

# Home

## About JDO

Java Data Objects (JDO) is a standard way to access persistent data in databases, using plain old Java objects (POJO) to represent persistent data. The approach separates data manipulation (done by accessing Java data members in the Java domain objects) from database manipulation (done by calling the JDO interface methods). This separation of concerns leads to a high degree of independence of the Java view of data from the database view of the data.

Interfaces are defined for the user's view of persistence:

- **PersistenceManager**: the component responsible for the life cycle of persistent instances, Query factory, and Transaction access
- **Query**: the component responsible for querying the datastore and returning persistent instances or values
- **Transaction**: the component responsible for initiating and completing transactions

Commercial and open source implementations of JDO are available for relational databases, object databases, and file systems.

JDO is being developed as a Java Specification Request in the Java Community Process. The original JDO 1.0 is JSR-12 <http://www.jcp.org/en/jsr/detail?id=12> and the current JDO 2.0 is JSR-243 <http://www.jcp.org/en/jsr/detail?id=243>. JSR-243 has been approved by the Executive Committee of the JCP <http://jcp.org/en/jsr/results?id=3663>

JSR-243 is now in maintenance. For a list of issues to be resolved in maintenance, please see the [ChangeLog](#).

## Current Development Activities

Please look at [CurrentDevelopment](#) for a list of tasks currently under development or planned. Volunteer by expressing interest at [jdo-dev@db.apache.org](mailto:jdo-dev@db.apache.org).

## Design Work

Designing a [ScalableBackEnd](#) for an implementation of Apache JDO similar to FOSTore but scalable using a queryable back end.

## About Apache JDO

Sun Microsystems has decided to donate Java Data Objects (JDO) to the open source community. Both of the specifications will be developed in the Apache JDO project. We envision several sub-projects to be developed as part of this project:

- **JDO 1.0 API**. This is the standard definition of the JDO API as defined by the published JSR-12 standard.
- **JDO 1.1 Reference Implementation**. This is a file-based, single-user store for persistent Java objects. The version number is 1.1 to reflect that the Java package has changed from `com.sun.jdori` to `org.apache.jdo`. The license terms have also changed.
- **JDO 1.1 Technology Compatibility Kit**. This set of programs tests that a JDO 1.0 implementation is in compliance with the standard. The version number is 1.1 to reflect that the test framework has changed from [JavaTest](#) (a proprietary Sun package not suitable for open source projects) to [JUnit](#), an open source test framework. The license terms have also changed.
- **JDO 2.0 API**. This is the standard definition of the JDO API as defined by the JSR-243 standard currently under development.
- **JDO 2.0 Technology Compatibility Kit**. This set of programs tests that a JDO 2.0 implementation is in compliance with the standard.
- **JDO 2.0 FOSTore (File Object Store) implementation**. This is an implementation of JDO 2.0 based on the JDO 1.0 Reference Implementation. It does not support relational data access, so it is inappropriate for the JDO 2.0 Reference Implementation. We are investigating [SleepyCat](#) for the back end. See [SleepyCatIntegration](#) for discussion of the JDO/SleepyCat integration.
- **JDO 2.0 Geronimo Integration**. This project will build the glue between the JDO 2.0 Reference Implementation and the J2EE compliant application server. We expect to use a common persistence framework (Tranql) to provide both EJB3 and JDO persistence services.

Due to timing constraints, the JDO 2.0 Reference Implementation is not being built as an Apache project, but will be built as a JPOX release <http://jpox.org>.

## Project Structure

For more information on the structure of the project, see [ProjectStructure](#).

## Tools

This project will use:

- [Wiki](#) for general project communications;
- [ASF JIRA](#) for issue tracking;
- [subversion](#) for source control;
- [maven](#) for build;
- test first philosophy.

# Participation

If you are interested in participating in the JDO project, please subscribe to a [mailing list](#) to find out what's happening.

If you're interested in committing to the project:

- You'll need to fill out some legal paperwork and go through a process to get an apache committer account: See [New Committers Guide](#), [Contributors](#), and [Committers](#) for more details.
- After you've received an email from root@apache.org with your committer account information, change your initial password: Login by 'ssh -l <username> cvs.apache.org' (soon 'people.apache.org', as by apache email announcement); run 'passwd'; run 'svnpasswd'. See [Committer Subversion Access](#).
- Check out the JDO sources and test your [subversion](#) account.
- Sign up for a [Wiki](#) account.
- Sign up for an [ASF JIRA](#) account.

We have a communications conference call every Friday at 9:00 AM Pacific time. The conference call number is 866 230-6968 (international 865 544-7856) code 294-0479#. During this call we discuss project status, issues, concerns, and strategy. Everyone interested in Apache JDO is welcome and encouraged to participate.

## Technology Compatibility Kit Developer's Page

Information for developers of the JDO TCK is found at [TechnologyCompatibilityKit21](#)

The list of TCK20 challenges is found at [TCKChallenges](#)

## Releases

Apache JDO 2.3 is currently under development.

The release testing instructions are at [ReleaseTesting2dot3](#).

The release manager must have a key to sign the release. Learn about key signing and creating a key at [KeySigning](#).

Test instructions of previous releases: [ReleaseTesting2dot1](#), [ReleaseTesting2dot1dot1](#), [ReleaseTesting2dot2](#).

## General Developer Information

- JDO uses the same [coding standards](#) as the Geronimo project.
- There are a number of items on the [ToDoList](#). You can volunteer for any of them...
- The subversion repository has been set up. You can take a look at <http://db.apache.org/jdo/svn.html> for more information.
- The apache JDO project uses maven for build. You can take a look at the [MavenInfo](#) page for more information.
- The JDO sub-projects ri11 and tck11 make use of the apache commons logging package. You can take a look at the [LoggingDetails](#) page for more information.
- The objective of JDO is to support J2SE 1.3 for runtime. You find details about running with J2SE 1.3 at the [J2SESupport](#) page.
- A [DeveloperScratchpad](#) is available for uncategorized issues.

## JDO Resources

[Sun's official JDO site](#)

[Apache JDO site](#)

[JDO-related resources](#) - Articles, Books, Presentations, Tutorials.

[JavaOne2006](#) and [ApacheCon2006](#) presentations in progress

## 'Special' Wiki pages

### TitleIndex

A list of all pages on this wiki.

### HelpContents

A basic guide to the [MoinMoin](#) wiki (including information about wiki syntax).

### WordIndex

A list of all the words that appear in the titles of the pages on this wiki, with links to pages that include that word.

### FindPage

A full-text search of the wiki.

**WantedPages**

All the "broken links" – a list of all the pages on this wiki that are linked to, but do not exist.

**OrphanedPages**

All pages on this wiki that are not linked to from anywhere else (and are thus very hard to reach).

**RandomPage**

Generates a list of 75 random pages on this wiki.

**PageSize**

Generates a graph and some statistics about the sizes of pages on this wiki.

**EventStats/HitCounts**

Generates a graph of page views and page visits.

**EventStats/UserAgents**

Generates a graph of the web browsers used in visiting this page.

**SystemInformation**

Shows basic information about this wiki installation, the extensions it has installed, etc.