Query Database Requirements

Introduction

The query_database tool allows SQL queries to be run from within IDL code. To support operational use the query_database library will maintain the highest standards of quality, documentation, agility, testing, and rapid deployment.

To aid agility and reduce costs, library functionality will be achieved with a reasonable minima of software code and entities.

IDL API

Version 1.1 of the query_database API will continue to be supported. Changes must not break backward compatibility.

When no data are found for a select query, query_database shall return an undefined data variable and zero rows.

For inserts, updates, and deletes query_database shall return an undefined data variable and a row count for the affected rows. (This can be zero.)

Performance

Must maintain version 1.1 performance.

The /dev/random problem is solved on Unix platforms.

Databases Supported

Database:

- Sybase
- Oracle

Database Actions Supported

The following operations are supported:

- select
- insert
- update
- delete
- stored procedures (currently not all the return values are present for Oracle: planned fix in version 2.2)

IDL and Java Versions

- The query_database tool is supported for versions of IDL currently in use at the lab (at the time of this writing, 7.x and 8.x).
- For compatibility with older versions of IDL (and the IDL-Java bridge), the Java code in query database is compile with Java 1.6
- The software development team will monitor Oracle, Sybase, Java, and IDL releases. If a new release addresses an urgent bug or security problem, we will issue a quick release of query_database with the upgraded component, and notify ops. For non-urgent updates, we will fold latest component release into existing (quarterly) query_database release schedule.
- Support for down rev versions will not be dropped until key users (including ops and solar_tools library) are ready.

User Credentials

Multiple user credentials must be supported for each user. (tsis_reader, tcte_reader, etc.) The file formats for credentials are detailed in Query Database single and multi-database resource file formats

- Login information (user names/passwords) shall not be stored in anything distributed to the user.
- The query_database calls must support automated, unattended operation.

- Location of configuration files is configurable by the user.
- The order of precedence for user credentials shall be as follows:
    1. Credentials explicitly provided through IDL keywords such as 'user', 'password', 'dbloginfile', etc. passed to the IDL query_database procedures
    2. Credentials found in a file whose full path (including filename) is specified by an environment variable named 'DB_LOGIN_FILE'
    3. Credentials found in the current working directory in a file named ".qdbResources"
    4. Credentials found in the current working directory in a file named ".dbLogin"
    5. Credentials found in the user's home directory in a file named ".qdbResources"
    6. Credentials found in the user's home directory in a file named ".dbLogin"
- Once a connection is opened, it remains open until reset or explicitly closed.

Settings

Default settings for all connections

Although the effect is only noticed for Oracle connections made from a Linux client, the following Java Virtual Machine (JVM) property is always set at system initialization:

1. java.security.egd=file:///dev/urandom

Default settings for Sybase connections

1. TEXTSIZE=2147483647
2. HOSTNAME=system:user.name:os.name
3. SQLINITSTRING=set quoted_identifier off
4. ENCRYPT_PASSWORD=false

Default settings for Oracle connections

1. oracle.net.READ_TIMEOUT=30000
2. oracle.jdbc.READ_TIMEOUT=30000
3. oracle.net.CONNECT_TIMEOUT=60000

Changes in the way settings will be handled in version 2.3

1. System properties may be set in the query_database .qdbResources file. (version 2.3)
2. Fetch-size and other JDBC connection properties are user configurable. (version 2.3)
3. When system settings are set, previous values of the settings are captured and restored when possible. (version 2.3)

Supported Platforms

- Linux
- Windows
- Mac

Documentation

- Documentation is on confluence.
- Documentation will be provided in the release as a PDF.

Distribution

- The query_database package will be versioned, tested, documented, and released.
- New revisions have drop-in capability and are fully backward compatible.
- Each release contains accurate and complete documentation, including release notes.
- The query_database will be released on a quarterly schedule.
- The ops group has additional distribution requirements which will be met with a special additional distribution in the near term.
- For the ops team distribution, all IDL code must live in a single file, called query_database2.pro.
- Jar files and IDL code will not change names between releases.
- Neither the Sybase nor the Oracle database jar files are not need for the normal distribution. At the ops team's request, jar files are not distributed to them.

Testing

- The query_database package is used operationally at LASP. It is necessary that the software be thoroughly tested. See Query Database Jenkins Unit testing strategy
- Continuous integration (Jenkins) is used by developers.
- Package is tested on Linux and Windows.
- Package is tested with Oracle and IDL JREs.
- Package is tested with IDL 8.
- Full test records are kept and are available to users.