

QpidProposal

Qpid Proposal (renamed from Blaze & Glasgow)

Concern: Glasgow was well-known as the Sun codename for a revision of the [JavaBeans](#) Specification.

Based on incubator threads, the new project name will be Qpid and used as qpid in the code. The proposal will be updated once the thread "dies" on g@i.a.o

RATIONALE

Qpid provides multiple language implementations of the Advanced Message Queuing Protocol (AMQP) specification and related technologies including PGM, transaction management, queuing, distribution, security, management and heterogeneous multi-platform support for messaging (links to these specifications are in the "Initial Source" section of this proposal.) Qpid's overall goal is to create an open and interoperable implementation for messaging which implements the emerging AMQP specification, in keeping with the philosophy of the Foundation. This implementation will provide a messaging solution that will be language and platform agnostic by using a well defined wire specification. Providing both libraries for the framing and protocol in addition to brokers in both Java and C/C++ allows for integration with Apache and non-Apache projects in a manner that facilitates heterogeneous deployment with full interoperability for SOA & distributed systems. The seed code for the project will consist of in-progress C/C++ and Java implementations of the AMQP specification that we intend to continue development on in conjunction with other Apache communities. More information on the scope of the seed code can be found in subsequent sections of this proposal.

CRITERIA

Meritocracy:

The Qpid committers recognize the desirability and necessity of running this project as a full meritocracy; indeed, the scope of the project's technical aspects are so varied that we find it hard to envision success any other way. One of the most important lessons that can be derived from the historic evolution of middleware is that specifications architected in isolation from real usable code that has been developed to solve tangible, real world problems or amongst a narrowly restricted list of contributors often do not see widespread adoption. Our goal in crafting this implementation and providing our learning to the specification team is to develop the best possible language agnostic advanced message queuing platform. We understand that in order to do so, we will need to engage all interested members of the community and operate to the standard of meritocracy that characterizes successful Apache projects.

Community:

The project's primary objective is to build a vibrant community of users and active contributors. Although Qpid is not based on an existing open source community, many of the initial contributors have experience participating in and building other open source communities. Several of the contributors have previously participated in Apache communities. We understand that Apache's community governance protocols are a unique contributor to the success of Apache's project communities and we are eager to learn from our Incubator mentors so that we can evolve Qpid into a healthy and sustainable community.

Core Developers:

Most of the initial committers are members of Red Hat, IONA, and JP Morgan Chase's (JPMC) development teams. Additional developers will be added through the Apache community process.

Alignment:

An initial implementation has been written in Java and C++, which will be refactored into this project to form the initial code base. We have had a few exploratory conversations about integration with individuals of other communities such as Apache Geronimo, Tuscany, ActiveMQ, Fedora and [ObjectWeb's](#) Celtix and hope to work towards future collaboration with these communities. Our current implementation makes extensive use of projects from the Apache Jakarta Commons, Mina and other Apache infrastructure projects. A compatibility binding for JMS also exists. It is however important to note that this is NOT a JMS project and aims to solve a different problem space, providing language and platform independent and interoperable messaging, driven by a protocol specification which may ultimately be commoditized in hardware.

The scope of the project is broader than just Java and C++ as the project will also look at providing bindings in other languages such as PHP and Python. Additionally, bindings have already been created for test automation. As Qpid's broad goal is to create a standardized, widely available, and interoperable messaging solution based on the AMQP protocol, there are numerous potential collaboration opportunities with many other Apache projects including:

- Transport support for Geronimo
- Interoperability integration with ActiveMQ(JMS)
- Integration with Axis for SOAP messaging over an asynchronous transport
- Language/platform neutral interoperable messaging for projects like Synapse and [ServiceMix](#)

AVOIDING THE WARNING SIGNS

Orphaned products:

The initial code submission is based on active code developed and we believe that through its continued evolution in an open community will lead to a stronger, more widely accepted foundation for development of middleware and be valuable to many other Apache and community projects.

Inexperience with open source:

Many of the initial committers have experience working on open source projects and several are committers on other Apache projects. Each of the companies involved in the initial submission has prior success building or contributing to open source projects. Moreover, some of the initial companies involved focus exclusively on developing open source software. This depth and diversity of experience fosters a deep understanding of managing and running open source projects.

Homogenous developers:

The current list of committers includes developers from several different companies who are geographically distributed across the U.S. and Europe. They are experienced with working in a distributed environment and with resolving technical differences outside the scope of a common employer.

Reliance on salaried developers:

Most of the initial developers are paid by their employers to contribute to this project; however, this submission includes employers with track records for ongoing investment in open source communities (including Apache, Eclipse, [ObjectWeb](#) and Fedora).

No ties to other Apache products:

As described in the Alignment section, this framework already leverages existing Apache projects. By making use of other Apache projects for infrastructure building blocks. The initial codebase will be licensed under the Apache License 2.0.

A fascination with the Apache brand:

The committers are intent on developing a strong open source community around what we hope will be a best-in-class, enterprise-grade high performance messaging framework. We believe that the Apache Software Foundation's emphasis on community open development makes it the most suitable choice for such a project. We understand that the Apache brand has become synonymous with the values of quality, meritocracy, and community, and we endeavor to make our project worthy of such an affiliation. We also commit to working proactively with the Public Relations Committee to ensure that any marketing or promotional activities we pursue are in compliance with the ASF's policies.

SCOPE OF SUBPROJECTS

The initial contributors envision an active community of related projects sharing a common of commodity and interoperable middleware but targeting specific technical areas:

Qpid will be seeded with several projects based on donated material (see the next section):

- a Java implementation of the wire level framing
- a C++ implementation of the wire level framing
- a Java implementation of a broker
- a Java implementation of a JMS interface
- a C++ implementation of a portability layer, which will be refactored to be pluggable
- an implementation of the broker which will be refactored into C++, for existing work and possible use of GCJ

To assist in community building, the committers have identified several key technology areas that will allow new contributors points of entry to actively engage in the project. These include:

- integration with other Apache projects (Tuscany, ActiveMQ, [ServiceMix](#), Apache Axis)
- integration with security and both local and distributed transactions (XA)
- support heterogeneous API bindings in C, C++, Java, PHP, Python and BPEL
- support for cross memory or RDMA transports
- support for in process IPC clients or IPC transport bindings
- support for broadcast and relay from PGM <--> AMQP
- integration with payload marshalling toolkits
- Declarative policy based API's

These initial projects are intended merely as starting points and should not be taken as bounding the scope of the Qpid project as a whole. Some other potential projects may include:

- Integration with rich middleware frameworks (such as Celtix or [ServiceMix](#)).
- Support and integration of Security.
- Management tools.
- Support for additional class frames such as tunneling

INITIAL SOURCE

A group of companies are developing a set of specifications relating to the creation of commodity enterprise class messaging, collectively called Advanced Message Queuing Protocol (AMQP). In progress versions are available at:

- <http://www.envoytech.org/spec/amqp/>

- <http://www.iona.com/opensource/amqp/>
- <http://www.imatix.net/amqp>
- <http://www.redhat.com/solutions/specifications/amqp/>
- <http://www.twiststandards.org/tiki-index.php?page=AMQ>
- <http://www.faqs.org/rfcs/rfc3208.html>

The initial contributors have been developing Java and C++ code bases (licensed under the Apache License 2.0) which implement aspects of these specifications, and intend to donate it to Apache. The current working svn is available at:

<https://etp.108.redhat.com/source/browse/etp/trunk/blaze>

userid: etpguest password: guest

Although the Qpid project expects to bootstrap using these materials and in the case of specifications, to provide feedback that will be incorporated into their ongoing development, we expect and encourage the open source community to take the project in new directions not envisioned by them to create a world class implementation of the AMQP specification and related technologies.

Interactions with the specifications

The specification is being developed by group of companies, under a contract that requires the resulting work to be published to a standards body. This model has been chosen to assure that anyone that contributes to the specification grants a copyright and patent license to all contributions made to the specification on every publication (draft or final). This ensures that the specification will always be open and implementable by anyone without royalties or commercial limitations. We feel that this is a very strong model for keeping this work entirely open and will fit well with the Apache project enabling innovations to pass in both directions across the extended community.

Dealing with feedback from the Qpid project to specifications It is key that the best implementation and specifications be created based on technical merit and practicalities for adoption by both the parties developing the specification and the committers within the Apache community. Given this, one of the important aspects is how issues discovered during the development of this implementation are incorporated back into the specifications. The following feedback loop exists to ensure that any specification input including the Qpid community can have their feedback incorporated into the specifications.

<http://wiki.apache.org/incubator/AMQPlicenses> Copy of the license for published work and to a standards body as requested on the list

MECHANISMS FOR FEEDBACK

a.) Anyone can issue a "JIRA" to the specification working group through the RLA (Reviewer License Agreement). This agreement provides a license to that IP so that the specification team can incorporate it and the specification as they like and the specifications can remain entirely open and royalty free.

b.) If an individual has shown understanding of the project and substantive contribution to the specification, a vote based on technical merit and understanding of the goals of the work can be initiated to have that party join the specification working group. On such acceptance the employer is required to sign an agreement to make sure that employer also grants the ongoing and consistent licenses to the work as posted in specifications.

c.) Additional options for committers on this project to interact with the Working group will be pursued. (see list for context). Quite a wide range of options have been given, these and any others suggested in this process will be explored.

The Reviewer License Agreement (RLA) can be viewed from the AMQP specification page of any of the members as listed above.

ASF resources to be created

mailing list(s)

- qpid-dev
- qpid-commits

Subversion repository

- <https://svn.apache.org/repos/asf/incubator/qpid>

Jira

- qpid (QPID)

INITIAL COMMITTERS

- Rajith Attapattu (Red Hat)
- Mark Atwell (JPMC)
- Bela Ban (Red Hat)
- Bhupendra Bardwaj (JPMC)
- Alan Conway (Red Hat)
- Tejeswar Das (IONA)
- Ovidiu Feodorov (Red Hat)
- Tim Fox (Red Hat)
- Paul Fremantle (WSO2)
- Eoghan Glynn (IONA)
- Robert Greig (JPMC)
- Chamikara Jayalath (WSO2)

- Sam Joyce (IONA)
- John O'Hara (JPMC)
- Frank Lynch (IONA)
- Marnie [McCormack](#) (JPMC)
- Martin Ritchie (JPMC)
- Rafael Schloming (Red Hat)
- Archit Shah (Red Hat)
- Steven Shaw (JPMC)
- Gordon Sim (Red Hat)
- James Strachan (LogicBlaze)
- Manik Surtani (Red Hat)
- Carl Trieloff (Red Hat)
- Kim van der Riet (Red Hat)
- Steve Vinoski (IONA)
- Sergey Yedrikov (IONA)

APACHE SPONSOR

The Qpid team will make the submission to the incubator as the sponsor for incubation.

Champion

- Cliff Schmidt (consultant to Red Hat)

Mentors:

- Cliff Schmidt (consultant to Red Hat)
- Paul Fremantle