User's Guide for RDA/SQL Validation Tests (Version 1.0)

Kevin Brady
Joan Sullivan

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001
User's Guide for RDA/SQL Validation Tests (Version 1.0)

Kevin Brady
Joan Sullivan

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001

December 1996
ABSTRACT: The RDA/SQL Validation Tests, developed by NIST, consist of a set of C programs designed to test an RDA/SQL server for conformance to the international standards for Remote Database Access with an SQL Specialization. Testing is limited to the RDA Basic Application Context/Immediate Execution profile defined by the NIST OIW Stable Agreements for OSI Protocols. The validation tests use software tools provided in the public domain ISO Development Environment (ISODE) and has been operating at NIST on TCP/IP networks using the Internet Engineering Task Force (IETF) specification RFC 1006.

KEYWORDS: conformance testing; database standards; interoperability; remote database access; RDA; SQL; RFC 1006; ISODE; testing of software; validation of software

DISCLAIMER: Because of the nature of this report, it is necessary to mention vendors and commercial products. The presence or absence of a particular trade name product does not imply criticism or endorsement by the National Institute of Standards and Technology, nor does it imply that the products identified are necessarily the best available.
Table of Contents

1. INTRODUCTION .................................................. 1
   1.1 Background ............................................... 1
   1.2 Scope of Testing ....................................... 4

2. STRUCTURE OF RDA/SQL VALIDATION TESTS ..................... 5
   2.1 Description of RDA/SQL Validation Tests ................. 5
   2.2 System Requirements ..................................... 6
   2.3 Directory Structure ..................................... 7
   2.4 File Naming Conventions .................................. 7

3. INSTALLATION .................................................. 7

4. REPORTING ...................................................... 8

5. MAINTENANCE ................................................... 8

6. REFERENCES .................................................... 9

APPENDIX A  Sample RDA/SQL Test Program ......................... 11
APPENDIX B  Sample Test Program Report ........................ 14
APPENDIX C  Sample Test Program Log ............................ 15
APPENDIX D  Sample Summary Report .............................. 24
1. INTRODUCTION

This user's guide describes the RDA/SQL Validation Tests and the procedures needed to test and evaluate an RDA Server implementation for conformance to international standards for Remote Database Access with an SQL Specialization (RDA/SQL).

1.1 Background

The purpose of the RDA specification is to standardize a method for SQL clients to access and modify the data on a remote SQL server. Many SQL products support the client/server architecture, where the user application and the SQL data server reside on separate computers. Each SQL software developer has proprietary formats and protocols for client/server communications. As a result, client and server products from different SQL software developers do not interoperate.

RDA is a standard that describes the communications protocol used between database clients and servers; i.e., what flows "over the wire". An implementation of RDA for a client application can be described as a layer of software supporting the application by formatting the application's SQL requests for data into standard messages. The SQL statement itself, along with input parameter values for the SQL statement, is formatted as a structured message. The data returned to the client is then received as a structured message. The returned message could contain little more than a status code to indicate success or failure. Or, the returned message could additionally contain multiple rows of data, along with a table description (including column names, data types, precision, character set, etc.). RDA allows the client application to request the server to convert data values into data types or character sets appropriate to the client application.

An implementation of an RDA server can be described as a layer of software written on top of the SQL server. The RDA server layer of code unpacks the RDA message structure, moving the SQL statement and parameter values into local variables and then interacting with the SQL server to obtain data or to update the database. The data to be returned to the client is reformatted and restructured into an RDA message by the RDA server. Although the RDA server code could be written using a standard Embedded SQL C or CLI interface
to an SQL database, a native (proprietary) interface to the SQL database is typically chosen by the software developer.

RDA is an international standard that specifies the functionality of a database server within a distributed open systems environment. RDA also specifies the communication service and protocol for accessing its capabilities. RDA is a two-part Standard [1]. Part 1 specifies service and protocol for a generic model. Part 2 specifies the SQL specialization of the service and protocol and refers to the SQL standard [2] to define the SQL language contained in the RDA messages. The structured messages between the client and server use ASN.1 (Abstract Syntax Notation 1, defined in ISO/IEC 8824) to specify their format and the associated binary encoding rules.

RDA is positioned in the Application Layer of the Reference Model of Open Systems Interconnection (OSI) and relies on services provided by lower layers.

Because of this layered approach, RDA service requirements can also be mapped to other communications protocols, such as TCP/IP. The current NIST laboratory environment for the RDA/SQL Validation Tests uses the Internet Engineering Task Force (IETF) specification RFC 1006 to interoperate with other RDA/SQL clients and servers over the Internet. Figure 1 shows the RDA service layers for client/server architectures, although, in theory, an RDA client or service product could support both TCP/IP and OSI protocols, it is probable that each product will support only one of the protocols.
Although a thorough test of the database server's support for the SQL language is also needed, this set of tests does not attempt that level of testing. Instead, a separate evaluation of the SQL server should be conducted using the NIST SQL Test Suite [3].

The NIST RDA/SQL laboratory environment in which the validation tests were developed is part of an RDA testbed announced in the February 8, 1994, Federal Register. As a result of research into RDA by NIST and other organizations, additional prototype tools and demonstration software have been developed:

- distributed query processor
- GUI monitor for distributed query processor
- RDA server for Informix
- RDA server for Oracle
- RDA server for Sybase
- RDA sniffer
- RDA GUI client
- RDA applications

The distributed query processor and GUI monitor, as well as RDA
servers for SQL products (Informix, Oracle, and Sybase), are enhancements of software donated by NRaD (Naval Ocean Surveillance R&D, San Diego, CA).

1.2 Scope of Testing

All of the test programs validate standard behavior of an RDA server with access to an SQL database; i.e., the RDA server layer of code is not tested in stand-alone mode. Also, an RDA client product that may be sold with the RDA server, is not tested by these programs.

Fourteen different message types are defined in the RDA standard to provide five services. These services and messages are:

- **Dialogue Management services** (to start and end RDA dialogues)
  - Initialize
  - Synchronize
  - Terminate

- **Transaction Management services** (to start and end RDA transactions)
  - Begin Transaction
  - Commit
  - Rollback

- **Control services** (to report the status or cancel existing operations)
  - Cancel
  - Status

- **Resource Handling services** (to enable or disable access by RDA clients to data resources)
  - Open
  - Close

- **Database language services** (to access and modify data resources)
  - Execute DBL (Immediate execution)
  - Define DBL
  - Invoke DBL
  - Drop DBL

Two application contexts are defined for RDA dialogues, depending on whether or not the application requires distributed transactions (transactions spanning two or more servers). Transaction processing services (TP), invoking two-phase commit protocols, are
defined in a separate TP standard [4]. The two application contexts are called:

RDA Basic application-context
RDA TP application-context

Testing is limited to a minimal profile, selecting from the above options only those needed for the simplest access to the full SQL language. The profile selected for testing was defined cooperatively in the NIST Open Systems Environment (OSE) Implementors Workshops (OIW) Stable Implementation Agreements for OSI Protocols [5]. The profile used in RDA/SQL testing is called "Immediate Execution." This profile requires the following functional units: dialogue initialization, dialogue termination, transaction management, resource handling, and immediate execution DBL. Testing is also restricted to RDA Basic Application Context. The OIW RDA agreements provide additional restrictions and specify min-max limits.

Although the RDA standards and the OIW agreements specify access over OSI networks, the RDA/SQL Validation Tests were developed in a TCP/IP laboratory environment and remote testing of products was accomplished over the Internet. In theory, all the software developed by NIST for RDA/SQL testing can be deployed in a strict OSI environment after minor configuration steps.

The SQL/RDA test programs are not comprehensive, due to limited resources. The goal of the set of tests is to ensure that straight-forward SQL access will be successful. The focus of testing has been on exercising the options of the Execute DBL message type (which carries the SQL request and the SQL data) and verifying support for min-max limits. Only modest effort was expended on testing other message types and error conditions.

2. STRUCTURE OF RDA/SQL VALIDATION TESTS

2.1 Description of RDA/SQL Validation Tests

The RDA/SQL Validation Tests are a collection of 74 C-language programs, each testing a specific feature or requirement of the RDA standard. The test programs make liberal use of a library of subroutines, provided in a separate directory. The testbed of tables and data used by the test programs is created by an
initialization program containing SQL statements to create tables and insert rows. Since test programs may alter the testbed, there is another program to assist in restoring the testbed to its initial state. This program simply deletes SQL objects, in order to allow the initialization program to rerun cleanly. Reporting for the RDA/SQL Validation Tests consists of capturing the "PASS" or "FAIL" lines from the program logs and combining these lines into a summary printout.

The RDA/SQL Validation Tests were developed using ISODE tools. Softcopy of the RDA SQL Specialization ASE ASN.1 Module, defined on page 35 of the standard ISO/IEC 9579-2:1993, was used by the ISODE tools to generate C-language header files and encoding/decoding subroutines. These subroutines are used by the test programs. ISODE tools also provide the RFC 1006 communications mapping software needed to run over TCP/IP protocols in the NIST laboratory and over the Internet. In order to execute the RDA/SQL Validation Tests, it is necessary to install the ISODE tools first.

Makefiles and scripts to assist in creating and running the test executables in a Unix environment are provided with the validation tests. For non-Unix platforms, the tester will need to create similar makefiles and scripts. Since Unix platforms are widely available, we recommend installing the validation tests (the RDA/SQL client) on a Unix platform. Keep in mind that the RDA/SQL client and the RDA/SQL server to be tested do NOT need to be compatible platforms. They need have nothing in common except the ability to support the RDA/SQL protocol over TCP/IP (or some other communications protocol).

### 2.2 System Requirements

In order to install and execute the validation tests successfully, the following are suggested:

- ISO conforming RDA server (on the server side)
- ANSI/ISO conforming SQL database (on the server side)
- ISODE Version 8.0 (installed on the client side)
- ANSI/ISO conforming C compiler (on the client side)

Since the RDA/SQL Validation Tests are designed to exercise the features of the RDA server rather than the other components above, a C compiler or SQL database which is not fully conforming may
still be adequate.

2.3 Directory Structure

The files are organized in two directories. The top directory is called CONF (for conformance testing), and it contains only one subdirectory, LIB. Directory CONF contains all the test programs, a Makefile, and a header file (testing.h) for the test programs. The subdirectory LIB contains all the support subroutines called by the test programs to initialize, terminate, encode, decode, build customized messages, check return values, etc.

2.4 File Naming Conventions

The following are naming conventions used in directories CONF and LIB:

- **RDAxxx.c**: a test program
- **RDAxxx.rpt**: a summary of the log created by the program
- **RDAxxx.log**: the log created by executing the program

Please refer to the appendicies for examples as follows:
- Appendix A: Sample RDA/SQL test Program
- Appendix B: Sample Test Program Report
- Appendix C: Sample Test Program Log

Makefiles, subroutines, and header files follow C-language naming conventions. The program to initialize the SQL tables and data is called **CREATE.c**. The program to drop tables prior to a clean restart is called **DROP.c**. The text file **README.out** is documentation of the test specifications for the test programs. **README.out** is generated by program **build_readme.c** which extracts documentation coded as comments in the individual test programs.

3. INSTALLATION

The RDA/SQL Validation Tests software is publicly available over the World Wide Web from the URL

http://www.nist.gov/itl/div897/ctg/rda_form.htm

The test software is a compressed Unix TAR archive. Since the test software is built using ISODE tools, the ISODE compressed Unix TAR archive should also be downloaded from the same site. After downloading, uncompressing, and extracting from the TAR archives,
the next step is to build the test programs.

Install the ISODE tools according to the instructions provided.

There are two files that need to be edited to identify the RDA server(s) to be tested. File isoentities (in the root directory of ISODE) must specify P-selector, S-selector, T-selector, IP address and (optional) port. File load.c (in directory LIB) must specify host name (IP address) and service.

Review the Makefiles in directories LIB and CONF to point to the ISODE library header files. In directory LIB, type "make" to create the C-language library file libconf.a. In the directory CONF, type "make" to create all the C test programs and to execute them in the correct order. Typically, this order is DROP and CREATE, followed by all of the test programs in alphabetical order. Program REPORT is run last to summarize the results from executing the test programs. For additional information about the results of an individual program, read the ".log" and ".rpt" text files created by the test program.

To make a single executable, e.g., RDA001, type "make RDA001". To execute a single executable, e.g., RDA001, type "RDA001 hostname", where hostname matches one of the host names in file load.c.

4. REPORTING

Program REPORT summarizes the results in the log files. The test programs are listed in ascending order with a result value of pass, fail, missing, or aborted. Subtotals for each category are printed at the end of the report. See Appendix D for a sample report.

Program REPORT reads the README.out file to obtain one-line descriptions for the test programs, so program build_readme should be run prior to program REPORT. The REPORT output file name is REPORT.log.

5. MAINTENANCE

Although the test programs have been debugged and executed on several RDA servers, errors in interpretation or in coding are
The program availability is possible. Questions about the RDA/SQL Validation Tests should be directed to:

Joan M. Sullivan  
National Institute of Standards and Technology  
Bldg. 820, Room 562  
Gaithersburg, MD 20899  
phone: (301) 975-3258  
FAX: (301) 948-6213  
joan.sullivan@nist.gov

Maintenance of corrected or enhanced programs will consist of replacing the original program in the online Unix TAR archive. A maintenance log, containing the date and description of each modification, will be available online in the file named update.log.

6. REFERENCES


APPENDIX A  Sample RDA/SQL Test Program

#include "testing.h"

char *name = "RDA025";

/**********************************************************
4.1.7.1.1 R-ExecuteDBL Service
NOTE: The Sniffer will wait on an r-ExecuteDBL-RC for each r-
ExecuteDBL-RI sent to the server. The server "fails" if it
does not respond within the timeout period (some large
default value) and tester does not justify failure to
respond (e.g., network went down).

0025 CT 4 SQLDBLArgumentValues -> 4 SQLDBExceptions, DELETE

DELETE FROM RX
WHERE DRUG_ID = :H AND PAYMENT_METHOD = :H
in: [ArgVal:
  7444 'cash'
  7444 'xxxx'
  0000 'cash'
  7004 'insu']

out: [SQLCODE/SQLSTATE:
  0 '00000'
  100 '02000'
  100 '02000'
  0 '00000']

ROLLBACK WORK

/***********************************************************/

main (argc, argv)

int argc;
char **argv;
{
  int get_host ();

  /***
  /************ Connect to the host ***********************
  /***** Make sure host_under_test is set to host *******/
  /***
    connect_to_host (argv);
  /***/

  /***
  /*** Send an initialize and wait for a response ***/
  /***/

    fill_initialize_ri ();

  /***/

    return 0;
}
strcpy (APDU_0.UserAuData, username);
strcpy (APDU_0.IdeoOfUser, password);
encode_initialize_ri ();
delay (CA);
check_errors ("Initialize-request");

/***
 *** Send an open and wait for a response
 ***
/***/
fill_open_ri ();
strcpy (APDU_15.DataResName, username);
strcpy (APDU_15.UserAuData, password);
encode_open_ri ();
delay (CO);
check_errors ("Open-request");

/***
 *** Send a begin Transaction, no response
 ***
/***/
fill_begintrans_ri ();
encode_begintrans_ri ();
check_errors ("Begin-transaction-request");

/***
 *** Send an EXECUTE and wait for a response
 ***
/***/
fill_executeDBL_ri ();
check_sql_string (APDU_19.StatTex);
APDU_19.multiple_args = TRUE;
add_to_value_list (choice_RDA_17_integerType, "7444", NULL, 4, int_RDA_precisionBase_decimal, NULL);
add_to_value_list (choice_RDA_17_characterType, "cash", NULL, 4, 1, NULL);
add_to_value_list (choice_RDA_17_integerType, "7444", NULL, 4, int_RDA_precisionBase_decimal, NULL);
add_to_value_list (choice_RDA_17_characterType, "xxxx", NULL, 4, 1, NULL);
add_to_value_list (choice_RDA_17_integerType, "0000", NULL, 4, int_RDA_precisionBase_decimal, NULL);
add_to_value_list (choice_RDA_17_characterType, "cash", NULL, 4, 1, NULL);
add_to_value_list (choice_RDA_17_integerType, "7004", NULL, 4, int_RDA_precisionBase_decimal, NULL);
add_to_value_list (choice_RDA_17_characterType, "insu", NULL, 4, 1, NULL);
encode_execute_ri ();
wait_for_response ();
check_resval (1, NULLCP, NULLINT, 0, "00000");
check_resval (2, NULLCP, NULLINT, 100, "02000");
check_resval (3, NULLCP, NULLINT, 100, "02000");
check_resval (4, NULLCP, NULLINT, 0, "00000");
check_errors ("Execute-request");

/***
 *** Rollback, close, terminate and disconnect
 ***
/***/
/***/
    shutdown (host,"R");
    summary();
    check_test_status();
}
APPENDIX B  Sample Test Program Report

9/23 15:47:28 RDA025  05904 (kevin ) Test RDA025 starting - using host = rda
9/23 15:47:31 RDA025  05904 (kevin ) Initialize-request: Passed
9/23 15:47:32 RDA025  05904 (kevin ) Open-request: Passed
9/23 15:47:33 RDA025  05904 (kevin ) Execute-request: Passed

9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Initialize Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Initialize Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Synchronize Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Terminate Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Terminate Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Begintrans Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Begintrans Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Commit Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Commit Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Rollback Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Rollback Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Cancel Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Cancel Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Status Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Status Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Open Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Open Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Close Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Close Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Execute Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 1 Execute Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Define Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Define Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Invoke Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Invoke Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Drop Requests found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) Checked 0 Drop Confirms found 0 bad.
9/23 15:47:33 RDA025  05904 (kevin ) RDA025 : Completion Status: PASSED.
APPENDIX C  Sample Test Program Log

9/23 15:47:28 RDA025 05904 (kevin ) Test RDA025 starting - using host = rda
9/23 15:47:28 RDA025 05904 (kevin ) A-Associate-REQUEST sent to rda

9/23 15:47:28 RDA025 05904 (kevin ) A-ASSOCIATE.REQUEST:
9/23 15:47:28 RDA025 05904 (kevin ) Context : "iso sql rda"
9/23 15:47:28 RDA025 05904 (kevin ) AEI : <1.1.3.9999.9579,,>
9/23 15:47:28 RDA025 05904 (kevin ) Paddress: '0415'\H/Internet=129.6.59.29
9/23 15:47:30 RDA025 05904 (kevin ) A-ASSOCIATE.CONFIRMATION:
9/23 15:47:30 RDA025 05904 (kevin ) File Descriptor : 5
9/23 15:47:30 RDA025 05904 (kevin ) Reason: Accepted
9/23 15:47:30 RDA025 05904 (kevin ) Diagnostic: Rejected by service-user: null
9/23 15:47:30 RDA025 05904 (kevin ) Context : 1.0.9579.2.2.1
9/23 15:47:30 RDA025 05904 (kevin ) Connected to to rda

9/23 15:47:30 RDA025 05904 (kevin ) wrote ---------- RDA-Initialize-Request ----------
RDA-APDU {
    RDA-APDU {
        r-Initialize-RI {
            operationID 1,
            r-Initialize-req {
                dialogueIDSuffix {
                    ostring "SQL1"
                },
                identityOfUser "dqp",
                userAuthenticationData {
                    ostring "dqp"
                },
                controlServiceDataRequested TRUE,
                functionalUnitsRequested { termination, transaction, cancel, status, resource, immediate-DBL, stored-DBL },
                SQLInitializeArgument {
                    SQLConformanceLevelDefault "ISO-SQL-1992-LOW-CONFORMANCE",
                    userData "NIST RDA Client"
                }
            }
        }
    }
}
----------
9/23 15:47:30 RDA025 05904 (kevin ) ---------- RDA-Initialize-Request ----------
9/23 15:47:31 RDA025 05904 (kevin ) STATE: CA
9/23 15:47:31 RDA025 05904 (kevin ) read ---------- RDA-Initialize-Result ----------, context 1
RDA-APDU {
    RDA-APDU {
        r-Initialize-RC {
            operationID 1,
            res-or-err {
                r-Initialize-res {
                    RDA-Initialize-Result
                }
            }
        }
    }
}

---
controlServiceData {
    controlServicesAllowed FALSE
}

functionalUnitsAllowed { termination, transaction, cancel, status, resource
immediate-DBL, stored-DBL },

sQLInitializeResult { 
    userData "NIST RDA Server: rda.ncsl.nist.gov - Oracle 7.2.1"
}

---------- RDA-Initialize-Result ----------
9/23 15:47:31 RDA025 05904 (kevin ) Operation ID : 1
9/23 15:47:31 RDA025 05904 (kevin ) control service allowed : FALSE
9/23 15:47:31 RDA025 05904 (kevin ) Functional Units Present :
9/23 15:47:31 RDA025 05904 (kevin ) Termination
9/23 15:47:31 RDA025 05904 (kevin ) Transaction
9/23 15:47:31 RDA025 05904 (kevin ) Cancel
9/23 15:47:31 RDA025 05904 (kevin ) Status
9/23 15:47:31 RDA025 05904 (kevin ) Resource
9/23 15:47:31 RDA025 05904 (kevin ) Immediate__DBL
9/23 15:47:31 RDA025 05904 (kevin ) Stored__DBL
9/23 15:47:31 RDA025 05904 (kevin ) Initialize-request: Passed
9/23 15:47:31 RDA025 05904 (kevin ) wrote ---------- RDA-Open-Request ----------

RDA-APDU {
    RDA-APDU {
        r-Open-RI {
            operationID 2,
            r-Open-req {
                dataResourceHandle 1,
                dataResourceName "dqp",
                sQLAccessControlData {
                    ostring "dqp"
                },
                sQLUsageMode 1,
                sQLOpenArgument {
                    charSet "ISO-OIW-CHARACTER-SET",
                    sQLConformanceLevel "ISO-SQL-1992-LOW-CONFORMANCE"
                }
            }
        }
    }
}

---------- RDA-Open-Request ----------
9/23 15:47:31 RDA025 05904 (kevin ) read ---------- RDA-Open-Result ----------, context 1
r-Open-RC {
  operationID 2,
  res-or-err {
    r-Open-res {
      sQLOpenResult {
        charSet "ISO-OIW-CHARACTER-SET",
        charSetNotSupported FALSE (DEFAULT BOOLEAN)
      }
    }
  }
}

--------
9/23 15:47:32 RDA025 05904 (kevin ) ----------- RDA-Open-Result  -----------
9/23 15:47:32 RDA025 05904 (kevin ) RDA Open Result:
  Operation Id : 2
  Character Set  : 1.0.9568.2.0.1
  Open-request: Passed

RDA-APDU {
  RDA-APDU {
    r-BeginTransaction-RI {
      operationID 4,
      r-BeginTransaction-req NULL
    }
  }
}--------
9/23 15:47:32 RDA025 05904 (kevin ) ----------- RDA-BeginTransaction-Request  -----------
9/23 15:47:32 RDA025 05904 (kevin ) wrote ----------- RDA-Execute-Request  -----------

RDA-APDU {
  RDA-APDU {
    r-ExecuteDBL-RI {
      operationID 5,
      r-ExecuteDBL-req {
        dataResourceHandle 1,
        sQLDBLStatement {
          statementText "DELETE FROM RX WHERE DRUG_ID = :H AND PAYMENT_METHOD = :H"
        },
        sQLDBLArgumentSpecification {
          listOfSQLDataTypeDescriptor {
            SQLDataTypeDescriptor {
              nullable FALSE,
              typeDescriptor {
                integerType {
                  precision 4,
                  precisionBase 1
                }
              }
            }
          }
        }
      }
    }
  }
}
nullable FALSE,
typeDescriptor {
  characterType {
    length 4,
    fixedLengthEncoding TRUE
  }
} },
dblArguments {
  multipleArgument {
    listOfSQLDBLArgumentValues {
      SQLDBLArgumentValues {
        listOfSQLValue {
          SQLValue {
            dataItem {
              integerItem 7444
            }
          },
          SQLValue {
            dataItem {
              characterItem "cash"
            }
          }
        }
      }
    },
    SQLDBLArgumentValues {
      listOfSQLValue {
        SQLValue {
          dataItem {
            integerItem 7444
          }
        },
        SQLValue {
          dataItem {
            characterItem "xxxx"
          }
        }
      }
    },
    SQLDBLArgumentValues {
      listOfSQLValue {
        SQLValue {
          dataItem {
            integerItem 0
          }
        },
        SQLValue {
          dataItem {
            characterItem "cash"
          }
        }
      }
    }
  }
}

<table>
<thead>
<tr>
<th>Time</th>
<th>User</th>
<th>ID</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>9/23 15:47:32</td>
<td>kevin</td>
<td>05904</td>
<td>RDA-Execute-Request</td>
</tr>
<tr>
<td>9/23 15:47:33</td>
<td>kevin</td>
<td>05904</td>
<td>STATE: CT read RDA-Execute-Result, context</td>
</tr>
</tbody>
</table>

RDA-APDU {
  RDA-APDU {
    r-ExecuteDBL-RC {
      operationID 5,
      res-or-err {
        r-ExecuteDBL-res {
          listOfResultValues {
            ResultValues {
              SQLDBLException {
                SQLSTATE "00000",
                SQLCODE 0,
                SQLErrorText "no data: (no subclass)"
              },
            },
            ResultValues {
              SQLDBLException {
                SQLSTATE "02000",
                SQLCODE 100,
                SQLErrorText "ORA-01403: no data found"
              },
            },
          },
        },
      },
    },
  },
}
sQLOCODE 100,
sQLErrorText "ORA-01403: no data found
"
}
}
ResultValues {
    sQLDBLException {
        sQLSTATE "00000",
        sQLOCODE 0,
        sQLErrorText "ORA-01403: no data found
"
    }
}
}
}

-------
9/23 15:47:33 RDA025 05904 (kevin ) --------------- RDA-Execute-Result ---------------
9/23 15:47:33 RDA025 05904 (kevin ) <---------------------------
9/23 15:47:33 RDA025 05904 (kevin ) SQLSTATE expected = 00000 : successful completion
(no subclass)
9/23 15:47:33 RDA025 05904 (kevin ) SQLSTATE received = 00000 : successful completion
(no subclass)
9/23 15:47:33 RDA025 05904 (kevin ) SQLcode expected = 0 : successful completion.
9/23 15:47:33 RDA025 05904 (kevin ) SQLcode received = 0 : successful completion.
9/23 15:47:33 RDA025 05904 (kevin ) SQL Error Text = no data: (no subclass)
9/23 15:47:33 RDA025 05904 (kevin ) <---------------------------
9/23 15:47:33 RDA025 05904 (kevin ) <---------------------------
9/23 15:47:33 RDA025 05904 (kevin ) Checking Result # 2
9/23 15:47:33 RDA025 05904 (kevin ) SQLSTATE expected = 02000 : no data: (no subclass)
9/23 15:47:33 RDA025 05904 (kevin ) SQLSTATE received = 02000 : no data: (no subclass)
9/23 15:47:33 RDA025 05904 (kevin ) SQLcode expected = 100 : no data.
9/23 15:47:33 RDA025 05904 (kevin ) SQLcode received = 100 : no data.
9/23 15:47:33 RDA025 05904 (kevin ) SQL Error Text = ORA-01403: no data found
9/23 15:47:33 RDA025 05904 (kevin ) <---------------------------
9/23 15:47:33 RDA025 05904 (kevin ) <---------------------------
9/23 15:47:33 RDA025 05904 (kevin ) Checking Result # 3
9/23 15:47:33 RDA025 05904 (kevin ) SQLSTATE expected = 02000 : no data: (no subclass)
9/23 15:47:33 RDA025 05904 (kevin ) SQLSTATE received = 02000 : no data: (no subclass)
9/23 15:47:33 RDA025 05904 (kevin ) SQLcode expected = 100 : no data.
9/23 15:47:33 RDA025 05904 (kevin ) SQLcode received = 100 : no data.
9/23 15:47:33 RDA025 05904 (kevin ) SQL Error Text = ORA-01403: no data found
9/23 15:47:33 RDA025 05904 (kevin ) <---------------------------
9/23 15:47:33 RDA025 05904 (kevin ) <---------------------------
9/23 15:47:33 RDA025 05904 (kevin ) Checking Result # 4
9/23 15:47:33 RDA025 05904 (kevin ) SQLSTATE expected = 00000 : successful completion:
(no subclass)
9/23 15:47:33 RDA025 05904 (kevin ) SQLSTATE received = 00000: successful completion:
(no subclass)
9/23 15:47:33 RDA025 05904 (kevin ) SQLcode expected = 0: successful completion.
9/23 15:47:33 RDA025 05904 (kevin ) SQLcode received = 0: successful completion.
9/23 15:47:33 RDA025 05904 (kevin ) SQL Error Text = ORA-01403: no data found

----------
RDA-APDU {
    RDA-APDU {
        r-Rollback-RI {
            operationID 6,
            r-Rollback-req NULL
        }
    }
}
----------
9/23 15:47:33 RDA025 05904 (kevin ) STATE: CA
9/23 15:47:33 RDA025 05904 (kevin ) read ---------- RDA-Rollback-Result ----------, context 1
RDA-APDU {
    RDA-APDU {
        r-Rollback-RC {
            operationID 6,
            res-or-err {
                r-Rollback-res NULL
            }
        }
    }
}
----------
9/23 15:47:33 RDA025 05904 (kevin ) wrote ---------- RDA-Rollback-Request ----------
9/23 15:47:33 RDA025 05904 (kevin ) read RDA-Close-Result context 1
RDA-APDU {
    RDA-APDU {
        r-Close-RI {
            operationID 7,
            r-Close-req {
                listOfDataResourceHandle {
                    DataResourceHandle 1
                }
            }
        }
    }
}
----------
9/23 15:47:33 RDA025 05904 (kevin ) wrote ---------- RDA-Close-Request ----------
9/23 15:47:33 RDA025 05904 (kevin ) read RDA-Close-Result context 1
operationID 7,  
res-or-err {  
  r-Close-res {}  
}  
}
}

9/23 15:47:33 RDA025 05904 (kevin )  ----------- RDA-Close-Result  -----------
9/23 15:47:33 RDA025 05904 (kevin ) wrote  ----------- RDA-Terminate-Request  -----------
RDA-APDU {  
  RDA-APDU {  
    r-Terminate-RI {  
      operationID 8,  
      r-Terminate-req NULL  
    }  
  }  
}

9/23 15:47:33 RDA025 05904 (kevin )  ----------- RDA-Terminate-Request  -----------
9/23 15:47:33 RDA025 05904 (kevin ) STATE: Inactive
9/23 15:47:33 RDA025 05904 (kevin ) read  ----------- RDA-Terminate-Result  -----------, context
1  
RDA-APDU {  
  RDA-APDU {  
    r-Terminate-RC {  
      operationID 8,  
      res-or-err {  
        r-Terminate-res NULL  
      }  
    }  
  }  
}

9/23 15:47:33 RDA025 05904 (kevin )  ----------- RDA-Terminate-Result  -----------
9/23 15:47:33 RDA025 05904 (kevin ) A-RELEASE.CONFIRMATION :
9/23 15:47:33 RDA025 05904 (kevin ) Status: released - normal
9/23 15:47:33 RDA025 05904 (kevin ) Released conection to rda

9/23 15:47:33 RDA025 05904 (kevin )

9/23 15:47:33 RDA025 05904 (kevin )  Summary RDA Syntax Checker Report

9/23 15:47:33 RDA025 05904 (kevin )

9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Initialize Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Initialize Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Syncronize Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Terminate Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Terminate Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Begintrans Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Begintrans Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Commit Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Commit Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Rollback Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Rollback Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Cancel Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Cancel Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Status Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Status Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Open Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Open Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Close Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Close Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Execute Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 1 Execute Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Define Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Define Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Invoke Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Invoke Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Drop Requests found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  Checked 0 Drop Confirms found 0 bad.
9/23 15:47:33 RDA025 05904 (kevin )  RDA025 : Completion Status: PASSED.
## APPENDIX D  Sample Summary Report

RDA/SQL Test Suite  
for RDA Server on rda  
Mon Sep 23 16:42:14 1996

<table>
<thead>
<tr>
<th>PROGRAM</th>
<th>RESULT</th>
<th>Description of Test Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDA001</td>
<td>pass</td>
<td>singleArg, multipleArg missing -&gt; 1 listOfResultValues</td>
</tr>
<tr>
<td>RDA002</td>
<td>pass</td>
<td>repetitionCount default is 1</td>
</tr>
<tr>
<td>RDA003</td>
<td>fail</td>
<td>repetitionCount = 10 -&gt; 10 SQLDBExceptions, FETCH</td>
</tr>
<tr>
<td>RDA004</td>
<td>pass</td>
<td>repetitionCount = 3 -&gt; SQLDBExceptions, INSERT</td>
</tr>
<tr>
<td>RDA005</td>
<td>pass</td>
<td>4 SQLDBLArgumentValues -&gt; 4 SQLDBExceptions, SELECT</td>
</tr>
<tr>
<td>RDA006</td>
<td>pass</td>
<td>OPEN undeclared cursor name (syntax error)</td>
</tr>
<tr>
<td>RDA007</td>
<td>fail</td>
<td>cursor declaration survives CLOSE</td>
</tr>
<tr>
<td>RDA008</td>
<td>pass</td>
<td>cursor declaration survives ROLLBACK</td>
</tr>
<tr>
<td>RDA009</td>
<td>pass</td>
<td>cursor declaration survives COMMIT</td>
</tr>
<tr>
<td>RDA010</td>
<td>pass</td>
<td>null charSet on R-Open -&gt; each charSet has objId, CURSOR</td>
</tr>
<tr>
<td>RDA011</td>
<td>pass</td>
<td>null charSet on R-Open -&gt; each charSet has objId, SELECT</td>
</tr>
<tr>
<td>RDA012</td>
<td>pass</td>
<td>ConformanceLevel &lt;1987&gt;, objId 1.0.9075.0</td>
</tr>
<tr>
<td>RDA013</td>
<td>aborted</td>
<td>ConformanceLevel &lt;1989&gt; without IEF, objId 1.0.9075.1.0.1</td>
</tr>
<tr>
<td>RDA014</td>
<td>pass</td>
<td>ConformanceLevel &lt;1989&gt; with IEF, objId 1.0.9075.1.1.1</td>
</tr>
<tr>
<td>RDA015</td>
<td>pass</td>
<td>ConformanceLevel &lt;1992&gt; low, objId 1.0.9075.2.0</td>
</tr>
<tr>
<td>RDA016</td>
<td>fail</td>
<td>Table10.5b&amp;6a: DECLARE ArgSpec, OPEN ResSpec</td>
</tr>
<tr>
<td>RDA017</td>
<td>fail</td>
<td>Table10.5a&amp;6a: DECLARE ArgSpec override by OPEN, OPEN ResSpec</td>
</tr>
<tr>
<td>RDA018</td>
<td>fail</td>
<td>Table10.5a&amp;6b: OPEN ArgSpec, OPEN ResSpec</td>
</tr>
<tr>
<td>RDA019</td>
<td>fail</td>
<td>Table10.2a.2b.3b: DECLARE ArgSpec, FETCH overrides OPEN ResSpec</td>
</tr>
<tr>
<td>RDA020</td>
<td>fail</td>
<td>Table10.2a.2b.3b: OPEN overrides DECLARE ArgSpec, FETCH overrides</td>
</tr>
<tr>
<td>RDA021</td>
<td>fail</td>
<td>Table10.2a.2b.3b: OPEN ArgSpec, FETCH overrides OPEN ResSpec</td>
</tr>
<tr>
<td>RDA022</td>
<td>pass</td>
<td>Table10.4a: SELECT ResSpec sent by client</td>
</tr>
</tbody>
</table>
Table 10.4b: SELECT ResSpec omitted by client

- repetitionCount = 3 -> SQLDBExceptions, UPDATE
- 4 SQLDBLArgumentValues -> 4 SQLDBExceptions, DELETE
- ERROR for ROLLBACK WORK in r-ExecuteDBL-RI
- ERROR for COMMIT WORK in r-ExecuteDBL-RI
- ERROR for INSERT with explicit SQLUsageMode = retrieval
- ERROR for CURSOR UPDATE explicit SQLUsageMode = retrieval
- ERROR for UPDATE WHERE explicit SQLUsageMode = retrieval
- ERROR for DELETE WHERE explicit SQLUsageMode = retrieval
- ERROR for CURSOR DELETE explicit SQLUsageMode = retrieval
- SIZING SQL statementText 4000 octets long
- SIZING charSet 16 elements long in ArgSpec
- SIZING 100 entries in ArgSpec and ArgVal
- Check that NULLS can be inserted and retrieved.
- SIZING 100 entries of SQLDataTypeDescriptor in ResSpec-RI
- SIZING repetitionCount = 64
- SIZING 64 entries of SQLDBLArgumentValues in ArgVal
- SIZING smallintType decimal precision 4
- SIZING charSet 16 elements long in ResSpec-RI
- SIZING integerType decimal precision 9
- SIZING 64 entries of ResultValueslist
- SIZING 100 entries of SQLValue in SQLDBLResultValues
- SIZING colName 18 characters long
- SIZING length (characters) 240 (Including null terminator)
- SIZING numericType (15, 0) (cut to (2**31 -1))
- SIZING decimalType precision 8 scale 8
- SIZING integerType decimal precision 9
User's Guide for RDA/SQL Validation Tests (Version 1.0)

Kevin Brady
Joan Sullivan
September 1996

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Gaithersburg, MD 20899
User's Guide for the RDA/SQL Validation Tests (Version 1.0)

Kevin Brady
Joan Sullivan
September 1996

ABSTRACT: The RDA/SQL Validation Tests, developed by NIST, consist of a set of C programs designed to test an RDA/SQL server for conformance to the international standards for Remote Database Access with an SQL Specialization. Testing is limited to the RDA Basic Application Context/Immediate Execution profile defined by the NIST OIW Stable Agreements for OSI Protocols. The validation tests use software tools provided in the public domain ISO Development Environment (ISODE) and has been operating at NIST on TCP/IP networks using the Internet Engineering Task Force (IETF) specification RFC 1006.

KEYWORDS: conformance testing; database standards; interoperability; remote database access; RDA; SQL; RFC 1006; ISODE; testing of software; validation of software
Summary of Test Cases

26
tests passed: 0051
tests failed: 0021
tests missing: 0001
tests aborted: 0001

TOTAL tests: 0074