JSR 343 (JMS 2.0) status update to JCP EC

Nigel Deakin

11 January 2012

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- Introducing JMS
- JMS 2.0 (JSR 343) Goals
- Current status
- Roadmap with detailed milestones
- Issues
- Detailed status (will skip quickly over...)
- Community and transparency status

Introducing JMS

- Java Message Service (JMS)
  - an API for accessing enterprise messaging systems from Java programs
  - A key component of Java EE - but also stands alone
- Last maintenance release (1.1) was in 2003
  - Does not mean JMS is moribund!
  - Multiple active commercial and open source implementations
  - Shows strength of existing spec
- Meanwhile
  - Java EE has moved on since, and now Java EE 7 is planned
  - Time for JMS 2.0

JMS 2.0 (JSR 343) Goals

- Changes to improve ease of development
    - simpler API, facilitate use of CDI, clarify ambiguity
- Clarification of the relationship between the JMS and other Java EE specifications
    - Much of the JMS spec is overridden by the EJB and Java EE specs – perhaps consolidate in JMS spec
- Definition of a mandatory API to allow any JMS provider to be integrated with any Java EE application server.
- Extensions to support Java EE 7
- Other enhancements as requested by the community

Expert Group Membership

Organisations	
Nigel Deakin (lead), Tom Barnes	Oracle
Shivajee Samdarshi	TIBCO
Matthew White	IBM
Emran Sharif	Pramati Technologies
Reza Rahman	Caucho
Bruce Snyder	VMWare
Clebert Suconic	RedHat
Sastry Maladi	eBay
Robert Davies	FuseSource

Individuals
Julien Dubois
John Ament
Rüdinger zu Dohna
Nicholas Wright (C2B2)
Masoud Kalali
Adam Bien

Current status

- Expert group formed and in operation
- Development of Early Draft in progress
  - Scope now defined and frozen
  - Most content agreed in principle
  - Currently drafting spec, API and javadocs
  - Early Draft expected in Jan/Feb 2012 (3 months late)
- RI and TCK work will start after early draft
- Additional content will be curtailed to try to maintain planned final release date
  - Must release in time for Java EE 7

Roadmap with detailed milestones

Task	Estimate	Milestone	Planned (Feb 2011)	Status (Jan 2012)
Formation of Expert Group		**Expert Group formed**	**End March 2011**	**End May 2011** (actual)
Preparation of Early Draft	**7 months**	**Start of Early Draft Review**	**End Oct 2011**	**Jan/Feb 2011** (expected)
Early Draft Review During early draft review period the EG may update the draft	**1 month**	**End of Early Draft Review**	**End Nov 2011**	
Preparation of Public Draft	**4 months**	**Start of Public Review**	**End Mar 2012**	
Public Review During this period the EG may update the draft. During the final week the draft is frozen and the EG conducts a Public Draft Specification Approval Ballot to decide if the draft may proceed.	**1 month**	**End of Public Review**	**End Apr 2012:**	
Completion of RI and TCK If this uncovers deficiencies in the spec,the spec will be updated. Comments will also be considered.	**5 months**	**Start of Final Approval Ballot**	**Mid Sept 2012:**	
Final Approval Ballot	**2 weeks**	**Final Release**	**End Sept 2012:**	

`http://java.net/projects/jms-spec/lists/jsr343-experts/archive/2011-05/message/0`

Issues

- Took longer than expected to form expert group
  - Perhaps unsurprising for a completely new JSR
  - But now up and running with most key players represented
- Taking longer than expected to provide simplified API
  - First attempt to add a CDI layer on top of existing JMS API was withdrawn as not technically feasible
  - Second approach in progress using modified JMS API
- Several changes required to other specs
  - Java EE 7 spec
  - EJB 3.2 spec
  - JCA spec (no JSR so may incorporate changes in JMS spec)

Detailed Status

- Classified by
  - Improved Java EE integration and JMS portability
  - Spec clarifications
  - Ease of use
  - New features
  - Cloud/PaaS

JIRA ID	Summary	Purpose	Agreed in principle	Added to early draft	Comments

- Most significant changes in **bold**
  - Note that some changes require changes to other specs

Detailed Status:

Improved Java EE integration and JMS portability

JIRA ID	Summary	Purpose	Agreed in principle	Added to early draft	Comments
JMS_SPEC-25 & JMS_SPEC-32	**Standardise the interface between a JMS provider and a Java EE application server**	**Java EE integration**	**Yes**	**Not yet**	
	Define JCA spec changes needed to support JMS	Java EE Integration	Not started	Not yet	**No JCA JSR, so may need to add to JMS spec**
JMS_SPEC-26	Decide on the future of the optional Chapter 8 API "JMS Application Server Facilities"	Java EE integration	Not yet discussed	Will probably defer	
JMS_SPEC-27	Clarify the relationship between the JMS and other Java EE specifications	Java EE integration	Yes	Not yet	
JMS_SPEC-28	Clarify how the JMS provider should interact with Transaction Managers.	Java EE integration	Not yet discussed	Will probably defer	
JMS_SPEC-30	Define mandatory activation config properties for JMS MDBs	Java EE integration	Yes, passed to EJB 3.2 EG	Will be part of EJB 3.2	**Waiting for EJB 3.2 EG**
JMS_SPEC-46	Define standard API to create and configure a ConnectionFactory	Java EE integration	Proposals under discussion	Will probably defer	
JMS_SPEC-54	Define a standard way to configure the destination on which a MDB consumes messages	Java EE integration	Yes, passed to EJB 3.2 EG	Will be part of EJB 3.2	**Waiting for EJB 3.2 EG**
JMS_SPEC-55	Define a standard way to configure the connection factory used by a MDB	Java EE integration	Yes, passed to EJB 3.2 EG	Will be part of EJB 3.2	**Waiting for EJB 3.2 EG**

Detailed Status:

Spec clarifications

JIRA ID	Summary	Purpose	Agreed in principle	Added to early draft	Comments
JMS_SPEC-48	Specify that connection.stop() or close() may not be called from a MessageListener	Clarification	Yes	Yes	
JMS_SPEC-49	Improve specification of ExceptionListener	Clarification	Yes	Yes	
JMS_SPEC-50	Clarify that JMS providers must implement both P2P and Pub-Sub	Clarification	Yes	Yes	
JMS_SPEC-52	Clarify that a message may be sent using a different session from that used to create the message	Clarification	Yes	Yes	
JMS_SPEC-34	Calling setJMSDeliveryMode or setJMSPriority on a Message before it is sent don't have any effect	Clarification	Yes	Yes	
JMS_SPEC-65	Clarify use of NoLocal arg when using createDurableSubscriber	Clarification	Yes	Yes	

Detailed Status:

Ease of use

JIRA ID	Summary	Purpose	Agreed in principle	Added to early draft	Comment
JMS_SPEC-33	**Improving the JMS API with API simplifications, annotations and CDI**	**Ease of use**	**No**	**Abandoned**	**Proposals agreed to be unworkable**
JMS_SPEC-64	**Defined simplified API**	**Ease of use**	**Under discussion**	**In progress**	**Second attempt after JMS_SPEC-33 was abandoned**
JMS_SPEC-39	Make clientId optional when creating a durable subscription	Ease of use	Yes	Yes	
JMS_SPEC-45	Clarify and improve Connection.createSession	Ease of use	Yes	In progress	
JMS_SPEC-63	Introduce concept of platform default connection factory in Java EE	Ease of use	Yes, passed to Java EE 7	Will be part of Java EE 7	**Waiting for Java EE EG**
JMS_SPEC-51	Change Session.createDurableSubscriber() to return a MessageConsumer	Ease of use	Yes	Yes	
JMS_SPEC-53	Make Connection and other interfaces implement AutoCloseable	Ease of use	Yes	Yes	

Detailed Status:

New features

JIRA ID	Summary	Purpose	Agreed in principle	Added to early draft	Comments
JMS_SPEC-40	Allow multiple consumers to be created on the same topic subscription	New feature	Yes	Not yet	
JMS_SPEC-41	Support topic hierarchies	New feature	Not yet discussed	Will probably defer	Waiting for proposals
JMS_SPEC-42	Make support for JMSXDeliveryCount mandatory	New feature	Yes	Yes	
JMS_SPEC-43	New API to send a message with async acknowledgement from server	New feature	Yes	Yes	
JMS_SPEC-44	New API to specify delivery delay	New feature	Yes	Yes	
JMS_SPEC-36	Allow messages to be delivered asynchronously in batches	New feature	Yes	Yes	
JMS_SPEC-56	Enhanced EJB spec to support the delivery of messages in batches to MDBs	New feature	Yes, passed to EJB 3.2 EG	Will be part of EJB 3.2	Waiting for EJB 3.2 EG

Detailed Status:

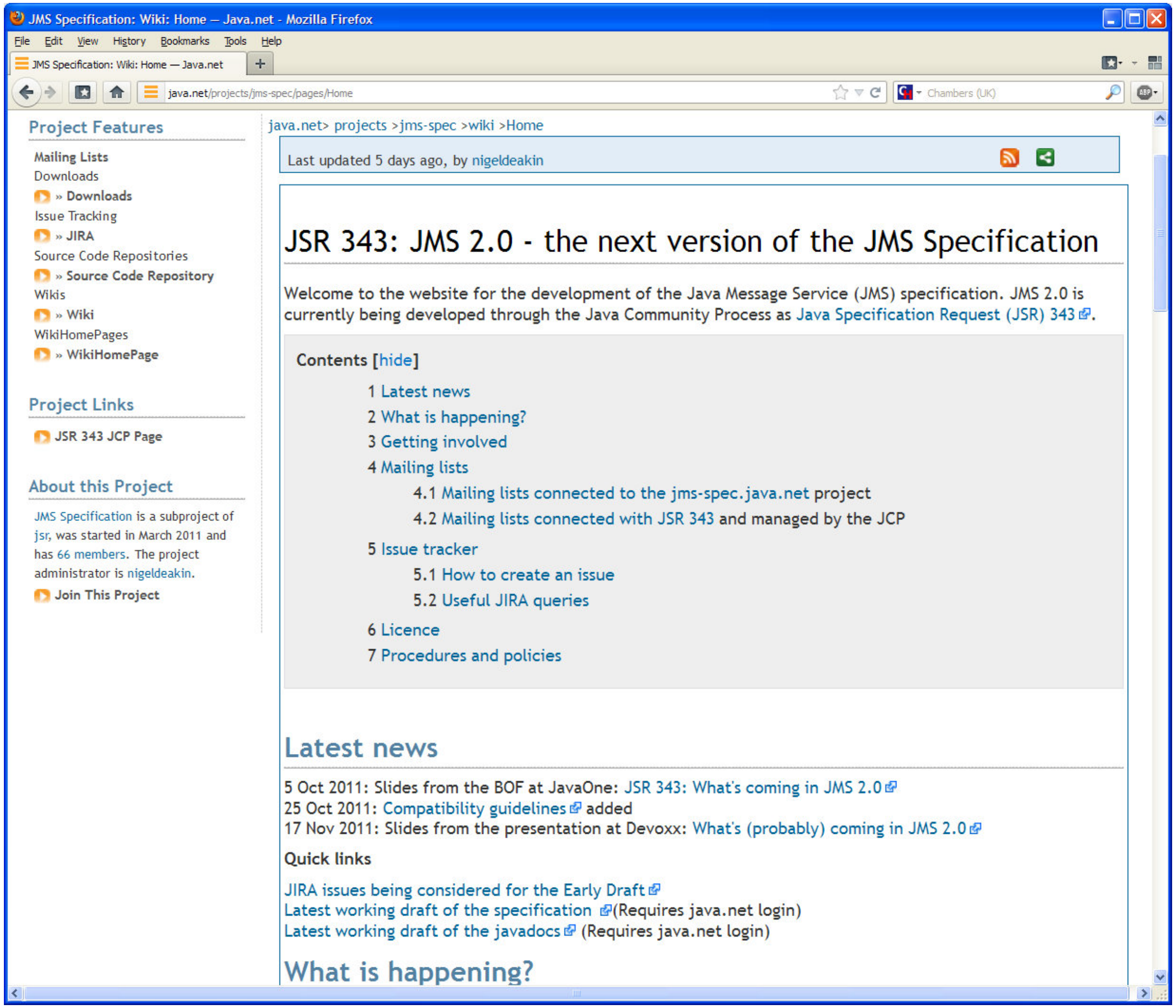
Cloud/PaaS

JIRA ID	Summary	Purpose	Agreed in principle	Added to early draft	Comments
JMS_SPEC-57	Support for multi-tenancy	Cloud/PaaS	Under discussion	Will probably defer	Joint work with platform spec
	Support for Java EE resource metadata	Cloud PaaS	Under discussion	Will probably defer	Joint work with platform spec

Community and transparency

- JSR 343 uses JCP 2.7 but has operated transparently from the start
- Project website jms-spec.java.net
  - Public wiki
  - Public issue tracker
  - Public user mailing list
  - Public expert group mailing list (forwarded to user list)
- Intention to move to JCP 2.8 (with other Oracle JSRs)

jms-spec.java.net


JMS Specification: Wiki: Home – Java.net - Mozilla Firefox
File Edit View History Bookmarks Tools Help
JMS Specification: Wiki: Home — Java.net
java.net/projects/jms-spec/pages/Home
Chambers (UK)

Project Features

Mailing Lists
Downloads
» Downloads
Issue Tracking
» JIRA
Source Code Repositories
» Source Code Repository
Wikis
» Wiki
WikiHomePages
» WikiHomePage

Project Links

JSR 343 JCP Page

About this Project

JMS Specification is a subproject of jsr, was started in March 2011 and has 66 members. The project administrator is nigeldeakin.

Join This Project

java.net> projects >jms-spec >wiki >Home

Last updated 5 days ago, by nigeldeakin

JSR 343: JMS 2.0 - the next version of the JMS Specification

Welcome to the website for the development of the Java Message Service (JMS) specification. JMS 2.0 is currently being developed through the Java Community Process as Java Specification Request (JSR) 343.

Contents [hide]

Latest news

5 Oct 2011: Slides from the BOF at JavaOne: JSR 343: What's coming in JMS 2.0
25 Oct 2011: Compatibility guidelines added
17 Nov 2011: Slides from the presentation at Devoxx: What's (probably) coming in JMS 2.0

Quick links

JIRA issues being considered for the Early Draft
Latest working draft of the specification (Requires java.net login)
Latest working draft of the javadocs (Requires java.net login)

What is happening?

Community and transparency - issues

- Moderate activity from expert group
  - only three or four are regularly active
  - several new/replacement members this month may imporve things
  - planning more one-on-one/group phone calls
- Moderate activity from user community
  - several people have submitted new issues using JIRA
  - much interest at JavaOne and Devoxx
  - user email alias very quiet

JCP 2.7 Transparency Checklist

(JSR 343 uses JCP 2.7)

Checklist question	Current status (Jan 2012)	Details
The public can read the names of the people on the Expert Group (ie, not just JCP Members)	No	Only the standard list of members provided at http://jcp.org/en/jsr/summary?id=343 is available
The Expert Group business is regularly reported on a publicly readable alias	Yes	EG emails are forwarded to public user alias users@jms-spec.java.net and archived at http://java.net/projects/jms-spec/lists
The schedule for the JSR is publicly available, it's current, and I update it regularly.	No	Initial schedule was published by email only and has not been updated since
The public can read/write to a wiki for my JSR.	Public may read wiki but only EG members may update it	Public wiki is at http://jms-spec.java.net
I have a discussion board for my JSR and I regularly read and respond to posts on that board.	Yes	Public email alias users@jms-spec.java.net and archived at http://java.net/projects/jms-spec/lists
There is an issue-tracker for my JSR that the public can read.	Yes	http://java.net/jira/browse/JMS_SPEC
I have spoken at conferences and events about my JSR recently.	Yes	BOF at JavaOne Oct 2011 USA Session at Devoxx Nov 2011 Belgium
I am using open-source processes for the development of the RI and/or TCK.	Yes for RI . Unknown for TCK	Development of RI and TCK has not formally started. RI will be developed at mq.java.net
The Community tab for my JSR has links to and information about all public communication mechanisms and sites for the development of my JSR.	Yes	Community tab has link to http://jms-spec.java.net which has full details

JCP 2.8 Transparency Checklist

(JSR 343 uses JCP 2.7 but plans to move to 2.8)

Checklist question	Current status (Jan 2012)	Details
Is the schedule for the JSR publicly available, current, and updated regularly?	No	Initial schedule was published by email only but has not been updated since
Can the public read and/or write to a wiki for the JSR?	Public may read wiki but only EG members may update it	Public wiki is at http://jms-spec.java.net
Is there a publicly accessible discussion board for the JSR that you read and respond to regularly?	Yes	Public email alias users@jms-spec.java.net and archived at http://java.net/projects/jms-spec/lists
Have you spoken at conferences and events about the JSR recently?		BOF at JavaOne Oct 2011 USA Session at Devoxx Nov 2011 Belgium
Are you using open-source processes for the development of the RI and/or the TCK?	Yes for RI . Unknown for TCK	Development of RI and TCK has not formally started. RI will be developed at mq.java.net
What are the Terms of Use required to use the collaboration tools you have prepared to use with the Expert Group, so that prospective EG members can judge whether they are compatible with the JSPA?	**JSR 343 uses java.net collaboration tools. Terms of use are at http://home.java.net/javanet-web-site-terms-use**	
Does the Community tab for my JSR have links to and information about all public communication mechanisms and sites for the development of my JSR?	Yes	Community tab has link to http://jms-spec.java.net which has full details

Any questions?