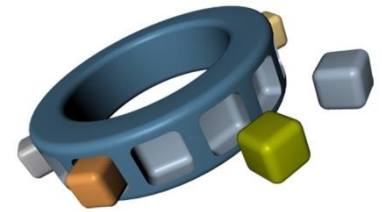


OpenESB Newsletter

Edition2, volume1 - May 2013



The best integration environment

Editorial

Last month the community issued OpenESB Version 2.3. That was a great event for us. Except for a bug due to a bad FTP transfer (we apologise) we made a nice announcement on Internet and your initial feedback was really positive. Thanks to you, we had several very nice weeks with hundreds download in few days. We are very happy to see the interest aroused by OpenESB. We were eager to read other comments. Feel free to contact us at contact@open-esb.net.



In this issue, we will talk about one interesting OpenESB project in France for a one of the top DIY companies in Europe Leroy-Merlin. This thousands of work days project has been developed in partnership INEAT and Pymma.

The tip and tricks this month explains the concept of atomicity in the BPEL engine.

We would like to thanks our sponsor this month [LogiCoy](http://www.logicoy.com) who takes a large part of the community effort to make "The best open source integration environment on the market".

Thank for our English reviewer Kevan Moran

OpenESB Version 2.3

Last month the community announced OpenESB version 2.3 with the following message

Today, OpenESB community (www.open-esb.net) is pleased to release the latest version (2.3) of OpenESB for Linux, Windows and Mac OS. This version is a major improvement to the project and also an important step for the community since it is the first version released by the community after Glassfish ESB 2.2 issued by Sun Microsystems 3 years ago. It marks the end of the consolidation step and the beginning of new extension staged on new platforms and architectures.

Thanks

We would like to thank our main sponsors and contributor Logicoy (www.logicoy.com) and Pymma (www.pymma.com) for the effort they made to reorganise and promote the OpenESB. Thanks to David, Fred, Nicolas, Paul, Prabhu, Vishnu, Param, Jay and Yoram for their involvement and dedicaton. Thanks to OpenESB's users who give us feedback and support and thanks for their patience. We would like to thanks the companies that support us and dedicated time to achieve this version: AT-Consulting , Integrated-Apps and Neva-Open.

Useful link

OpenESB Download and documentation at: <http://www.open-esb.net/>
Open-ESB forum and discussion: Forum Nabble: <http://openesb-users.794670.n2.nabble.com/>

LinkedIn Group: http://www.linkedin.com/groups?home=&gid=126145&trk=anet Ug_hm
Bugs report: <https://openesb.atlassian.net/secure/Dashboard.jspx>

OpenESB V2.3 Release note

- More than one hundreds bugs fixed on OpenESB Core, Components and IDE
- Support Netbeans 7.2.1 platform
- Add Netbeans plugins and components for Intelligent Event Process application
- Add Netbeans plugins and components for Work List Management
- Add Netbeans plugins and components for ETL (Extract, Transform and Load)
- Add Netbeans plugins and component for HL7

OpenESB use case: Leroy Merlin

As a major actor on the worldwide DIY market, Leroy Merlin (www.leroymerlin.com) helps residents and homeowners with their home-improvement projects. It specialises in sales of products and solutions and, in doing so, makes a unique commitment: to provide home improvement solutions tailored to each customer's specific needs. Leroy Merlin is present in 13 countries around the world. INEAT Conseil (www.ineat-conseil.com) is the main contributor for the project. Pymma (www.pymma.com) provided Leroy Merlin with consulting on architecture and training at the beginning of the project. This use case has been written by Simon Steinmetz from Ineat.



Several years ago hampered by point-to-point connections between its website and its back office, Leroy Merlin faced maintenance and evolution issues when business processes changed. Each evolution (new feature, new service) was planned with a long “time to market” due to low reuse rate during design and development.

Leroy Merlin decided to review its website architecture and set up a services bus (OpenESB) as the communication channel between the web server and its back office. Leroy Merlin leverages OpenESB for access and orchestration of back office services (client repository, loyalty, order management, inventory management and project management) and external services (Address recovery management planning...). After few months of work, the project was deployed in production. The first feedback showed that services and orchestration reduced Leroy Merlin's time to market and dramatically increased development agility. OpenESB and its “development process” improved Services Oriented Development and integration in the company.

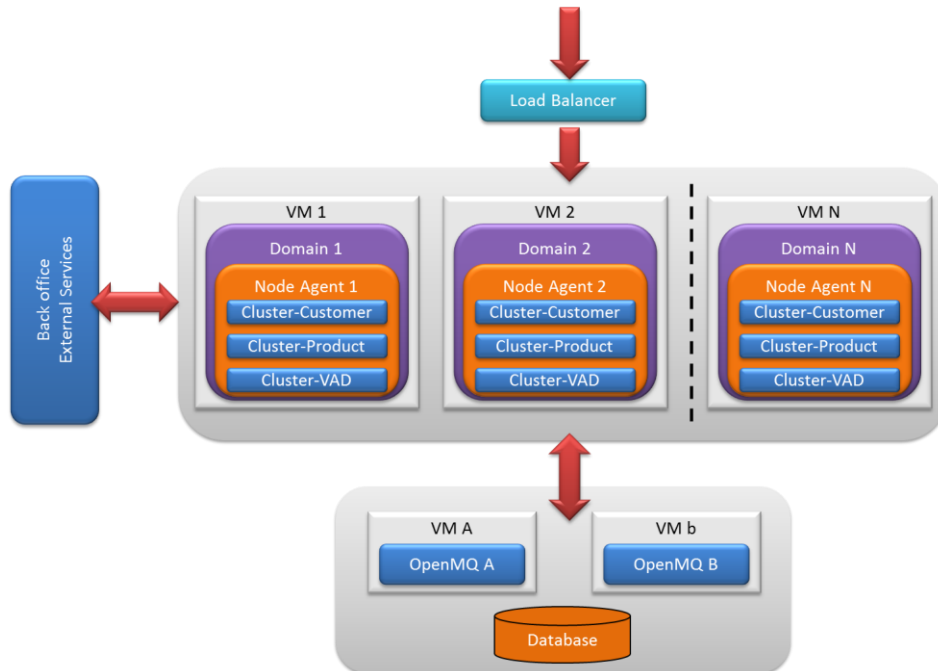
At the technical level:

Technical teams created three clusters across many machines (Customers, Product and VAD for “Vente A Distance” in French). A load balancer stands in front and spreads inbound messages. BPEL engine does not use persistence for recovery mainly because the processes were not critical enough. Development teams use compensation and error management to recover information and context in case of failure. An external Open MQ cluster has been set for asynchronous purposes. The project has been developed with OpenESB v2.2. A migration to OpenESB v2.3 on Glassfish is planned for this year (2013).

Today more than thirty composite applications and 150 BPEL processes run in production. Despite a heavy load on the website, the ESB platform performs well and proves its scalability. Every month, Support teams deploy new services on OpenESB without (major) issues. Following this project, Leroy Merlin decided to extend OpenESB's scope to other projects (product publication, inventory, management, e-commerce...)

Ineat developed a nice and efficient supervision system for OpenESB and Open MQ brokers. It interfaces OpenESB with Nagios/Opsview through Jolokia.

The picture below describes the global architecture at Leroy Merlin



OpenESB resources

Pymma (www.pymma.com) wrote two interesting papers on OpenESB. The first dedicated to IT architect and management is a comparison between Mule ESB and OpenESB views on integration. The second paper explains how to improve and build OpenESB components with Netbeans.

Hubert is teaching at the University of Denmark. He recorded videos on OpenESB for his students on Youtube. Links and references below. Thanks Hubert ☺

Mule vs. OpenESB: http://pymma.com/index.php?option=com_easyblog&view=entry&id=5&Itemid=678

Improve and build components: http://pymma.com/index.php?option=com_content&view=article&id=137:improve-and-build-opensb-components&catid=79:opensb-technical-paper&Itemid=572

Hubert's videos:

http://www.open-esb.net/index.php?option=com_content&view=category&layout=blog&id=84&Itemid=472

Previous newsletters:

http://www.open-esb.net/index.php?option=com_content&view=category&id=90&Itemid=589



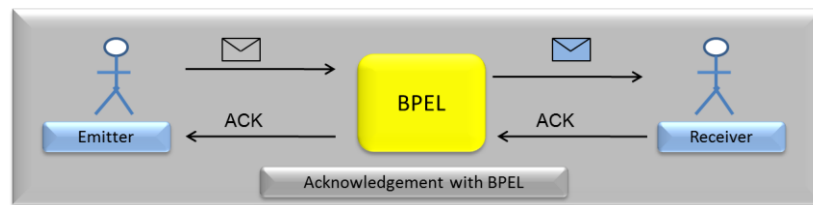
OpenESB tips and tricks: BPEL Atomic or not.

OpenESB BPEL engine proposes an interesting feature named "Atomic". In this "Tip and Trick" we will touch on what Atomic means.

OpenESB relies on JBI specifications defined by Sun Microsystems. JBI forces a receiver to send back an acknowledgment to the emitter when he receives a message.

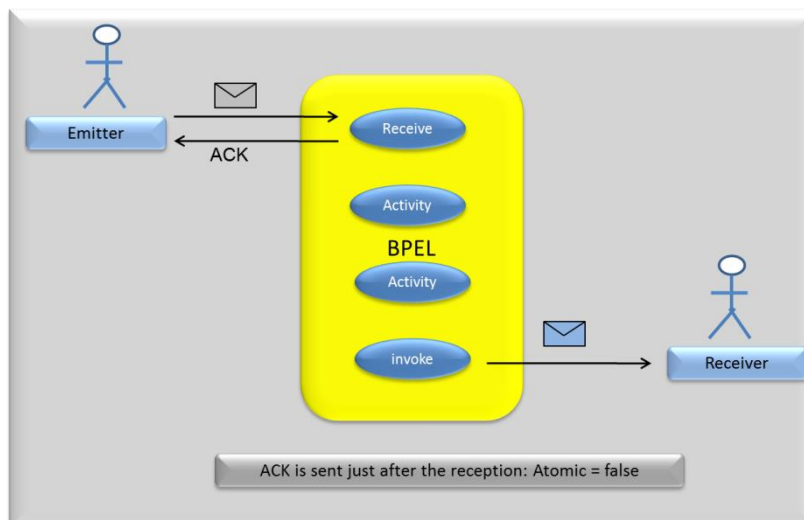


The figure below describes how emitter and receiver exchange messages and acknowledgments on a JBI platform.

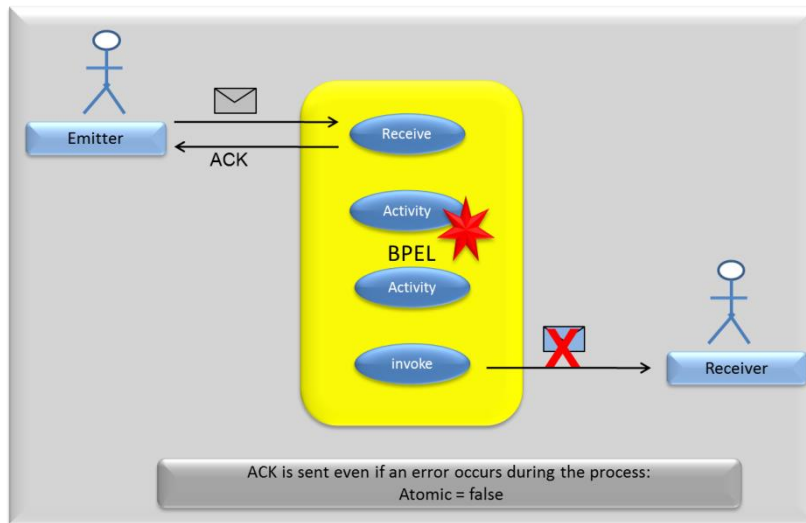


The acknowledgment is an important feature for Quality of Service (QOS). The emitter is responsible for the inbound message until he receives an ACK. Then the receiver has the outbound message responsibility.

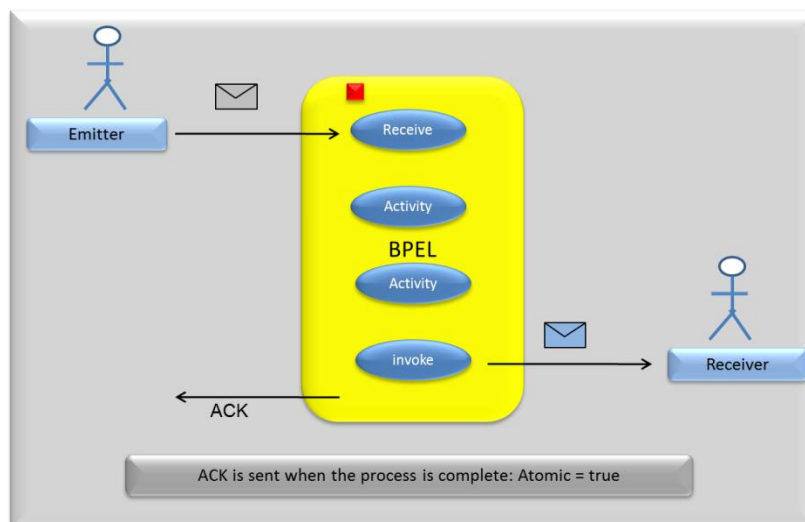
So we understand that when a BPEL receives a message it has to send back an acknowledgment to the emitter. The JBI specification is not precise about when the ACK must be issued? Does the BPEL return an acknowledgment just after receiving an inbound message or after completing the process? OpenESB BPEL property "Atomic" defines the moment when the BPEL returns an ACK. If "Atomic=false" OpenESB BPEL returns an acknowledgment after receiving a message. If "Atomic=true", the BPEL sends back an acknowledgement when the process is over. In either case, if an error occurs, no ACK is sent back.



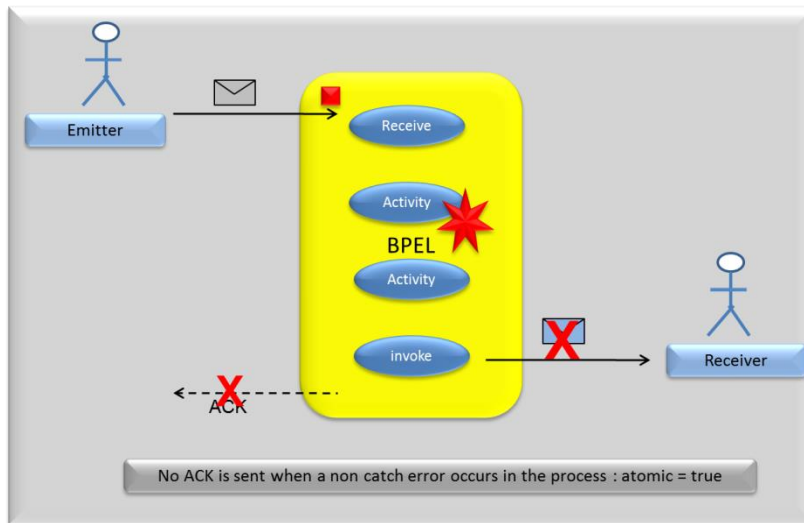
Case 1: Atomic=false + process OK. The BPEL engine receives a message and immediately returns an ACK. When the emitter receives an ACK, it frees the resources linked to the inbound message which becomes unavailable. Once the process is complete, the receiver gets the outbound message.



Case 2: Atomic=false + process KO. The BPEL engine receives a message and immediately returns an ACK. When the emitter receives the ACK, it frees the resources linked to the inbound message. After a logical or technical error, the process stops and loses the message. Since the emitter can no longer retrieve or resend the inbound message and the receiver does not yet get the outbound message it is lost.



Case 3: Atomic=true + process OK. The BPEL engine receives an inbound message and waits for the process to end before sending back an ACK. The receiver gets the outbound then the emitter receives the ACK and frees the resources linked to the inbound message.



Case 4: Atomic=true + process KO. The BPEL engine receives an inbound message and waits for a successful process end to return an ACK. An error occurs during the process. The Receiver does not receive the outbound message and the Emitter does not receive an ACK but since the Emitter can retrieve the inbound message, it can replay the process.

Conclusion

So when do we set Atomic to false or true? The response is more depending on your business case than on a technical requirement. The right answer requires analysis by an OpenESB architect. However, we can suggest a rule of thumb for the atomic properties. With atomic=false, we get better performance since a prompt acknowledgement release resources used to keep the message available at the emitter level. However the message can be lost if an error occurs during the process. If the inbound message is critical, it cannot be lost during the process. Setting Atomic to true is a good option.

OpenESB partners

INEAT Conseil is a consulting services company specialising in Enterprise Information Management. With the objective of optimising the management and use of corporate information, INEAT consultants guide the support, IT and business departments in their information needs. The expertise is focused on dematerialisation projects, document management and archiving projects, implementation of collaborative tools, portals, workflow and search engines and mobile. A fast-growing company with 120 employees, INEAT has 3 main locations, in the Paris region, in Northern France (Lille), and in Brussels. Its partnership with Alfresco and Nuxeo and Pymma aligns with an approach of growing its functional positioning.



Our sponsor:

Logicoy incorporated in February 2009 with headquarters in Los Angeles and 24/7 offices in US, and India, is a Global Information Technology and Services company providing products and services in integration and middleware across various industries including financial, healthcare, telecommunications, manufacturing, and government. Logicoy was created from the original architects, authors, developers and engineering managers/directors for Seebeyond eGate, ICAN, Sun JCAPS, Open/GlassFish ESB products from inception to end.

Today Logicoy is one of the main OpenESB sponsors and its engineers are committers in the OpenESB codebase and actively working on new JBI components, and enhancements and bug fixes to existing components and new versions of OpenESB. Logicoy provides World Class Global 24x7 Production Support and Monitoring and Migration Services for OpenESB. Please see <http://logicoy.com/support> for further details.

Contact Us: info@logicoy.com

