Java Community Process

Community Development of Java Technology Specifica

My JCP

Sign-in

Register for Site

Use of JCP site is subject to the JCP Terms of Use and the Oracle Privacy Policy

JCP Info

» **About JCP**   ▶

» **Get Involved**   ▶

» **Community Resources**   ▶

» **Community News**   ▶

» **FAQ**

» **Contact Us**

Download Java Software for Your Computer

| JSR | Community | Expert Group |

Summary | Proposal | Detail (Summary & Proposal)

JSRs: Java Specification Requests
JSR 156: Java API for XML Transactions

Stage	Access	Start	Finish
Withdrawn		18 Feb, 2010	
Expert Group Formation		06 Nov, 2001	26 Aug, 2005
JSR Review Ballot	View results	23 Oct, 2001	05 Nov, 2001

Status: Withdrawn
Reason: Withdrawn at the request of the Spec Lead.
JCP version in use: 2.1
Java Specification Participation Agreement version in use: 1.0

Description:
JAXTX provides an API for packaging and transporting ACID transactions (as in JTA) and extended transactions (e.g., the BTP from OASIS) using the protocols being defined by OASIS, W3C.

Please direct comments on this JSR to the Spec Lead(s)

| Team |

Specification Leads

Jonathan Halliday	Red Hat

Expert Group

Choreology Ltd.	Cotton III, Ben D.	IBM
Micro Focus Ltd.	Oracle	Pope, William Z.
Progress Software	Red Hat	SAP SE
Sun Microsystems, Inc.	Verma, Manish	

This JSR has been Withdrawn
Reason: Withdrawn at the request of the Spec Lead.

Updates to the Original JSR

The following has been updated from the original request.

2005.08.03: The name of the JSR was changed from "XML Transactioning API for Java (JAXTX)" to "Java API for XML Transactions (JAXTX)".

Original Java Specification Request (JSR)

Identification | Request | Contributions

Section 1. Identification

Submitting Member: Hewlett-Packard

Name of Contact Person: Bob Bickel

E-Mail Address: bob_bickel@hp.com

Telephone Number: +1 856 727 4600 x1210

Fax Number: +1 856 787 9395

Specification Lead: Mark Little

E-Mail Address: mark_little@hp.com

Telephone Number: +44 191 206 4538

Fax Number: +44 191 206 4203

Initial Expert Group Membership:

HP
IBM
IONA Technologies
Choreology Limited

Supporting this JSR:

HP
IBM
IONA Technologies
Choreology Limited
TalkingBlocks

Section 2: Request

2.1 Please describe the proposed Specification:

This JSR requests the creation of the XML Transactioning API's for Java 1.0 specification (JAXTX).

This specification will describe Java API's designed specifically for the management (creation and lifetime) and exchange of transaction information between participating parties in a loosely coupled environment. Rather than communicating via IIOP, such parties will typically use SOAP and XML document exchange to conduct business "transactions". If these "transactions" are to be conducted in an ACID transaction manner, then information (e.g., the transaction context) will need to accompany these XML documents and be managed appropriately.

In addition, there is work going on within OASIS (the Business Transactions protocol) and the JCP community (JSR 95) to provide extended non-ACID transactions to users that will enable applications to run business transactions that span many organisations, last for hours, days, and weeks, and yet still retain some of the fault-tolerant and consistency aspects of traditional ACID transactions.

The business community is working to converge on a set of standard message headers and industry-specific message payloads. It is planned that this JSR will leverage work currently under way in the OASIS, W3C, and potentially other relevant and open standards bodies to produce an API that will accomodate both strict ACID and relaxed ACID transactional applications.

This JSR does not aim to define either XML messaging standards or XML schemas for particular tasks. These networking and formatting standards belong in networking standards bodies such as Oasis or IETF. Instead this JSR aims to define standard Java APIs to allow convenient access from Java to emerging XML messaging standards.

2.2 What is the target Java platform? (i.e., desktop, server, personal, embedded, card, etc.)

The JAXTX 1.0 specification will be provided initially as a standard extension but will be incorporated into the Java 2 Enterprise Edition platform as soon as this is practical and there is sufficient demand to warrant such integration.

2.3 What need of the Java community will be addressed by the proposed specification?

This specification will provide consistent Java APIs for using emerging XML based transaction standards, as well as allowing existing ACID transaction standards to be used in a loosely coupled environment.

2.4 Why isn't this need met by existing specifications?

Existing transaction standards such as JTA/JTS assume a closely coupled environment, with participants communicating via IIOP. In loosely coupled environments such as the web, the individual components of these transaction standards (e.g., transaction coordinator, transcional resource) would be offered as services, communicating via SOAP and XML. There is no industry standard for how these services may be presented to a Java programmer, and how XML exchanges between distributed spaces may be augmented with transactional payloads.

In addition, there are newly emerging business transaction standards specifically designed for loosely coupled environments where relaxation of ACID properties is a requirement. For example, BTP from OASIS. However, these standards tend to address the issues of message communication in a language neutral manner. Therefore, once again their is no standard API for Java programmers to use.

2.5 Please give a short description of the underlying technology or technologies:

The JAXTX 1.0 specification will most likely specify a low-level abstract Java interface specifically targeting the management of loosely coupled transaction systems.

2.6 Is there a proposed package name for the API Specification? (i.e., `javapi.something`, `org.something`, etc.)

To Be Determined.

2.7 Does the proposed specification have any dependencies on specific operating systems, CPUs, or I/O devices that you know of?

No.

2.8 Are there any security issues that cannot be addressed by the current security model?

No.

2.9 Are there any internationalization or localization issues?

No.

2.10 Are there any existing specifications that might be rendered obsolete, deprecated, or in need of revision as a result of this work?

No. Existing transaction APIs such as JTA and JTS are most suitable for the closely coupled environments in which they operate.

2.11 Please describe the anticipated schedule for the development of this specification.

The final schedule will need to be determined by the expert group, and will also be dependent on the progress of the relevant industry standards. However, it is anticipated that a reasonably solid draft should be available in the early summer.

2.12 Please describe the anticipated working model for the Expert Group working on developing this specification.

It is anticipated there will be a face-to-face kick-off meeting. Subsequent work will be done by email. The goal will be to attempt to develop a consensus in the expert group over the main APIs and technologies.

Section 3: Contributions

3.1 Please list any existing documents, specifications, or implementations that describe the technology. Please include links to the documents if they are publicly available.

BTP specification: http://www.oasis-open.org/committees/business-transactions/

JSR 95: http://www.jcp.org/jsr/detail/95.jsp

JTA/JTS: http://jcp.org/jsr/detail/907.jsp

3.2 Explanation of how these items might be used as a starting point for the work.

The key transaction technology for J2EE is JTA/JTS. If used correctly, implementations of these specifications can be made available to loosely coupled services operating in the Web. Achieving this

will be an important goal for this JSR.

The Business Transactions Protocol from OASIS defines a relaxed transaction capability tailored for those applications and services that can tolerate a degree of inconsistency or require forward/backward compensation. BTP does not, however, define any language API. It is expected that many of the services that will use BTP will be written in Java, so standardising on the API will be extremely important.

ORACLE®