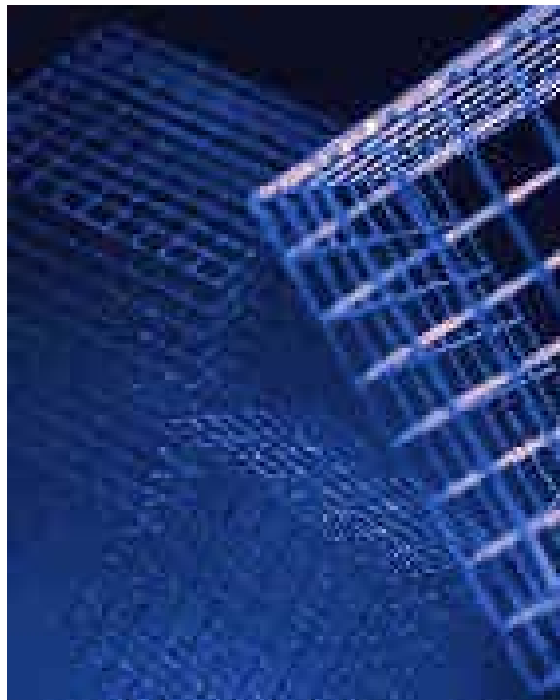


SQL Database Connectivity for Application and Interface Development: SAP DB



© Copyright 2003 SAP AG.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation.

For more information on the GNU Free Documentation License see
<http://www.gnu.org/copyleft/fdl.html#SEC4>.

Contents

SQL Database Connectivity for Application and Interface Development

How SQLDBC Works	4
Architecture	5
Features	6
Installation of SQLDBC	6

SQL Database Connectivity for Application and Interface Development

SQL Database Connectivity (SQLDBC) is a runtime library that enables applications to execute SQL statements in the database, and to access and modify data. The object-oriented structure of SQLDBC was developed in a similar way to the architecture of ODBC and JDBC.

Due to their complexity, ODBC and JDBC have been considered as slow. SQLDBC, on the other hand, has a thin layer of abstraction and a correspondingly high level of performance. This makes it an ideal basis for native interfaces such as ODBC or Perl. However, you can also implement SQLDBC for pure application developments.

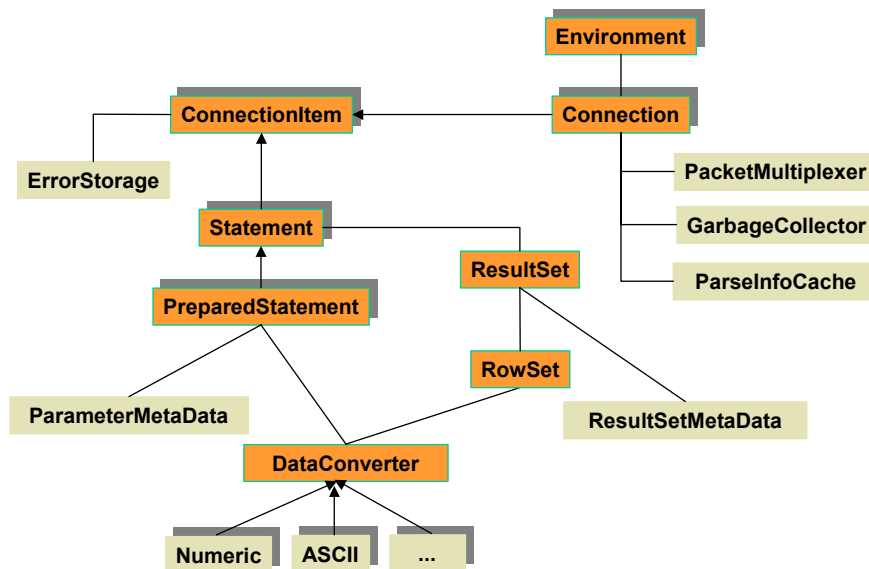
When the SQLDBC classes were being designed, particular attention was paid to guaranteeing the highest possible level of downward or upward compatibility with new developments. For this reason, if you implement the SQLDBC as a shared library, you will probably not need to recompile the application.

SQLDBC supports all database versions from 7.3.00.

How SQLDBC Works

SQLDBC encapsulates the database communication protocol, so simplifying direct access to the database. The SAP DB communication protocol (the Order Interface) uses data structures for its communication, which are dispatched as TCP/IP packets. The application receives the response packets from the database, interprets them, and converts them. This simplifies the processing of SQL statements. For this reason, any applications and interfaces that are based on SQLDBC do not need to recognize the SAP DB communication protocol to be able to use the SQLDBC classes and methods. Also, these applications and interfaces do not need to be adjusted if any changes are made to the communication protocol in the future.

Architecture



Schematic Representation of the SQLDBC Class Diagram

Just like ODBC and JDBC, the SQLDBC shown above has a tree-like structure with classes of the types Environment, Connection, Statement, PreparedStatement, ResultSet, RowSet, ParameterMetaData, and ResultSetMetadata.

The Environment class manages the database connections. A separate interface for memory management is also available.

You can use the Connection class to set up connections to databases, while at the same time defining various properties for the connections (such as SQLMODE, ISOLATIONLEVEL, and AUTOCOMMIT). Within this class, the administration structures of SQL statements are cached (in the ParseInfoCache), which significantly reduces the time needed to execute the statements.

The Connection class also manages the Statement and PreparedStatement objects needed for executing SQL statements. The methods in the Statement and PreparedStatement classes can be used to execute SQL statements and process the result sets of SELECTs and database procedures. The application can use row sets (RowSets) to navigate within the result sets (ResultSets).

You can execute SQL statements either as single commands or mass commands (in JDBC: batchExecute function). Mass commands improve the runtime responses of the application and minimize the volume of data in the network.

The data converter converts the C++ program variables and data structures that are used in the application into the parameters and columns embedded in the SQL statements, and also converts these parameters and columns back into program variables and data structures.

The ParameterMetaData and ResultSetMetadata classes provide information about the parameters and columns of SQL statements, such as their data type and length.

Features

- Object-oriented architecture
- Available in both 32-bit and 64-bit variants
- Multithreading
This allows to develop applications, which can execute SQL statements in parallel or terminate long-running SQL statements.
- Synchronization within a multithreaded environment at the connection level
This makes multiple database connections independent of each other.
- No exception errors
Instead, if errors occur, return codes, error messages, and warnings are displayed.
- Unlimited number of connections, and an unlimited number of SQL statements for each connection
- You can execute SQL statements as single commands or mass commands.
- You can navigate within result sets.
- You can modify the result sets.
- A wide range of SQL data types and C data types is supported (such as data types for Native C, UCS2, UTF8, ODBC, and R/3).
- All character data types, including LONGs and binaries, can be written and read piece by piece.
- Complete UNICODE compliance (UCS2/UTF8)
- PreparedStatements are cached internally in the ParseInfoCache to minimize the number of requests.
- A packet multiplexer limits the amount of copy operations.
- Packet optimizer
 - Use of several segments
 - Utilization of free packet space for clean-up operations (such as PARSEIDs and CLOSE cursors)
 - Space optimization with mixed modes (UNICODE/ASCII)

Installation of SQLDBC

1. Download the software from the ftp server. check whether your browser changes the package extension from `tgz` to `tar` during the download. If so, rename the package to `tgz` before installing it.
Microsoft Windows:
ftp://ftp.sap.com/pub/sapdb/bin/win/sapdb-sqldbc-win-32bit-i386-7_4_4_3beta.tgz
Linux:
ftp://ftp.sap.com/pub/sapdb/bin/linux/sapdb-sqldbc-linux-32bit-i386-7_4_4_3beta.tgz
2. Download the interface documentation from
ftp://ftp.sap.com/pub/sapdb/misc/sapdb-sqldbc-htmldoc-7_4_4_3beta.tgz
3. Unpack all the software in a local directory

For working with SAP DB SQLDBC, open the documentation file

`<local_directory>/html/index.html`.

Then choose *Related Pages -> Application development with SQLDBC*.