



D H A R M A **SQL**tm

Query Technology for Database Developers

Dharma/SQL

Dharma/SQL is a suite of advanced query technology components for database developers who need a high performance SQL 92 interface to provide integration or access to their database. Dharma/SQL is comprised of the SQLengine, ODBCnet, JDBCnet and SQLtools.

These components are designed for databases that need to support high volume transaction systems and query intensive data warehouse type applications. They are production proven by a large installed base of users and offer performance parity with industry leading database vendors.

Dharma/SQL provides a high performance, functionally-complete SQL interface compatible with the SQL-92, ODBC 3.5, OLE DB, and JDBC 3.0 standard, to any relational or non-relational database management system (DBMS). The philosophy of Dharma/SQL storage system interface is to provide a framework that takes maximum advantage of the underlying DBMS. It includes optimizations that take advantage of the strength of your database and allows you to preserve your existing investment in tools and applications that run on it.

Dharma/SQL offers full-featured SQL technology for a fraction of the cost of enhancing and maintaining your own implementation, with added extras like Java Stored Procedures and Java Triggers. These innovative SQL components are distributed in source-code format, providing developers with the flexibility to create a wide variety of custom solutions.

Highlights

SQLengine is a high performance optimizing engine that is the core component in Dharma's advanced query technology.

ODBCnet provides ODBC 3.5 and OLE DB compatibility with hundreds of Windows applications and development tools

JDBCnet is a Type 4 driver that provides a standard interface to any Java application

SQLtools include server-based interfaces and utilities that support application development and administration for your DBMS.

Our commitment to you is continued conformance with evolving SQL standards and interoperability with current and future ODBC, JDBC and OLE DB applications. This dedication guarantees high performance, optimized query execution and added functionality will be available to you and your customers in the future.

Dharma/SQL frees you to focus on your core strengths, confident that the SQL functionality is on par with the leading database vendors.

DHARMA SQLENGINE

Optimized join performance

Dynamic indexing creates indices 'on the fly' where no index exists for faster column joins.

Fast SQL execution

System performance features like SQL and metadata caching eliminate accessing and re-optimizing either the SQL commands or the system metadata for every query.

Minimal system impact

For instance, data pipelining streamlines the system processing by working on one piece of data at a time - instead of trying to process the whole command at once.

Reduced query times

Optimization techniques such as join-order optimization automatically calculate the order and priority of the tables to maximize query speed.

Optimal use of your data source capabilities

Capabilities like push-down restrictions take advantage of the strengths of your data source.

Dharma SQLengine, the core component in our advanced query technology, is a high performance optimizing SQL engine that provides the ultimate in power and flexibility. As part of our Dharma/SQL product, it has been designed to plug into any relational or non-relational database architecture and provide a functionally-complete SQL interface compatible with the SQL-92 standard.

The SQLengine component allows development organizations to build more scalable, high performance, open systems by providing the flexibility to access the full power of the database using SQL. Faster development and deployment is the result of giving your developers and users more flexible and easy, standard access to your database.

Tailored for high demanding production environments, the SQLengine includes optimizations found in industry-leading commercial DBMS solutions. Sophisticated query rewrite techniques improve nested query performance. Join-order optimization improves performance of queries joining many tables. The SQLengine extensively caches and buffers information for maximum transaction and query throughput.

The SQLengine component provides innovative optimizations geared towards the needs of your database. In environments where index creation is not possible, Dharma SQLengine dynamically builds indexes to dramatically improve join performance. In environments with built-in navigation elements, the Dharma SQLengine optimizer uses hash joins to take advantage of the proprietary navigation structures.

Dharma SQLengine is known for its reliability and for our continuing product enhancements and available support.

DHARMA ODBCNET

The Dharma ODBCnet component brings ODBC and OLE DB compliant access to your database in conjunction with the Dharma SQLengine component.

It provides wide compatibility with hundreds of Windows query, reporting, Web and development tools across a variety of platforms. ODBCnet supports an ODBC Version 3.5 driver on Windows and UNIX clients and provides networking and server-side software that manages interaction with the SQL interface to your database.

Dharma's extensive ODBC regression tests total over 200,000 lines of code. These tests insure continuing compliance and robustness with the latest standards. Dharma maintains marketing relationships with key tool vendors and tests pre-release versions of their products to insure continued compatibility.

Dharma ODBCnet supports "Extended SQL Grammar", the broadest level of the ODBC standard. This includes nested queries, recursive views, case expressions, outer joins, positioned updates, union, intersect and minus set operations, ODBC-

compatible scalar functions, extended data types, and procedure calls. Extended SQL provides high-end decision-support functionality with the power to answer complex business questions.

Dharma ODBCnet is tailored for high performance production environments, including online transaction processing and decision support applications. Specialized stored procedures and client-side caching give you efficient metadata access. Optimized client/server interactions provide efficient online transaction processing. Extensive client-side buffering reduces network traffic resulting in fast ad-hoc queries. Dharma's performance tests show that ODBCnet gives equal or superior performance compared to other leading vendors.

DHARMA JDBCNET

Dharma JDBCnet provides an industry standard means of connecting to the SQLengine component. Java Database Connectivity (JDBC) has emerged as the preferred method for Web browser access to databases. With more developers using the Java programming language, the need to access databases from Java will continue to grow. JDBC lets database vendors enhance their DBMS for the Internet. JDBC does for thin clients what Microsoft's ODBC (Open Database Connectivity) did for Windows: it provides access to your DBMS from Web-browser or Java based applications.

Dharma was among the first vendors to embrace the JDBC standard and supply a JDBC driver as part of its SQL component technology. Dharma JDBCnet continues that commitment to JDBC. The latest release provides a "Type 4" JDBC Driver written completely in Java. This type of driver requires no software on client browsers and offers superior performance over other types of JDBC Drivers.

JDBC specifies a single Application Programming Interface for Java programs to access any data source. JDBC does for Intranets what ODBC did for Windows®: it provides a way for programmers to develop a single Web tool that can access different vendors' databases.

JDBC masks the differences between different data sources and provides a standard interface to any Java application. With JDBC and Java, developers can quickly develop and deploy applications that are portable across various databases, including yours.

100% Pure Java™ JDBC Driver

Dharma JDBCnet is a 100% Pure Java JDBC driver. Unlike other types of JDBC drivers, Dharma JDBCnet is written completely in Java and does not rely on third-party network middleware.

Instead, the JDBCnet converts JDBC calls directly into the Java network protocol. This type of driver has the following advantages:

- Because there is no client-resident software, the installation and administration hassles of ODBC drivers are a thing of the past.
- Because there is no third-party network middleware, JDBCnet provides superior performance to multi-tier JDBC drivers.

DHARMA SQLTOOLS

Dharma SQLtools is a suite of tools that supports application development and administration for your database. These tools provide the flexibility to develop server-based applications as well as providing server-based utilities. Embedded SQL and interactive SQL interfaces allow the creation of custom administrative tools and three-tier applications. Quick, interactive or embedded SQL access speeds development and allows simple reporting. Data loading and extracting utilities are administration tools that allow you to automate loading and exporting data and exporting schema definitions - giving you the flexibility to develop server-based batch applications. All of these tools help to preserve your investment in existing SQL applications or build new ones.

DHARMA SQL ENGINE

TECHNICAL DATA SHEET

Optimizations:

Transaction Processing Optimizations

- SMP (symmetric multi-processor) support provides transaction scalability.
- Compiled/cached SQL for repeated queries.
- Metadata and security information caching.
- Execution plan data pipelining.
- Pre-computed joins take advantage of built-in navigation elements.

Data Warehousing Optimizations

- Star-join schema optimization.
- Join order/method selection for complex decision support queries.
- Query rewrite techniques improve nested query performance.
- Join hints for influencing query execution plans.
- Multi-index/bitmap support for compound-predicate queries.
- Dynamic Indexing for multi-table joins.
- Index usage for scalar functions.

Statistics & Cost-Based Optimizations

- Table cardinality
- Query selectivity
- Query access costs
- Data distribution histograms
- Dynamic database statistics

Standards Conformance:

SQL-92

Complete SQL-92 entry level support and support for these SQL-92 Intermediate features:

Dynamic SQL	GET DIAGNOSTICS support
CASE expressions	INTERSECT, MINUS set operators
ALTER INDEX	Extensive character-string functions
SET SCHEMA	Date/time/interval types
CAST functionality	Variable-length character strings
Nested queries	Transaction isolation levels
Recursive views	Schema manipulation statements

SQL-92 Internationalization features:

Multi-byte characters	Multiple character-set support
Unicode support	Explicit collation sequences
Translation functions	

Full SQL-92 data type support, including unstructured LONG types

Server-side Java Functionality:

Java Stored Procedures

Compiled and stored in the DBMS in bytecode format for direct execution without recompilation and parsing. Can import any JDK-supplied and third-party Java libraries. SQL statements are compiled and cached for optimum performance. Java stored procedures can return result sets to calling applications or procedures.

Java Triggers

Execute as a result of specified database modifications made through SQL or other interfaces.

Security

GRANT and REVOKE support: column, table, and view level
Inherits underlying DBMS-based user names
Inherits underlying DBMS-based security and authentication

ODBC 3.5

ODBC extended SQL grammar
ODBC Core, Level 1, and most Level 2 APIs
Extensive scalar functions

OLE DB

OLE DB v1.1 Driver Manager

JDBC 3.0

Fully supported, pure-Java driver

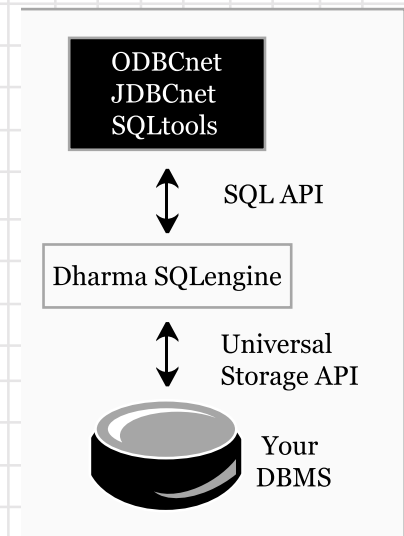
Universal Storage API

Index and navigation properties

- Mandatory components for multi-component indexes
- Supported sort orders
- Row-level locking support
- Stability of scan cursors
- Row-id sorting behavior
- Aggregate function capabilities
- Push-down predicate operator processing
- Multi-record append capability
- Multi-record fetch capability
- Supported relational operators
- Ability to update index

Statistics

Information about cost of access methods and distribution of data among and within tables



Current Version: 09.00.0000

Latest Release:

March, 2003

Distribution Format:

Source Code

Supported Operating

Systems: Windows, Sun Solaris, IBM AIX, HP-UX, Linux

DHARMA ODBCNET TECHNICAL DATA SHEET

Flexible Configuration Support

Dharma ODBCnet supports both client/server and three-tier configurations for Internet/Intranet environments. The ODBCnet driver runs on Windows. This flexibility insures ODBCnet supports your present and future needs.

By providing a client/server environment, Windows tools can access previously-unavailable data in proprietary databases. Three-tier configurations give you the flexibility to access proprietary data through a Web browser, and push business-logic processing to the application server.

Data Type Support

Dharma ODBCnet supports all ODBC data types, including LONG types of arbitrary length and structure.

ODBC 3.5 Support

The ODBCnet Driver supports Core, Level 1, and Level 2 API functions, plus extensive scalar functions.

Scalar Functions

String Scalar Functions

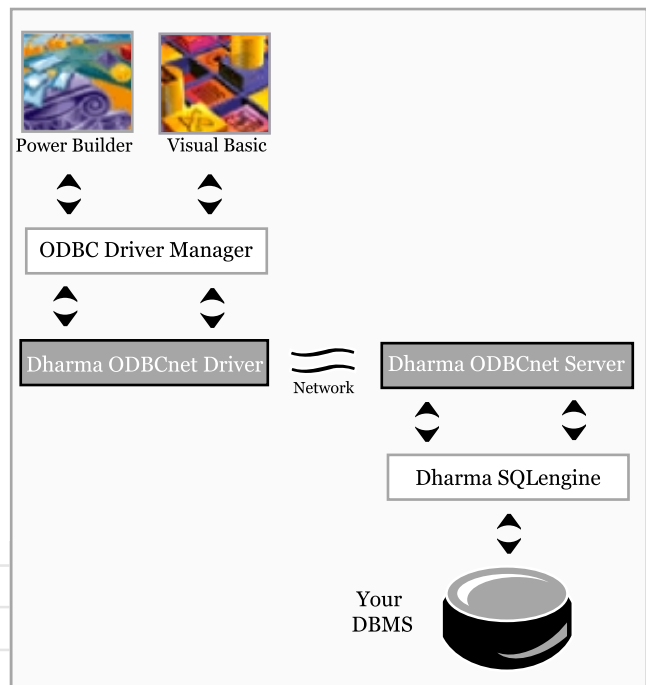
ASCII	CHAR	CHAR_LENGTH	CHARACTER_LENGTH
CONCAT	DIFFERENCE	INSERT	LCASE
LEFT	LENGTH	LOCATE	LTRIM
REPEAT	REPLACE	RIGHT	SOUNDEX
SPACE	SUBSTRING	UCASE	

Time and Date Functions

CURDATE	CURTIME	DAYNAME
DAYOFMONTH	DAYOFWEEK	DAYOFYEAR
HOUR	MINUTE	MONTH
MONTHNAME	NOW	QUARTER
SECOND	WEEK	YEAR
TIMESTAMPADD	TIMESTAMPDIFF	

Numeric Functions

ABS	ACOS	ASIN
ATAN	ATAN2	CEILING
COS	DEGREES	EXP
FLOOR	LOG	LOG10
MOD	PI	POWER
RADIANS	RAND	SIGN
SIN	SQRT	TAN



DHARMA JDBCNET

TECHNICAL DATA SHEET

Supported Functions

JDBCnet supports an extensive set of scalar functions, including the following:

Supported Numeric Functions:

ABS	ATAN	COS	COS	FLOOR	PI
RAND	SQRT	ACOS	ATAN2	DEGREES	LOG10
DEGREES	SIGN	POWER	TAN	ASIN	EXP
EXP CEILING	MOD	RADIANS	SIN	TRUNCATE	

Supported String Functions:

ASCII	CHAR	CONCAT	DIFFERENCE	INSERT	LCASE
LEFT	LENGTH	LOCATE	LTRIM	REPEAT	REPLACE
RTRIM	SOUNDEX	SPACE	SUBSTRING	UCASE	

Supported TimeDate Functions:

CURDATE	CURTIME	DAYOFMONTH	DAYOFWEEK	DAYOFYEAR	HOUR
MINUTE	MONTHNOW	QUARTER	SECOND	WEEK	YEAR

Supported System Functions:

USERNAME	IFNULL	DBNAME
----------	--------	--------

Supported Configurations

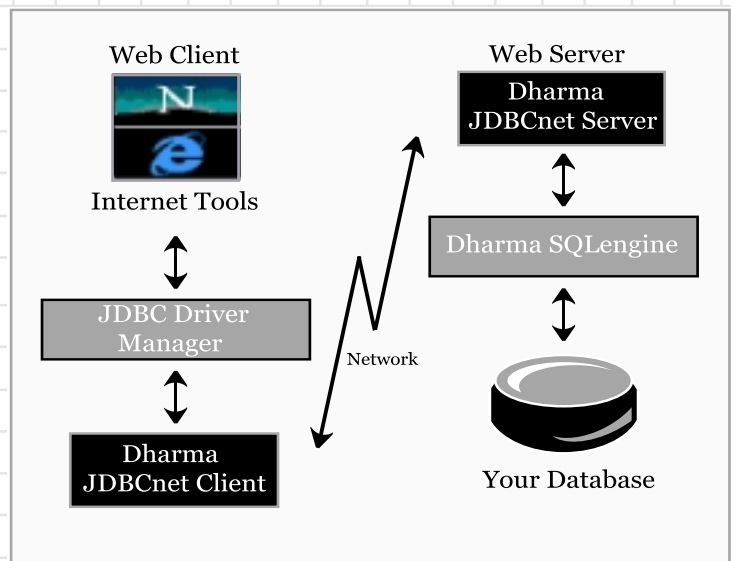
JDBCnet supports both two-tier and three-tier configurations. In a two-tier configuration, the JDBC driver runs on the Web browser and communicates with the DBMS running on a server. In a three-tier configuration, the JDBC Driver runs on a middle tier (typically called the application server) and communicates with the DBMS running on the third tier.

Because Dharma's JDBCnet is a pure-Java JDBC Driver, it can support either configuration with no modifications.

Supported Data Types

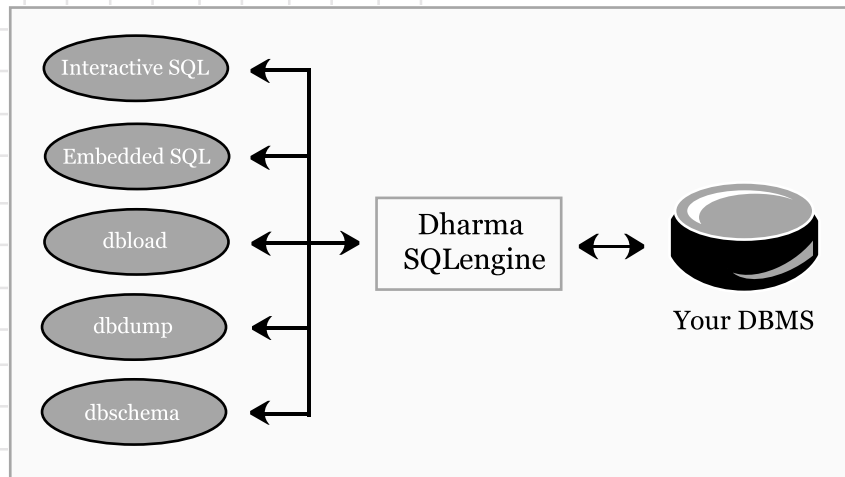
Dharma JDBCnet supports all JDBC data types, including LONG types of arbitrary length and structure:

JDBC type	Java type
CHAR	String
LONGVARCHAR	String
DECIMAL	java.math.BigDecimal
TINYINT	byte
INTEGER	int
REAL	float
DOUBLE	double
VARBINARY	byte
DATE	java.sql.Date
TIMESTAMP	java.sql.Timestamp
VARCHAR	String
NUMERIC	java.math.BigDecimal
BIT	boolean
SMALLINT	short
BIGINT	long
FLOAT	double



DHARMA SQLTOOLS

TECHNICAL DATA SHEET



```

$ isql sampledb
Dharma/ISQL Version 09.00.0000
Dharma Systems Inc (C) 1988-2003.
Dharma Computers Pvt Ltd (C) 1988-2003.
ISQL> CREATE TABLE TEST (C1 CHAR(30));
ISQL> INSERT INTO TEST VALUES('Dharma');
1 record inserted.
ISQL> SELECT * FROM TEST;
C1
—
Dharma
1 record selected
  
```

```

EXEC SQL
SELECT 1
INTO :result
FROM systpe.syscalctable
WHERE :ship_date > SYSDATE
AND :ship_date < ADD_MONTHS (:order_date, 1) ;
if (sqlca.sqlcode == 0)
printf ("ship_date valid ") ;
else
if (sqlca.sqlcode == SQL_NOT_FOUND)
printf ("ship_date invalid ") ;
else
printf ("Error ") ;
  
```

SQLtools/Interactive SQL

You can use interactive SQL to port SQL scripts to your database, generate formatted reports and develop administrative scripts.

Invoke the interactive SQL tool at the shell prompt by typing isql and a database name, as shown in the example to the right.

Interactive SQL includes an online help facility with syntax and descriptions of the supported statements.

SQLtools/Embedded SQL

The embedded SQL compiler lets you port applications from other DBMSs to your DBMS.

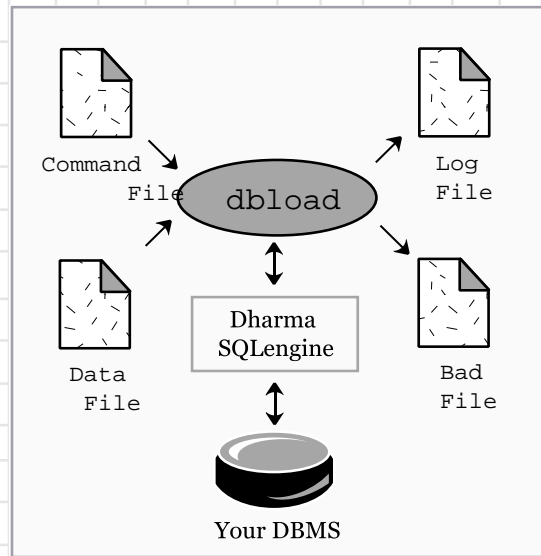
The precompiler processes C language program source files with embedded SQL statements. It converts the SQL statements to C language calls and invokes the C compiler to generate object or executable files.

Precede embedded SQL statements with EXEC SQL flags and terminate them with a semicolon (;). You can insert the SQL statements among C constructs, as shown in the program excerpt to the right.

SQLtools/dbload

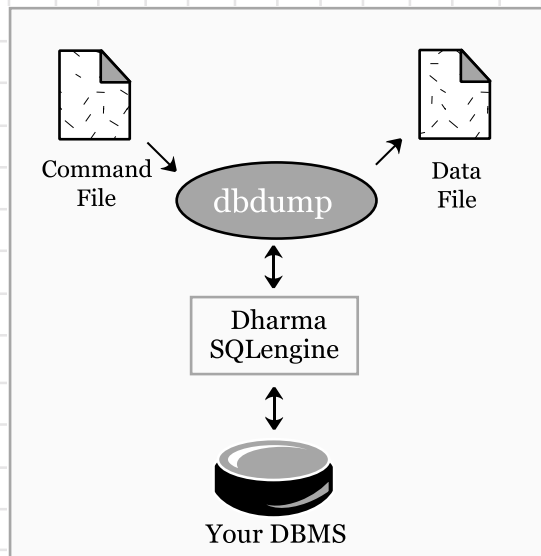
The dbload utility loads records from an input data file into your DBMS. The dbload utility reads the input data file and processes its contents into data records that it loads into the database. The format of the data file is specified by a record description given in an input command file to dbload.

Dbload generates a log file with error messages and statistics, and a badfile to hold any records that do not comply with the format specified in the command file.



SQLtools/dbdump

The dbdump utility exports the contents of tables in your DBMS. The format of the exported data is specified by the record description given in an input command file to dbdump. The command file specifies the data file, the format of data records, and the destination (or source) database columns and tables for the data.



SQLtools/dbschema

The dbschema utility generates schema definition information for specified tables and views, or all the objects in your DBMS. Invoke dbschema at the shell prompt by typing dbschema. The following example writes SQL statements to define and load the table assignments in the database tested to the file assignments.sql:

```
$ dbschema -d -o assignments.sql -t assignments testdb
Dharma/DBSCHEMA Version 09.00.0000
Dharma Systems Inc (C) 1988-2003.
Dharma Computers Pvt Ltd (C) 1988-2003.
$ more assignments.sql
create table systpe.assignments (
c1 integer,
c2 integer,
c3 character
) pctfree 20;
insert into assignments values(1,100,'r');
```