

Annotated Bibliography

Almaer, Dion. (2002). Using Java Data Objects. *ONJava.com* 2/6/2002. URL: <http://www.onjava.com/lpt/a/1372>

Java Data Objects (JDO) is a recent addition to the suite of APIs available for accessing databases from a Java-based environment. This article provides some simple examples of how the JDO technology is used to build Java objects from relational databases.

Baldwin, Richard T. (2003). Views, Objects, and Persistence for Accessing a High Volume Global Data Set. *Digest of Papers IEEE Symposium on Mass Storage Systems (MSS'03)* p 77-81

This paper describes lessons learned by the National Climate Data Center (NCDC) as it decides how to deploy extremely large data sets for use by scientists and researchers world-wide. Prototype systems were developed using direct file access from Java programs, a key/value database bundled with Unix systems, and Java Data Objects (JDO) and Enterprise JavaBeans (EJB) interfaces to RDBMS and ODBMS.

Brown, Jeff. (2002). An Introduction To Java Data Objects. *Object Computing, Inc. - Java News Brief* June 2002. URL: <http://www.ocweb.com/jnb/jnbJun2002.html>

This web article provides a general overview of the Java Data Objects (JDO) API and how it relates to the prior Java Database Connectivity (JDBC) API. The XML descriptions of the RDBMS-to-Java object mappings are also discussed.

Cecchet, Emmanuel ; Marguerite, Julie and Zwaenepoel, Willy. (2002). Performance and Scalability of EJB Applications. *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA, 2002*, p 246-261

The paper describes a study of the performance of an e-commerce application utilizing different combinations of Enterprise JavaBeans. The conclusion of the study is that stateless session beans with bean managed persistence out perform alternate combinations of entity beans with container managed persistence.

Eisenberg, Andrew and Melton, Jim. (1998). SQLJ Part O, now known as SQL/OLB (Object-Language Bindings). *SIGMOD Record*, Vol. 27, No. 4, December 1998. p 94-100

SQLJ/OLB is a standard for embedding SQL code directly into Java source code. This paper describes the standard and illustrates how it merges with the Java Database Connectivity (JDBC) API.

Gorton, Ian and Liu, Anna. (2003). Evaluating the Performance of EJB Components. *IEEE Internet Computing*, v 7, n 3, May/June, 2003, p 18-23

A report comparing the performance of two common Java 2 Enterprise Edition's (J2EE) Enterprise JavaBean (EJB) application architectures. J2EE allows for management of object persistence either by the server container or the EJBs themselves.

Jordan, David and Russell, Craig. (2003). JDO or CMP? *ONJava.com* 05/21/2003. URL: <http://www.onjava.com/lpt/a/3763>

This web article provides a bullet comparison of the capabilities of object persistence using Java Data Objects (JDO) and Container Managed Persistence (CMP) of the Java 2 Enterprise Edition standard. It is excerpted from the book *Java Data Objects* published by O'Reilly & Company by the same authors.

Salo, T. and Hill, J. (2000). Building Enterprise Web Applications with Java. *JOOP - Journal of Object-Oriented Programming*, v 13, n 2, May, 2000, p 28-29+47

Five architectures for web applications utilizing different combinations of Java-based technologies are examined. Systems using components comprised of servlets, Java Server Pages (JSP), Java Beans and Enterprise JavaBeans, and combinations are compared.

Tost, A. and Johnson, V.M. (2000). Using JavaBeans Components as Accessors to Enterprise JavaBeans Components. *IBM Systems Journal*, v 39, n 2, 2000, p 293-300

The paper describes how using JavaBean components in a Enterprise JavaBean allows for a clean , three-tier architecture separating client-side and server-side implementations.