RDB Connect Software Documentation

Version 5.0

Contents

1 Introduction

RDB Connect is a collection of commands and functions allowing IBM i users connection to remote databases. RDB Connect is pre-configured to access MySQL, Microsoft SQL Server, Oracle, PostGre SQL, DB2 8.1 and above, and DB2 for the IBM i. Other databases can be configured manually (see RDBJARCFG). An ODBC connection is also available via an included PC component.

RDB Connect will run on IBM i with an operating system version of V6R1M0 and above. It requires an IP connection to the remote server that is running the database to access.

This document will cover the usage of RDB Connect functions and the commands that are shipped with the software. Service program function prototypes and example usage are available in the source file **RDB50/RDBSRC**.

After installing RDB Connect, a subsystem called **RDBSBS** will be created in the RDB50 library. This subsystem must be active to use RDB Connect. The subsystem will contain a job called **RDB CONNECT** jobs that handle the processing of the requests.

Technical support is available M-T 8:00 am - 5:00 pm CST. Friday 8:00 am – 4:00 pm CST (except holidays)

Email – help@prodatacomputer.com
Phone – 1.800.228.6318 option 2

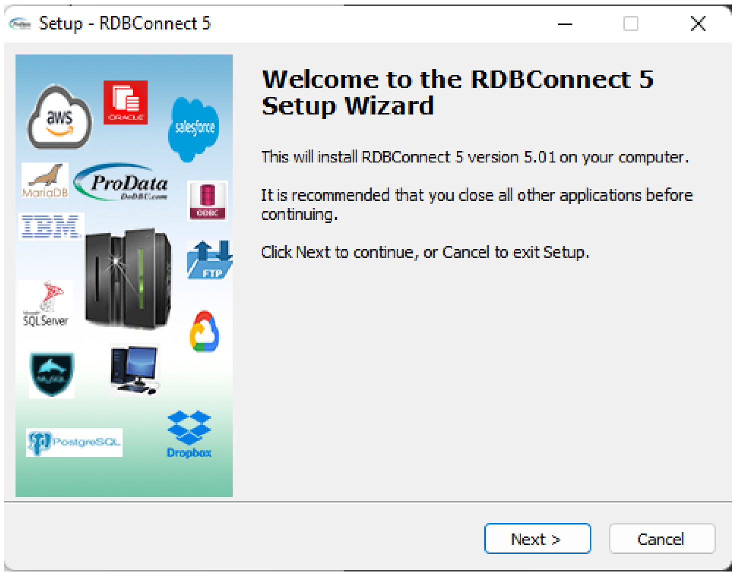
NOTE: Before installing RDBConnect 5, check if Java 1.6 or above is installed and running on your System for better performance and JDBC driver compatibility.

This version of RDBConnect requires version V6R1M0 or above.

2
Software Installation

1 Install the software using .exe file

A splash screen will appear and a series of notices informing you of the process being performed. After which the following screen should appear. Click the **Next** button to continue the installation process.



Welcome to the RDBConnect 5 Setup Wizard

This will install RDBConnect 5 version 5.01 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure 2-1

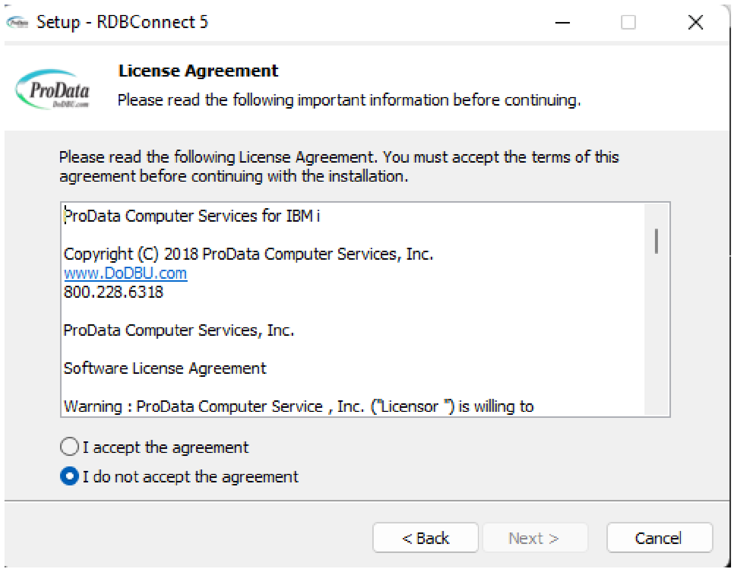
2 Installation wizard requires access to IBMi

The installation process requires a connection being established to your iSeries (AS/400) host computer. The following notice may appear informing you for the need of a connection.
Once you have verified the connection to your iSeries (AS/400) host.

You must have *IOSYSCFG authority. Once you have verified your authority, click the OK button to continue the installation process.

3 License Agreement

Please read the *License Agreement* and upon accepting the agreement, click the **Next** button to continue the installation process.



Setup - RDBConnect 5

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

ProData Computer Services for IBM i

Copyright (C) 2018 ProData Computer Services, Inc.
www.DoDBU.com
800.228.6318

ProData Computer Services, Inc.

Software License Agreement

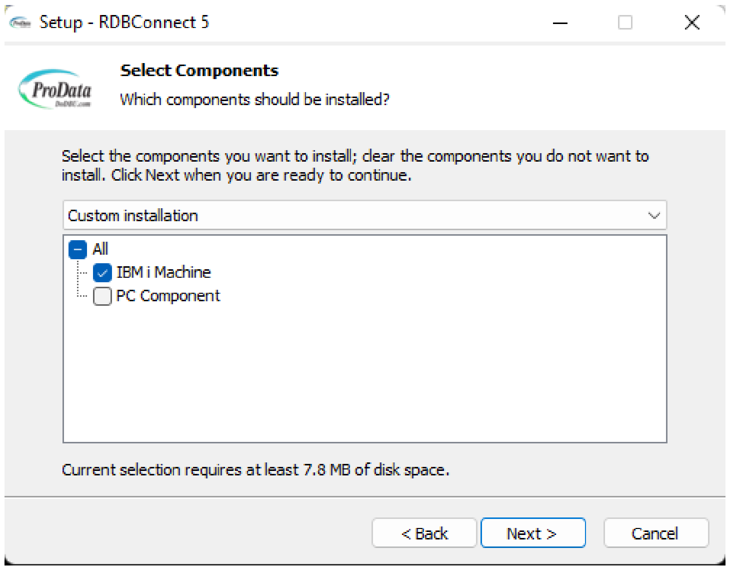
Warning : ProData Computer Service , Inc. ("Licensor ") is willing to

I accept the agreement [✓ I do not accept the agreement]

Figure 2-2

4 Select Components

This screen gives you the option to select which components you would like to install. You must select IBM i Machine to complete the installation.



Setup - RDBConnect 5

Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Custom installation

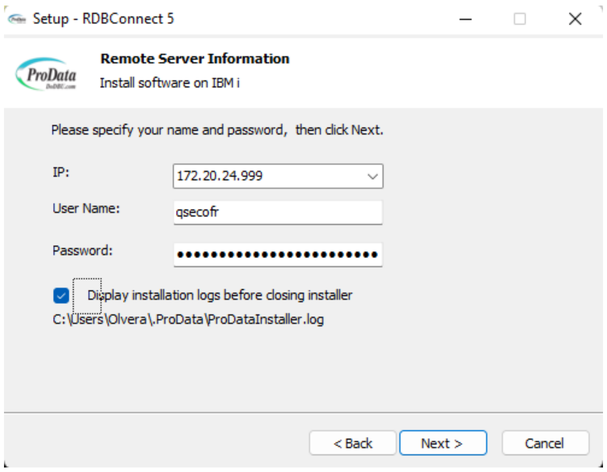
- All
  - ✓ IBM i Machine
  - ☐ PC Component

Current selection requires at least 7.8 MB of disk space.

Figure 2-3

5 IBMi Credentials required

Type the **IP Address, User and Password** in the space provided of the iSeries (AS/400) host computer. Once you have completed the iSeries (AS/400) host computer selection, click the **Nex**t button to continue.

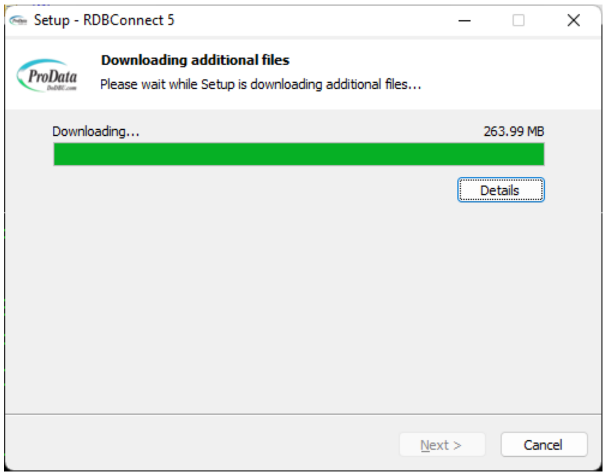


Remote Server Information
Install software on IBM i

Please specify your name and password, then click Next.

IP: 172.20.24.999
User Name: qsecofr
Password: ••••••••••••••••••••••••

✓ Display installation logs before closing installer
C:\Users\Olvera\.ProData\ProDataInstaller.log



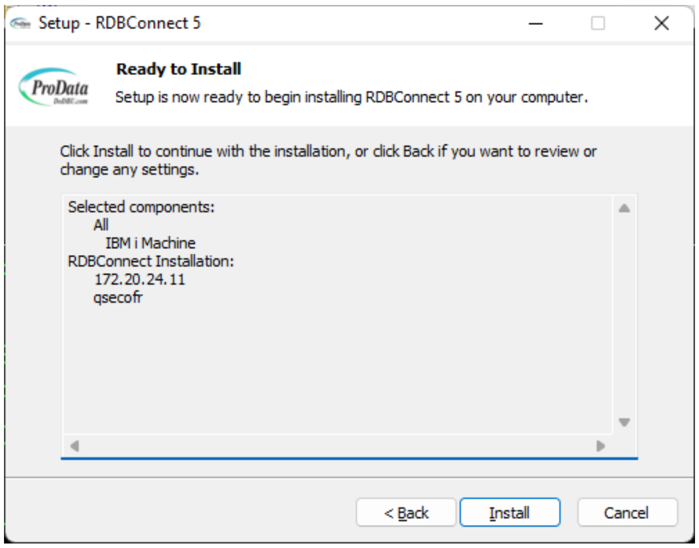
Downloading additional files
Please wait while Setup is downloading additional files...

Downloading... 263.99 MB

Figure 2-4 **Figure 2-5**

6 Ready to install on your IBMi

Click Next to install RDBConnect on the iSeries(AS/400) host computer.



Ready to Install
Setup is now ready to begin installing RDBConnect 5 on your computer.

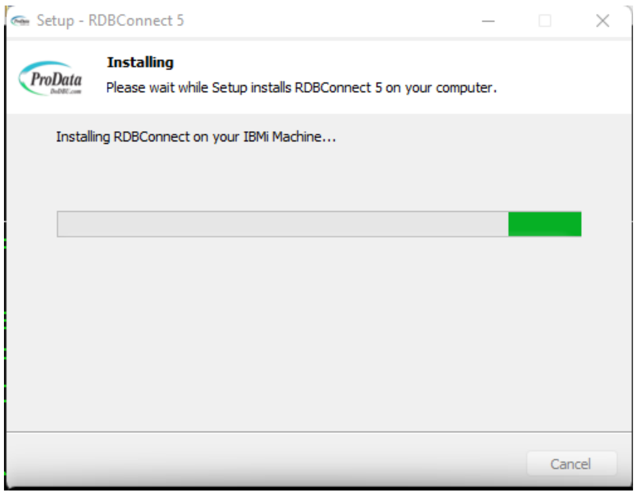
Click Install to continue with the installation, or click Back if you want to review or change any settings.

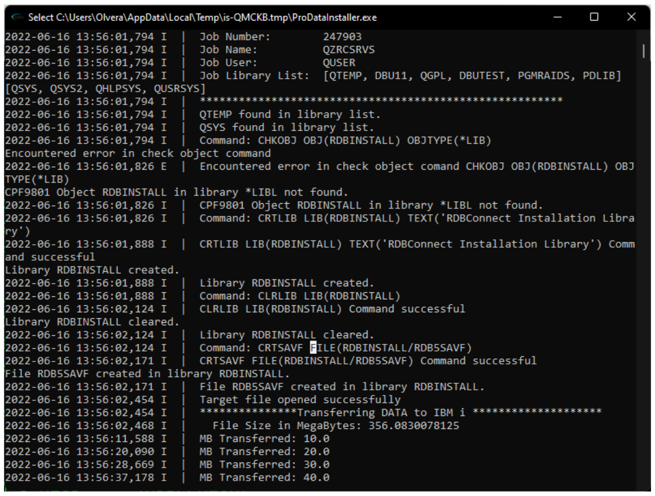
```
Selected components:
    All
      IBM i Machine
RDBConnect Installation:
    172.20.24.11
    qsecofr
```

Figure 2-5

7 Ready to install on your IBMi

A Windows DOS will appear displaying the transfer process.


Setup - RDBConnect 5
Installing
Please wait while Setup installs RDBConnect 5 on your computer.
Installing RDBConnect on your IBMi Machine...
Cancel


Select C:\Users\Olvera\AppData\Local\Temp\is-QMCKB.tmp\ProDataInstaller.exe

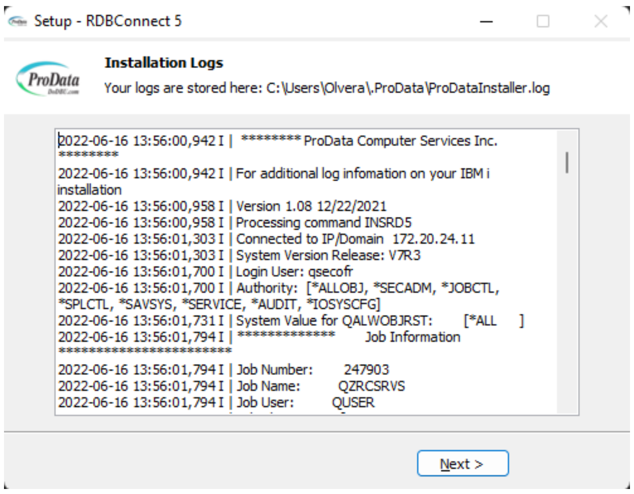
```
2022-06-16 13:56:01,794 I  |  Job Number:        247903
2022-06-16 13:56:01,794 I  |  Job Name:          QZRCSRVS
2022-06-16 13:56:01,794 I  |  Job User:          QUSER
2022-06-16 13:56:01,794 I  |  Job Library List:  [QTEMP, DBU11, QGPL, DBUTEST, PGMRAIDS, PDLIB]
[QSYS, QSYS2, QHLPSYS, QUSRSYS]
2022-06-16 13:56:01,794 I  |  ****************************************************
2022-06-16 13:56:01,794 I  |  QTEMP found in library list.
2022-06-16 13:56:01,794 I  |  QSYS found in library list.
2022-06-16 13:56:01,794 I  |  Command: CHKOBJ OBJ(RDBINSTALL) OBJTYPE(*LIB)
Encountered error in check object command
2022-06-16 13:56:01,826 E  |  Encountered error in check object comand CHKOBJ OBJ(RDBINSTALL) OBJ
TYPE(*LIB)
CPF9801 Object RDBINSTALL in library *LIBL not found.
2022-06-16 13:56:01,826 I  |  CPF9801 Object RDBINSTALL in library *LIBL not found.
2022-06-16 13:56:01,826 I  |  Command: CRTLIB LIB(RDBINSTALL) TEXT('RDBConnect Installation Libra
ry')
2022-06-16 13:56:01,888 I  |  CRTLIB LIB(RDBINSTALL) TEXT('RDBConnect Installation Library') Comm
and successful
Library RDBINSTALL created.
2022-06-16 13:56:01,888 I  |  Library RDBINSTALL created.
2022-06-16 13:56:01,888 I  |  Command: CLRLIB LIB(RDBINSTALL)
2022-06-16 13:56:02,124 I  |  CLRLIB LIB(RDBINSTALL) Command successful
Library RDBINSTALL cleared.
2022-06-16 13:56:02,124 I  |  Library RDBINSTALL cleared.
2022-06-16 13:56:02,124 I  |  Command: CRTSAVF FILE(RDBINSTALL/RDB5SAVF)
2022-06-16 13:56:02,171 I  |  CRTSAVF FILE(RDBINSTALL/RDB5SAVF) Command successful
File RDB5SAVF created in library RDBINSTALL.
2022-06-16 13:56:02,171 I  |  File RDB5SAVF created in library RDBINSTALL.
2022-06-16 13:56:02,454 I  |  Target file opened successfully
2022-06-16 13:56:02,454 I  |  ***************Transferring DATA to IBM i ********************
2022-06-16 13:56:02,468 I  |    File Size in MegaBytes: 356.0830078125
2022-06-16 13:56:11,588 I  |  MB Transferred: 10.0
2022-06-16 13:56:20,090 I  |  MB Transferred: 20.0
2022-06-16 13:56:28,669 I  |  MB Transferred: 30.0
2022-06-16 13:56:37,178 I  |  MB Transferred: 40.0
```

Figure 2-6

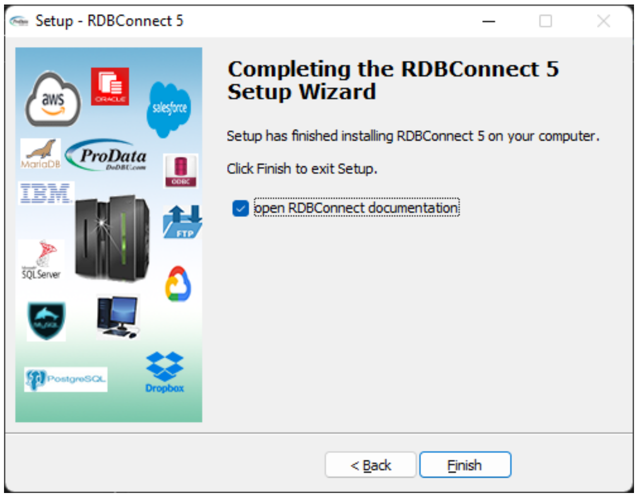
Figure 2-7

8 Installation logs

Next screen shows the installation logs and last screen is the **Completion** screen. Press the **Finish** button to complete the installation process.


Setup - RDBConnect 5
Installation Logs
Your logs are stored here: C:\Users\Olvera\.ProData\ProDataInstaller.log
Next >

```
2022-06-16 13:56:00,942 I |  ********* ProData Computer Services Inc.
*********
2022-06-16 13:56:00,942 I | For additional log infomation on your IBM i
installation
2022-06-16 13:56:00,958 I | Version 1.08 12/22/2021
2022-06-16 13:56:00,958 I | Processing command INSRD5
2022-06-16 13:56:01,303 I | Connected to IP/Domain  172.20.24.11
2022-06-16 13:56:01,303 I | System Version Release: V7R3
2022-06-16 13:56:01,700 I | Login User: qsecofr
2022-06-16 13:56:01,700 I | Authority:  [*ALLOBJ, *SECADM, *JOBCTL,
*SPLCTL, *SAVSYS, *SERVICE, *AUDIT, *IOSYSCFG]
2022-06-16 13:56:01,731 I | System Value for QALWOBJRST:      [*ALL     ]
2022-06-16 13:56:01,794 I | **************       Job Information
*************************
2022-06-16 13:56:01,794 I | Job Number:      247903
2022-06-16 13:56:01,794 I | Job Name:        QZRCSRVS
2022-06-16 13:56:01,794 I | Job User:        QUSER
```



Completing the RDBConnect 5 Setup Wizard

Setup has finished installing RDBConnect 5 on your computer.

Click Finish to exit Setup.

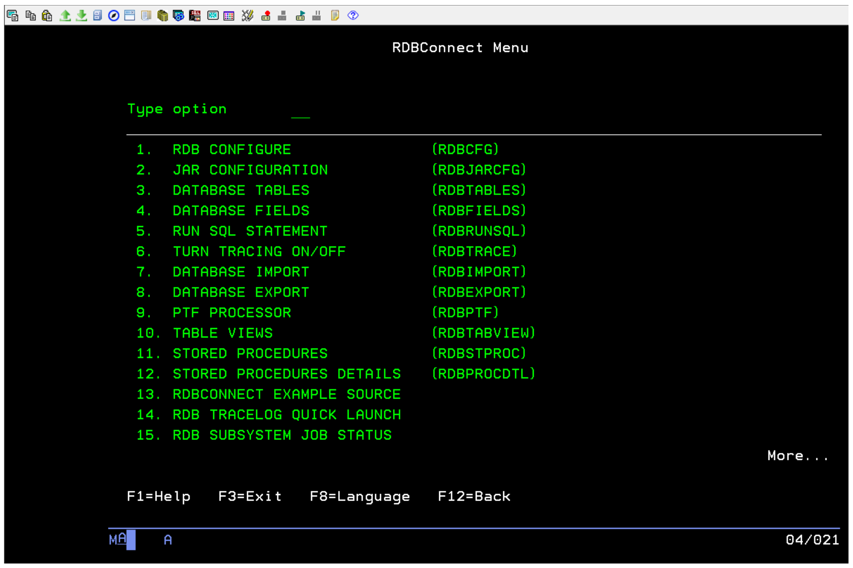
[✓] open RDBConnect documentation

Figure 2-8

Figure 2-9

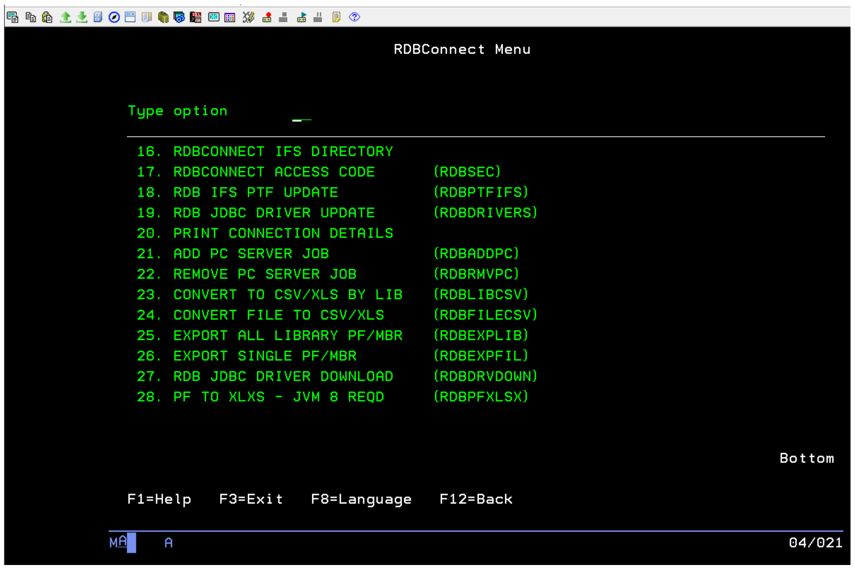
3 RDBConnect Menu

To start the RDB Connect menu, from a IBM i command line execute the command **ADDLIBLE RDB50 .** Execute the command **RDBMNU.** Type an option on the option field to execute the program.



```
                          RDBConnect Menu


     Type option        __

     1.  RDB CONFIGURE                 (RDBCFG)
     2.  JAR CONFIGURATION             (RDBJARCFG)
     3.  DATABASE TABLES               (RDBTABLES)
     4.  DATABASE FIELDS               (RDBFIELDS)
     5.  RUN SQL STATEMENT             (RDBRUNSQL)
     6.  TURN TRACING ON/OFF           (RDBTRACE)
     7.  DATABASE IMPORT               (RDBIMPORT)
     8.  DATABASE EXPORT               (RDBEXPORT)
     9.  PTF PROCESSOR                 (RDBPTF)
     10. TABLE VIEWS                   (RDBTABVIEW)
     11. STORED PROCEDURES             (RDBSTPROC)
     12. STORED PROCEDURES DETAILS     (RDBPROCDTL)
     13. RDBCONNECT EXAMPLE SOURCE
     14. RDB TRACELOG QUICK LAUNCH
     15. RDB SUBSYSTEM JOB STATUS
                                                          More...

     F1=Help    F3=Exit    F8=Language    F12=Back

 MA     A                                                  04/021
```

Figure 3-1



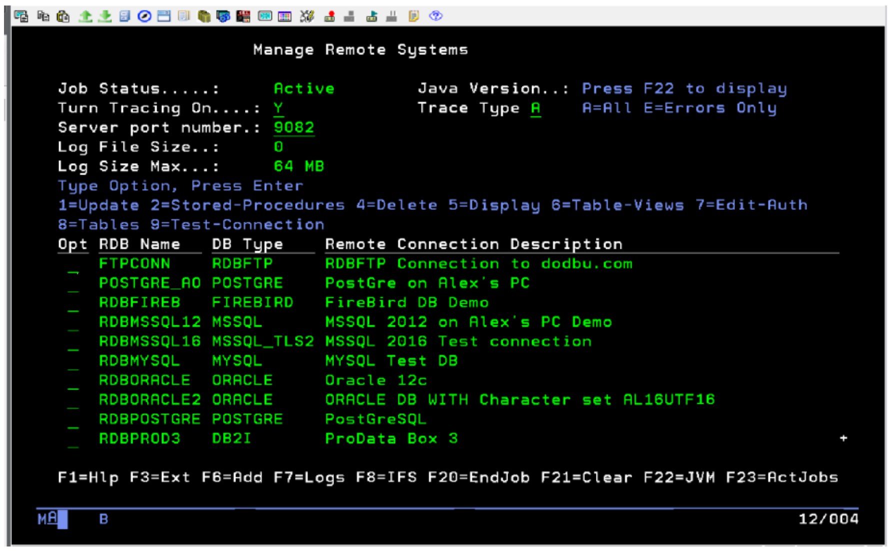
```
                          RDBConnect Menu


     Type option        __

     16. RDBCONNECT IFS DIRECTORY
     17. RDBCONNECT ACCESS CODE        (RDBSEC)
     18. RDB IFS PTF UPDATE            (RDBPTFIFS)
     19. RDB JDBC DRIVER UPDATE        (RDBDRIVERS)
     20. PRINT CONNECTION DETAILS
     21. ADD PC SERVER JOB             (RDBADDPC)
     22. REMOVE PC SERVER JOB          (RDBRMVPC)
     23. CONVERT TO CSV/XLS BY LIB     (RDBLIBCSV)
     24. CONVERT FILE TO CSV/XLS       (RDBFILECSV)
     25. EXPORT ALL LIBRARY PF/MBR     (RDBEXPLIB)
     26. EXPORT SINGLE PF/MBR          (RDBEXPFIL)
     27. RDB JDBC DRIVER DOWNLOAD      (RDBDRVDOWN)
     28. PF TO XLXS - JVM 8 REQD       (RDBPFXLSX)


                                                          Bottom

     F1=Help    F3=Exit    F8=Language    F12=Back

 MA     A                                                  04/021
```

Figure 3-2

Easy access options added.

4 Configuring RDB Connect

To start configuring RDB Connect, from a IBM i command line execute the command **RDBCFG** or option 1 from the RDB Connect Menu. This will take you to the **Manage Remote Systems** screen.



```
                        Manage Remote Systems

   Job Status.....:     Active         Java Version..: Press F22 to display
   Turn Tracing On....: Y              Trace Type A    A=All E=Errors Only
   Server port number.: 9082
   Log File Size..:     0
   Log Size Max...:     64 MB
   Type Option, Press Enter
   1=Update 2=Stored-Procedures 4=Delete 5=Display 6=Table-Views 7=Edit-Auth
   8=Tables 9=Test-Connection
   Opt RDB Name    DB Type    Remote Connection Description
    _  FTPCONN     RDBFTP     RDBFTP Connection to dodbu.com
    _  POSTGRE_AO  POSTGRE    PostGre on Alex's PC
    _  RDBFIREB    FIREBIRD   FireBird DB Demo
    _  RDBMSSQL12  MSSQL      MSSQL 2012 on Alex's PC Demo
    _  RDBMSSQL16  MSSQL_TLS2 MSSQL 2016 Test connection
    _  RDBMYSQL    MYSQL      MYSQL Test DB
    _  RDBORACLE   ORACLE     Oracle 12c
    _  RDBORACLE2  ORACLE     ORACLE DB WITH Character set AL16UTF16
    _  RDBPOSTGRE  POSTGRE    PostGreSQL
    _  RDBPROD3    DB2I       ProData Box 3                                  +

   F1=Hlp F3=Ext F6=Add F7=Logs F8=IFS F20=EndJob F21=Clear F22=JVM F23=ActJobs

 MA     B                                                            12/004
```

Figure 4-1

Maximum Connections – This is the maximum number of threads the **RDB CONNECT** job will create. The default number of jobs is 50.

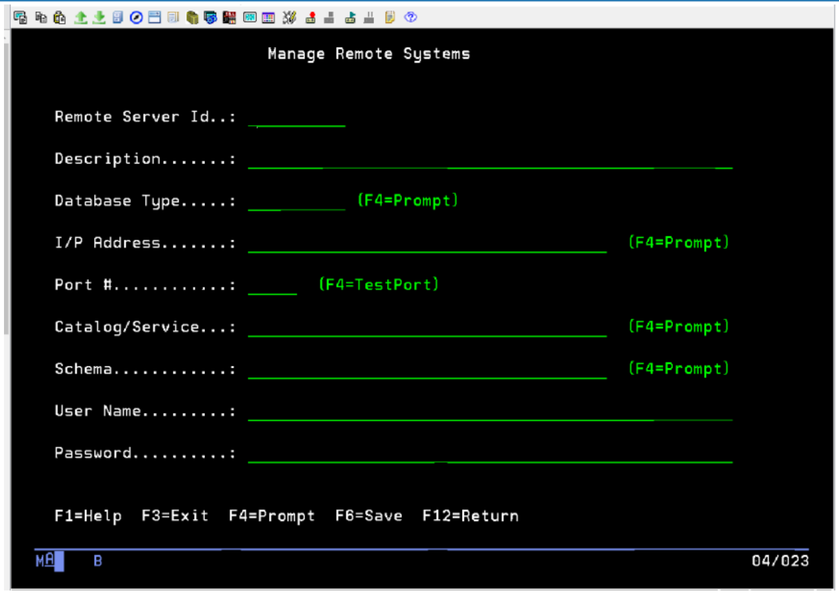
Turn Tracing On – Default is N. Set to Y when troubleshooting with ProData Tech Support. This value can be changed at any time from this program. The command **RDBTRACE** can also be used for this function.

*Trace Type – Specify A for all trace logs, E for errors only.
Server port number – This is the IP port that the **RDB CONNECT** job will use to answer requests for RDB transactions. This port must be available. 9082 is the default. Press **F6** to add a remote connection.

***Job Status -** This indicates the RDBConnect job status in subsystem RDBSBS
***Log File Size** – This is the actual size of the log tracing file in IFS
***Log Size Max** – This is the maximum file size before a warning message is sent to the operator if the warning flag is set to 'Y'.

*Function Keys – This will explain the addition of new function keys.

F1=Hlp – This will display the help file.
F3=Exit – This will exit.
F6=Add – This will allow you to add a new connection.
F7=Trace Logs – This will display the trace logs stored in IFS.
F8=IFS – This will take you to the IFS directory
F20=End Job – This will end the subsystem job RDBConnect in RDBSBS
F21= Clear Logs – This will clear the IFS trace logs.
F22=JVM – This will display the java version active on your system.



```
                         Manage Remote Systems

 Remote Server Id..: __________
 Description.......: __________________________________________________
 Database Type.....: __________  (F4=Prompt)
 I/P Address.......: _____________________________________  (F4=Prompt)
 Port #............: _____  (F4=TestPort)
 Catalog/Service...: _____________________________________  (F4=Prompt)
 Schema............: _____________________________________  (F4=Prompt)
 User Name.........: __________________________________________________
 Password..........: __________________________________________________

 F1=Help  F3=Exit  F4=Prompt  F6=Save  F12=Return
 MA     B                                                        04/023
```

Figure 4-2

Enter the information for the remote database.

Remote Server ID – This is the name that will be used when referencing this database connection.

Description – The description for this remote connection.

Database Type – The type of database being configured. Preconfigured databases:
MYSQL – www.mysql.org (an open source database)
MSSQL – Microsoft SQL Server 2000 and above (JTDS driver) **NOTE: If you are on Java 6 or below please use the following driver jtds-1.2.8.zip by changing file RDB50/RDBJVMDRVS row number 7**
MSSQL2 – Microsoft SQL Server 2000 and above (Microsoft driver). Requires i5/OS V6R1 and above using JRE 6 and above. **NOTE: If you are on Java 6 or below please use the following driver sqljdbc4.zip by changing file RDB50/RDBJVMDRVS row number 10**
ORACLE – Oracle server 9i and above
POSTGRE – www.postgresql.org (an open source database)
DB2 – IBM DB2 8 and above (not OS/400 or i5/OS)
DB2I – IBM DB2 for OS/400 and i5/OS
FIREBIRD – www.firebirdsql.com (an open source database)
ODBC – A PC based database using a System DSN. Must have RDB PC module loaded.

I/P Address – The I/P address used to access the remote database. This can also be an entry from the host table.

Port # – The port number to be used with the above I/P address to connect to the remote database. Each database type has a default port number, make sure to check your remote database port. ***NEW** feature to Test Port will allow check if that port is open or blocked by a firewall.

Catalog/Service – Some databases require a Catalog or Service to be specified on the connection. If one is required for the database specified, enter it here.

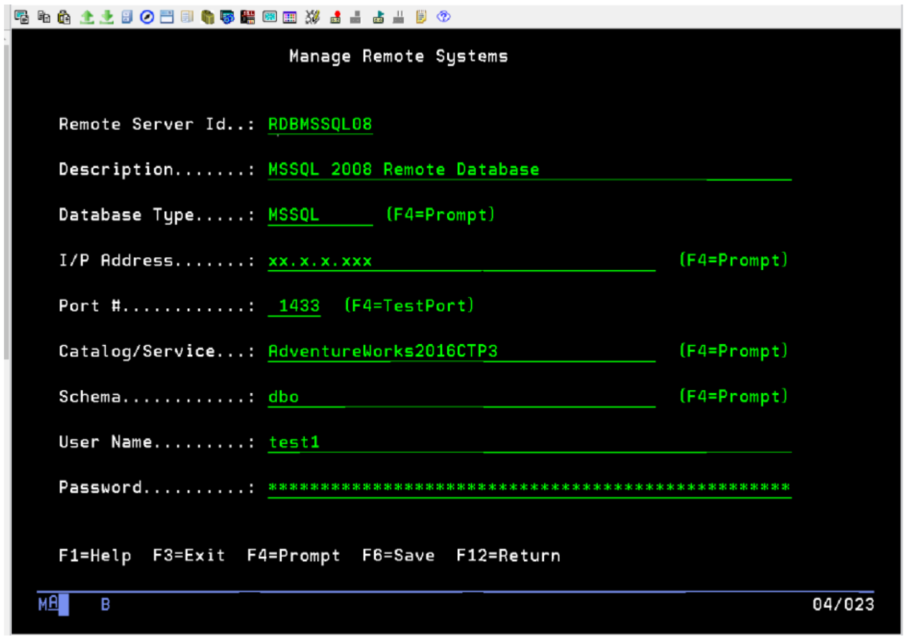
Schema – The schema for the remote database.

User Name – The user to be used when connecting. This user id will control the authority on the remote database when using this connection. This user name must be configured on the remote database. ** For Microsoft SQL Server using Windows (NTLM) authentication instead of the usual SQL Server authentication, the User name should be in the format of domain/user. This allows non-Windows clients to log in to servers which are only configured to accept Windows authentication.

Password – The password for the above entered user name.

To save the information, press **F6**. The information entered will be encrypted 256 bit and stored in a IBM i object. The object will be named the same as the **Remote Database ID** and be placed in the RDB library. IBM i object level authority can be used, with this object, to add another level of security to the remote connection.

The example below is for a SQL Server configuration.



```
                         Manage Remote Systems


   Remote Server Id..: RDBMSSQL08

   Description.......: MSSQL 2008 Remote Database

   Database Type.....: MSSQL       (F4=Prompt)

   I/P Address.......: xx.x.x.xxx                              (F4=Prompt)

   Port #............:  1433   (F4=TestPort)

   Catalog/Service...: AdventureWorks2016CTP3                  (F4=Prompt)

   Schema............: dbo                                     (F4=Prompt)

   User Name.........: test1

   Password..........: **************************************************


   F1=Help   F3=Exit   F4=Prompt   F6=Save   F12=Return

 MA      B                                                         04/023
```

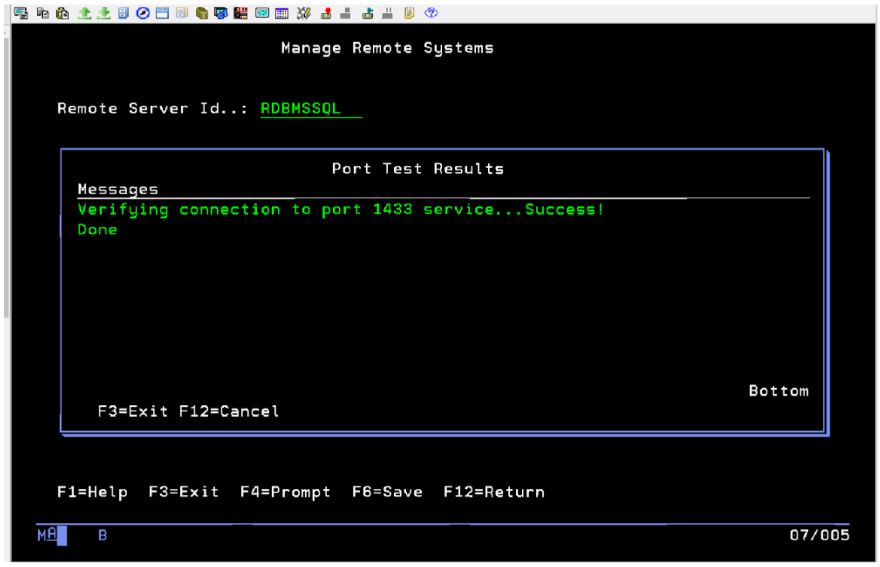
Figure 4-3

Cloud Services

Cloud databases are a pivotal component of modern data management, offering scalable and flexible storage solutions that leverage the power of cloud computing. These databases operate in virtualized environments hosted on cloud platforms like Amazon AWS, Google Cloud, and others. One notable tool that facilitates seamless connectivity to these cloud-based relational databases is RDBConnect.

RDBConnect serves as a versatile bridge, allowing users to establish connections to any relational database residing in the cloud. Whether it's harnessing the immense capabilities of Amazon's AWS infrastructure or tapping into Google Cloud's robust database offerings, RDBConnect excels in providing a unified interface, enabling users to interact with their data stored on the cloud effortlessly. Its compatibility with various cloud platforms makes it an invaluable asset for businesses seeking efficient and streamlined access to their cloud-based databases, thereby enhancing data retrieval and analysis capabilities.

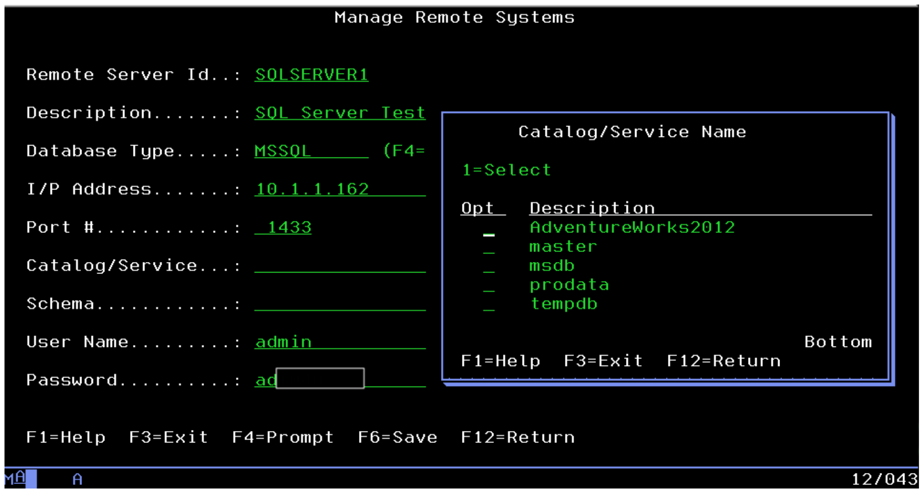
5
New prompt windows added to RDBCFG



```
                           Manage Remote Systems

   Remote Server Id..: RDBMSSQL

                                Port Test Results
       Messages
       Verifying connection to port 1433 service...Success!
       Done

                                                                 Bottom
         F3=Exit F12=Cancel

   F1=Help  F3=Exit  F4=Prompt  F6=Save  F12=Return
 MA    B                                                          07/005
```

Figure 5-1

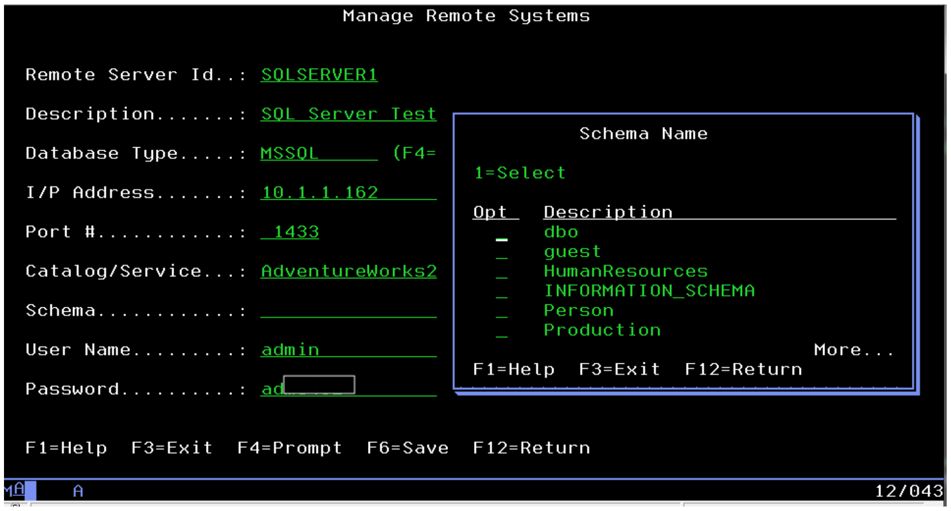
New Port Test in ADD Mode allows you to check if the port you're using to connect isn't blocked (Compatible with Java 1.7 and above)



```
                              Manage Remote Systems

  Remote Server Id..: SQLSERVER1
  Description.......: SQL Server Test    Catalog/Service Name
  Database Type.....: MSSQL      (F4=
  I/P Address.......: 10.1.1.162         1=Select
  Port #............:  1433              Opt  Description
                                          _   AdventureWorks2012
  Catalog/Service...:                     _   master
                                          _   msdb
  Schema............:                     _   prodata
                                          _   tempdb
  User Name.........: admin                                          Bottom
  Password..........: ad                 F1=Help  F3=Exit  F12=Return

  F1=Help  F3=Exit  F4=Prompt  F6=Save  F12=Return
MA    A                                                              12/043
```

Figure 5-2

New prompt field added to display all Catalog/Service Names. A User ID, Password, valid IP Address and Port must be provided for this option to work.



```
                              Manage Remote Systems

 Remote Server Id..: SQLSERVER1
 Description.......: SQL Server Test   Schema Name
 Database Type.....: MSSQL      (F4=
                                      1=Select
 I/P Address.......: 10.1.1.162
                                      Opt  Description
 Port #............:  1433              _   dbo
                                         _   guest
 Catalog/Service...: AdventureWorks2    _   HumanResources
                                         _   INFORMATION_SCHEMA
 Schema............:                     _   Person
                                         _   Production
 User Name.........: admin                                      More...
                                      F1=Help  F3=Exit  F12=Return
 Password..........: ad

 F1=Help  F3=Exit  F4=Prompt  F6=Save  F12=Return
MA    A                                                          12/043
```

Figure 5-3

New Schema prompt – Retrieve a list of schemas. A valid Catalog/Service name must be provided for this option to work.

NOTE: Currently the Catalog prompt will work on the following remote databases

MSSQL, MYSQL and POSTGRES

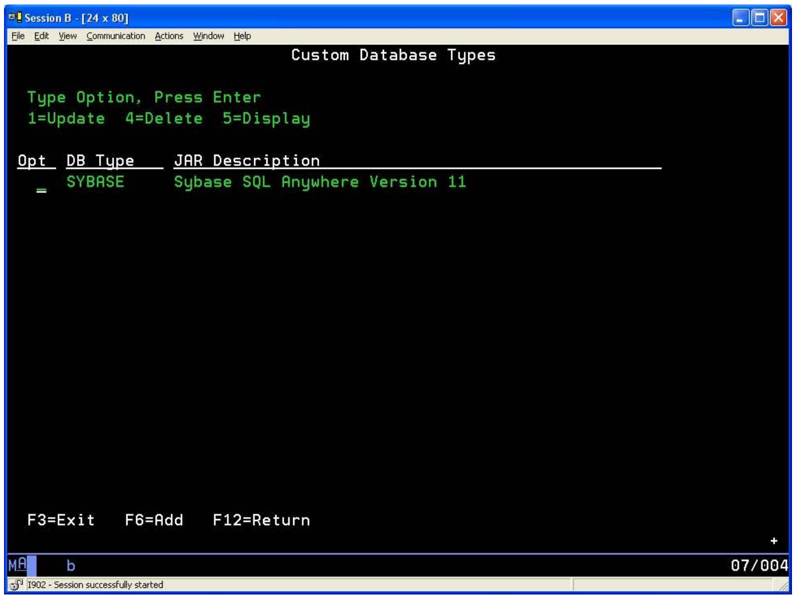
The Schema prompt will work on the following remote databases

MSSQL, ORACLE and POSTGRES

6

Configuring a RDB Custom Database

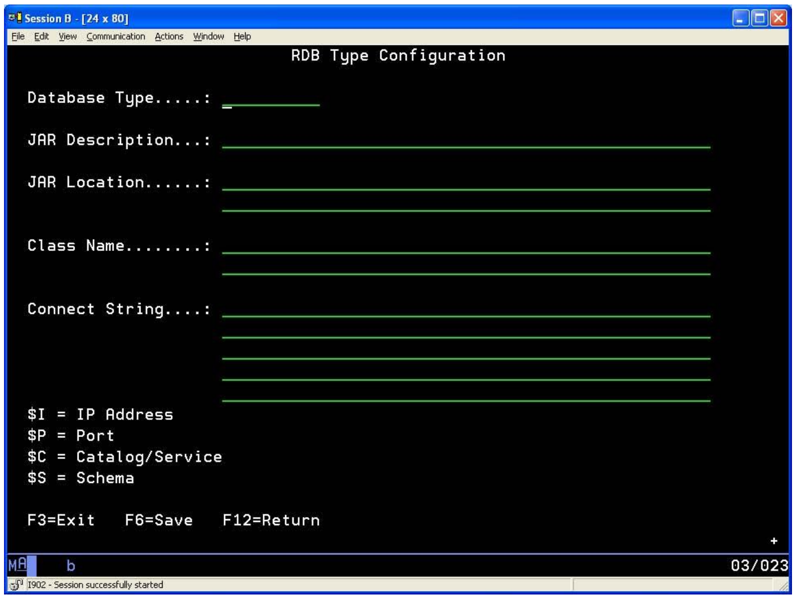
RDB Connect can be configured to access any database of your choosing. You must have a JDBC driver for the database and it must be in a directory on the IFS. To configure a custom database execute the command **RDBJARCFG**. This will take you to the **Custom Database Types** screen.



```
                         Custom Database Types

 Type Option, Press Enter
 1=Update  4=Delete  5=Display

Opt  DB Type    JAR Description
 _   SYBASE     Sybase SQL Anywhere Version 11

 F3=Exit   F6=Add   F12=Return
```

Figure 6-1

Press **F6** to add a custom database.



```
                         RDB Type Configuration

 Database Type.....: __________

 JAR Description...: __________________________________________________

 JAR Location......: __________________________________________________
                     __________________________________________________

 Class Name........: __________________________________________________
                     __________________________________________________

 Connect String....: __________________________________________________
                     __________________________________________________
                     __________________________________________________
                     __________________________________________________
                     __________________________________________________

 $I = IP Address
 $P = Port
 $C = Catalog/Service
 $S = Schema

 F3=Exit   F6=Save   F12=Return
```

Figure 6-2

Enter the information for the custom database.

Database Type – This is the name that will be used when referencing this custom database type. It will appear in the prompt list when configuring a connection.

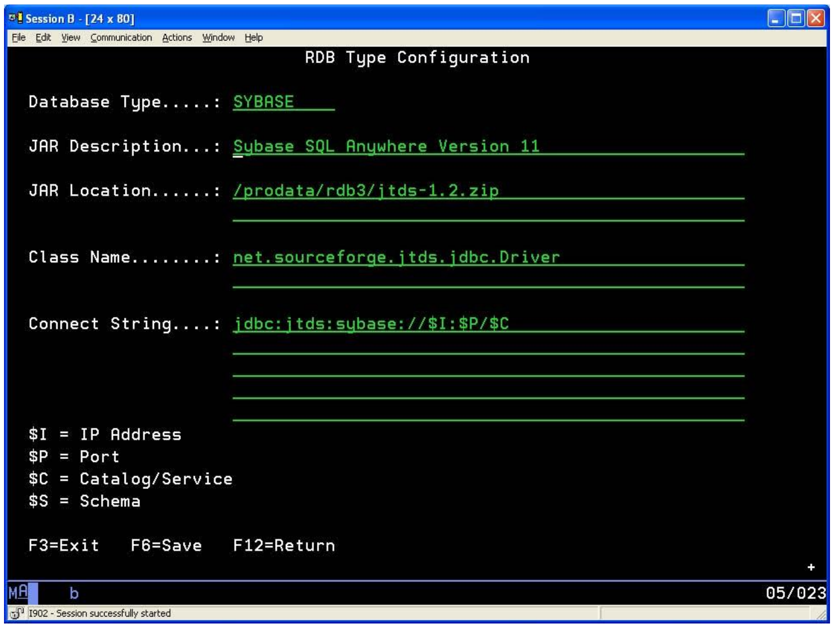
JAR Description – The description for this custom connection.

JAR Location – The path on the IFS for the jar file containing the JDBC driver of this database.

Class Name – The JDBC drive class name. This must be the complete jar path

Connect String – The string used to connect to the custom database. This string must contain the $I substitution variable and optionally the $P, $C, and $S.
$I – The IP address of the remote database will be placed at this point in the
connection string.
$P – The port number of the remote database will be placed at this point in the connection string.
$C – The Catalog/Service of the remote database will be placed at this point in the connection string.
$S – The Schema of the remote database will be placed at this point in the connection string.

The example below is for a Sybase configuration.



```
Session B - [24 x 80]
                              RDB Type Configuration

  Database Type.....: SYBASE

  JAR Description...: Sybase SQL Anywhere Version 11

  JAR Location......: /prodata/rdb3/jtds-1.2.zip


  Class Name........: net.sourceforge.jtds.jdbc.Driver


  Connect String....: jdbc:jtds:sybase://$I:$P/$C





  $I = IP Address
  $P = Port
  $C = Catalog/Service
  $S = Schema

  F3=Exit   F6=Save   F12=Return
                                                                              +
MA     b                                                                 05/023
I902 - Session successfully started
```

Figure 6-3

7
Configuring a PC Database

RDB Connect can be configured to access a database on a PC that does not have a JDBC driver. The database must have an available ODBC driver. This function requires two things. The RDB service running on the PC and a DSN defined for that database.

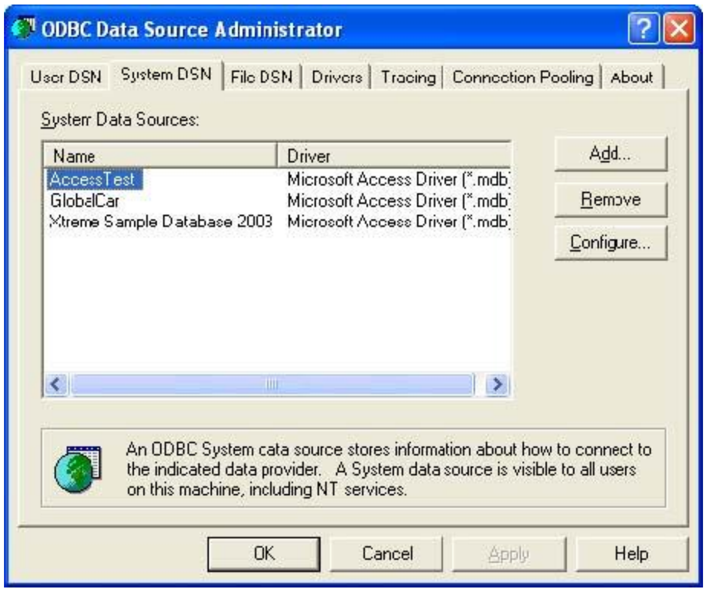
When installing the RDB software, there is a checkbox for the "PC Component". Select this option to install the PC software on the appropriate computer. This will create a directory "Program Files/Prodata/RDB5" and place the required programs in that directory. Two batch files will also be placed in the directory. The file "RDBServiceStart.bat" must be modified. The first line in the file is the configuration command for the service. It contains the path to the JVM on your PC. This path must be correct. If it is not, please change it. Example:

```
RDBService -i -J "C:\\ProData\\RDB50\\Java\\jre1.6.0_05\\bin\\client\\jvm.dll" -P 9082 -L 100 –T
```

(Installer includes a directory that is placed in C:\ProData\RDB50\Java)
The portion that needs to be modified is following the "-J" parameter. This is the path to the jvm.dll object. The "-P" parameter controls the port that will be used for communication to the PC. The default is 9082. The "-L" parameter controls the number of listeners that will process requests on this PC. The default is 100. The "-T" parameter controls the use of the trace log function. This is useful to troubleshoot issues. The "-i" parameter causes the service to be installed.

After the batch file has been changed to contain the correct values, it can be executed to start the service. To end the service, execute the "RDBServiceStop.bat" file.

To configure a connection the PC database you must setup a DSN on the PC. This is done via the Control Panel->Administrative Tools->Data Sources (ODBC).



Name	Driver
AccessTest	Microsoft Access Driver (*.mdb
GlobalCar	Microsoft Access Driver (*.mdb
Xtreme Sample Database 2003	Microsoft Access Driver (*.mdb

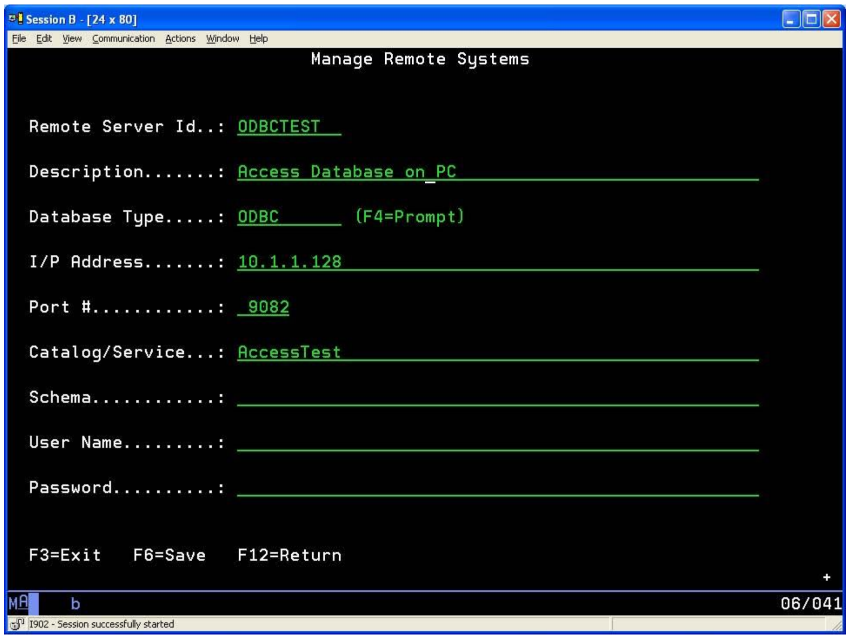
An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Figure 7-1

You can also open Starts > Type ODBC and click on the 32bit or 64bit version

The DSN needs to be defined as a “System DSN”. This will allow the RDB Service to access the definition.

On the IBM i, use the **RDBCFG** to configure the connection to the ODBC database. The screen below shows an example ODBC configuration.



```
Session B - [24 x 80]
File  Edit  View  Communication  Actions  Window  Help
                             Manage Remote Systems


 Remote Server Id..: ODBCTEST

 Description.......: Access Database on PC

 Database Type.....: ODBC       (F4=Prompt)

 I/P Address.......: 10.1.1.128

 Port #............:  9082

 Catalog/Service...: AccessTest

 Schema............:

 User Name.........:

 Password..........:


 F3=Exit   F6=Save   F12=Return
                                                                            +
MA     b                                                               06/041
I902 - Session successfully started
```

Figure 7-2

The “System DSN” that was defined on the PC is placed in the “Catalog/Service” field.

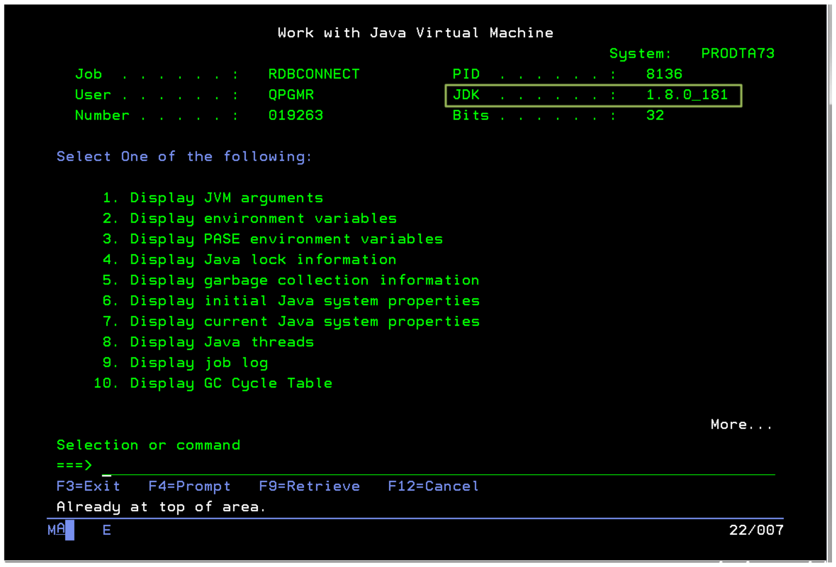
Configuring RDBConnect to run under a specific Java version

RDBConnect can run under a different JAVA version if needed. You must add the IFS path where java JDK is located.
Example: /QOpenSys/QIBM/ProdData/JavaVM/jdk70/32bit (case sensitive)

You can switch between 32 bit and 64 bit.

Update DTAARA RDBJVMPATH with the JDK path and restart RDBConnect.

Use command WKRJVMJOB to check what version of java RDConnect is using.



```
                         Work with Java Virtual Machine
                                                          System:   PRODTA73
  Job . . . . . . . :   RDBCONNECT          PID . . . . . . :   8136
  User . . . . . . . :   QPGMR               JDK . . . . . . :   1.8.0_181
  Number . . . . . . :   019263              Bits . . . . . . :   32

Select One of the following:

     1. Display JVM arguments
     2. Display environment variables
     3. Display PASE environment variables
     4. Display Java lock information
     5. Display garbage collection information
     6. Display initial Java system properties
     7. Display current Java system properties
     8. Display Java threads
     9. Display job log
    10. Display GC Cycle Table

                                                                       More...
Selection or command
===> _

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
Already at top of area.
MA     E                                                                22/007
```

Figure 7-3

8
The Commands

RDBPTF (RDB PTF Processor)

The RDB PTF Processor (RDBPTF) command allows you to retrieve updates for RDB Connect. The process makes a connection to Prodata Computer Services using port **2809**. If problems occur during the running of this command, verify your firewall is not blocking this transaction. This function will only retrieve the programs that have been updated since your last update.

Parameters

Keyword	**Description**	**Default**	**Notes**
LIB	The library to receive the updated programs	RDB50	This must be the library that currently contains RDB Connect

RDBSEC (RDB Security Code)

The RDB Security Code (RDBSEC) command provides an interface to enter the permanent and temporary access codes for RDB Connect.

Parameters

N/A

RDBTRACE (Set RDB Tracing)

The Set RDB Tracing (RDBTRACE) command turns the logging process on and off in the RDB CONNECT job. This is the same function as the "Turn Tracing On" in the RDBCFG screen. The flag on the RDBCFG screen is used at startup of the RDB CONNECT job. This command can be used anytime the RDB CONNECT job is running.

Parameters

Keyword	Description	Default	Notes
TRACE	Turn tracing on or off.	*ON	Valid values are *ON or *OFF
Log File Size	Set Max log file size	Number	Number of MegaBytes
Send Msg	Send Warning Msg	N	Valid values are 'Y' or 'N'
User Profile	Warning msg is sent here	Character	Valid profile ID

Example

The following is a RPG Example to turn on Tracing in RPG

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
//Definition
D nTrace       s          n   inz('1')
D vAE          s          1A inz
   vAE = 'A';    //A will retrieve all logs – E will only retrieve errors
   rc =  RDBTRACE(nTrace: vAE);
```

RDBFIELDS (Retrieve field information)

The Retrieve field information (RDBFIELDS) command provides the field definitions from the remote database. The definitions are based on the select statement provided.

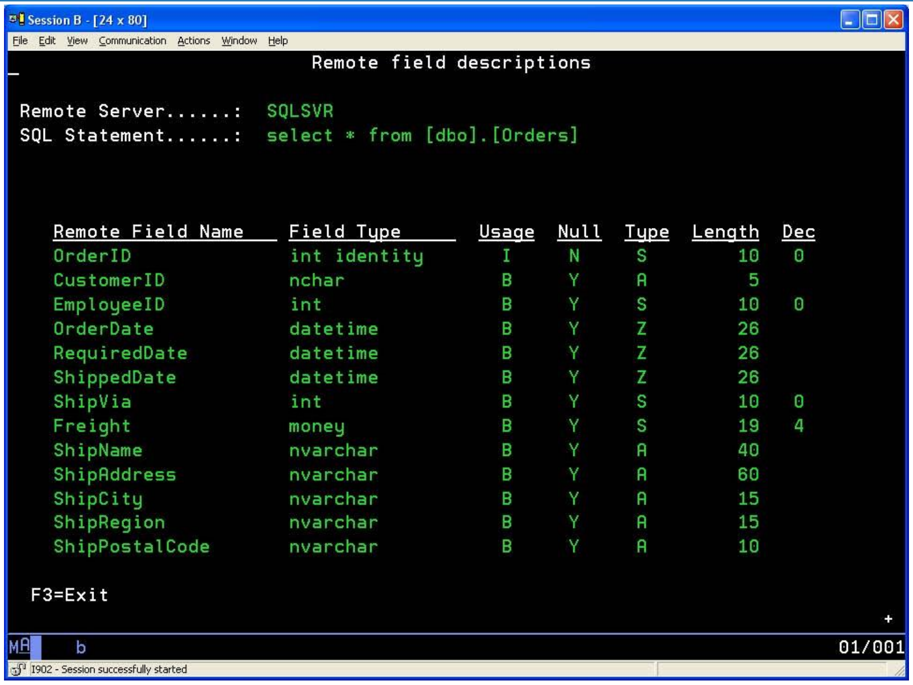
Parameters

Keyword	Description	Default	Notes
SERVER	The remote server id that was created using **RDBCFG**.		This must be a valid connection. You need to be authorized to use the connection object.
USER	The user id used in the connection process	*CONFIG	The default will get the user from the RDBCFG of the selected server.
PASSWORD	The password used in the connection process	*CONFIG	The default will get the password from the RDBCFG of the selected server.
STM	The SQL statement that will be used to retrieve the field listing.		Only SELECT statements are allowed.

Example

The following command might generate a screen similar to the one below.

```
RDBFIELDS STM('select * from [dbo].[Orders]') SERVER(SQLSVR)
```



```
                         Remote field descriptions

 Remote Server......:  SQLSVR
 SQL Statement......:  select * from [dbo].[Orders]



    Remote Field Name     Field Type        Usage  Null  Type  Length  Dec
    OrderID               int identity        I     N     S       10    0
    CustomerID            nchar               B     Y     A        5
    EmployeeID            int                 B     Y     S       10    0
    OrderDate             datetime            B     Y     Z       26
    RequiredDate          datetime            B     Y     Z       26
    ShippedDate           datetime            B     Y     Z       26
    ShipVia               int                 B     Y     S       10    0
    Freight               money               B     Y     S       19    4
    ShipName              nvarchar            B     Y     A       40
    ShipAddress           nvarchar            B     Y     A       60
    ShipCity              nvarchar            B     Y     A       15
    ShipRegion            nvarchar            B     Y     A       15
    ShipPostalCode        nvarchar            B     Y     A       10

  F3=Exit
                                                                         +
MA    b                                                             01/001
I902 - Session successfully started
```

Figure 8-1

RDBRUNSQL (Run RDB SQL Statement)

The Run RDB SQL Statement (RDBRUNSQL) command provides an interface to execute SQL Statements on the remote database. A **SELECT** will run, however it will not generate any output. All other SQL Statements will be executed on the remote database (a warning message will appear before continuing).

Parameters

Keyword	Description	Default	Notes
SERVER	The remote server id that was created using **RDBCFG**.		This must be a valid connection. You need to be authorized to use the connection object.
USER	The user id used in the connection process	*CONFIG	The default will get the user from the RDBCFG of the selected server.
PASSWORD	The password used in the connection process	*CONFIG	The default will get the password from the RDBCFG of the selected server.
STM	The SQL statement to process.		All statements except SELECT are allowed.

Example

RDBRUNSQL STM('drop [dbo].[Orders]') SERVER(SQLSVR)

RDBIMPORT (Import Remote Database)

The Import Remote Database (RDBIMPORT) command provides an interface to execute commands on the remote database and return the results to a local database file. The statement must be a **SELECT** statement.

Parameters

Keyword	Description	Default	Notes
CRTADD	Create the local file	Y	If 'Y' is specified the local file cannot exist. If 'N' is specified, the local file must exist.
OUTFILE	The name of the local IBM I file.		The file that will contain the results of the select.
LIB	The name of the library where the file will be created.	*CURRENT	This must be specified if *CURRENT library is not set.
SERVER	The remote server id that was created using **RDBCFG**.		This must be a valid connection that you are authorized.
USER	The user id used in the connection process	*CONFIG	The default will get the user from the RDBCFG of the selected server.
PASSWORD	The password used in the connection process	*CONFIG	The default will get the password from the RDBCFG of the selected server.
STM	The statement to execute against remote database		A SELECT statement to retrieve rows from remote server.

Example

```
RDBIMPORT    STM('select    *    from    [dbo].[Orders]')    SERVER(SQLSVR)
CRTADD(*YES) OUTFILE(MYLIB/ORDERS)
```

The above statement will create a file called ORDERS in the library MYLIB and write the selected records from the remote database to the file.
NOTE: when using MSSQL2 connection the max allowed combined characters for import to IBMi is 32765.

RDBCFG (Remote Database Configuration)

The Remote Database Configuration (RDBCFG) command provides an interface to configure the remote database connections. The definitions are created as objects in the RDB50 library. IBM i security can be applied to these objects to better secure your remote connections.

For usage of this command, see "Configuring RDB Connect".

Parameters N/A

RDBJARCFG (Custom Database Configuration)

The Custom Database Configuration (RDBJARCFG) command provides an interface to configure any database that has a JDBC driver.

For usage of this command, see “Configuring a RDB Custom Database”.

Parameters
N/A

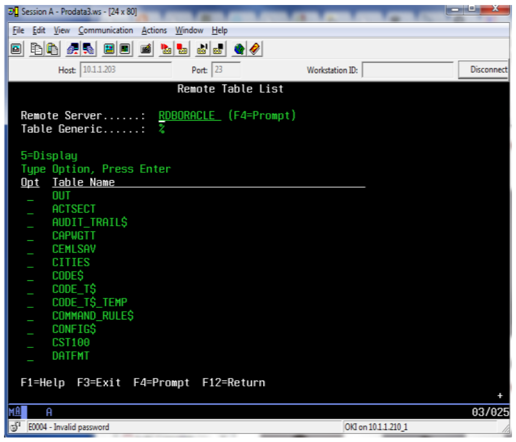
RDBTABLES (Retrieve table list)

The Retrieve table list (RDBTABLES) command provides a list of available tables on the remote server. The list is based on the table parameter.

Parameters

Keyword	Description	Default	Notes
TABLE	The subset of tables to be listed.	*ALL	The % symbol is used as a wildcard in this parameter. Example: For a list of all tables beginning with “f”, the parameter would be specified as ‘f%’.
SERVER	The remote server id that was created using **RDBCFG**.		This must be a valid connection. You need to be authorized to use the connection object.
USER	The user id used in the connection process	*CONFIG	The default will get the user from the RDBCFG of the selected server.
PASSWORD	The password used in the connection process	*CONFIG	The default will get the password from the RDBCFG of the selected server.

Option 5 was added to display fields for each table (RDBFIELDS).



Session A - Prodata3.ws - [24 x 80]
File Edit View Communication Actions Window Help
Host: 10.1.1.203
Port: 23
Workstation ID:
Disconnect

```
                              Remote Table List

 Remote Server......:  RDBORACLE   (F4=Prompt)
 Table Generic......:  %

 5=Display
 Type Option, Press Enter
 Opt  Table Name
  _   OUT
  _   ACTSECT
  _   AUDIT_TRAIL$
  _   CAPWGTT
  _   CEMLSAV
  _   CITIES
  _   CODE$
  _   CODE_T$
  _   CODE_T$_TEMP
  _   COMMAND_RULE$
  _   CONFIG$
  _   CST100
  _   DATFMT

 F1=Help  F3=Exit  F4=Prompt  F12=Return
                                                                              +
MA    A                                                                  03/025
E0004 - Invalid password                         OKI on 10.1.1.210_1
```

Figure 8-2

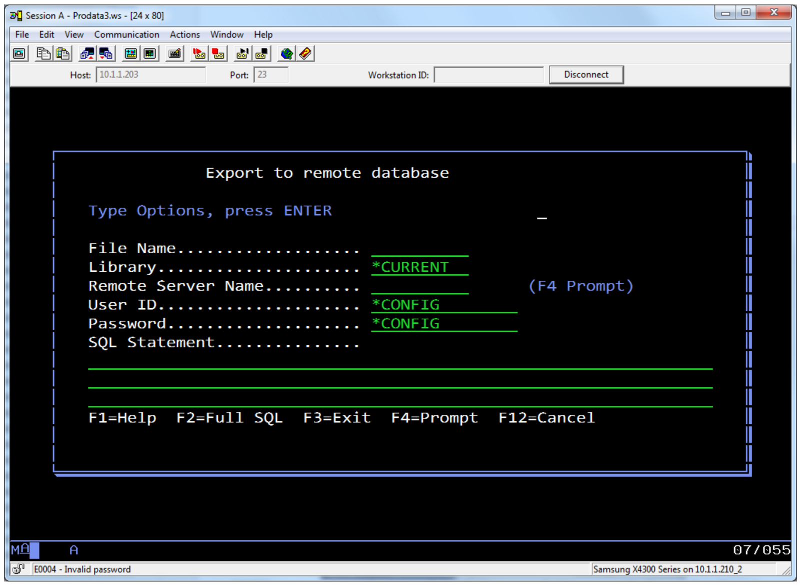
RDBEXPORT (Export to Remote Database)

The Export to Remote Database (RDBEXPORT) command provides an interface to export a local file from IBM i to the remote database. This command is not currently designed to add records to an existing table. It will only create a new table with records the first time.

Parameters

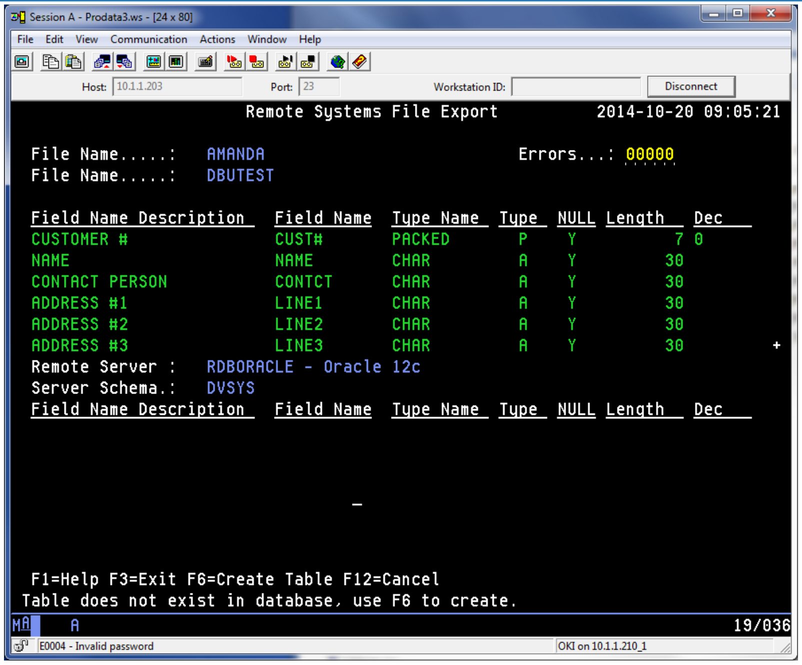
Keyword	Description	Default	Notes
EXPORT FILE	Export File Name in iSeries		Specify a valid File Name
LIBRARY	LIBRAY NAME where object resides		File and Library must be correct in order to run.
SERVER	Remote Server Name specified in **RDBCG** command.		The default will get the user from the RDBCFG of the selected server.
USER	The user id used in the connection process	*CONFIG	The default will get the user from the RDBCFG of the selected server.
PASSWORD	The password used in the connection process	*CONFIG	The default will get the password from the RDBCFG of the selected server.
REMOTE DB SQL STATEMENT	Use an SQL Select statement for all fields or individual fields.		Must specify a schema where the table resides. Select statement can include WHERE, ORDER BY and HAVING.

Example


Session A - Prodata3.ws - [24 x 80]
File Edit View Communication Actions Window Help
Host: 10.1.1.203
Port: 23
Workstation ID:
Disconnect

```
                 Export to remote database

    Type Options, press ENTER

    File Name...................  __________
    Library.....................  *CURRENT__
    Remote Server Name..........  __________      (F4 Prompt)
    User ID.....................  *CONFIG_________
    Password....................  *CONFIG_________
    SQL Statement...............
    ________________________________________________________________
    ________________________________________________________________
    ________________________________________________________________
    F1=Help  F2=Full SQL  F3=Exit  F4=Prompt  F12=Cancel

MA     A                                                      07/055
E0004 - Invalid password                Samsung X4300 Series on 10.1.1.210_2
```

Figure 8-3



```
                         Remote Systems File Export         2014-10-20 09:05:21

  File Name.....:   AMANDA                             Errors...: 00000
  File Name.....:   DBUTEST

  Field Name Description   Field Name   Type Name  Type  NULL Length   Dec
  CUSTOMER #               CUST#        PACKED       P    Y        7  0
  NAME                     NAME         CHAR         A    Y       30
  CONTACT PERSON           CONTCT       CHAR         A    Y       30
  ADDRESS #1               LINE1        CHAR         A    Y       30
  ADDRESS #2               LINE2        CHAR         A    Y       30
  ADDRESS #3               LINE3        CHAR         A    Y       30         +
  Remote Server :   RDBORACLE - Oracle 12c
  Server Schema.:   DVSYS
  Field Name Description   Field Name   Type Name  Type  NULL Length   Dec







  F1=Help F3=Exit F6=Create Table F12=Cancel
 Table does not exist in database, use F6 to create.
MA    A                                                                  19/036
```

Figure 8-4

RDBExport screen – Contains all the field names, field types, null allowed, length and decimal.
The top part of the screen pertains to the IBMi Database and the bottom part pertains to the remote database if a connection is established.

The errors counter is a warning to identify when the field types and lengths do not match.

RDBTABVIEW (Table Views)

The Table Views abase (RDBEXPORT) command provides an interface to execute commands on the remote database and return the results.

Keyword	Description	Default	Notes
TABLE VIEW	Table View Name	*ALL	Specify a table view, if *ALL then all views available will be retrieved.
SERVER	Remote Server Name specified in **RDBCG** command.		The default will get the user from the RDBCFG of the selected server.
USER	The user id used in the connection process	*CONFIG	The default will get the user from the RDBCFG of the selected server.
PASSWORD	The password used in the connection process	*CONFIG	The default will get the password from the RDBCFG of the selected server.

RDBSTPROC (Stored Procedures)

The Stored Procedure (RDBSTPROC) command provides an interface to execute commands on the remote database and return the results.

Keyword	Description	Default	Notes
STORED PROCEDURE NAME	Stored Procedure Name	*ALL	Specify a stored procedure, if *ALL then all procedures available will be retrieved.
SERVER	Remote Server Name specified in **RDBCG** command.		The default will get the user from the RDBCFG of the selected server.
USER	The user id used in the connection process	*CONFIG	The default will get the user from the RDBCFG of the selected server.
PASSWORD	The password used in the connection process	*CONFIG	The default will get the password from the RDBCFG of the selected server.

RDBPROCDTL (Stored Procedures Details)

The Stored Procedure (RDBPROCDTL) command provides an interface to execute commands on the remote database and return the results.

Keyword	Description	Default	Notes
SERVER	Remote Server Name specified in **RDBCG** command.		The default will get the user from the RDBCFG of the selected server.
STORED PROCEDURE NAME	Stored Procedure Name		Specify a stored procedure.
USER	The user id used in the connection process	*CONFIG	The default will get the user from the RDBCFG of the selected server.
PASSWORD	The password used in the connection process	*CONFIG	The default will get the password from the RDBCFG of the selected server.

RDBSEC (RDBConnect Security Details)

The Security (RDBSEC) command provides your product license information details.

RDBPTFIFS (RDB PTF Processor for IFS objects)

The RDB PTF Processor (RDBPTFIFS) command allows you to retrieve updates for RDB Connect IFS objects. The process makes a connection to ProData Computer Services using port **2809**. If problems occur during the running of this command, verify your firewall is not blocking this transaction. This function will only retrieve the programs that have been updated since your last update.

Parameters

Keyword	Description	Default	Notes
LIB	The library to receive the updated programs	RDB50	This must be the library that currently contains RDB Connect

RDBDRIVERS (RDB JDBC Drivers)

The (RDBDRIVERS) command allows you to update JDBC drivers used when RDB Connect job starts. You can change/remove any JDBC driver from an interactive screen.

Parameters

Keyword	Description	Default	Notes

RDBADDPC (Add PC Server Job to RDBConnect SubSystem Description)

The (RDBADDPC) command adds an entry to the RDBConnect SubSystem description.

Parameters

Keyword	Description	Default	Notes

RDBRMVPC (Remove PC Server Job from RDBConnect SubSystem Description)

The (RDBRMVPC) command removes entry from the RDBConnect SubSystem description if it has been added previously.

Parameters

Keyword	Description	Default	Notes

RDBLIBCSV (Convert Entire Library Files/Members to CSV or XLS Format)

The (RDBLIBCSV) command converts an entire library's Files and Members to CSV or XLS format. All files are sent to IFS '/prodata' directory

Parameters

Keyword	Description	Default	Notes
LIBRARY	Library Name		
EXTENSION	File Extension		Valid options CSV & XLS
CCSID	IBM Character Set	0037	

RDBFILECSV (Convert Single File/Member to CSV or XLS Format)

The (RDBFILECSV) command converts a single Files and Members to CSV or XLS format. All files are sent to IFS '/prodata' directory

Parameters

Keyword	Description	Default	Notes
FILELIB	File Name/Library Name		
EXTENSION	File Extension		Valid options CSV & XLS
CCSID	IBM Character Set	0037	

RDBEXPLIB (Converts All PF and Members to XLS or CSV and exports all to remote Database)

The (RDBEXPLIB) command exports an entire library Files and Members to a remote Database.

Parameters

Keyword	Description	Default	Notes
LIBRARY	Library Name		
ID	Remote Connection ID		

RDBEXPFIL (Convert Single File/Member to XLS or CSV and transfer to remote Database)

The (RDBEXPFIL) command converts to XLS-CSV and exports a single File and Members to a remote Database.

Parameters

Keyword	Description	Default	Notes
FILELIB	File Name/Library Name		
ID	Remote Connection ID		

RDBDRVDOWN (Download/Import JDBC Drivers)

The (RDBDRVDOWN) Download/Import JDBC Driver Files to IFS automatically.

Parameters

Keyword	Description	Default	Notes
URL	File URL Path		
ZIPNAME	Zip File Name		

RDBPFXLSX (Convert Single File/Member to XLSX)

The (RDBPFXLSX) command converts a single File and Members to a XLSX file and stores it in IFS directory. A new directory is created with the same name as the library name.

Parameters

Keyword	Description	Default	Notes
FILENAM	File Name for XLSX		You must specify .xlsx
FILELIB	File/Library to be converted		LIBRARY/FILE
ALLMBR	Include all Members	N	Valid values Y/N
GENCSV	Generate CSV File	N	Valid values Y/N
SQLSTM	Specify a SQL Statement		SELECT * FROM SCHEMA.TABLE

RDBCRTABLE (Create IBMi Empty Table on Remote Database)

The (RDBCRTABLE) command creates a copy of your IBM i File/Table on a remote database with the option to use SQL Statement to specify a SELECT statement.

Parameters

Keyword	Description	Default	Notes
SELECT	SQL SELECT Statement		SQL Statement required
OUTFILE	IBMi File Name		IBMi File Name required
	Library Name	*CURRENT	Library Name required
SERVER	Connection ID	*PROMPT	

RDBENCRYPT (Encrypt/Decrypt using AES)

The (RDBENCRYPT) program (not a command) can be used to Encrypt/Decrypt any string using Advanced Encryption Standard.

Parameters

Keyword	Description	Default	Notes
Mode	E=Encrypt D=Decrypt	1A	
Text	Text to Encrypt/Decrypt	65535A Varying	Original Text
Key	Secured Key used to Encrypt/Decrypt	256A	Key use to Encrypt/Decrypt
Return	Return value	65535A Varying	Return value

RDBDROPBOX (DropBox Cloud Data Transfer)

The (RDBDropBox) command can be used to transfer data seamlessly between IBM i and DropBox cloud platform, ensuring smooth and reliable exchange of information.

Parameters

Keyword	**Description**	**Default**	**Notes**
TOKEN	Command for Token	5A	Default to TOKEN
CMD	Command	6A	SubCommand to Execute
DBPATH	DropBox Path	256A	DropBox Path to file/s
IBMIPATH	IBM I Path	256A	IBMi Path

RDBCRTSQLF (Create .SQL File from Physical File)

The (RDBCRTSQLF) command can be used create .SQL File with scripts to create the table and insert data on a remote database.

Parameters

Keyword	**Description**	**Default**	**Notes**
CRTTABLE	Create Table Script	1A	Y=Yes N=NO
FILENAME	IBM i PF Name	10A	Physical File Name
DBTYPE	Database Type	10A	Specify Database where script will run
CATALOG	Remote DB Catalog Name	256A	Catalog Name to remote database
SCHEMA	Remote DB Schema Name	256A	Schema Name to remote database

9 The Functions

RDB Connect provides you with a service program to access your remote databases. The service program is called RDB2000 in the library RDB50. An example program and the prototypes for the supplied functions can be found in RDB50/RDBSRC. A binding directory called RDB2000 is supplied with RDB Connect to assist in the compiling of your programs. It can be found in the library RDB50.

RDBConnect (Connect to the remote server)

Purpose

RDB Connect sends the connection information to the remote server and returns an ID to be used in future transactions for this database. The returned ID is valid until it is closed.

Syntax

ID = RDBConnect(RemoteId: {user}: {password}:{port})

Function Arguments

Data Type	**Argument**	**Use**	**Description**
Char(10)	Remote ID	Input	The RemoteID parameter is the name that was used when creating the configuration of the remote server. See "Configuring RDB Connect".
Char(20)	User	Input(Optional)	The user parameter is used during the connection process to validate the connection to the database. If it is not specified, the user from the configuration will be used.
Char(20)	Password	Input(Optional)	The password parameter is used during the connection process to validate the connection to the database. If it is not specified, the password from the configuration will be used.
Signed(4)	Port	Input(Optional)	The port number the RDB server is listening on. This is only used when multiple RDB servers are being ran at the same time. Omitting this parameter causes the connect process to use the port number from the RDBCFG screen.
Int(10)	ID	Output	An ID is returned - will be used throughout the process to maintain the connection. To connect to a database multiple times or to multiple databases, use multiple IDs. A non-negative number signifies a valid connection.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
 *
 * Connect to the remote system using the Id created in the RDB
 * configuration screen (RDBCFG)
C                   Eval      Id = RDBConnect('SQLSVR')
```

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
 *
 * Connect to the remote system using the Id created in the RDB
 * configuration screen (RDBCFG) with a user and password
C                  Eval        Id = RDBConnect('SQLSVR': :'me':'mypass')
```

RDBClose (Close any open connection)

Purpose

RDBClose closes any open connection.

Syntax

RDBClose(Id)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	Connection ID	Input	The Connection ID parameter previously given when RDBConnect was used.

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
  *
  * Close the connection.
  C              Callp            RdbClose(Id)
```

RDBExec (Execute a SQL statement on the remote server)

Purpose

RDBExec directly executes the specified SQL statement on the remote server. Any valid SQL statement can be executed. The syntax for the statement must be valid on the remote server.

RDBConnect() must be called before calling this function.

If a previous statement has been executed for this connection, RDBFreeStmt() must be called to close the cursor, before calling RDBExec().

Syntax

rc = RDBExec(ID: Statement: Update)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Char(32767) Varying	Statement	Input	The statement to be processed on the remote server.
Boolean	Update	Input (Optional)	Is the statement updatable. Valid values: *OFF – Statement is read only *ON – Statement is updatable (default)
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
             CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
     *
     *   After the connecton is established, prepare a select statement.
C                       Eval       rc = RdbExec (Id:
C                                  'Select * from [dbo].[Orders]':*OFF)
```

RDBPrepStmt (Create a prepared SQL statement on the remote server)

Purpose

RDBPrepStmt sends the SQL statement to the remote server to be prepared.. Any valid SQL statement can be prepared. The syntax for the statement must be valid on the remote server.
The SQL statement string may contain parameter markers. A parameter marker is represented by a “?” character, and indicates a position in the statement where the value of an application variable is to be substituted, when RDBPrepExec() is called. RDBSetStr(), RDBSetDate(), RDBSetNull(), and RDBSetNum() are used to associate a application variable or constant value to each parameter marker.

RDBConnect() must be called before calling this function.

If a previous statement has been executed for this connection, RDBFreeStmt() must be called to close the cursor, before calling RDBPrepStmt ().

Syntax

rc = RDBPrepStmt (ID: Statement: Update)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Char(32767) Varying	Statement	Input	The statement to be prepared on the remote server.
Boolean	Update	Input (Optional)	Is the statement updatable. Valid values: *OFF – Statement is read only *ON – Statement is updatable (default)
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
     *
     * After the connection is established, prepare a select statement.
       C          Eval rc = RdbPrepStmt(Id:
       C               'Select * from [dbo].[Orders]' +
       C               'where ShippedDate = ?':*OFF)
```

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
   *
   * After the connection is established, prepare a select statement.
    C                 Eval rc = RdbPrepStmt(Id:
    C                      'Select * from [dbo].[Orders]' +
    C                      'where ShippedDate = ?':*OFF)
```

RDBPrepExec (Execute a previously prepared SQL statement on the remote server)

Purpose

RDBPrepExec executes a statement, that was successfully prepared using RDBPrepStmt(), once or multiple times. The statement is executed using the current values of any application variables that were bound to parameter markers by RDBSetStr(), RDBSetStr(), and RDBSetStr().

Syntax

rc = RDBPrepExec(ID)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
    *
    * After the statement is prepared, execute the prepared statement.
    C              Eval rc = RdbPrepExec(Id)
```

RDBFreeStmt (Free the previously executed statement)

Purpose

RDBFreeStmt ends processing on the previously executed statement. The connection to the remote system will remain open.

Syntax

RDBFreeStmt(ID)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
  *
  * Free the statement that was last executed.
  C             Callp            RdbFreeStmt(Id)
```

RDBError (Returns the errors that occurred)

Purpose

RDBError returns the error code and error text that were generated by the last executed RDB function. The output parameters will only be generated when a negative one (-1) is returned from a function.

Syntax

RDBError(Error:ErrorText)

Function Arguments

Data Type	Argument	Use	Description
Char(7)	Error	Output	The error code that was generated by the previously executed function. Error codes can be found in the RDBMSGF message file.
Char(100)	ErrorText	Output	The additional message information for the error

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
 *
 * Retrieve the error that was generated.
C         Callp                RdbError(Error:ErrorText)
```

RDBFetchNxt (Fetch the next available record)

Purpose

RDBFetchNxt moves the statement cursor on the remote database SELECT to the next available record. The function is only valid when a RDBExec has been used for a SELECT statement. If the fetch fails, a negative one (-1) will be returned from the function. A zero will be returned upon successful completion.

Syntax

rc = RDBFetchNxt(ID)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
* Fetch the next record of the result set that was generated with the
* previously executed statement.
C                      Eval    rc = RdbFetchNxt(Id)
```

RDBFetchPrv (Fetch the previous record)

Purpose

RDBFetchPrv moves the statement cursor on the remote database SELECT to the previously available record. The function is only valid when a RDBExec has been used for a SELECT statement. If the fetch fails, a negative one (-1) will be returned from the function. A zero will be returned upon successful completion.

Syntax

rc = RDBFetchPrv(ID)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
   * Fetch the previous record of the result set that was generated with
   * the previously executed statement.
   C           Eval rc = RdbFetchPrv(Id)
```

RDBFetchAbs (Fetch the absolute record)

Purpose

RDBFetchAbs moves the statement cursor on the remote database SELECT to the record requested. The function is only valid when a RDBExec has been used for a SELECT statement. If the fetch fails, a negative one (-1) will be returned from the function. A zero will be returned upon successful completion.

Syntax

rc = RDBFetchAbs(ID: RecNum)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	RecNum	Input	The record number in the result set that was generated by a previously executed SELECT.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Fetch a record of the result set using the absolute position of the
* result set. This will return the 4th record of the result set.
C                   Eval      rc = RdbFetchAbs(Id:4)
```

RDBGetNum (Get a numeric field from a record)

Purpose

RDBGetNum retrieves the data from a numeric field in the record of the remote database. The function is only valid when RDBExec has been used for a SELECT statement and a fetch has been used.

Syntax

number = RDBGetNum(ID: FieldNum)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	The field number in the result set that was generated by a previously executed SELECT.
Number(30,9)	number	Output	The value of the field in the corresponding result set is returned. Zero is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Fetch the first field of the result set and return it as a number.
C              Eval      Field1 = RdbGetNum(Id:1)
```

RDBAscNum (Get a numeric field from a record using field name)

Purpose

RDBAscNum retrieves the data from a numeric field in the record of the remote database using the associated field name. The function is only valid when RDBExec has been used for a SELECT statement and a fetch has been used.

Syntax

number = RDBAscNum(ID: FieldName)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Char(50)	FieldName	Input	The name of the field in the result set that was generated by a previously executed SELECT.
Number(30,9)	number	Output	The value of the field in the corresponding result set is returned. Zero is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Fetch the OrderID field of the result set and return it as a number.
C Eval Field1 = RdbAscNum(Id:’OrderID’)
```

RDBGetStr (Get a character field from a record)

Purpose

RDBGetStr retrieves the data from a character field in the record of the remote database. The function is only valid when RDBExec has been used for a SELECT statement and a fetch has been used.

Syntax

String = RDBGetStr(ID: FieldNum)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	The field number in the result set that was generated by a previously executed SELECT.
Char(32767) Varying	String	Output	The value of the field in the corresponding result set is returned. Blank is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
  * Fetch the first field of the result set and return it as a character
  * field.
C        Eval      Field1 = RdbGetStr(Id:1)
```

RDBAscStr (Get a character field from a record using field name)

Purpose

RDBAscStr retrieves the data from a character field in the record of the remote database using the associated field name. The function is only valid when RDBExec has been used for a SELECT statement and a fetch has been used.

Syntax

String = RDBAscStr(ID: FieldName)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Char(50)	FieldName	Input	The name of the field in the result set that was generated by a previously executed SELECT.
Char(32767) Varying	String	Output	The value of the field in the corresponding result set is returned. Blank is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Fetch the CompanyName field of the result set and return it as a
* character field.
C      Eval      Field1 = RdbAscStr(Id:'CompanyName')
```

RDBGetDate (Get a date/time/timestamp field from a record)

Purpose

RDBGetDate retrieves the data from a date/time/timestamp field in the record of the remote database. The function is only valid when RDBExec has been used for a SELECT statement and a fetch has been used.

Syntax

Timestamp = RDBGetDate(ID: FieldNum)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	The field number in the result set that was generated by a previously executed SELECT.
Timestamp	Timestamp	Output	The value of the field in the corresponding result set is returned. An initialized timestamp is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Fetch the first field of the result set and return it as a timestamp
* field.
C       Eval       Field1 = RdbGetDate(Id:1)
```

RDBAscDate (Get a date/time/timestamp field from a record using field name)

Purpose

RDBAscDate retrieves the data from a date/time/timestamp field in the record of the remote database using the associated field name. The function is only valid when RDBExec has been used for a SELECT statement and a fetch has been used.

Syntax

Timestamp = RDBAscDate(ID: FieldName)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Char(50)	FieldName	Input	The name of the field in the result set that was generated by a previously executed SELECT.
Timestamp	Timestamp	Output	The value of the field in the corresponding result set is returned. An initialized timestamp is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Fetch the ShipDate field of the result set and return it as a
* timestamp field.
C      Eval      Field1 = RdbAscDate(Id:’ShipDate’)
```

RDBSetNum (Set a numeric field to a parameter marker)

Purpose

RDBSetNum associates a numeric application variable or constant value to a parameter marker in an SQL statement. When the statement is executed, the content of the variable is sent to the database server.

Syntax

rc = RDBSetNum(ID: FieldNum: Value)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	Parameter marker number, ordered sequentially left to right, starting at 1.
Number(30,9)	Value	Input	The value to use in the corresponding parameter marker.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
*  Set  the  value  of  the  second  parameter  as  a  number  with  a  value  of  5
C       Eval       rc = RdbSetNum(Id: 2: 5)
```

RDBSetStr (Set a string field to a parameter marker)

Purpose

RDBSetStr associates a character application variable or constant value to a parameter marker in an SQL statement. When the statement is executed, the content of the variable is sent to the database server.

Syntax

rc = RDBSetStr(ID: FieldNum: Value)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	Parameter marker number, ordered sequentially left to right, starting at 1.
Char(32767) Varying	Value	Input	The value to use in the corresponding parameter marker.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
*  Set  the  value  of  the  third  parameter  as  a  string  with  a  value  of  VINET
C       Eval       rc = RdbSetStr(Id: 3: 'VINET')
```

RDBSetDate (Set a timestamp field to a parameter marker)

Purpose

RDBSetDate associates a timestamp application variable or constant value to a parameter marker in an SQL statement. When the statement is executed, the content of the variable is sent to the database server.

Syntax

rc = RDBSetDate(ID: FieldNum: Value)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	Parameter marker number, ordered sequentially left to right, starting at 1.
Timestamp	Value	Input	The value to use in the corresponding parameter marker.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Set the value of the first parameter as a Date
C         Eval        rc = RdbSetDate(Id: 1: ShipDate)
```

RDBSetNull (Set a NULL value to a parameter marker)

Purpose

RDBSetNull associates a NULL indicator to a parameter marker in an SQL statement. When the statement is executed, the database server field will be set to NULL.

Syntax

rc = RDBSetNull(ID: FieldNum: FieldType)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	Parameter marker number, ordered sequentially left to right, starting at 1.
Int(10)	FieldType	Input	The type of field being set to NULL by this function. Valid types are: Rdb_Array Rdb_Boolean Rdb_Char Rdb_Clob Rdb_Date Rdb_Decimal Rdb_Double Rdb_Float Rdb_Integer Rdb_Null Rdb_Numeric Rdb_Real Rdb_SmallInt Rdb_Time Rdb_TimeStamp Rdb_VarChar
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
* Set the value of the first parameter as a NULL
C       Eval       rc = RdbSetNull(Id: 1: Rdb_Char)
```

RDBSetCommit(Set commitment control)

Purpose

RDBSetCommit set the automatic commitment control value. By default it is set to on, meaning all transaction are automatically committed. Setting this to off will for the need to either commit or rollback any transactions that are performed.

Syntax

rc = RDBSetCommit(ID: AutoCommit)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Boolean	AutoCommit	Input	Tells the remote DB engine if AutoCommit is true or false. The default is true.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Set the commitment control status.
C       Eval       rc = RdbSetCommit(Id: *Off)
```

RDBCommit(Commit all transactions)

Purpose

RDBCommit commits all transactions that have been performed since that last commit or rollback. Transaction commit only applies to the transactions issued for the current ID. Closing the ID without a commit will rollback the transactions.

Syntax

rc = RDBCommit(ID)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
* Commit the tranactions.
C       Eval       rc = RdbCommit(Id)
```

RDBRollback(Rollback all transactions)

Purpose

RDBRollback reverses all transactions that have been performed since that last commit or rollback. Transaction rollback only applies to the transactions issued for the current ID. Closing the ID without a commit will rollback the transactions.

Syntax

rc = RDBRollback (ID)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Rollback the tranactions.
C       Eval       rc = RdbRollback(Id)
```

RDBAddRec(Add a record to the remote database)

Purpose

RDBAddRec will add a record to the remote database, based on a previously executed SELECT statement. The record structure **must** match the field definitions from the RDBFIELDS command. An external datastructure can be created for this process by issuing the RDBIMPORT command to an output file and using that file as a datastructure.

The fields selected by the SELECT statement **must** match the datastructure.

Syntax

rc = RDBAddRec(ID: Record)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Char(*)	Record	Input	A datastructure that represents the record to be written. The structure must match the field definitions from the RDBFIELDS command.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
D @Orders E DS ExtName(Orders) Inz
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
*     After the connection is established, execute a select statement.
C      Eval           rc = RdbExec IId:
                      'Select * From [dbo].[Orders]')
*     Add a record to the remote file
       Eval           rc = RdbAddRec(id: @Orders)
```

RDBUpdRec(Update a record in the remote database)

Purpose

RDBUpdRec will update a last record read in the remote database, based on a previously executed SELECT statement. The record structure **must** match the field definitions from the RDBFIELDS command. An external datastructure can be created for this process by issuing the RDBIMPORT command to an output file and using that file as a datastructure.

The fields selected by the SELECT statement **must** match the datastructure.

Syntax

rc = RDBUpdRec(ID: Record)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Char(*)	Record	Input	A datastructure that represents the record to be written. The structure must match the field definitions from the RDBFIELDS command.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
D @Orders E DS ExtName(Orders) Inz
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++

* After the connection is established, execute a select statement.
C      Eval      rc = RdbExec(Id:
C                'Select * from [dbo].[Orders]')

* Fetch the next record of the result set that was generated with the
* previously executed statement.
C      Eval      rc = RdbFetchNxt(Id)

* Update the current record.
C      Eval      rc = RdbUpdRec(Id: @Orders)
```

RDBDelRec(Delete a record in the remote database)

Purpose

RDBDelRec will delete the last record read in the remote database, based on a previously executed SELECT statement.

Syntax

rc = RDBDelRec(ID)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++

* After the connection is established, execute a select statement.
C       Eval       rc = RdbExec(Id:
C                  'Select * from [dbo].[Orders]')

* Fetch the next record of the result set that was generated with the
* previously executed statement.
C       Eval       rc = RdbFetchNxt(Id)

* Delete the current record.
C       Eval       rc = RdbDelRec(Id)
```

RDBNextSet(Move the cursor to the next result set)

Purpose

RDBNextSet will move the cursor to the next result set of a multiple result set call. If a second result set does not exist, an error will be returned. Once the cursor has been moved, the previous result set can not be accessed again.

Syntax

rc = RDBNextSet(ID)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++

* After the connection is established, execute a select statement.
C       Eval       rc = RdbExec(Id:
C                  '{ Call MultiResultSet() }')

* Fetch the next record of the result set that was generated with the
* previously executed statement.
C       Eval       rc = RdbFetchNxt(Id)

* Move to the next result set
C       Eval       rc = RdbNextSet(Id)

* Fetch the next record of the result set that the cursor was just
* moved to.
C       Eval       rc = RdbFetchNxt(Id)
```

RDBStoredProc (Create a SQL statement to execute a stored procedure on the remote server)

Purpose

RDBStoredProc will execute a stored procedure on the remote server. The syntax for the statement must be valid on the remote server.

The SQL statement string may contain parameter markers. A parameter marker is represented by a "?" character, and indicates a position in the statement where the value of an application variable is to be substituted, when RDBPrepExec() is called. RDBSetStr(), RDBSetDate(), and RDBSetNum() are used to associate a application variable or constant value to each parameter marker.

RDBConnect() must be called before calling this function.

If a previous statement has been executed for this connection, RDBFreeStmt() must be called to close the cursor, before calling RDBPrepStmt ().

Syntax

rc = RDBStoredProc (ID: Statement)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Char(32767) Varying	Statement	Input	The stored procedure to be ran on the remote server.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
 *
 * After the connection is established, prepare the stored
 * procedure.
C       Eval      rc = RdbStoredProc(Id:
C                 '{ Call CreateFullName(?,?,?) }')
```

RDBRegOutput (Register an output parameter of a stored procedure)

Purpose

RDBRegOutput register a parameter with an output marker. The function is only valid when RDBStoredProc has been used.

Syntax

rc = RDBRegOutput(ID: FieldNum: FieldType: FldScale)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	The field number of the parameter that was used by a previously executed RDBStoredProc.
Int(10)	FieldType	Input	The type of field being registered by this function. Valid types are: Rdb_Array Rdb_Boolean Rdb_Char Rdb_Clob Rdb_Date Rdb_Decimal Rdb_Double Rdb_Float Rdb_Integer Rdb_Null Rdb_Numeric Rdb_Real Rdb_SmallInt Rdb_Time Rdb_TimeStamp Rdb_VarChar
Int(10)	FieldScale	Input	The number of decimal places to be returned by the stored procedure. This number must be zero or more.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Set the value of the third parameter as a output field of type
* character with a scale of 0.
C      Eval      rc = RdbRegOutput(Id: 3: Rdb_VarChar: 0)
```

RDBGetParmNum (Get a numeric field from a stored procedure parameter)

Purpose

RDBGetParmNum retrieves the data from a numeric field in the stored procedure call. The function is only valid when RDBStoredProc has been used for an execution and a RDBRegOutput has been set for the requested field.

Syntax

number = RDBGetParmNum(ID: FieldNum)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	The field number of the parameter that was used by a previously executed RDBStoredProc.
Number(30,9)	number	Output	The value of the field in the corresponding result set is returned. Zero is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
* Fetch the first field of the parameters and return it as a number.
C         Eval       Field1 = RdbGetParmNum(Id:1)
```

RDBGetParmStr (Get a character field from a stored procedure parameter)

Purpose

RDBGetParmStr retrieves the data from a numeric field in the stored procedure call. The function is only valid when RDBStoredProc has been used for an execution and a RDBRegOutput has been set for the requested field.

Syntax

String = RDBGetParmStr(ID: FieldNum)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	The field number of the parameter that was used by a previously executed RDBStoredProc.
Char(32767) Varying	String	Output	The value of the field in the corresponding result set is returned. Blank is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Fetch the first field of the parameters and return it as a character
* field.
C      Eval      Field1 = RdbGetParmStr(Id:1)
```

RDBGetParmDate (Get a date/time/timestamp field from a stored procedure parameter)

Purpose

RDBGetParmDate retrieves the data from a date/time/timestamp field in the stored procedure call. The function is only valid when RDBStoredProc has been used for an execution and a RDBRegOutput has been set for the requested field.

Syntax

Timestamp = RDBGetParmDate(ID: FieldNum)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	The field number of the parameter that was used by a previously executed RDBStoredProc.
Timestamp	Timestamp	Output	The value of the field in the corresponding result set is returned. An initialized timestamp is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
* Fetch the first field of the parameters and return it as a timestamp
* field.
C      Eval      Field1 = RdbGetParmDate(Id:1)
```

RDBCrtTable(Create a table in any Database configured in RDBCFG).

Purpose

RDBCrtTable is used to export/create IBMi iSeries tables into any database configured in RDBCFG.

Syntax

rc = RDBCrtTable(id: vCreateTableStmt)

Notes: Do not include a ‘CREATE TABLE‘ in the prepared statement, the API adds it automatically.

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDBConnect Statement.
Char(4096)	CreateTable command	Input	This parameter must contain the correct syntax for the appropriate database where the table is being created.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
//Prepare Statement
vCreateTableStmt =  ‘PERSON.TESTZIP (CITY CHAR(25), STATE CHAR(25), TEST_OPEN
CHAR(8), ROWID CHAR(10)) ‘

// Create Table
C          Eval        rc =  RdbCrtTable(id: vCreateTableStmt );
```

RDBSetIsoDate(Set ISO Date in result set)

Purpose

RDBSetIsoDate sets the date under ISO Format to transfer from IBMi to a compatible remote database. The function is only valid when RDBPrepStmt has been previously executed.

Syntax

rc = RDBSetIsoDate(ID: FieldNum: ISODATE)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDBConnect Statement.
Int(10)	FieldNum	Input	The field number of the parameter that was used by a previously executed RDBPrepStmt.
Date (10)	Date in ISO Format (2014-01-01)	Input	The value of the field in the corresponding result set.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
// Set the ISO Date value
C          Eval     rc =  RdbSetIsoDate(id: iQ: isoDate);
```

RDBSetIsoTime(Set ISO Time in result set)

Purpose

RDBSetIsoTime sets the time under ISO Format to transfer from IBMi to a compatible remote database. The function is only valid when RDBPrepStmt has been previously executed.

Syntax

rc = RDBSetIsoTime(ID: FieldNumber: TIME)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDBConnect Statement.
Int(10)	FieldNum	Input	The field number of the parameter that was used by a previously executed RDBPrepStmt.
Time (8)	Time in ISO Format (10.59.59)	Input	The value of the field in the corresponding result set.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
 // Set the ISO Time value
C           Eval      rc =  RdbSetIsoTime(id: iQ: IsoTime);
```

RDBSetCharStr(Set characters stream after executing RDBPrepStmt)

Purpose

RDBSetCharStr(Set character stream after a prepared statement. This allows up to 32767 characters.)

Syntax

rc = RDBSetCharStr(ID: FieldNum: CharacterVariable)

Function Arguments

Data Type	Argument	Use	Description
Int(10,0)	ID	Input	The ID that was returned from the RDBConnect Statement.
Int(10,0)	FieldNum	Input	Prepared statement placeholder sequence number
Char(32767)	Character variable	Input	Character variable containing results
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
 // Get list of Catalogs in remote database
 C          Eval            rc =  RDBSetCharStr(id: 1: vCharStream);
```

RDBGetCharStr(Get characters stream after executing RDBPrepExec)

Purpose

RDBGetCharStr(Get character stream after a prepared statement. This allows up to 32767 characters to be retrieved from remote database.)

Syntax

vRetValue = RDBGetCharStr(ID: FieldNum)

Function Arguments

Data Type	**Argument**	**Use**	**Description**
Int(10,0)	ID	Input	The ID that was returned from the RDBConnect Statement.
Int(10,0)	FieldNum	Input	Prepared statement placeholder sequence number
Char(32767)	Character variable	Output	Character variable receiving results

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
// Get list of Catalogs in remote database
  C                 Eval          VarString  =  RDBGetCharStr(id: 1);
```

RDBGetIsoTime(Get ISO Time in result set)

Purpose

RDBGetIsoTime gets the time value under ISO Format from a compatible remote database to IBMi. The function is only valid when RDBPrepStmt or RDBExec has been previously executed.

RPGLE Definition

```
  D   ISOTime     S         T
```

Syntax

```
 ISOTime = RDBGetIsoTime(ID: FieldNum)
```

Function Arguments

Data Type	Argument	Use	Description
Int(10,0)	ID	Input	The ID that was returned from the RDBConnect Statement.
Int(10)	FieldNumber	Input	The Field name /Column name from remote database table used in previous RDBPrepStmt or RDBExec
Time(8)	ISOTime	Output	The ISO Time field returned from the RDBConnect Statement.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
// Set the ISO Time value
   ISOTime =  RdbGetIsoTime(id: FieldNumber);
```

RDBGetIsoDate(Get ISO Date in result set)

Purpose

RDBGetIsoDate gets the date value under ISO Format from a compatible remote database to IBMi. The function is only valid when RDBPrepStmt or RDBExec has been previously executed.

RPGLE Definition

```
  D   ISODate     S         T
```

Syntax

```
 ISODate = RDBGetIsoDate(ID: FieldNumber)
```

Function Arguments

Data Type	**Argument**	**Use**	**Description**
Int(10,0)	ID	Input	The ID that was returned from the RDBConnect Statement.
Int(10)	FieldNumber	Input	The Field name /Column name from remote database table used in previous RDBPrepStmt or RDBExec
Date(10)	ISODate	Output	The ISO Date field returned from the RDBConnect Statement.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
 // Set the ISO Time value
   ISODate =  RdbGetIsoDate(id: FieldNumber);
```

RDBAscIsoTime(Get ISO Time in result set by Field Name)

Purpose

RDBAscIsoTime gets the time value under ISO Format from a compatible remote database to IBMi. The function is only valid when RDBPrepStmt or RDBExec has been previously executed.

RPGLE Definition

```
D   ISOTime     S         T
```

Syntax

```
ISOTime = RDBAscIsoTime(ID: FieldName)
```

Function Arguments

Data Type	Argument	Use	Description
Int(10,0)	ID	Input	The ID that was returned from the RDBConnect Statement.
Int(10)	FieldName	Input	The Field name /Column name from remote database table used in previous RDBPrepStmt or RDBExec
Time(8)	ISOTime	Output	The ISO Time field returned from the RDBConnect Statement.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
 // Set the ISO Time value
   ISOTime =  RdbAscIsoTime(id: FieldName);
```

RDBAscIsoDate(Get ISO Date in result set by Field Name)

Purpose

RDBAscIsoDate gets the date value in ISO Format from a compatible remote database to IBMi. The function is only valid when RDBPrepStmt or RDBExec has been previously executed.

RPGLE Definition

```
  D   ISODate     S         D
```

Syntax

```
 ISODate = RDBAscIsoDate(ID: FieldName)
```

Function Arguments

Data Type	**Argument**	**Use**	**Description**
Int(10,0)	ID	Input	The ID that was returned from the RDBConnect Statement.
Int(10)	FieldName	Input	The Field name /Column name from remote database table used in previous RDBPrepStmt or RDBExec
Date(10)	ISODate	Output	The ISO Date field returned from the RDBConnect Statement.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
// Set the ISO Date value
   ISODate =  RdbAscIsoDate(id: FieldName);
```

RDBGetBlob(Get Blob in result set by column number)

Purpose

RDBGetBlob gets the raw BLOB value in from a compatible remote database to IBMi. The function is only valid when RDBPrepStmt or RDBExec has been previously executed.

RPGLE Definition

```
  D   BLOB      S                         SQLTYPE(BLOB: 15728640)    //SQL DS
 D  vRetBlob   S                      A    len(15728640) varying
```

Syntax

```
 rc = RDBGetBlob(ID: FieldNum: BLOB_DATA);
```

Function Arguments

Data Type	Argument	Use	Description
Int(10,0)	ID	Input	The ID that was returned from the RDBConnect Statement.
Int(10)	FieldNumber	Input	The Field/Column number from remote database table used in previous RDBPrepStmt or RDBExec
Char(15728640)	BLOB_DATA	Output	The returned BLOB data will be return.
Int(10,0)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
 // Set the ISO Date value

C          Eval   rc =  RdbGetBlob(id: FieldNnumber: vRetBlob);
C          Eval   BLOB_LEN = %len(%trim(vRetBlob));
C          Eval   BLOB_Data = %trim(vRetBlob);

  exec SQL
  INSERT INTO DBUTEST/BLOBS
  VALUES(:Field1, :BLOB, :Field2);          //Insert BLOB into IBMi File.
```

RDBSetInt (Set a Integer field to a parameter marker)

Purpose

RDBSetInt associates an integer application variable or constant value to a parameter marker in an SQL statement. When the statement is executed, the content of the variable is sent to the database server.

Syntax

rc = RDBSetInt(ID: FieldNum: NumValue)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	Parameter marker number, ordered sequentially left to right, starting at 1.
Int(10)	NumValue	Input	The value to use in the corresponding parameter marker. This parameter can take up to a max of 10 numbers
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Set the value of the third parameter as a numeric value without decimal position
up to a max of 10 numbers.
 /Free
    rc = RdbSetInt(Id: 2: 5);
 /End-Free
```

RDBSetNegInt (Set a negative Integer field to a parameter marker)

Purpose

RDBSetNegInt associates a negative integer application variable or constant value to a parameter marker in an SQL statement. When the statement is executed, the content of the variable is sent to the database server.

Syntax

rc = RDBSetNegInt(ID: FieldNum: NumValue)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	Parameter marker number, ordered sequentially left to right, starting at 1.
Int(10)	NumValue	Input	The value to use in the corresponding parameter marker. This parameter can take up to a max of 10 numbers including a negative sign
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
* Set the value of the third parameter as a negative numeric value without decimal
position up to a max of 10 numbers.
 /Free
    rc = RdbSetNegInt(Id: 2: -516);
 /End-Free
```

RDBCpyToIFS (Copy Blob Data Directly to IFS)

Purpose

RDBCopyToIFS copies BLOB data to a specific directory in IFS. When the statement is executed, the content of the variable is sent to IFS.

Syntax

nIndicator = RDBCpyToIFS(vIfsDir: vFileName: Blob);

Function Arguments

Data Type	Argument	Use	Description
Char(100)	Path	Input	IFS Path where the file will be added
Char(100)	File Name	Input	Name of file to be added.
BLOB	SQLTYPE(BLOB)	Input	Actual raw BLOB data to be inserted
Boolean	nIndicator	Returns	True or False is returned to the output Boolean indicator.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
*
D BlobField   S                       SQLTYPE(BLOB:15728640)     //SQL BLOB DS
/Free
   nIndicator = RdbCpyToIFS('/ProData/Temp/':
                            vFileName:
                            BLOBField;
 /End-Free
```

RDBCpyFromIFS (Copy Blob Data Directly from IFS)

Purpose

RDBCpyFrmIFS retrieves BLOB from a specific directory in IFS. When the statement is executed, the content of the BLOB is available to insert it in a Table containing a BLOB column.

Syntax

nIndicator = RDBCpyFromIFS(vIfsDir: vFileName: Blob);

Function Arguments

Data Type	Argument	Use	Description
Char(100)	Path	Input	IFS Path
Char(100)	File Name	Input	Name of file to be added.
BLOB	SQLTYPE(BLOB)	Input	Actual raw BLOB data to be inserted
Boolean	nIndicator	Returns	True or False is returned to the output Boolean indicator.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
*
D BlobField   S                       SQLTYPE(BLOB:15728640)     //SQL BLOB DS
/Free
   nIndicator = RdbCpyFromIFS('/ProData/Temp/':
                              vFileName:
                              BLOBField;
 /End-Free
```

RDBExecFetchFirst (Execute a SQL statement on the remote server and fetch first row only)

Purpose

RDBExecFetchFirst directly executes the specified SQL statement on the remote server. Any valid SQL SELECT statement can be executed and the cursor is set to the first row. The syntax for the statement must be valid on the remote server. (`ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY`)

RDBConnect() must be called before calling this function.

If a previous statement has been executed for this connection, RDBFreeStmt() must be called to close the cursor, before calling RDBExecFetchFirst().

Syntax

rc = RDBExecFetchFirst(ID: Statement)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Char(32767) Varying	Statement	Input	The SELECT statement to be processed on the remote server.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
          CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
         *   After the connecton is established, prepare a select statement.
/Free
   rc = RdbExecFetchFirst(Id: ‘Select * from [dbo].[Orders]’);

    // Check for errors and must follow with RDB Getters

/End-Free
```

RDBWriteLog (Write to IFS TraceLog from IBMi)

Purpose

RDBWriteLog writes custom logs to the IFS file located in this directory '/prodata/rdb5'. This will help you keep track of your customized messages. Make sure the file exists in /prodata/rdb5 before you start using this API.

vFileName	S	100A	inz('rdbtracelog')
vText	S	100A	inz

Syntax

rc = RDBWriteLog(vFileName: %trim(vText))

Function Arguments

Data Type	Argument	Use	Description
Char(1024)	FileName	Input	File name where the log is going to be written.
Char(1024)	Text	Input	Text to be added to vFileName
Boolean	True/False	Output	True will return if no errors were encountered while writing to the file specified.

Examples

```
          CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
                                    *Write logs from any program.

/Free
   nBoolean = RdbWriteLog(FileName: 'Example notes');

    //

 /End-Free
```

RDBGetStrCs (Get string by specific IBM I character set code page)

Purpose

RDBGetStrCs – Get string by specific IBM I character set code page for double byte characters, example CP933 for Korean, CP935 for Simplified Chinese, CP420 for Arabic, CP1025 Russian, etc.

Notes:

1. Switch your Client Access host code page to the specific code page you are requesting your data, this will allow you to view the data. Change settings in Client Access (Communications -> Configure -> Host Code Page) Enable Unicode data Stream and Enable DBCS in Unicode Fields.
2. You must create the IBM I file column with data type 'G' and CCSID corresponding to the character set you wish to retrieve.
3. By default CCSID (13488 ISO/IEC 10646 Universal Coded Character Set Level 2 (UCS-2) for most double byte characters.
4. Change your job's CCSID (Use CMD CHGJOB) to match the language used in RDBGetStrCS

Syntax

vReturnString = RDBGetStrCs(vId: iFieldNum: 'CP935');

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	Connection ID.
Int(10)	Field Number	Input	Field Number from result set.
Char(10)	Character code page	Input	Character code page from IBM I
Char(32767)	Value	Return	Output from remote DB in requested code page.

Examples

```
          CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
                 *Retrieve String from remote DB in IBM I code page.

/Free
   vCharField = RdbGetStrCs(Id: FldNum: 'CP935');

    //

/End-Free
```

RDBAddBatch (Add individual statements to batch)

Purpose

RDBAddBatch – Add individual statements to batch after a prepared statement. By using batch processing, these queries can be sent to the database in one call, thus improving performance.

Syntax

rc = RDBAddBatch(vId);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	Connection ID.
Int(10)	Value	Return	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
          CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
                 *AddBatch must be called after a prepared statement.

/Free
   rc = RdbAddBatch(Id);

    //

/End-Free
```

RDBExecBatch (Execute batch)

Purpose

RDBExecBatch – Execute batch after a prepared statement and RDBAddBatch previously called. By using batch processing, these queries can be sent to the database in one call, thus improving performance.

Syntax

rc = RDBExecBatch(vId);

Function Arguments

Data Type	Argument	Use	Description
Dec(10)	ID	Input	Connection ID.
Int(10)	Value	Return	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
          CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
               *RDBExecBatch must be called after a prepared statement.

/Free
   rc = RdbExecBatch(Id);

    //

/End-Free
```

*RDBSetClob (Set CLOB Field)

Purpose

RDBSetClob – Set CLOB field after a prepared statement called.

Syntax

rc = RDBSetClob(vId, ColumnNumber, vClobValue);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	Connection ID.
Int(10)	Value	Input	Field Number to set after preparing SQL Statement.
Char(15728640) Varying	CLOB Value to be set	Input	CLOB Value to be set
Int(10)	rc	Return	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
          CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
               *RDBPrepExec must be called after a setting all fields.

Id        s   10I 0 Inz
vClob     S      A     len(15728640) Varying

/Free

   rc = RdbSetClob(Id, 1, %trim(vClob));

    //

 /End-Free
```

*RDBFTPPUT (Put single File via FTP)

Purpose

RDBFTPPUT – Transfer single file to remote FTP Server.

Syntax

rc = RDBFtpPut(vId, `vFileName, vLocPath, vRemPath`);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	Connection ID.
Char(99)	FileName	Input	File Name of the file getting transferred
Char(512)	Directory	Input (Optional)	Local Directory Path where the file being transferred resides. (Defaults to the connection local path if not passed in)
Char (512)	RemotePath	Input (Optional)	Remote Directory Path where the file is being transferred. (Defaults to the connection remote path if not passed in)
Int(10)	rc	Return	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
          CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
                  *RDBConnect must be called prior to using this API
Id        s   10I 0 Inz

/Free
   vFileName = ‘demo.zip’;
   vLocPath  = ‘/home/prodata’;
   vRemPath  = ‘/Uploads’;

   rc = RdbFTPPut(Id: vFileName: vMode: vLocPath: vRemPath);

    //

/End-Free
```

**RDBFTPGet (Get single File from remote server via FTP)*

Purpose

RDBFTPGET – Transfer single file from remote FTP Server.

Syntax

rc = RDBFtpGet(vId, `vFileName: vLocPath: vRemPath`);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	Connection ID.
Char(99)	FileName	Input	File Name of the file getting transferred
Char(512)	Directory	Input (Optional)	Local Directory Path where the file being transferred resides. (Defaults to the connection local path if not passed in)
Char (512)	Path	Input (Optional)	Remote Directory Path where the file is being transferred. (Defaults to the connection remote path if not passed in)
Int(10)	rc	Return	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
   *RDBConnect must be called prior to using this API
Id        s   10I 0 Inz

/Free
   vFileName = 'demo.zip';
   vLocPath  = '/home/prodata';
   vRemPath                                        =                    '/Uploads';
   rc = RdbFTPGet(Id: vFileName: vLocPath: vRemPath);
  //
/End-Free
```

RDBFTPGetDir (Get Directory from a remote server via FTP)

Purpose

RDBFTPGETDir – Transfer a directory from a remote FTP Server.

Syntax

rc = RDBFtpGetDir(vId, `vDirName: vLocPath: vRemPath`);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	Connection ID.
Char(99)	DirName	Input	Directory Name that's getting transferred.
Char(512)	Directory	Input (Optional)	Local Directory Path where the file being transferred resides. (Defaults to the connection local path if not passed in)
Char (512)	Path	Input (Optional)	Remote Directory Path where the file is being transferred. (Defaults to the connection remote path if not passed in)
Int(10)	rc	Return	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++++
   *RDBConnect must be called prior to using this API

Id        s    10I 0 Inz
/Free
   vDirName = 'demo';
   vLocPath  = '/home/prodata';
   vRemPath='/Uploads';
   rc = RdbFTPGetDir(Id: vDirName: vLocPath: vRemPath);
    //
/End-Free
```

**RDBFtp2Txt (Convert PF to CVS,XLX or XLSX and transfer to remote server via FTP)*

Purpose

RdbFtp2TXT – Convert PF to CVS or XML and transfer to remote server via FTP

Syntax

rc = RdbFtp2Txt (vId, `vFileName, vLibName, vRemPath, vExt, vFCCSID, vTCCSID`);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	Connection ID.
Char(10)	FILE	Input	PF Name getting converted
Char(10)	LIB	Input	Library Name where the file resides
Char (512)	PATH	Input (Optional)	Remote Directory Path where the file is being transferred. (Defaults to the connection remote path if not passed in)
Char(3)	EXT	Input (Optional)	Desired Extension (XML or CSV) Defaults to CSV
Char(5)	FCCSID	Input (Optional)	From CCSID Values (1-65533) Defaults to '37'
Char(9)	TCCSID	Input (Optional)	To CCSID Values (*STMF, *PCASCII, *STDASCII) Defaults to *PCASCII
Char(1)	Value	Input (Optional)	Include column headers. (Only supported on V7R2+)
Int(10)	rc	Return	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
   *RDBConnect must be called prior to using this API
Id        s   10I 0 Inz
/Free
   vFileName = 'demopf';
   vLib = 'IBMLIB';
   vRemPath  = '/Uploads';
   vColHeader                              =                              'Y';
   rc = RDBFtp2Txt(Id: vFileName: vLib: vRemPath); //extra parms below
   rc  =  RDBFtp2Txt(Id:  vFileName:  vLib:  vRemPath:  vExt:  vFCCID:  vTCCID:
vColHeader);
 /End-Free
```

RDBFTPPutDir (Put Directory on a remote server via FTP)

Purpose

RDBFTPPutDir – Transfer a directory to a remote FTP Server from IBM i

Syntax

rc = RDBFtpPutDir(vId, `vDirName: vLocPath: vRemPath`);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	Connection ID.
Char(512)	DirName	Input (Optional)	Local Directory Path where the file being transferred resides. (Defaults to the connection local path if not passed in)
Char (512)	Path	Input (Optional)	Remote Directory Path where the file is being transferred. (Defaults to the connection remote path if not passed in)
Int(10)	rc	Return	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
   *RDBConnect must be called prior to using this API
Id        s   10I 0 Inz

/Free
   vRemPath   = '/Uploads';
   vLocPath   = '/home/prodata';

   rc = RdbFTPPutDir(Id: vRemPath: vLocPath);

    //

 /End-Free
```

**RDBPCSNDFILE (Transfer file from IFS to PC directly)*

Purpose

RDBPCSNDFILE uploads a specific file from IFS to your PC

Syntax

rc = RdbPcSndFile(Id: FileName)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDBConnect statement
Char(512)	File Name	Input	File name
Int(10)	Return	Output	The value of the field in the corresponding result set is returned. Negative result is returned upon failure.

**RDBPC2TXT (Convert Physical File to CSV, XLX or XLSX and Transfer to PC)*

Purpose

RDBFTP2TXT Convert PF to CSV and transfer to remote PC

Syntax

rc = RdbFtp2Txt (Id: PF_Name: LibraryName: RemoteDir: Ext: FCCSID: TCCSID: HEADER);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDBConnect statement
Char(10)	Physical File Name	Input	Physical File name
Char(10)	Library Name	Input	Library name where the PF resides.
Char(512)	Remote Directory	Input	Remote directory where the file will be transferred. (*Optional)
Char(3)	Extension Type	Input	Specify the text file extension (CSV,TXT,XLS) (*Optional)
Char(5)	From CCSID	Input	Specify the CCSID
Char(9)	To CCSID	Input	Specify the To CCSID
Char(1)	Include Headers	Input	Specify if table header should be included (Y,N) (*Optional)
Int(10)	Return	Output	The value of the field in the corresponding result set is returned. Negative result is returned upon failure.

RDBRUNSQLFILE (Run SQL File against remote server)

Purpose

RDBRUNSQLFILE – Run any SQL file that includes the SQL statement to be executed on a remote server

Syntax

rc = RdbRunSqlFile(Id: IFSFilePath)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDBConnect statement
Char(512)	IFS File Path	Input	File path to .sql file
Int(10)	Return	Output	The value of the field in the corresponding result set is returned. Negative result is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
   *RDBConnect must be called prior to using this API

Id         s    10I 0 Inz
vIfsPath s   512a    Inz
/Free
   Id = RdbConnect('CONNID');
   vIfsPath   =  '/prodata/SqlDemo.sql';

   rc  =  RdbRunSQLFile(Id: vIfsPath);

    //

 /End-Free
```

RDBRUNCSVFILE (Run CSV File against remote server)

Purpose

RDBRUNCSVFILE – Execute CSV file from IBM i, Table must be created prior to executing this API

Syntax

rc = RdbRunCSVFile(Id: vTableName: IFSFilePath)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDBConnect statement
Char(256)	Table Name	Input	Schema and Table Name where table resides on remote DB
Char(512)	CSV File Path	Input	File path to .csv file
Int(10)	Return	Output	The value of the field in the corresponding result set is returned. Negative result is returned upon failure.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
   *RDBConnect must be called prior to using this API

Id        s   10I 0 Inz
vIfsPath s  512a    Inz

/Free
   Id = RdbConnect('CONNID’);
   vIfsPath   = ‘/prodata/csvDemo.csv’;

   rc = RdbRunCSVFile(Id: ‘Schema.TableName‘: vIfsPath);

    //

/End-Free
```

*RDBGETNVARCHAR (Get NVARCHAR String from MS SQL and Oracle Servers *ONLY)*

Purpose

RDBGETNVARCHAR – Get NVARCHAR String value back from MS SQL or ORACLE server only. Do not attempt to use with other databases that are not supported.

Syntax

rc = RdbGetNVarChar(Id: ColumnNum);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDBConnect statement
Int(10)	ColumnNum	Input	Column Number to
Int(10)	Return	Output	Returned value of rc

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
  * Fetch the first field of the result set and return it as a character
  * field.
C       Eval      Field1 = RdbGetNVarChar(Id: ColNum)
```

*RDBASCNVARCHAR (Get NVARCHAR String from MS SQL and Oracle Servers *ONLY)*

Purpose

RDBASCNVARCHAR – Get NVARCHAR String value back from MS SQL or ORACLE server only. Do not attempt to use with other databases that are not supported.

Syntax

rc = RdbAscNVarChar(Id: ColunName);

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDBConnect statement
Int(10)	ColumnName	Input	Column Name to retrieve
Int(10)	Return	Output	Returned value interger less than zero is an error

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2++++++++++++++++
  * Fetch the first field of the result set and return it as a character
  * field.
C       Eval      Field1 = RdbAscNVarChar(Id: “ColumnName”)
```

RDBSetFloat4 (Set a single byte float)

Purpose

RDBSetFloat4 associates a float application variable or constant value to a parameter marker in an SQL statement. When the statement is executed, the content of the variable is sent to the database server.

Syntax

rc = RDBSetFloat4(ID: FieldNum: Value)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	Parameter marker number, ordered sequentially left to right, starting at 1.
Float(4)	Value	Input	The value to use in the corresponding parameter marker.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Set the value of the second parameter as a number with a float value
D vFloat S    4f  inz
C      Eval       rc = RdbSetFloat(Id: 2: vFloat)
```

RDBSetFloat8 (Set a double byte float)

Purpose

RDBSetFloat8 associates a float application variable or constant value to a parameter marker in an SQL statement. When the statement is executed, the content of the variable is sent to the database server.

Syntax

rc = RDBSetFloat8(ID: FieldNum: Value)

Function Arguments

Data Type	Argument	Use	Description
Int(10)	ID	Input	The ID that was returned from the RDB Connect statement.
Int(10)	FieldNum	Input	Parameter marker number, ordered sequentially left to right, starting at 1.
Float(8)	Value	Input	The value to use in the corresponding parameter marker.
Int(10)	rc	Output	Zero is returned from a successful execution. Negative one (-1) is returned from a failed execution.

Examples

```
CL0N01Factor1+++++++Opcode&ExtExtended-factor2+++++++++++++++
* Set the value of the second parameter as a number with a float value
D vFloat8 S    8f   inz
C      Eval       rc = RdbSetFloat(Id: 2: vFloat8)
```