



## JDX 3.0

**JDX is a high-performance, cross-platform, developer-friendly and cost-effective solution for transactional persistence of Java objects.**

JDX seamlessly bridges the gap between Java object model and SQL relational model by providing a very natural object-oriented interface to store and retrieve Java objects using relational databases.

JDX increases programmer productivity tremendously (about 30 - 50%) by making the code simpler and yet efficient. It also allows mapping of existing (legacy) relational data to Java objects.

JDX provides your applications independence from operating systems, backend databases and JDBC drivers.

JDX itself is written in 100% pure Java, making it portable to every Java-enabled platform.

JDX is the ideal tool for developing and deploying modern object-oriented applications - be they standalone, JSP/Servlets based on EJB.

JDX software comes in different components that may be licensed individually.

**JDXClassic:** This is the core object-relational mapping technology consisting of the runtime mapping engine and basic tools (JDXSchema and JDXDemo).

**JDXServer:** This optional add-on to JDXClassic provides the capability of running the object-relational mapping functionality in a separate tier for load-balancing and scalability.

**JDXStudio:** A GUI program for designing and configuring the mapping system.

**JDXCombo:** This combines JDXClassic, JDXServer and JDXStudio components in one offering. JDXServer and JDXStudio need JDXClassic to run.



### JDXClassic 3.0 Features

- Simplifies integration of Java objects with relational data - application (or applet) programmers need not write any SQL statements
- Declarative mapping specification based on a simple grammar
- Can easily handle complex objects structures - one-to-one or one-to-many relationship models supported
- An object may contain other object(s) by value or by reference
- Object modeling with class-hierarchy supported
- Support for persistence by reachability
- No requirement to inherit from a special class
- No hassles of pre-processing or post-processing of your code
- Object viewing facility
- Flexible query options - deep, shallow and anything in-between
- Powerful object-streaming functionality
- Handy sequence generators
- Dynamic data routing
- Support for partial loading of objects
- Allow inline storage of the attributes of a contained object
- Support implicit attributes whose values are automatically initialized by JDX based on defined relationships
- Support for aggregated storage of inheritance objects in one table
- Support for using application provided dynamic class loaders
- Support for retrieving collection objects in a descending order
- Support for specifying a JNDI data source name to get database connection
- Allow specification for an external transaction manager during JXSession.open () call
- Utility components for pooling JX/JDX handles
- Option to filter referenced objects for query, update and delete operations
- Configurable logging destination
- Support for Java2 Collections
- Optionally generate default accessor (getter/setter) methods while reverse-engineering Java classes
- Intuitive and powerful APIs
- Efficient caching of metadata and database connections

- Implemented on top of standard JDBC interface. Can work with Microsoft SQL Server, Oracle, Sybase, IBM DB2, MS Access, Cloudscape, Pointbase and other relational databases and JDBC sources
- An application may work with multiple databases simultaneously
- Multiple invocations of JDX methods may be made part of a single transaction
- Simple and intuitive object-oriented client interface through open standard technologies
- Runs as a class library. Additionally, with one line of code change, you can also use the same mapping functionality in a separate high-performance, multithreaded server (using JDXServer)
- Tool (JDXSchema) provided for relational schema-generation given Java class definitions
- Tool (JDXSchema) provided to generate Java class definitions from legacy relational data
- Tool (JDXDemo) provided to verify object-relational mapping specification using a live database
- Modular design, high-performance and scalable implementation
- All-Java implementation makes it a cross-platform solution

### JDXServer 3.0 Features

- This optional add-on to JDXClassic provides the capability of running the object-relational mapping functionality in a separate tier for load-balancing and scalability
- JDXServer runs as an RMI server and can be located on a different machine that may provide advantages like better processing power or co-location with database servers
- One instance of the server may provide services to multiple clients
- Many instances of the server may be started simultaneously on different machines
- Promotes thin-clients applications

### JDXStudio 3.0 Features

- A Java Swing based GUI tool to define object-relational mapping (OR-Mapping)
- Create database schema (tables and constraints from an object model)
- Reverse-engineer Java classes from an existing database schema
- Verify OR-Mapping against a live database using tightly integrated JDXDemo program
- Multiple well laid out panels for ease-of-use
- Extensive online-help including tooltips



57, boulevard de l'Embouchure - 31075 Toulouse Cedex 2 - FRANCE  
Tel.: +33-5 34 27 90 03 - Fax: +33-5 34 27 90 04  
E-mail: [sales@kapitec.com](mailto:sales@kapitec.com) - URL: <http://www.kapitec.com>