Raima Database API for LabVIEW

By Wayne Warren, CTO – December 2013

Introduction

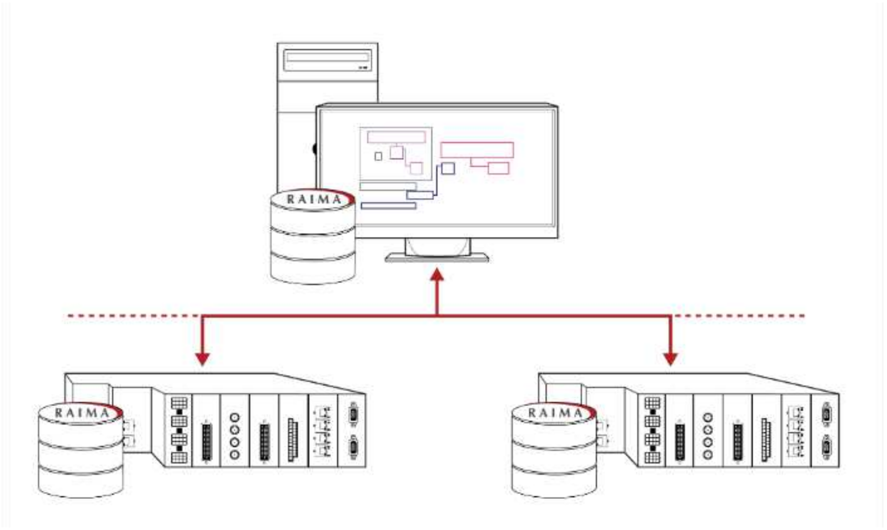
Raima Database API for LabVIEW is an interface package to Raima Database Manager (RDM), which is a high-performance database management system optimized for operating systems commonly used within the embedded market. Windows, NI Linux Real-Time and RT VxWorks (on the CompactRIO-9068, -9024 and Single-Board RIO) are supported in this package. The database engine has been developed to fully utilize multi-core processors and networks of embedded computers. It runs with minimal memory and supports both in-memory and on-disk storage. RDM provides Embedded SQL that is suitable for running on embedded computers with requirements to store live streaming data or sets of configuration parameters.

Current version:

Raima Database API for LabVIEW 2.0 - supports LabVIEW 2013, Raima Database Manager 12.0

Earlier versions:

Raima Database API for LabVIEW 1.0 - supports LabVIEW 2011 and 2012, Raima Database Manager 11.0



Where to Buy

The Raima Database API for LabVIEW can be downloaded and purchased at the LabVIEW Tools Network www.ni.com/labviewtools/raima, or directly from your NI or Raima sales representative.

Software Development Kit (SDK) and Distribution Licenses are priced separately.

This package is compatible with RDM 12.0 Workgroup Edition for Windows, which may be downloaded from the Raima web site: www.raima.com/downloads.

CONTENTS

1. OPERATIONAL OVERVIEW

The Database API consists of a set of primitive functions that are generally consistent with the ODBC standard. Databases are defined and manipulated in the SQL language. Each Database API function calls a Native C function through the Call Library Function Node VI. The Native C function will then call the RDM API (the same functions available directly to the C programmer using RDM 12.0 Workgroup Edition).

The basic call stack is shown in Figure 1:

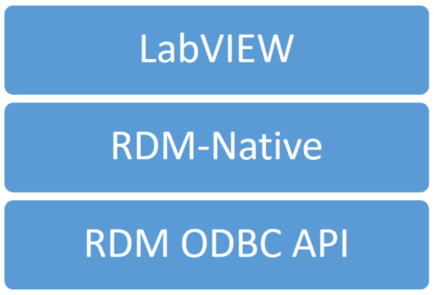


Figure 1: Basic Call Stack

The RDM ODBC API contains all of the database manipulation logic. The basic call stack is the same whether deployed in Windows, Linux or VxWorks.

Section 4 will discuss how to implement database sharing, but the basic concept of sharing is shown below:

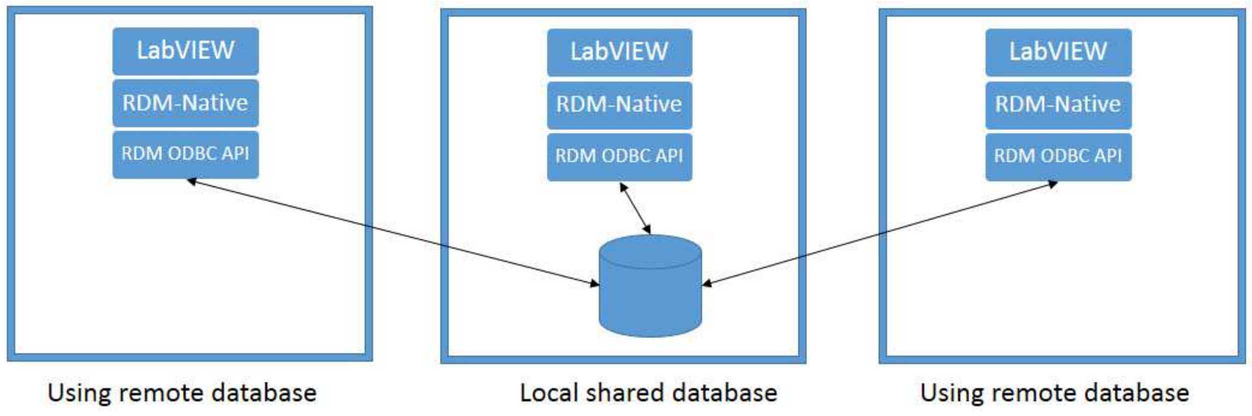


Figure 2: Sharing/Using a Database

2. PROGRAMMING WITH THE DATABASE API

Those familiar with the ODBC API will recognize the steps needed to work with databases. The following sections show the basic operations.

2.1 Allocate Handles

The following figure shows preparation work necessary for the rest of the steps. The Connection and/or Statement Handles are required inputs for the rest of the functions.

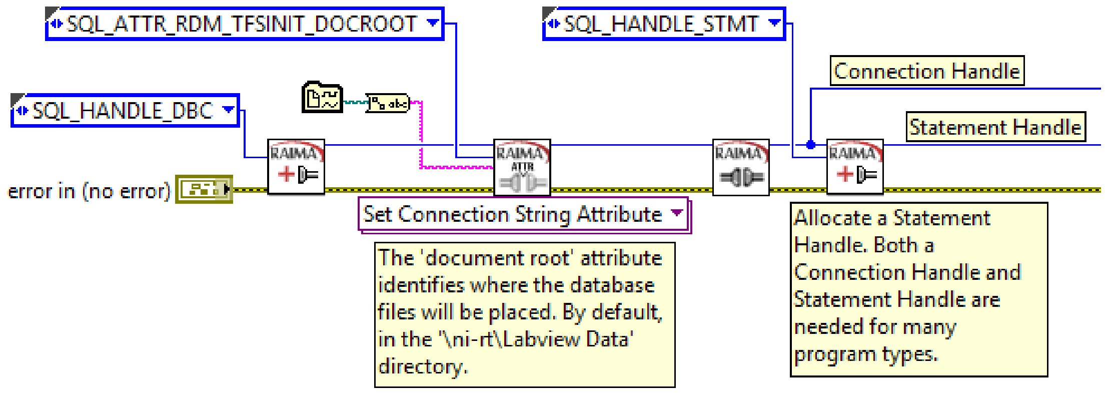

SQL_ATTR_RDM_TFSINIT_DOCROOT
SQL_HANDLE_STMT
Connection Handle
SQL_HANDLE_DBC
Statement Handle
RAIMA
ATTR
error in (no error)
Set Connection String Attribute
The 'document root' attribute identifies where the database files will be placed. By default, in the '\ni-rt\Labview Data' directory.
Allocate a Statement Handle. Both a Connection Handle and Statement Handle are needed for many program types.

Figure 3: Allocate Handles

From left to right, Figure 3 goes through the following steps:

- Allocate a Connection Handle (SQL_HANDLE_DBC).
- Set the location for storage of the database (SQL_ATTR_RDM_TFSINIT_DOCROOT). This example shows how to specify the "LabVIEW Data" directory. If this is not specified, the current directory of the executing program will be used.
- Connect.
- Allocate a Statement Handle (SQL_HANDLE_STMT). One connection can support many statements, but it is typical to use one.

2.2 Create or Open a Database

Creating a database in ODBC can be accomplished by executing a minimum of two SQL statements. In the following figure, two Execute SQL Statement Now (SQLExecDirect) functions are used to do just that:

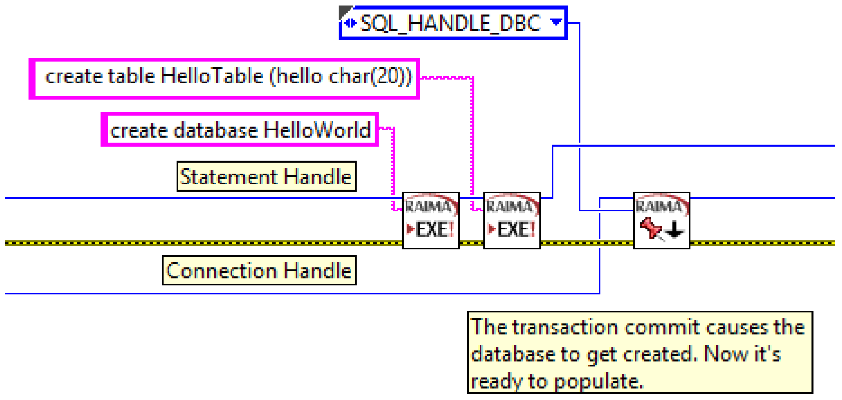

SQL_HANDLE_DBC
create table HelloTable (hello char(20))
create database HelloWorld
Statement Handle
RAIMA
EXE!
Connection Handle
The transaction commit causes the database to get created. Now it's ready to populate.

Figure 4: Create a New Database

Here is a description of the steps:

- Execute "create database" statement. Database named HelloWorld.
- Execute "create table" statement. Table is named HelloTable containing one character string column named hello.
- Commit the transaction. It is during the commit where the database is physically created on the storage media.

Note that when a database should always be created (rather than opening an existing database), it is best to delete the database before creating it.

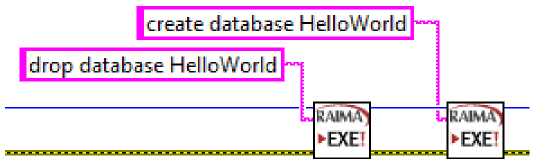


Figure 5: Dropping a Database

2.3 Populate and Read a Database

Insert and commit two rows:

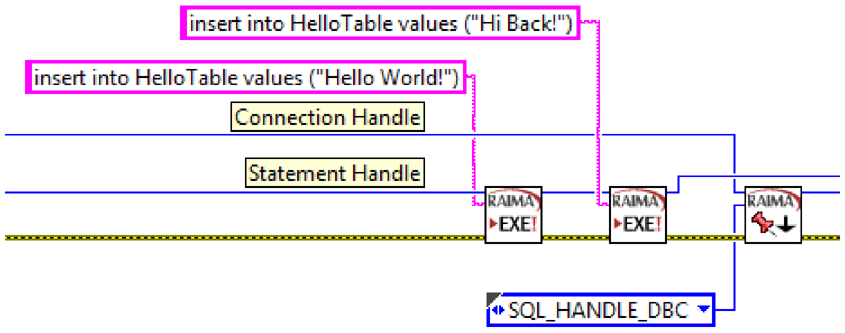

insert into HelloTable values ("Hi Back!")
insert into HelloTable values ("Hello World!")
Connection Handle
Statement Handle
RAIMA
EXE!
RAIMA
EXE!
RAIMA
SQL_HANDLE_DBC

Figure 6: Populate a Database

Database population frequently occurs in loops, but this simple example performs two inserts:

- Insert the value “Hello World!” into the string column in the table, creating one row.
- Insert the value “Hi Back!” into the string column in the table.
- Commit the two rows to the database.

2.4 Close a Database

The clean way to close a database is to make sure of the following:

- All transactions have been committed or aborted.
- All Statement Handles have been freed.
- The Connection Handle has been freed.

In this example, the transaction we committed after inserting the two rows, so there is nothing to complete in regards to transactions.

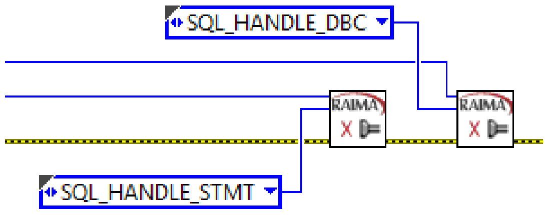


Figure 7: Clean Up / Close Database

2.5 Share/Use a Database

Standard connection to the Raima API is standalone. To allow a database to be shared, you must properly set some connection attributes.

Sharing a Database

A database that is used by one program can be made sharable to other programs by identifying the Server as “self” in the “Connect to a Data Source.vi”.

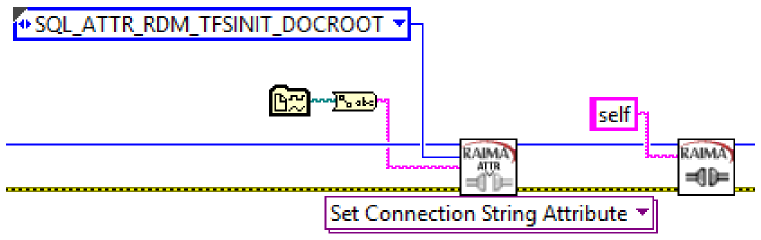


Figure 8: Sharing a Database

Once the “self” connection is established and a database open, this program must continue running in order for other programs to use the database. In the loop shown below, other work may be done on the database, but a soon as the loop terminates, the shared database will also be unavailable:

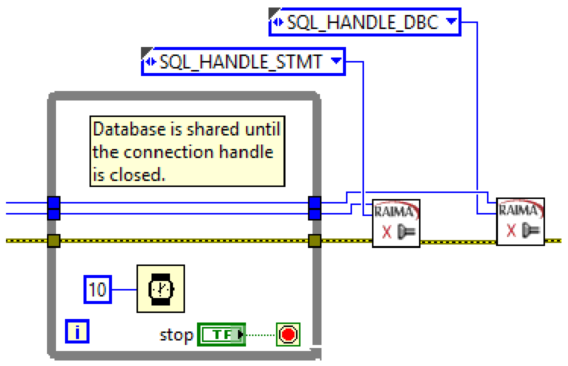


Figure 9: Keeping a Database Alive for Users

A rule about sharing is that a database must be “used” by another program running on the same architecture, e.g. cRIO-9024 to cRIO-9024 or Windows to Windows. Note below that another method exists to make databases sharable between cRIO devices and Windows. This requires running an external utility on either Windows or Linux.

Using a Database

This example assumes that two (or more) different computers are being used, with one LabVIEW program running on each. More advanced methods will allow multiple programs to run and share databases in the Windows environment, but that will not be covered here.

Once a database has been shared by another program, you need to know the name or IP address of the computer on which that program is running. Then, before the “Connect to a Data Source.vi” you need to set the SQL_ATTR_RDM_TFSINIT_TYPE to 1 and identify the other computer with the SQL_ATTR_RDM_TFS_NAME attribute:

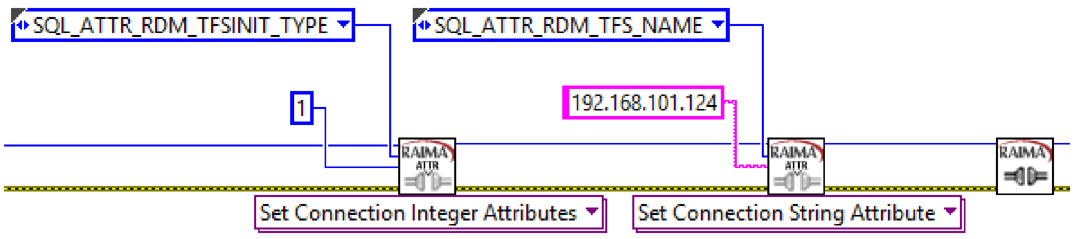

SQL_ATTR_RDM_TFSINIT_TYPE
SQL_ATTR_RDM_TFS_NAME
1
192.168.101.124
RAIMA ATTR
RAIMA ATTR
RAIMA
Set Connection Integer Attributes
Set Connection String Attribute

Figure 10: Using a Shared Database

Note that the SQL_ATTR_RDM_TFSINIT_DOCROOT is not necessary when using a database because the database location is established by the sharing program. Note also that it may be set in this program without negative consequences.

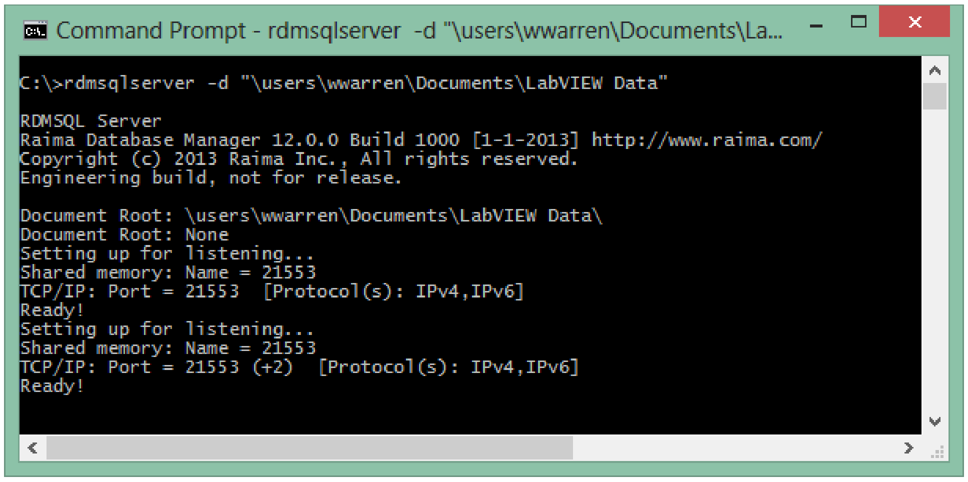
2.6 Connecting Real-time Programs to Windows Databases

There is yet-another connection method that has the following advantage/disadvantage:

- Advantage – Database compatibility between RT cRIO and Windows.
- Disadvantage – Database cannot be stored on the VxWorks-based cRIO.

On Windows and Linux, a program named `RDMSQLSERVER.EXE` (or just '`rdmsqlserver`') is available. When this program is running on a Windows or Linux computer within the domain of a specified document root, it can be accessed by LabVIEW programs running on RT systems concurrently with any database program running on Windows (LabVIEW or otherwise). This is because the communication method between the RDM runtime system on cRIO is heterogeneous.

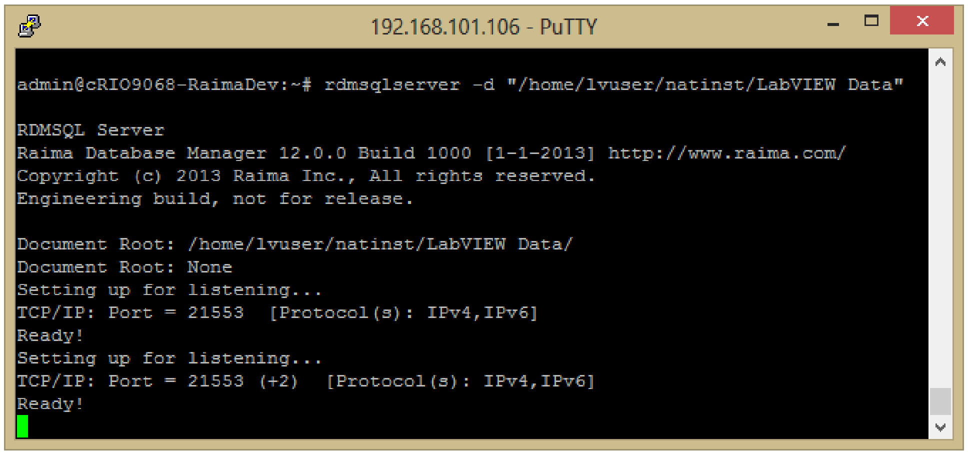
On Windows or Linux, run the utility, specifying the document root:


Command Prompt - rdmsqlserver -d "\users\wwarren\Documents\La...

```
C:\>rdmsqlserver -d "\users\wwarren\Documents\LabVIEW Data"

RDMSQL Server
Raima Database Manager 12.0.0 Build 1000 [1-1-2013] http://www.raima.com/
Copyright (c) 2013 Raima Inc., All rights reserved.
Engineering build, not for release.

Document Root: \users\wwarren\Documents\LabVIEW Data\
Document Root: None
Setting up for listening...
Shared memory: Name = 21553
TCP/IP: Port = 21553  [Protocol(s): IPv4,IPv6]
Ready!
Setting up for listening...
Shared memory: Name = 21553
TCP/IP: Port = 21553 (+2)  [Protocol(s): IPv4,IPv6]
Ready!
```



```
admin@cRIO9068-RaimaDev:~# rdmsqlserver -d "/home/lvuser/natinst/LabVIEW Data"

RDMSQL Server
Raima Database Manager 12.0.0 Build 1000 [1-1-2013] http://www.raima.com/
Copyright (c) 2013 Raima Inc., All rights reserved.
Engineering build, not for release.

Document Root: /home/lvuser/natinst/LabVIEW Data/
Document Root: None
Setting up for listening...
TCP/IP: Port = 21553  [Protocol(s): IPv4,IPv6]
Ready!
Setting up for listening...
TCP/IP: Port = 21553 (+2)  [Protocol(s): IPv4,IPv6]
Ready!
```

Figure 11: Starting RDMSQLSERVER on Windows and Linux

Then from cRIO, name the Server input in “Connect to Data Source.vi”. The name is the computer’s domain name or IP address.

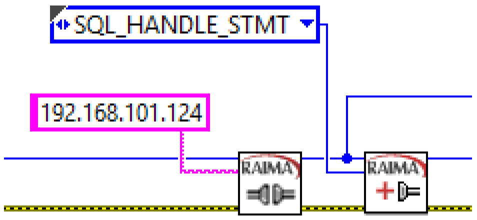


Figure 12: Connect to RDMSQLSERVER from cRIO

2.7 Arrays

Arrays of values are supported in the Database API, but they follow different rules.

1. Regardless of the basic data type, the column is defined as “long varbinary”.
2. A “Prepare” is required (cannot input array data with the “Execute SQL Statement Now” function).
3. The array is input after the “Execute SQL Statement” function using the “Set a Parameter … Array” function. The default length is the length of the array. You may enter a specific length.

When there are other non-array columns in the row, they must be input before the “Execute SQL Statement” function.

See below for the general flow:

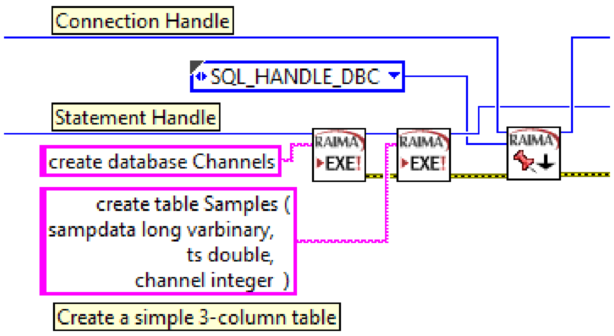


Figure 13: SQL DDL for Array Storage

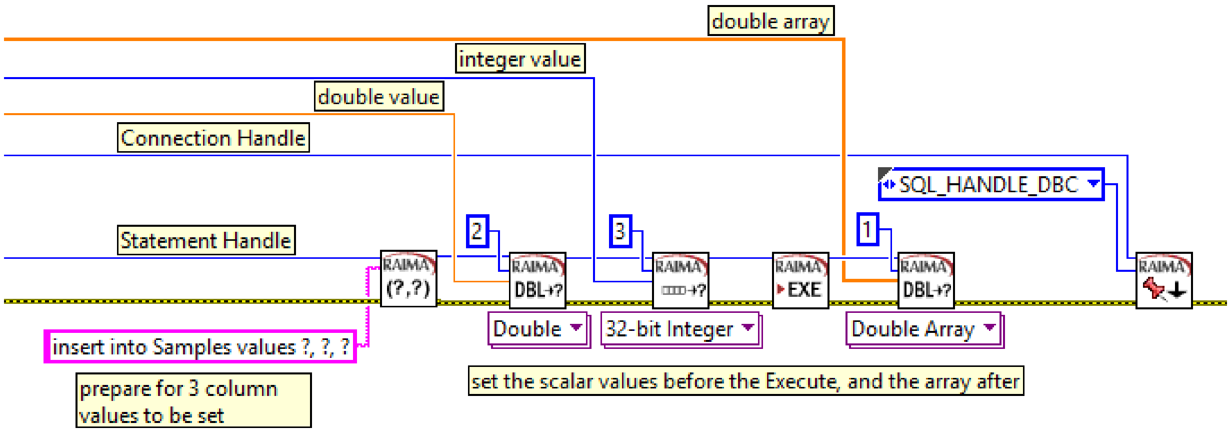


Figure 14: Storing Scalar and Array Columns

3. TECHNICAL DETAILS

This secton is meant as a quick look under the hood:

3.1 Windows

Advanced technical details.

- When installed, Raima Database API for LabVIEW will have a complete SDK for LabVIEW development on Windows, allowing programs to be run on Windows, cRIO-9068 or cRIO-9024. Here are the locations of several key directories:

  Toolkit VIs: `<LabVIEW>\vi.lib\addons\_Raima Inc\Raima Database API for LabVIEW\`

  Example VIs: `<LabVIEW>\examples\Raima Inc\Raima Database API for LabVIEW\`

  Error Codes (Raima-Database API-errors.txt): `<LabVIEW>\project\errors\`

  Help file (Raima-Database API for LabVIEW.chm): `<LabVIEW>\help\`

  RT Images: `<NI Home>\RT Images\RaimaDatabaseAPI\2.0\`

- Several Windows DLLs are required by LabVIEW programs running Raima VIs. The DLL named `rdmNative-12.dll` is the native interface module specifically created for the LabVIEW VIs to use. The remainder of the DLLs are part of the RDM 12.0 product. All of the Windows DLLs are stored in the `<Toolkit VIs>\source\Private\` directory, for example:

  ```
  C:\Program Files (x86)\National Instruments\LabVIEW 2013\vi.lib\addons\_Raima Inc\Raima
  Database API for LabVIEW\source\Private
  ```

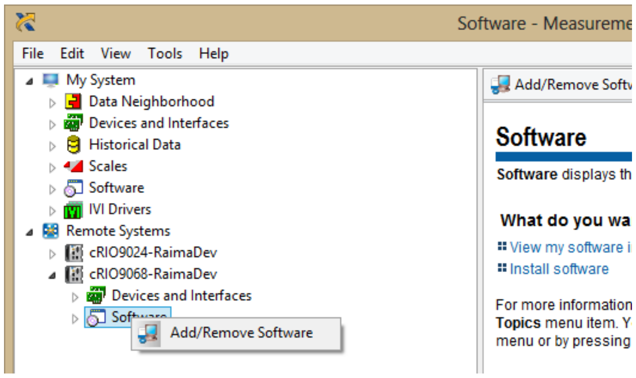
- Also in the directory containing the Windows DLLs are some executable programs. The most important are `tfserver.exe`, `rdmsqlserver.exe` and `rdmsql.exe`. Other utility programs have been included for convenience. Documentation for all of these programs can be found online on the Raima site. If you want to use these from a command prompt, be sure to include the above directory in your path.
- C/C++ program development is possible by downloading the complete RDM SDK from the Raima web site. Since RDM is a multi-user database management system, it is possible to create additional executable programs outside of the LabVIEW environment that interoperate with the LabVIEW programs at the database level.

3.2 Real-Time Installation using MAX

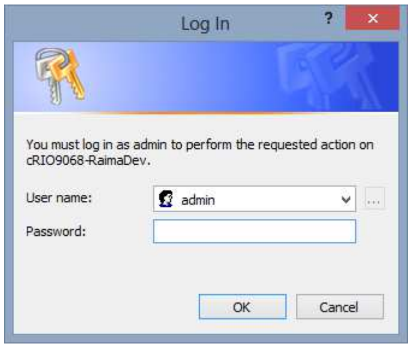
All of the binary code files needed for RT operation on the cRIO-9024 or cRIO-9068 is are stored in the `<RT Images>` directory identified above, in the Windows installation.

Before you can run Real-Time VIs on the CompactRIO/Single-Board RIO, you must make sure the RT modules have been installed. The following procedure will work for either VxWorks or NI Linux Real-Time based RIOs:

Open MAX and locate your target under Remote Systems, expand the tree of the target system onto which you are preparing to install, right-click on Software, and select Add/Remove Software to launch the wizard:



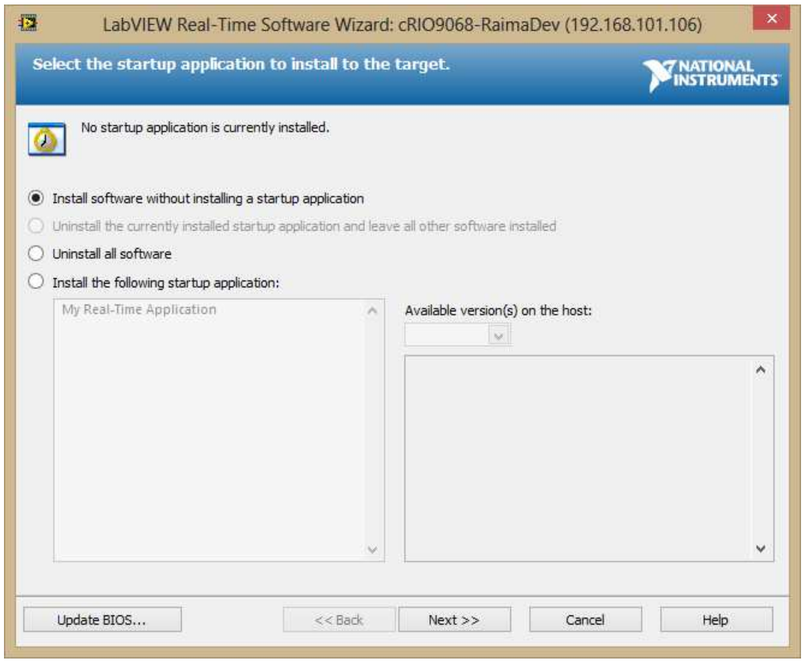
Skip this next step if your RT system is based on VxWorks. If your target is NI Linux Real-Time, you will be shown the following two dialogs. The Log In to the device is “`admin`” with no password by default. If you have changed this, use your new credentials. Upon successful login, it will move to the next screen. Use the default setting of “Install software without installing a startup application” and click Next:



You must log in as admin to perform the requested action on cRIO9068-RaimaDev.

User name: admin

Password:



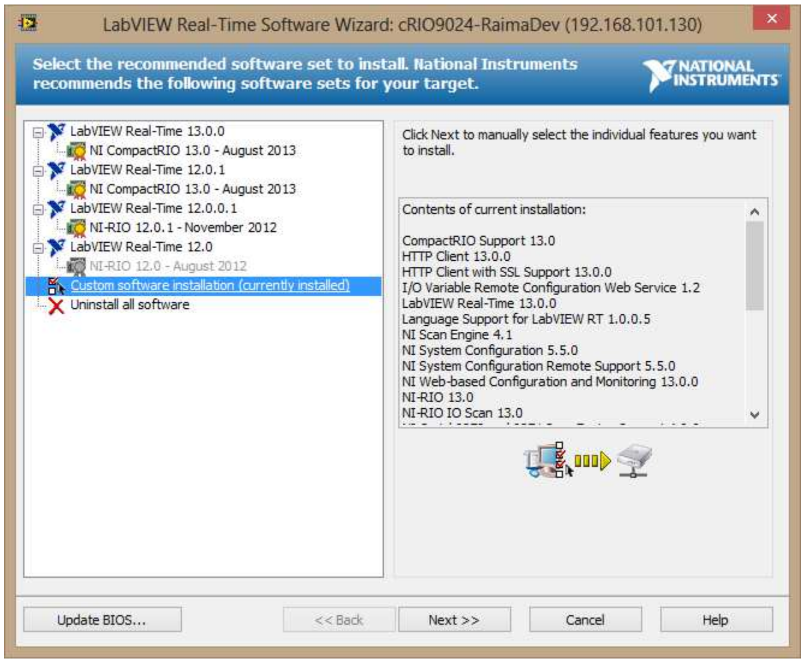
Select the startup application to install to the target.

No startup application is currently installed.

- [✓ Install software without installing a startup application]
- Uninstall the currently installed startup application and leave all other software installed
- Uninstall all software
- Install the following startup application:

Available version(s) on the host:

Select Custom Software Installation and click Next (then agree to the dialog/warning):


LabVIEW Real-Time Software Wizard: cRIO9024-RaimaDev (192.168.101.130)
NATIONAL INSTRUMENTS
LabVIEW Real-Time 13.0.0
NI CompactRIO 13.0 - August 2013
LabVIEW Real-Time 12.0.1
NI CompactRIO 13.0 - August 2013
LabVIEW Real-Time 12.0.0.1
NI-RIO 12.0.1 - November 2012
LabVIEW Real-Time 12.0
NI-RIO 12.0 - August 2012
Custom software installation (currently installed)
Uninstall all software
Update BIOS...
<< Back
Next >>
Cancel
Help

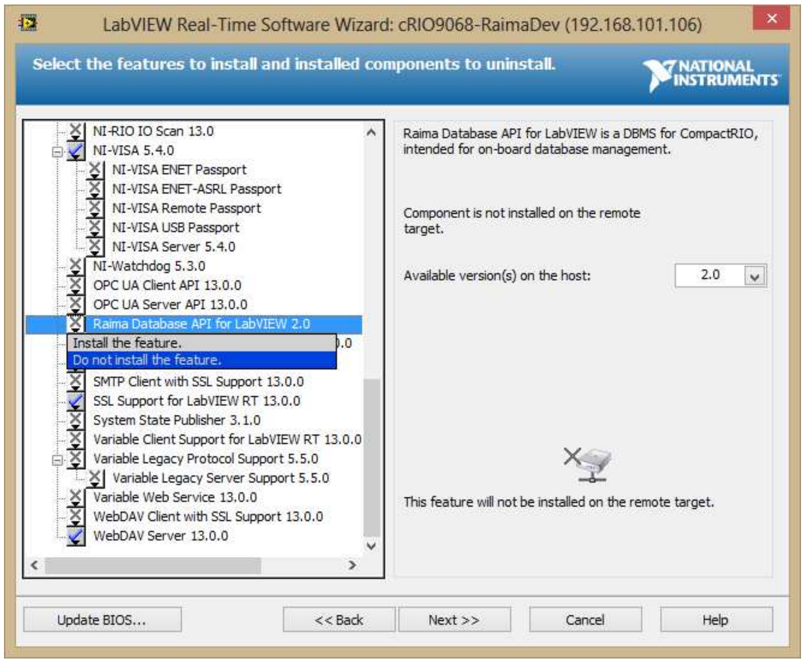
Select the recommended software set to install. National Instruments recommends the following software sets for your target.

Click Next to manually select the individual features you want to install.

Contents of current installation:

CompactRIO Support 13.0
HTTP Client 13.0.0
HTTP Client with SSL Support 13.0.0
I/O Variable Remote Configuration Web Service 1.2
LabVIEW Real-Time 13.0.0
Language Support for LabVIEW RT 1.0.0.5
NI Scan Engine 4.1
NI System Configuration 5.5.0
NI System Configuration Remote Support 5.5.0
NI Web-based Configuration and Monitoring 13.0.0
NI-RIO 13.0
NI-RIO IO Scan 13.0

You will now see the Raima module under the list of available features:


LabVIEW Real-Time Software Wizard: cRIO9068-RaimaDev (192.168.101.106)

Select the features to install and installed components to uninstall.

NI-RIO IO Scan 13.0
NI-VISA 5.4.0
NI-VISA ENET Passport
NI-VISA ENET-ASRL Passport
NI-VISA Remote Passport
NI-VISA USB Passport
NI-VISA Server 5.4.0
NI-Watchdog 5.3.0
OPC UA Client API 13.0.0
OPC UA Server API 13.0.0
Raima Database API for LabVIEW 2.0
Install the feature.
Do not install the feature.
SMTP Client with SSL Support 13.0.0
SSL Support for LabVIEW RT 13.0.0
System State Publisher 3.1.0
Variable Client Support for LabVIEW RT 13.0.0
Variable Legacy Protocol Support 5.5.0
Variable Legacy Server Support 5.5.0
Variable Web Service 13.0.0
WebDAV Client with SSL Support 13.0.0
WebDAV Server 13.0.0

Raima Database API for LabVIEW is a DBMS for CompactRIO, intended for on-board database management.

Component is not installed on the remote target.

Available version(s) on the host: 2.0

This feature will not be installed on the remote target.

Select “Install the feature” and press Next two times. MAX will put your RT system into “install mode”, telling you when the Real-Time target has been updated successfully.

3.3 NI Linux Real-Time

NI Linux Real-Time is a more advance environment than VxWorks and gives the LabVIEW developer additional tools and options to assist in development and application deployment.

Development recommendations:

- Take advantage of WebDAV, allowing you to mount the Linux file system as a Windows drive. To set it up, use a Windows Explorer and right-click on Computer or Network. Either location should show a menu containing “`Map network drive…`”. Select it, select the drive of your choice and then the Folder. The folder should be the network name or IP address of the target Linux device, for example “`cRIO9068-RaimaDev`”, with “`files`” as the shared folder name:

  ```
  \\cRIO9068-RaimaDev\files
  ```

- Use a tool such a PuTTY to connect with your device with SSH. Login as “`admin`” with an empty password, unless you have changed it.
- Command-line utilities are available. Those familiar with Linux will be able to use the common tools. Those familiar with RDM programming on Windows or Linux will be able to use the Raima utilities.

Advanced technical details:

- The “Default Data Directory” for LabVIEW on Linux is:

  ```
  /home/lvuser/natinst/LabVIEW Data
  ```

  (case is significant). Raima databases will be placed into subdirectories of the Data directory, with the directory names corresponding to the database names.

- Raima libraries are placed into:

  ```
  /usr/local/lib
  ```

- Raima programs (`rdmsql`, `rdmsqlserver` and `tfserver`) are placed into:

  `/usr/local/sbin`

- Additional Raima programs and libraries were installed with your package (into directory `…\National Instruments\RT Images`), but not installed onto Linux by default. These programs and the libraries required to support them may be copied into the `lib` and `sbin` directories mentioned above (using WebDAV) by advanced Linux developers. The following table shows the utilities that are available and the libraries required by them:

Program	Supporting Libraries
dbcheck	librdmutil-12.so
dbcrypt	librdmutil-12.so
dbexp	librdmutil-12.so, librdmdbexp-12.so
dbget	librdmdbget_tool-12.so, librdmdatamove-12.so, librdmhttpmon-12.so
dbimp	librdmdbimp_tool-12.so
dbmirror	librdmmirroring-12.so, librdmhttpmon-12.so, librdmdatamove-12.so
dbrep	librdmutil-12.so, librdmreplication-12.so, librdmrepcli-12.so, librdmhttpmon-12.so, librdmdatamove-12.so
dbrepair	librdmutil-12.so
initdb	librdmutil-12.so
keybuild	librdmutil-12.so
prdbd	librdmutil-12.so
tfsuser	librdmtfsuser_tool-12.so

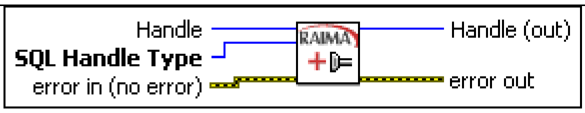
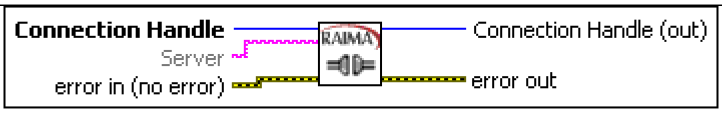
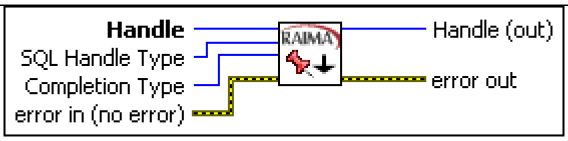
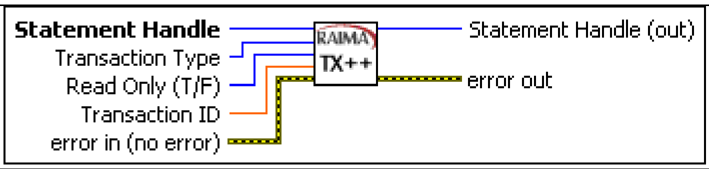
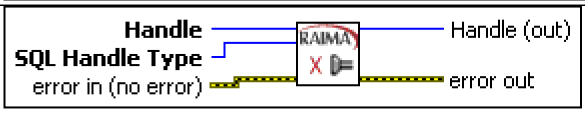
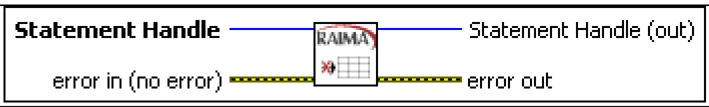
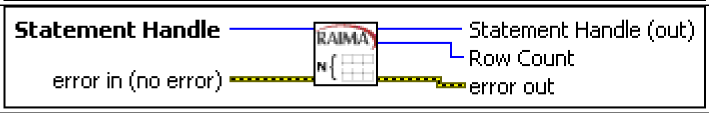
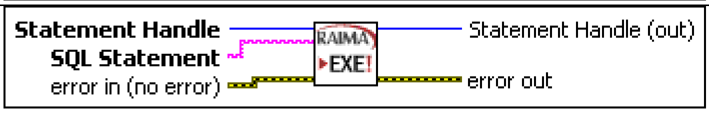
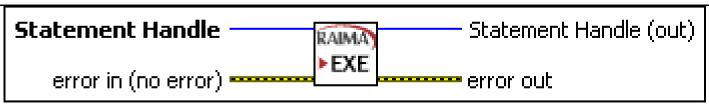
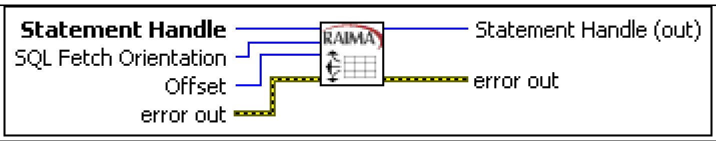
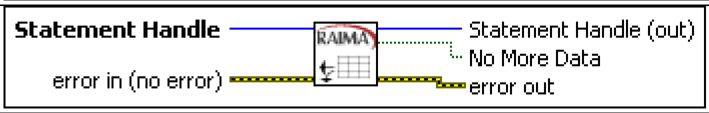
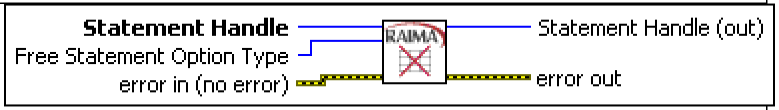
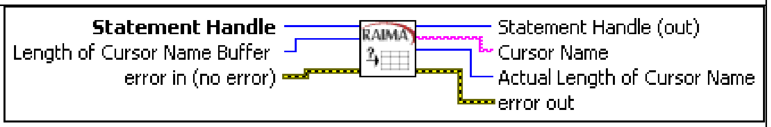
3.4 CompactRIO VxWorks

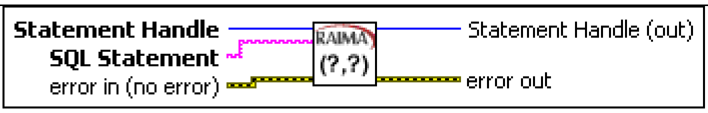
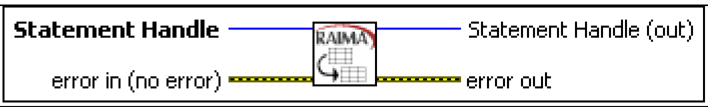
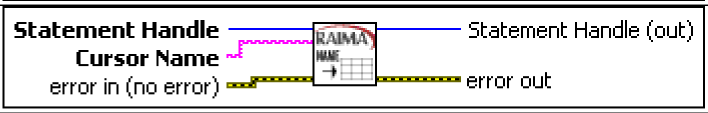
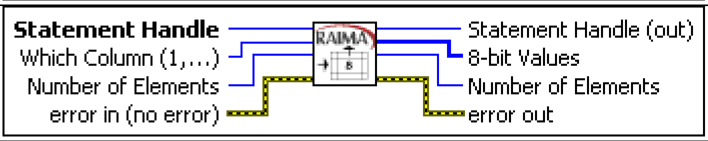
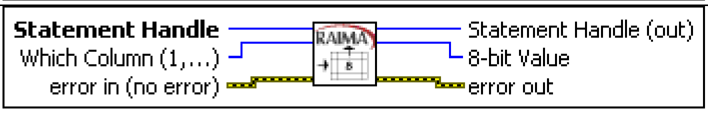
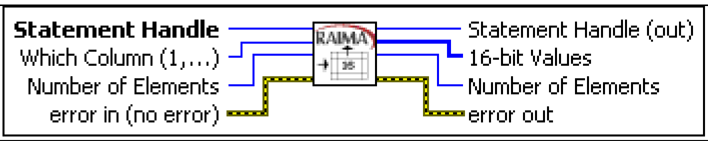
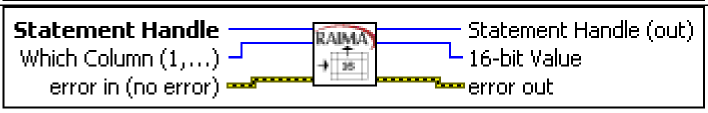
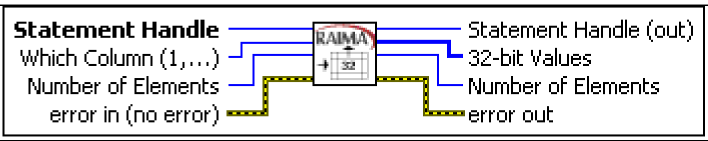
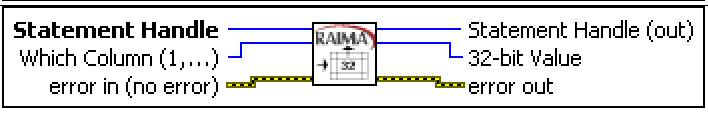
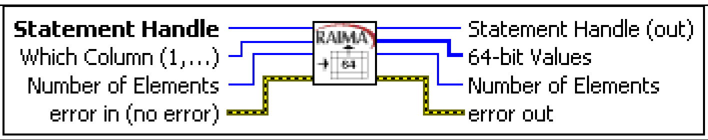
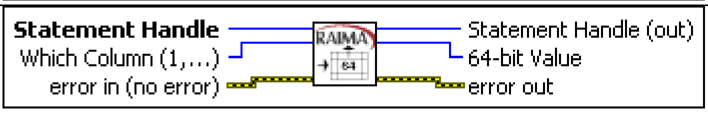
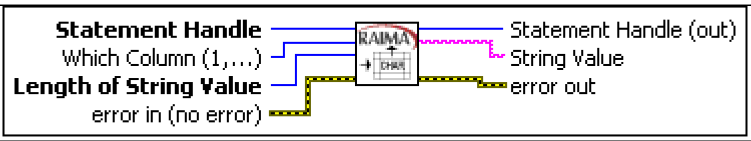
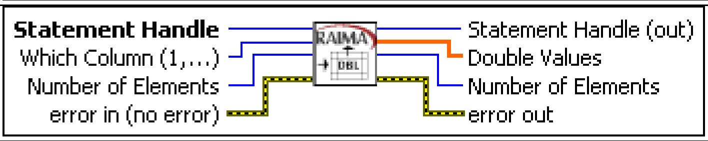
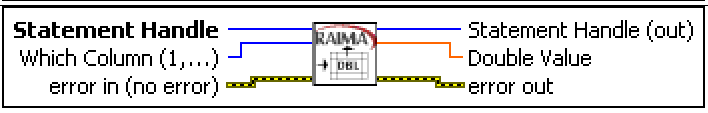
Advanced technical details:

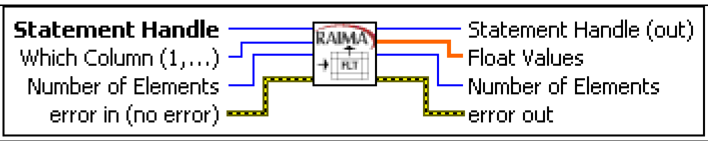
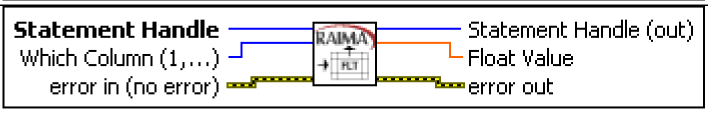
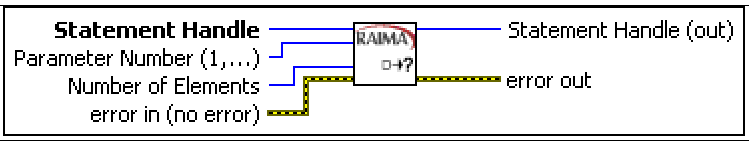
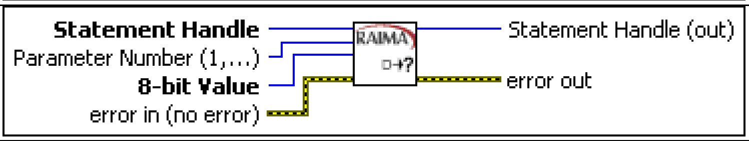
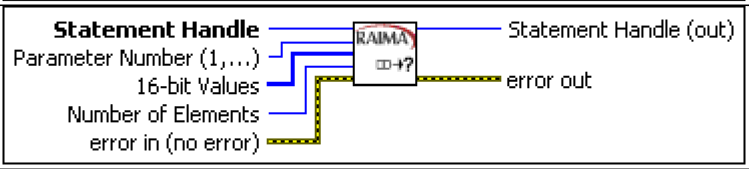
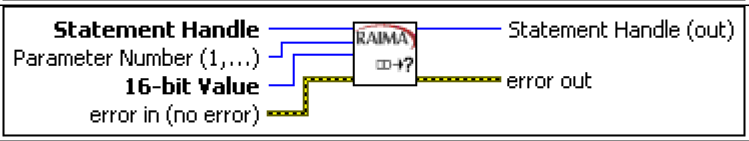
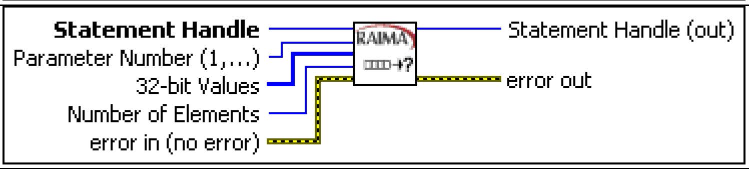
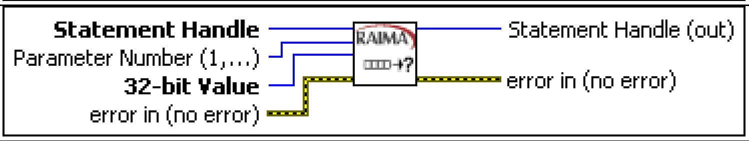
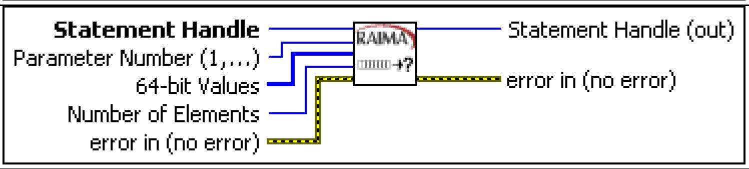
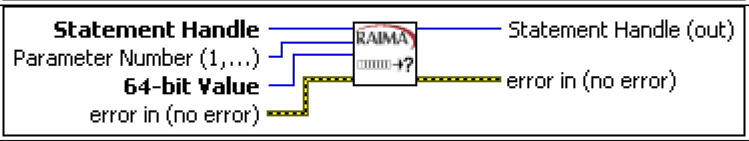
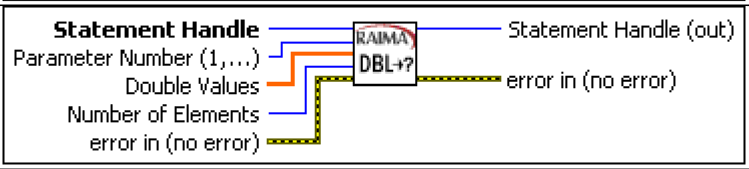
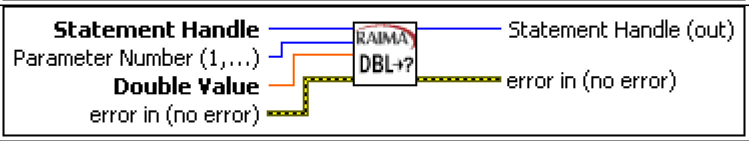
- A serial connection from Windows to cRIO-9024 can be useful during real-time testing. On the device, be sure that the switch “CONSOLE OUT” is turned on (left side). Between the two computers, connect a Null Modem serial cable. From Windows, use a remote connection program like PuTTY, select Serial connection with 9600 baud. On the console, you will see RDM print its document root every time it is activated. Use `^H` for backspace. Also, the commands “`pwd`”, “`cd`” and “`ls`” behave as expected.
- The directory onto which LabVIEW places its add-on object code modules is `/c/ni-rt/system`. You may use FTP to place `rdmNative-12.out` in this location if necessary (for example, if you obtain an update from www.raima.com/ni, FTP will allow you to overwrite the original file.
- If it appears that your configuration of VxWorks on cRIO does not have the right support modules installed for RDM to run, you may find out extra information from the console:
  - `cd “ni-rt/system”`
  - `ld < rdmNative-12.out`
- If there are any symbols missing or duplicated, they will be listed when you attempt to load the RDM library.
- In addition to other methods, you can use the console to reboot the cRIO-9024:
  - `reboot`
- Data, by default, is placed into the “LabVIEW Data” directory, located in /c/ni-rt/LabVIEW Data. You can view the database(s) contained there as follows:
  - `cd “ni-rt/LabVIEW Data”`
  - `ls`

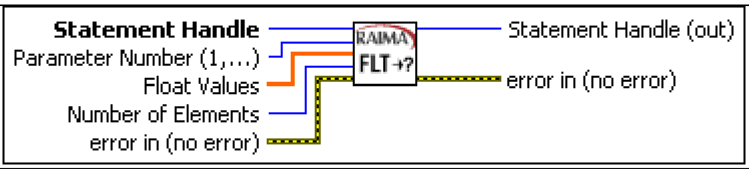
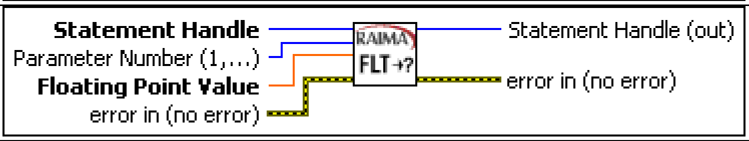
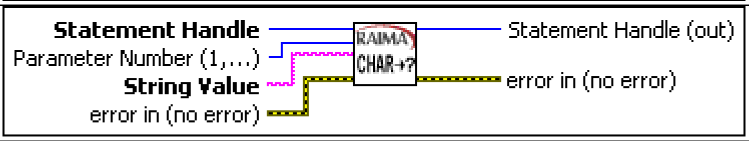
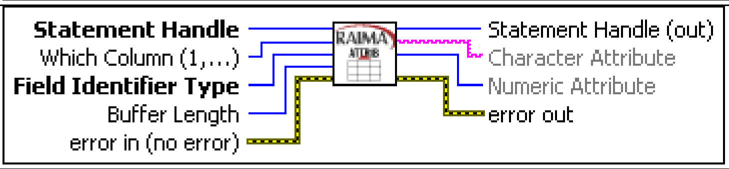
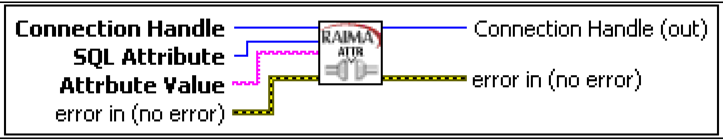
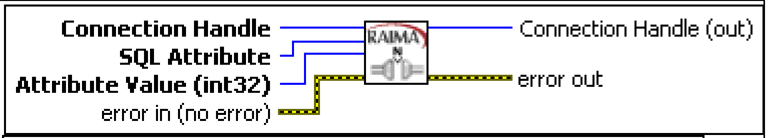
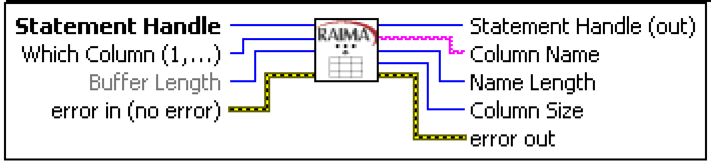
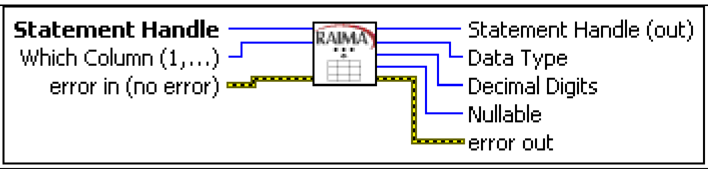
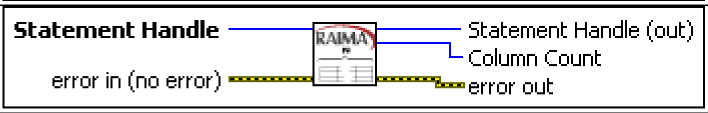
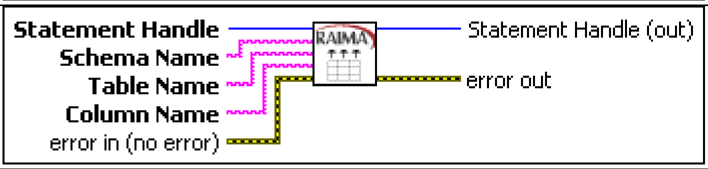
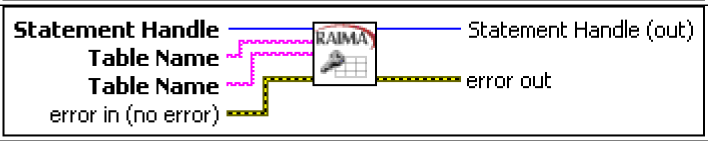
3.5 The Complete API

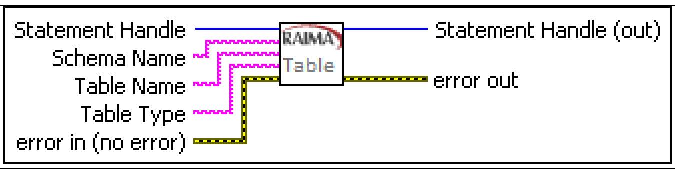
The following table shows the functions organized as they appear in the Functions Palette.

Palette	Function Name	Prototype
Main	Allocate a Handle	 Handle SQL Handle Type error in (no error) RAIMA Handle (out) error out
Main	Connect to Data Source	 Connection Handle Server error in (no error) RAIMA Connection Handle (out) error out
Main	End a Transaction	 Handle SQL Handle Type Completion Type error in (no error) RAIMA Handle (out) error out
Main	Extended Transaction Operation	 Statement Handle Transaction Type Read Only (T/F) Transaction ID error in (no error) RAIMA TX++ Statement Handle (out) error out
Main	Free a Handle	 Handle SQL Handle Type error in (no error) RAIMA Handle (out) error out
SQL	Close a Cursor	 Statement Handle error in (no error) RAIMA Statement Handle (out) error out
SQL	Count of Affected Rows	 Statement Handle error in (no error) RAIMA Statement Handle (out) Row Count error out
SQL	Execute a SQL Statement Now	 Statement Handle SQL Statement error in (no error) RAIMA EXE! Statement Handle (out) error out
SQL	Execute a SQL Statement	 Statement Handle error in (no error) RAIMA EXE Statement Handle (out) error out
SQL	Fetch Data and Move Cursor	 Statement Handle SQL Fetch Orientation Offset error out RAIMA Statement Handle (out) error out
SQL	Fetch the Next Row	 Statement Handle error in (no error) RAIMA Statement Handle (out) No More Data error out
SQL	Free a Statement	 Statement Handle Free Statement Option Type error in (no error) RAIMA Statement Handle (out) error out
Palette	**Function Name**	**Prototype**
SQL	Get a Cursor Name	 Statement Handle Length of Cursor Name Buffer error in (no error) RAIMA Statement Handle (out) Cursor Name Actual Length of Cursor Name error out

SQL	Prepare a Statement	 Statement Handle SQL Statement error in (no error) RAIMA (?,?) Statement Handle (out) error out
SQL	Request More Results	 Statement Handle error in (no error) RAIMA Statement Handle (out) error out
SQL	Set a Cursor Name	 Statement Handle Cursor Name error in (no error) RAIMA NAME Statement Handle (out) error out
Data	Get Rowset Data Polymorphic 8-bit Integer Array	 Statement Handle Which Column (1,...) Number of Elements error in (no error) RAIMA 8 Statement Handle (out) 8-bit Values Number of Elements error out
Data	Get Rowset Data Polymorphic 8-bit Integer	 Statement Handle Which Column (1,...) error in (no error) RAIMA 8 Statement Handle (out) 8-bit Value error out
Data	Get Rowset Data Polymorphic 16-bit Integer Array	 Statement Handle Which Column (1,...) Number of Elements error in (no error) RAIMA 16 Statement Handle (out) 16-bit Values Number of Elements error out
Data	Get Rowset Data Polymorphic 16-bit Integer	 Statement Handle Which Column (1,...) error in (no error) RAIMA 16 Statement Handle (out) 16-bit Value error out
Data	Get Rowset Data Polymorphic 32-bit Integer Array	 Statement Handle Which Column (1,...) Number of Elements error in (no error) RAIMA 32 Statement Handle (out) 32-bit Values Number of Elements error out
Data	Get Rowset Data Polymorphic 32-bit Integer	 Statement Handle Which Column (1,...) error in (no error) RAIMA 32 Statement Handle (out) 32-bit Value error out
Data	Get Rowset Data Polymorphic 64-bit Integer Array	 Statement Handle Which Column (1,...) Number of Elements error in (no error) RAIMA 64 Statement Handle (out) 64-bit Values Number of Elements error out
Data	Get Rowset Data Polymorphic 64-bit Integer	 Statement Handle Which Column (1,...) error in (no error) RAIMA 64 Statement Handle (out) 64-bit Value error out
Data	Get Rowset Data Polymorphic String	 Statement Handle Which Column (1,...) Length of String Value error in (no error) RAIMA CHAR Statement Handle (out) String Value error out
Data	Get Rowset Data Polymorphic Double Array	 Statement Handle Which Column (1,...) Number of Elements error in (no error) RAIMA DBL Statement Handle (out) Double Values Number of Elements error out
Data	Get Rowset Data Polymorphic Double	 Statement Handle Which Column (1,...) error in (no error) RAIMA DBL Statement Handle (out) Double Value error out

Data	Get Rowset Data Polymorphic Float Array	 Statement Handle Which Column (1,...) Number of Elements error in (no error) RAIMA FLT Statement Handle (out) Float Values Number of Elements error out
Data	Get Rowset Data Polymorphic Float	 Statement Handle Which Column (1,...) error in (no error) RAIMA FLT Statement Handle (out) Float Value error out
Data	Set a Parameter Value Polymorphic 8-bit Integer Array	 Statement Handle Parameter Number (1,...) Number of Elements error in (no error) RAIMA Statement Handle (out) error out
Data	Set a Parameter Value Polymorphic 8-bit Integer	 Statement Handle Parameter Number (1,...) 8-bit Value error in (no error) RAIMA Statement Handle (out) error out
Data	Set a Parameter Value Polymorphic 16-bit Integer Array	 Statement Handle Parameter Number (1,...) 16-bit Values Number of Elements error in (no error) RAIMA Statement Handle (out) error out
Data	Set a Parameter Value Polymorphic 16-bit Integer	 Statement Handle Parameter Number (1,...) 16-bit Value error in (no error) RAIMA Statement Handle (out) error out
Data	Set a Parameter Value Polymorphic 32-bit Integer Array	 Statement Handle Parameter Number (1,...) 32-bit Values Number of Elements error in (no error) RAIMA Statement Handle (out) error out
Data	Set a Parameter Value Polymorphic 32-bit Integer	 Statement Handle Parameter Number (1,...) 32-bit Value error in (no error) RAIMA Statement Handle (out) error in (no error)
Data	Set a Parameter Value Polymorphic 64-bit Integer Array	 Statement Handle Parameter Number (1,...) 64-bit Values Number of Elements error in (no error) RAIMA Statement Handle (out) error in (no error)
Data	Set a Parameter Value Polymorphic 64-bit Integer	 Statement Handle Parameter Number (1,...) 64-bit Value error in (no error) RAIMA Statement Handle (out) error in (no error)
Data	Set a Parameter Value Polymorphic Double Array	 Statement Handle Parameter Number (1,...) Double Values Number of Elements error in (no error) RAIMA DBL→? Statement Handle (out) error in (no error)
Data	Set a Parameter Value Polymorphic Double	 Statement Handle Parameter Number (1,...) Double Value error in (no error) RAIMA DBL→? Statement Handle (out) error in (no error)

Data	Set a Parameter Value Polymorphic Float Array	 Statement Handle Parameter Number (1,...) Float Values Number of Elements error in (no error) RAIMA FLT→? Statement Handle (out) error in (no error)
Data	Set a Parameter Value Polymorphic Float	 Statement Handle Parameter Number (1,...) Floating Point Value error in (no error) RAIMA FLT→? Statement Handle (out) error in (no error)
Data	Set a Parameter Value Polymorphic String	 Statement Handle Parameter Number (1,...) String Value error in (no error) RAIMA CHAR→? Statement Handle (out) error in (no error)
Attributes	Column Attributes	 Statement Handle Which Column (1,...) Field Identifier Type Buffer Length error in (no error) RAIMA ATTRIB Statement Handle (out) Character Attribute Numeric Attribute error out
Attributes	Set Connection Attributes	 Connection Handle SQL Attribute Attrbute Value error in (no error) RAIMA ATTR Connection Handle (out) error in (no error)
Attributes	Set Connection Attributes	 Connection Handle SQL Attribute Attribute Value (int32) error in (no error) RAIMA N Connection Handle (out) error out
Metadata	Describe a Column	 Statement Handle Which Column (1,...) Buffer Length error in (no error) RAIMA Statement Handle (out) Column Name Name Length Column Size error out
Metadata	Describe a Column Extended	 Statement Handle Which Column (1,...) error in (no error) RAIMA Statement Handle (out) Data Type Decimal Digits Nullable error out
Metadata	Number of Result Columns	 Statement Handle error in (no error) RAIMA N Statement Handle (out) Column Count error out
Metadata	Request Metadata for Columns	 Statement Handle Schema Name Table Name Column Name error in (no error) RAIMA Statement Handle (out) error out
Metadata	Request Metadata for Primary Keys	 Statement Handle Table Name Table Name error in (no error) RAIMA Statement Handle (out) error out

Metadata	Request Metadata for Tables	 Statement Handle Schema Name Table Name Table Type error in (no error) RAIMA Table Statement Handle (out) error out

3.4 Error Codes

A large number of error codes may potentially be returned by the RDM SQL module. The following table provides the error code, the SQL State (which may be used in a keyword search on the web for more detailed information), and the RDM-specific short description.

Code	SQL State	Description
402160	1004	data truncation
402161	RX003	psp subsystem initialization failure
402162	42000	syntax error
402163	42000	no CREATE DATABASE has been issued
402164	RX004	unable to open RDM core-level DDL file
402165	42000	column is not declared in referenced table
402166	42000	matching primary key does not exist in referenced table
402167	42000	foreign/primary key columns do not match
402168	42000	foreign key column is not declared in table
402169	42S21	column already declared in table
402170	42S21	table with the same name already been created
402171	3C000	duplicate cursor name
402172	42000	specified domain name not found
402173	42S02	table not found
402174	22008	date/time value overflow
402175	42000	key column not found
402176	RX005	unable to open file
402177	HY013	insuffcient memory available for operation
402178	42000	bad formatting specification
402179	22005	bad binary literal specification
402180	22005	bad literal specification
402181	22001	string literal too long
402182	42000	database does not exist

402183	42000	unable to open catalog file
402184	42000	unable to initialize database
402185	RX006	file I/O error
402186	8003	connection is not open
402187	HY009	invalid argument value
402188	HY009	invalid use of null pointer
402189	HY010	must free all connection handles first
402190	HY090	invalid string or buffer length
402191	7006	data type attribute violation
402192	7009	invalid descriptor index (column number)
402193	7009	invalid descriptor index (parameter number)
402194	HY010	function sequence error
402195	25000	transaction is active
402196	25000	transaction not active
402197	RX007	RDM runtime error
402198	8000	must call before connect
402199	42000	databases to be opened already specified
402200	42000	database not open
402201	21S01	insert value list does not match column list
402202	21S02	SELECT result columns do not match column list
402203	42S22	column not found
402204	7002	insufficient number of parameters specified
402205	42000	must specify value for column
402206	42000	table name not in FROM clause
402207	42S22	column name not found
402208	HY106	fetch type out of range
402209	HY107	row value out of range
402210	HY109	invalid cursor position
402211	24000	invalid cursor state
402212	24000	current/cursor's statement is not SELECT
402213	22003	numeric value out of range
402214	22003	significant data lost due to truncation

402215	23000	referential integrity error
402216	HY010	connection not closed
402217	HY001	driver memory allocation error
402218	HY024	invalid attribute value
402219	HY092	invalid attribute/option identifier
402220	01S01	error in row
402221	01S02	option value changed to default
402222	42000	data type mismatch
402223	22019	invalid escape character
402224	RX008	invalid statement state
402225	42000	aggregate functions not allowed in WHERE
402226	22012	division by zero
402227	42000	escape clause syntax error
402228	42000	prior prepared DDL statement not executed
402229	42000	invalid use of parameter marker
402230	42000	duplicate stored procedure name
402231	RX009	stored procedure file not found
402232	21000	invalid number of arguments specified
402233	42000	joined columns must match exactly
402234	42000	too many columns declared in foreign/primary key
402235	42000	no access path between outer joined tables
402236	8002	connection already in use
402237	42000	UDF/UDP/XTF already registered
402238	42000	all standard tables must be declared before first virtual table
402239	42000	data type not allowed for virtual table columns
402240	RX010	virtual table function error
402241	42000	reference to unregistered UDF/UDP/Virtual Table
402242	RX011	user-defined function error
402243	RX012	no result from user-defined function
402244	RX013	UDFLOADTABLE entry definition error
402245	21000	invalid funtion argument type
402246	21000	incorrect number of funtion arguments

402247	42000	SET NULL cannot be specified when nulls are not allowed
402248	42000	invalid column reference in INSERT expression
402249	42000	duplicate primary/unique key value
402250	42000	SET NULL not allowed with ON UPDATE
402251	42000	ON UPDATE CASCADE not allowed when foreign key column is used in a key
402252	42000	changes to referenced restricted primary/unique key not allowed
402253	42000	duplicate join column
402254	42000	no matching join columns
402255	42000	invalid order/group by column reference
402256	21000	invalid function argument
402257	42000	invalid number of specified function arguments
402258	RX014	operation is read only
402259	RX015	invalid statement type
402260	42000	only one DISTINCT aggregate function is allowed
402261	HY008	statement execution canceled by user
402262	42000	aggregate functions not allowed in GROUP BY
402263	7006	invalid C data type
402264	7006	invalid SQL data type
402265	7005	prepared statement is not a valid cursor
402266	7009	invalid descriptor index
402267	25S01	transaction state unknown
402268	HY007	associated statement is not prepared
402269	HY011	invalid operation at this time
402270	HY012	invalid transaction operation code
402271	HY016	cannot modify an implementation row descriptor
402272	HY017	invalid use of implicit descriptor handle
402273	HY021	inconsistent descriptor information
402274	HY091	invalid descriptor field identifier
402275	HY094	invalid scale value
402276	HY105	invalid parameter type
402277	HYT00	timeout expired

402278	HYC00	driver not capable
402279	HYC00	optional feature not implemented
402280	HYC00	invalid conversion
402281	22002	indicator variable required but not supplied
402282	IM001	function not supported
402283	42000	invalid TFS location spec - should be: "tfs*//*/" or "@hostname:port"
402284	42000	invalid TFS type
402285	RX016	import/export error
402286	RX017	TFS system error
402287	RX018	JNI/ADO.Net system error
402288	RX019	RPC communication error
402289	25000	read-only transaction is active
402290	25000	unlock not allowed in a transaction
402291	25000	table is not locked
402292	25000	operation not allowed due to active read locks
402293	42000	DDL requires that no databases be open
402294	25000	multiple database transactions are not allowed
402295	42000	invalid date format
402296	42000	invalid date separator
402297	42000	invalid db open mode
402298	RX020	operation requires exclusive database access
402299	42000	SELECT or column is not updateable
402300	42000	default values not allowed on long var{char|binary} columns
402301	42000	blobs cannot be referenced in expressions in deferred mode
402302	RX021	data-at-exec params only allowed with INSERT VALUES/UPDATE
402303	RX022	data-at-exec params only allowed for blob (long var...) columns
402304	RX023	data-at-exec param type not compatible with blob (long var...) column
402305	HY020	attempt to concatenate a null value
402306	HY095	function type out of range
402307	HY097	column type out of range
402308	HY098	scope out of range
402309	HY099	nullable type out of range

402310	HY100	Uniqueness option type out of range
402311	HY101	Accuracy option type out of range
402312	HY004	invalid SQL data type
402313	42000	circular tables cannot be referenced
402314	RX024	unable to connect to TFS
402315	42000	must specify '(length)' with variable size columns
402316	42000	cannot specify both MAXPGS and MAXROWS options
402317	42000	cannot delete rows from a circular table
402318	42000	maxrows can only be specified with CREATE CIRCULAR TABLE
402319	42000	maxrows must be specified with CREATE CIRCULAR TABLE
402320	42000	cannot refer to blob column in WHERE in deferred reading mode
402321	42000	database already exists
402322	RX025	TFS already initialized
402323	RX026	illegal locking mode
402324	42000	no columns have been updated
402325	42000	operation not allowed when autocommit is enabled
402326	42000	positioned UPDATE/DELETE table does not match cursor's
402327	42000	positioned UPDATE/DELETE not allowed in stored procedure
402328	25000	Inconsistent read-only transaction commit/rollback/end call
402329	42000	duplicate table reference in FROM clause
402330	42000	another database is already open in different mode
402331	42000	cannot call an aggregate function within an aggregate function
402332	42000	char/wchar type is required
402333	8001	unable to connect
402334	25000	cursor's read locks freed by intervening commit/rollback
402335	RX027	invalid transaction id
402336	42000	db union open invalid when other database is open
402337	42000	database unavailable due to exclusive access rules
402338	42000	database has already been opened
402339	42000	database is opened for read only
402340	42000	sorting on a blob column is not allowed
402341	42000	blob columns cannot be referenced in a SELECT with GROUP BY

402342	HY009	invalid argument type
402343	42000	statements from different connections
402344	42000	unable to process outer join specification
402345	42000	invalid access--use rsqlShowPlan/SQLShowPlan function
402346	42000	query() cannot be used in SELECT result column
402347	42000	virtual table access restricted to INSERT or SELECT
402348	42000	out of space in virtual table
402349	42000	blob columns cannot be used in WHERE clause of a SELECT with ORDER BY
402350	42000	invalid data type mapping
402351	7009	named parameter not found
402352	RX999	unused error code
402353	RX029	database is currently in use
402354	RX030	database is being used by other task(s)/user(s)
402355	42000	function cannot be called from a UDF
402356	42000	database not opened - use read-only mode to open core databases
402357	RX031	incompatible catalog version
402358	42000	ON condition only allows equi-join predicates
402359	42000	result column must have aggregate function call
402360	8001	the specified DOCROOT is already in use
402361	42000	no registered function interface for virtual table
402362	42000	database does not contain any virtual tables
402363	34000	invalid cursor name
402364	0T000	target table does not match cursor specification
402365	42000	FOR UPDATE column not in SELECT list
402366	RX032	stored procedure has an invalid version
402367	42000	distinct can only be specified with aggregate UDF
402368	42000	database does not contain any tables
402369	42000	core DDL keyword cannot be used in SQL DDL
402370	42000	invalid import/export file type
402371	42000	a slave database can only be opened in readonly mode
402372	42000	rowid columns can only be declared as primary or foreign keys

402373	42000	invalid rowid value
402374	42000	no last_insert_id is available
402375	42000	SELECT cannot have ORDER/GROUP BY clause
402376	42000	ON UPDATE CASCADE not allowed on rowid primary key references
402377	42000	UPDATE of rowid primary key column is now allowed
402378	42S21	duplicate table aliases
402379	42000	another update stats is already active
402380	42000	GROUP BY column expressions cannot have parameter markers
402381	42000	Variant argument value in INVAR aggregate function
402382	42000	bad function argument
402383	42000	bad wild all character (must be '%' or '*')
402384	42000	bad wild one character (must be '_', '.', or '?')
402385	42000	sort cost factor must be > 0.0 and < 1.0
402386	42000	cache size setting must be >= 100
402387	42000	valid debug mode values are 0,1,2,3
402388	42000	invalid DECIMAL precision. Must be > 0 and <= 32
402389	42000	invalid DECIMAL scale. Must be >= 0 and <= prec
402390	42000	DECIMAL value overflow
402391	42000	DEFAULT AUTO is only allowed with guid data types
402392	42000	invalid UUID/GUID value
402393	42000	binary DEFAULT value not correct length
402394	42000	invalid task ID
402395	42000	invalid encryption type
402396	42000	unsupported encryption type
402397	42000	encryption key not found
402398	42000	virtual table access restricted to one active hstmt per hconn
402399	RX028	yet to be implemented feature
402409	42000	system error
402410	42000	NULL/invalid connection or statement handle

4. ADDITIONAL RESOURCES

4.1 Support

Free “quickstart” support is available to help you get this package up and running. Call 206-748-5200 or write quickstart@raima.com.

To purchase full technical support, please contact Raima directly at www.raima.com.

Benefits of support include:

- Support provided through Raima Monday through Friday 8am PT to 4pm PT
- One Named Caller
- Unlimited Number of Incidents
- Response - Next Business Day
- Online Product Documentation
- Online Technical Forum
- Online Incident Tracking & Updates
- Customer Determined Incident Severity Level
- Incident Escalation Process
- Critical Patch Notification
- Patch Distribution
- Updates (i.e. 1.1 to 1.2 – change to right of decimal point)

National Instruments does not provide technical support for third-party add-ons for LabVIEW.

Please check out www.raima.com/ni for product information and any updates to RDM Native or the RDM DLLs.

4.2 LabVIEW Page

National Instruments provides tips, techniques and FAQ's at www.ni.com/labviewtools/raima. This page will have all of the informational resources specially designed for LabVIEW programmers.

Want to know more?

Please call us to discuss your database needs or email us at info@raima.com. You may also visit our website for the latest news, product downloads and documentation:

www.raima.com