



About Database Adapters

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.



Part No: 820-5069
07/08/08

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun[™] Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Contents

About Database Adapters	5
About Sun Adapter for DB2 Connect	5
About DB2 Connect	5
About the DB2 Connect Adapter	6
About Sun Adapter for DB2 Universal Database	6
About DB2 Universal Database	6
About the DB2 Universal Database Adapter	6
About Sun Adapter for Informix	7
About Informix	7
About the Informix Adapter	7
About Sun Adapter for JDBC/ODBC	7
About Java Database Connectivity (JDBC)	7
About the JDBC/ODBC Adapter	11
About Sun Adapter for Oracle	11
About Oracle Databases	11
About the Oracle Adapter	12
About Sun Adapter for SQL Server	12
About SQL Server	12
About the SQL Server Adapter	12
About Sun Adapter for Sybase	13
About Sybase	13
About the Sybase Adapter	13
About Sun Adapter for VSAM	13
About VSAM	14
About the VSAM Adapter	14

About Database Adapters

The topics listed here provide information about Database Adapters. If you have any questions or problems, see the Java CAPS web site at <http://goldstar.stc.com/support>.

- “About Sun Adapter for DB2 Connect” on page 5
- “About Sun Adapter for DB2 Universal Database” on page 6
- “About Sun Adapter for Informix” on page 7
- “About Sun Adapter for JDBC/ODBC” on page 7
- “About Sun Adapter for Oracle” on page 11
- “About Sun Adapter for SQL Server” on page 12
- “About Sun Adapter for Sybase” on page 13
- “About Sun Adapter for VSAM” on page 13

About Sun Adapter for DB2 Connect

This topic provides information about Sun Adapter for DB2 Connect.

About DB2 Connect

DB2 Connect, IBM's database management system, makes host data directly available to Personal Computers and LAN-based workstations. It connects desktop and palm-top applications to mainframe and minicomputer host databases, leveraging enterprise information no matter where it is.

DB2 Connect provides the application enablement and communication infrastructure for connecting Web, Windows, UNIX, Linux, OS/2 and mobile applications to S/390 and AS/400 data. It also enables secure access to legacy data through intranets, extranets or the public Internet, allowing you to build new Internet applications and extend existing applications.

DB2 Connect is included in many of the DB2 UDB products, providing extensive application programming tools for developing client-server and web applications using industry standard APIs such as ODBC, ADO, OLE DB, JDBC, SQLJ, DB2 CLI and Embedded SQL.

About the DB2 Connect Adapter

The Adapter enables Sun Java CAPS ESB Projects to exchange data with external DB2 databases. This document describes how to install and configure the Adapter.

Note – The DB2 Connect Adapter connects to DB2 via the IBM DB2 Connect driver, which is NOT packaged with the Adapter. The product must be separately installed and configured.

About Sun Adapter for DB2 Universal Database

This topic provides information about Sun Adapter for DB2 Universal Database.

About DB2 Universal Database

A database consists of a collection of information that is organized so that it can be easily accessed, managed, and updated. DB2 Universal Database is a database that handles the development and deployment of critical solutions such as:

- On demand business
- Business intelligence
- Content management
- Enterprise Resource Planning
- Customer Relationship Management

DB2 reduces the complexity of data management by eliminating, simplifying, and automating tasks associated with maintaining an enterprise-class database. It provides a foundation of information integration technologies, including federation, replication, Web services, and XML.

About the DB2 Universal Database Adapter

The Adapter enables Sun Java CAPS ESB Projects to exchange data with external DB2 databases. This document describes how to install and configure the Adapter.

Note – The DB2 Universal Adapter connects to DB2 via the DataDirect driver which is packaged with the Adapter.

About Sun Adapter for Informix

This topic provides information about Sun Adapter for DB2 Universal Database.

About Informix

Informix Dynamic Server (IDS) database is a multithreaded object-relational database server that manages data that is stored in rows and columns. It employs a single processor or symmetric multiprocessing (SMP) systems and dynamic scalable architecture to deliver database scalability, manageability and performance.

About the Informix Adapter

The Adapter is a component that connects Sun Java CAPS ESB and the Informix Dynamic Server (IDS) database. The Adapter is designed to handle all the communication details necessary to send and receive data between these components.

In addition to handling communications, the Adapter can also apply business logic within Collaboration Rules to perform any of Sun Java CAPS ESB's range of data identification, manipulation, and transformation operations.

About Sun Adapter for JDBC/ODBC

This topic provides information about Sun Adapter for JDBC/ODBC.

About Java Database Connectivity (JDBC)

Java Database Connectivity (JDBC) is an implementation of the Java programming language that dictates how databases communicate with each other. Through a standardized application programming interface (API), connectivity from database management systems (DBMS) to a wide range of SQL databases is accomplished. By deploying database drivers laced with JDBC technology, it is possible to connect to any database -- even in a heterogeneous environment -- and access tables, tabular data, flat files and more. When using JDBC, Java programmers have the ability to request connections to a database, send queries to the database using SQL statements, and receive results for advanced processing.

JDBC Drivers

To connect with individual databases, JDBC requires drivers for each database. Those drivers come in four varieties. Driver types 1 and 2 are typically intended for programmers that write applications. Driver types 3 and 4 are typically used by database and middleware vendors. The various driver types are described in the following sections:

- “Type I: JDBC-ODBC Bridge” on page 8
- “Type II: Partial Java driver” on page 9
- “Pure Java driver for database middleware” on page 9
- “Type Four Driver: Direct-to-database pure Java driver” on page 10

Type I: JDBC-ODBC Bridge

This combination provides JDBC access via ODBC drivers. ODBC binary code--and in many cases, database client code--must be loaded on each client machine that uses a JDBC-ODBC Bridge. A product called SequeLink from Data Direct Technologies provides a driver that supports some ODBC drivers (for example Microsoft Access).

Type one drivers provide JDBC access via one or more Open Database Connectivity (ODBC) drivers. ODBC, which predates JDBC, is widely used by developers to connect to databases in a non-Java environment.

Pros: A good approach for learning JDBC. May be useful for companies that already have ODBC drivers installed on each client machine— typically the case for Windows-based machines running productivity applications. May be the only way to gain access to some low-end desktop databases.

Cons: Not for large-scale applications. Performance suffers because there’s some overhead associated with the translation work to go from JDBC to ODBC. Doesn’t support all the features of Java. User is limited by the functionality of the underlying ODBC driver.

Type One Driver

A JDBC/ODBC bridge provides JDBC API access through one or more ODBC drivers. Some ODBC native code and in many cases native database client code must be loaded on each client machine that uses this type of driver.

The advantage for using this type of driver is that it allows access to almost any database since the database ODBC drivers are readily available.

Disadvantages for using this type of driver include the following:

- Performance is degraded since the JDBC call goes through the bridge to the ODBC driver then to the native database connectivity interface. The results are then sent back through the reverse process
- Limited Java feature set

- May not be suitable for a large-scale application

Type II: Partial Java driver

This type of driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

This type of driver converts the calls that a developer writes to the JDBC application programming interface into calls that connect to the client machine's application programming interface for a specific database, such as IBM, Informix, Oracle or Sybase.

Pros: Performance is better than that of Type 1, in part because the Type 2 driver contains compiled code that's optimized for the back-end database server's operating system.

Cons: User needs to make sure the JDBC driver of the database vendor is loaded onto each client machine. Must have compiled code for every operating system that the application will run on. Best use is for controlled environments, such as an intranet.

Type Two Driver

A native-API partly Java technology-enabled driver converts JDBC calls into calls on the client API for DBMSs. Like the bridge driver, this style of driver requires that some binary code be loaded on each client machine. An example of this type of driver is the Oracle Thick Driver, which is also called OCI.

Advantages for using this type of driver include the following:

- Allows access to almost any database since the databases ODBC drivers are readily available
- Offers significantly better performance than the JDBC/ODBC Bridge
- Limited Java feature set

Disadvantages for using this type of driver include the following:

- Applicable Client library must be installed
- Type 2 driver shows lower performance than type 3 or 4

Pure Java driver for database middleware

This style of driver translates JDBC calls into the middleware vendor's protocol, which is then translated to a DBMS protocol by a middleware server. The middleware provides connectivity to many different databases.

This driver translates JDBC calls into the middleware vendor's protocol, which is then converted to a database-specific protocol by the middleware server software.

Pros: Better performance than Types 1 and 2. Can be used when a company has multiple databases and wants to use a single JDBC driver to connect to all of them. Server-based, so no need for JDBC driver code on client machine. For performance reasons, the back-end server component is optimized for the operating system on which the database is running.

Cons: Needs some database-specific code on the middleware server. If the middleware must run on different platforms, a Type 4 driver might be more effective.

Type Three Driver

A net-protocol fully Java-enabled driver translates JDBC API calls into a DBMS-independent net protocol which is then translated to a DBMS protocol by a server. This net server middleware is able to connect all of its Java technology-based clients to many different databases. Many mainframe legacy non-relational databases use this kind of driver.

Advantages for using this type of driver include the following:

- Allows access to almost any database since the databases ODBC drivers are readily available
- Offers significantly better performance than the JDBC/ODBC Bridge and Type 2 Drivers
- Advanced Java feature set
- Scalable
- Caching
- Advanced system administration
- Does not require applicable database client libraries

The disadvantage for using this type of driver is that it requires a separate JDBC middleware server to translate specific native-connectivity interface.

Type Four Driver: Direct-to-database pure Java driver

This style of driver converts JDBC calls into a network protocol that sends the converted packets--in a proprietary format--to be used directly by DBMSs, thus allowing a direct call from the client machine to the DBMS server and providing a practical solution for intranet access. This type of driver has become very popular recently and is supported by most database software vendors. All JDBC drivers from Data Direct Technologies (driver vendor) are Type 4 drivers.

Pros: Better performance than Types 1 and 2. No need to install special software on client or server.

Cons: Not optimized for server operating system, so the driver can't take advantage of operating system features. (The driver is optimized for the database and can take advantage of the database vendor's functionality.) User needs a different driver for each different database.

A native-protocol fully Java technology-enabled driver converts JDBC technology calls into the network protocol used by DBMSs directly. This allows a direct call from the client machine to the DBMS server.

Advantages for using this type of driver include the following:

- Allows access to almost any database since the databases ODBC drivers are readily available
- Offers significantly better performance than the JDBC/ODBC Bridge and Type 2 Drivers
- Scalable
- Caching
- Advanced system administration
- Superior performance
- Advance Java feature set
- Does not require applicable database client libraries

The disadvantage for using this type of driver is that each database will require a driver

About the JDBC/ODBC Adapter

The JDBC/ODBC Adapter enables the Sun Java CAPS ESB system to exchange data with external databases. Sun Java CAPS ESB contains many database Adapters. You should use those Adapters to interface with the databases they support. The JDBC/ODBC Adapter should only be used when you have a driver or a database that is not supported by those Adapters.

The JDBC/ODBC Adapter uses Java Collaborations to interact with one or more external databases. By using a Java Collaboration Service it is possible for Sun Java CAPS ESB components such as Adapters to connect to external databases and execute business rules.

About Sun Adapter for Oracle

This topic provides information about the Oracle database and its Sun Adapter.

About Oracle Databases

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

About the Oracle Adapter

The Oracle Adapter enables Sun Java CAPS ESB Projects to exchange data with external Oracle databases.

About Sun Adapter for SQL Server

This topic provides information about Sun Adapter for SQL Server.

About SQL Server

SQL Server is Microsoft's® client-server Relational Data Base Management System (RDBMS), used for increased scalability, availability, and security of enterprise data and analytical applications while making them easier to create, deploy, and manage.

About the SQL Server Adapter

The SQL Server Adapter enables Sun Java CAPS ESB Projects to exchange data with external SQL Server databases. This user's guide describes how to install and configure the SQL Server Adapter.

The SQL Server Adapter uses JCDs (Java Collaboration Definitions) and BPEL (Business Process Execution Language) to perform the following:

- Query a database
- Automatically generate a graphical user interface (GUI) tree representation of database access objects
- Populate the structure with the actual data values during run time.

The SQL Server Adapter also uses the same GUI structure as the rest of the Sun Java CAPS ESB system to describe data flow through the entire enterprise. This feature enables business analysts to define the relationships between a database and relevant applications by dragging and dropping elements between graphical tree structures.

About Sun Adapter for Sybase

This topic provides information about Sun Adapter for Sybase.

About Sybase

Sybase Adaptive Server Enterprise (ASE) is a powerful data management platform for high performance business applications.

Sybase ASE's highly reliable data management technology provides a powerful data management platform that supports the demanding needs of mission critical enterprises, accelerating application development, securing critical company and customer data, and easing data administration tasks.

About the Sybase Adapter

The Sybase Adapter enables Sun Java CAPS ESB Projects to exchange data with external Sybase databases. This user's guide describes how to install and configure the Sybase Adapter.

The Sybase Adapter uses JCDs (Java Collaboration Definitions) and BPEL (Business Process Execution Language) to perform the following:

- Query or Update a database
- Automatically generate a graphical user interface (GUI) tree representation of database access objects
- Populate the structure with the actual data values during run time.

Though SQL coding is not required, the Sybase Adapter also supports a full set of SQL functions for advanced users.

The Sybase Adapter also uses the same GUI structure as the rest of the Sun Java CAPS ESB system to describe data flow through the entire enterprise. This feature enables business analysts to define the relationships between a database and relevant applications by dragging and dropping elements between graphical tree structures.

About Sun Adapter for VSAM

This topic provides information about Sun Adapter for VSAM.

About VSAM

Virtual Storage Access Method (VSAM) is one of several access methods that defines the technique by which data is stored and retrieved. It is a GET/PUT interface used to transfer data from a direct-access storage device (DASD) to an application program. VSAM does not support data stored on tape.

VSAM stores data as a collection of data sets. IBM uses a collection of three data-set organizations—sequential, indexed, and direct-access, together with the access methods and utilities used on mainframe operating systems.

VSAM data sets must be cataloged in an integrated catalog facility (ICF) structure. Records are arranged by an index key or by relative-byte addressing. VSAM uses direct or sequential processing of fixed and variable-length records stored on a DASD.

VSAM also provides the following features:

- Allows you to organize and access data (record management).
- Maintains information about this data, which is stored or referenced in a catalog (catalog management).
- Supports Entry-sequenced Data Set (ESDS) and Key-sequenced Data Set (KSDS) files.

The VSAM Adapter allows you to utilize and extend all of these features. For more information on VSAM, see IBM's Web site.

About the VSAM Adapter

The VSAM Adapter enables Sun Java CAPS ESB to exchange data with external IBM VSAM mainframe file systems. This Adapter is similar to other database Adapters, such as IAM, ADABAS, and IDMS. The Adapter uses its own properties settings and Sun Java CAPS ESB components to enable VSAM data integration.

Adapter Operation

This Adapter enables Sun Java CAPS ESB to communicate with VSAM files on a mainframe host system via TCP/IP. The Adapter utilizes an IBM WebSphere Information Integrator Classic Federation component that allows you to view and access VSAM in the same way as any standard relational database system.