

SOURCE FILE . . . . . CDJDBC/QRPGLESRC  
 MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
100      /*-
200      * Copyright (c) 2006-2010 Scott C. Klement
300      * All rights reserved.
400      *
500      * Redistribution and use in source and binary forms, with or without
600      * modification, are permitted provided that the following conditions
700      * are met:
800      * 1. Redistributions of source code must retain the above copyright
900      * notice, this list of conditions and the following disclaimer.
1000     * 2. Redistributions in binary form must reproduce the above copyright
1100     * notice, this list of conditions and the following disclaimer in the
1200     * documentation and/or other materials provided with the distribution.
1300     *
1400     * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
1500     * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
1600     * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
1700     * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABL
1800     * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
1900     * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
2000     * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
2100     * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
2200     * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
2300     * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
2400     * SUCH DAMAGE.
2500     *
2600     */
2700     * Service program to simplify the user of JDBC drivers from ILE RPG.
2800     *                               Scott Klement, April 18, 2006
2900     *
3000     * Before Compiling:
3100     *
3200     * - You need the JDBC drivers (use pure Java ones) for your
3300     * database to be installed on the iSeries.
3400     *
3500     * - This was originally written for, and most testing has been
3600     * done with the MySQL drivers. But, other drivers should
3700     * work as well. For a MySQL slanted article, see the following:
3800     * http://www.iseriesnetwork.com/article.cfm?id=52433
3900     *
4000     * - You need the System Openness Includes (5722-SS1 opt 13)
4100     * licensed program installed.
4200     *
4300     * - You need JDK 1.3 or later (JDK 1.4 recommended!)
4400     *
4500     *
4600     * To Compile:
4700     *> CRTRPGMOD JDBCR4 SRCFILE(QRPGLESRC) DBGVIEW(*LIST)
4800     *      ( Note: this is the JDBCR4.rpgle source member. )
4900     *
5000     *> CRTSRVPGM SRVPGM(JDBCR4) -
5100     *> EXPORT(*SRCFILE) SRCFILE(QSRVSRC)
5200     *      ( Note: this is the JDBCR4.bnd source member. )
5300     *

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
5400      *      CRTBNDDIR BNDDIR(mylib/JDBC)
5500      *      ADDBNDDIRE BNDDIR(mylib/JDBC) OBJ((JDBCR4 *SRVPGM))
5600      H NOMAIN
5700
5800      /define OS400_JVM_12
5900      /define JNI_COPY_CALL_METHOD_FUNCTIONS
6000      /copy qsysinc/qrpglesrc,jni
6100      /copy qrpglesrc,jdbc_h
6200
6300      D new_Driver      PR              like(Driver)
6400      D   driver_name      256A      varying const
6500      D jni_checkError  PR              1N
6600      D   errString      256A      varying options(*omit)
6700      D jdbc_get_jni_env...
6800      D               PR              *
6900      D jdbc_begin_object_group...
7000      D               PR              10I 0
7100      D   peCapacity      10I 0 value
7200      D jdbc_end_object_group...
7300      D               PR              10I 0
7400      D   peOldObj        like(jObject) const
7500      D               options(*nopass)
7600      D   peNewObj        like(jObject)
7700      D               options(*nopass)
7800      D attach_jvm      PR              *
7900      D start_jvm      PR              *
8000      D JniVersion      PR              10P 5
8100      D Timezone      PR              10I 0
8200
8300      D SndPgmMsg      PR              ExtPgm('QMHSNDPM')
8400      D   MessageID      7A      Const
8500      D   QualMsgF      20A      Const
8600      D   MsgData      80A      Const
8700      D   MsgDtaLen      10I 0 Const
8800      D   MsgType      10A      Const
8900      D   CallStkEnt      10A      Const
9000      D   CallStkCnt      10I 0 Const
9100      D   MessageKey      4A
9200      D   ErrorCode      32767A options(*varsize)
9300      D status      PR              1N
9400      D   peMsgTxt      256A      const
9500      D MsgKey      s              4A
9600      D ErrCode      s              8A inz(x'0000000000000000')
9700
9800      D Object      s              0 CLASS(*JAVA:
9900      D               'java.lang.Object')
10000     D Driver      s              0 CLASS(*JAVA:
10100     D               'java.sql.Driver')
10200     D BigDecimal      s              0 CLASS(*JAVA:
10300     D               'java.math.BigDecimal')
10400     D jsqlDate      s              0 CLASS(*JAVA:
10500     D               'java.sql.Date')
10600     D jsqlTime      s              0 CLASS(*JAVA:

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

```
SOURCE FILE : . . . . . SOURCEID:
MEMBER      : . . . . . JDBCR4
```

SEQNBR*	+...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
10700	D 'java.sql.Time')
10800	D jSqlTimestamp s 0 CLASS(*JAVA:
10900	D 'java.sql.Timestamp')
11000	
11100	D createStatement...
11200	D PR ExtProc(*JAVA:
11300	D 'java.sql.Connection':
11400	D 'createStatement')
11500	D like(Statement)
11600	
11700	D PrepareStatement...
11800	D PR ExtProc(*JAVA:
11900	D 'java.sql.Connection':
12000	D 'prepareStatement')
12100	D like(PreparedStatement)
12200	D sql like(jString) const
12300	
12400	D PrepareCall...
12500	D PR ExtProc(*JAVA:
12600	D 'java.sql.Connection':
12700	D 'prepareCall')
12800	D like(CallableStatement)
12900	D sql like(jString) const
13000	
13100	D executeQuery...
13200	D PR ExtProc(*JAVA:
13300	D 'java.sql.Statement':
13400	D 'executeQuery')
13500	D like(ResultSet)
13600	D query like(jString) const
13700	
13800	D executeUpdate...
13900	D PR ExtProc(*JAVA:
14000	D 'java.sql.Statement':
14100	D 'executeUpdate')
14200	D like(jInt)
14300	D query like(jString) const
14400	
14500	D getMetaData...
14600	D PR ExtProc(*JAVA:
14700	D 'java.sql.ResultSet':
14800	D 'getMetaData')
14900	D like(ResultSetMetaData)
15000	
15100	D getColumnCount...
15200	D PR ExtProc(*JAVA:
15300	D 'java.sql.ResultSetMetaData':
15400	D 'getColumnCount')
15500	D like(jInt)
15600	
15700	D getColumnName...
15800	D PR ExtProc(*JAVA:
15900	D 'java.sql.ResultSetMetaData':

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
16000      D                                     'getColumnName')
16100      D                                     like(jString)
16200      D      colIndex                       like(jInt) value
16300
16400      D getColumnDisplaySize...
16500      D                                     PR                               ExtProc(*JAVA:
16600      D                                     'java.sql.ResultSetMetaData':
16700      D                                     'getColumnDisplaySize')
16800      D                                     like(jInt)
16900      D      colIndex                       like(jInt) value
17000
17100      D getColumnTypeName...
17200      D                                     PR                               ExtProc(*JAVA:
17300      D                                     'java.sql.ResultSetMetaData':
17400      D                                     'getColumnTypeName')
17500      D                                     like(jString)
17600      D      colIndex                       like(jInt) value
17700
17800      D nextRec...
17900      D                                     PR                               1N  ExtProc(*JAVA:
18000      D                                     'java.sql.ResultSet':
18100      D                                     'next')
18200
18300      D prevRec...
18400      D                                     PR                               1N  ExtProc(*JAVA:
18500      D                                     'java.sql.ResultSet':
18600      D                                     'last')
18700
18800      D getColString...
18900      D                                     PR                               ExtProc(*JAVA:
19000      D                                     'java.sql.ResultSet':
19100      D                                     'getString')
19200      D                                     like(jString)
19300      D      colIndex                       like(jInt) value
19400
19500      D getColStrByName...
19600      D                                     PR                               ExtProc(*JAVA
19700      D                                     : 'java.sql.ResultSet'
19800      D                                     : 'getString')
19900      D                                     like(jString)
20000      D      colName                       like(jString)
20100
20200      D closeConn                          PR                               ExtProc(*JAVA:
20300      D                                     'java.sql.Connection':
20400      D                                     'close')
20500
20600      D commitTrn                          PR                               ExtProc(*JAVA:
20700      D                                     'java.sql.Connection':
20800      D                                     'commit')
20900
21000      D rollbackTrn                       PR                               ExtProc(*JAVA:
21100      D                                     'java.sql.Connection':
21200      D                                     'rollback')

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

```
SOURCE FILE : . . . . . SOURCEID:
MEMBER      : . . . . . JDBCR4
```

SEQNBR	*...+...	1 ...+...	2 ...+...	3 ...+...	4 ...+...	5 ...+...	6 ...+...	7 ...+...	8 ...+...	9 ...+...	0
21300											
21400	D	s		PR			like(jString)				
21500	D						EXTPROC(*JAVA				
21600	D						: 'java.lang.String'				
21700	D						: *CONSTRUCTOR)				
21800	D	create_from			32767A		VARYING const				
21900											
22000	D	new_StringC		PR			like(jString)				
22100	D						EXTPROC(*JAVA				
22200	D						: 'java.lang.String'				
22300	D						: *CONSTRUCTOR)				
22400	D	create_from			16383c		VARYING const				
22500											
22600	D	r		pr	32767A		varying				
22700	D						extproc(*JAVA:				
22800	D						'java.lang.String':				
22900	D						'getBytes')				
23000											
23100	D	c		pr	16383C		varying				
23200	D						extproc(*JAVA:				
23300	D						'java.lang.String':				
23400	D						'toCharArray')				
23500											
23600	D	new_Properties		PR			ExtProc(*JAVA				
23700	D						: 'java.util.Properties'				
23800	D						: *CONSTRUCTOR)				
23900	D						like(Properties)				
24000											
24100	D	setProperty		PR			ExtProc(*JAVA				
24200	D						: 'java.util.Properties'				
24300	D						: 'setProperty')				
24400	D						like(Object)				
24500	D	key					like(jString) const				
24600	D	value					like(jString) const				
24700											
24800	D	registerDriver		PR			ExtProc(*JAVA				
24900	D						: 'java.sql.DriverManager'				
25000	D						: 'registerDriver')				
25100	D						static				
25200	D	drv					like(Driver)				
25300											
25400	D	Driver_ConnProp...									
25500	D			PR			like(Connection)				
25600	D	drv					like(Driver) value				
25700	D	url			256A		varying const options(*varsize)				
25800	D	prop					like(Properties) value				
25900	D	Driver_connect		PR			like(Connection)				
26000	D	drv					like(Driver) value				
26100	D	url			256A		varying const options(*varsize)				
26200	D	user			50a		varying const options(*varsize)				
26300	D	pass			50a		varying const options(*varsize)				
26400											
26500	D	new_BigDecimal		PR			ExtProc(*JAVA				

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
26600      D                               : 'java.math.BigDecimal'
26700      D                               : *CONSTRUCTOR)
26800      D                               like(BigDecimal)
26900      D   str                         like(jString) const
27000
27100      D new_Date                     PR      ExtProc(*JAVA
27200      D                               : 'java.sql.Date'
27300      D                               : *CONSTRUCTOR)
27400      D                               like(jSqlDate)
27500      D   milli                      like(jLong) value
27600
27700      D new_Time                     PR      ExtProc(*JAVA
27800      D                               : 'java.sql.Time'
27900      D                               : *CONSTRUCTOR)
28000      D                               like(jSqlTime)
28100      D   milli                      like(jLong) value
28200
28300      D new_Timestamp                PR      ExtProc(*JAVA
28400      D                               : 'java.sql.Timestamp'
28500      D                               : *CONSTRUCTOR)
28600      D                               like(jSqlTimestamp)
28700      D   milli                      like(jLong) value
28800
28900      D setBigDecimal...
29000      D                               PR      ExtProc(*JAVA
29100      D                               : 'java.sql.PreparedStatement'
29200      D                               : 'setBigDecimal')
29300      D   idx                        like(jInt) value
29400      D   dec                        like(BigDecimal) const
29500
29600      D setDate...
29700      D                               PR      ExtProc(*JAVA
29800      D                               : 'java.sql.PreparedStatement'
29900      D                               : 'setDate')
30000      D   idx                        like(jInt) value
30100      D   date                      like(jSqlDate) const
30200
30300      D setTime...
30400      D                               PR      ExtProc(*JAVA
30500      D                               : 'java.sql.PreparedStatement'
30600      D                               : 'setTime')
30700      D   idx                        like(jInt) value
30800      D   time                      like(jSqlTime) const
30900
31000      D setTimestamp...
31100      D                               PR      ExtProc(*JAVA
31200      D                               : 'java.sql.PreparedStatement'
31300      D                               : 'setTimestamp')
31400      D   idx                        like(jInt) value
31500      D   ts                        like(jSqlTimestamp) const
31600
31700      D setDouble...
31800      D                               PR      ExtProc(*JAVA

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
31900      D                                     : 'java.sql.PreparedStatement'
32000      D                                     : 'setDouble')
32100      D      idx                             like(jInt) value
32200      D      dubba                           like(jDouble) value
32300
32400      D setInt...
32500      D                                     PR                               ExtProc(*JAVA
32600      D                                     : 'java.sql.PreparedStatement'
32700      D                                     : 'setInt')
32800      D      idx                             like(jInt) value
32900      D      int                             like(jInt) value
33000
33100      D setString...
33200      D                                     PR                               ExtProc(*JAVA
33300      D                                     : 'java.sql.PreparedStatement'
33400      D                                     : 'setString')
33500      D      idx                             like(jInt) value
33600      D      str                             like(jString) const
33700
33800      D setByte...
33900      D                                     PR                               ExtProc(*JAVA
34000      D                                     : 'java.sql.PreparedStatement'
34100      D                                     : 'setByte')
34200      D      idx                             like(jInt) value
34300      D      byte                           like(jByte) value
34400
34500      D registerOutParameter...
34600      D                                     PR                               ExtProc(*JAVA
34700      D                                     : 'java.sql.CallableStatement'
34800      D                                     : 'registerOutParameter')
34900      D      idx                             like(jInt) value
35000      D      type                           like(jInt) value
35100
35200      D getString...
35300      D                                     PR                               ExtProc(*JAVA:
35400      D                                     'java.sql.CallableStatement':
35500      D                                     'getString')
35600      D                                     like(jString)
35700      D      idx                             like(jInt) value
35800
35900      D getInt...
36000      D                                     PR                               ExtProc(*JAVA:
36100      D                                     'java.sql.CallableStatement':
36200      D                                     'getInt')
36300      D                                     like(jInt)
36400      D      idx                             like(jInt) value
36500
36600      D getShort...
36700      D                                     PR                               ExtProc(*JAVA:
36800      D                                     'java.sql.CallableStatement':
36900      D                                     'getShort')
37000      D                                     like(jShort)
37100      D      idx                             like(jInt) value

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

SEQNBR\*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0

```

37200
37300      D  getDouble...
37400      D                      PR                      ExtProc(*JAVA:
37500      D                      'java.sql.CallableStatement':
37600      D                      'getDouble')
37700      D                      like(jDouble)
37800      D      idx                      like(jInt) value
37900
38000      D  getBigDecimal...
38100      D                      PR                      ExtProc(*JAVA:
38200      D                      'java.sql.CallableStatement':
38300      D                      'getBigDecimal')
38400      D                      like(BigDecimal)
38500      D      idx                      like(jInt) value
38600
38700      D  getLong...
38800      D                      PR                      ExtProc(*JAVA:
38900      D                      'java.sql.CallableStatement':
39000      D                      'getLong')
39100      D                      like(jLong)
39200      D      idx                      like(jInt) value
39300
39400      D  getTimestamp...
39500      D                      PR                      ExtProc(*JAVA:
39600      D                      'java.sql.CallableStatement':
39700      D                      'getTimestamp')
39800      D                      like(jSqlTimestamp)
39900      D      idx                      like(jInt) value
40000
40100      D  getTime...
40200      D                      PR                      ExtProc(*JAVA:
40300      D                      'java.sql.CallableStatement':
40400      D                      'getTime')
40500      D                      like(jSqlTime)
40600      D      idx                      like(jInt) value
40700
40800      D  getDate...
40900      D                      PR                      ExtProc(*JAVA:
41000      D                      'java.sql.CallableStatement':
41100      D                      'getDate')
41200      D                      like(jSqlDate)
41300      D      idx                      like(jInt) value
41400
41500      D  getBoolean...
41600      D                      PR                      1N ExtProc(*JAVA:
41700      D                      'java.sql.CallableStatement':
41800      D                      'getBoolean')
41900      D      idx                      like(jInt) value
42000
42100      D  prepExecuteQuery...
42200      D                      PR                      ExtProc(*JAVA
42300      D                      : 'java.sql.PreparedStatement'
42400      D                      : 'executeQuery')

```



SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
42500      D                                like(ResultSet)
42600
42700      D  prepExecuteUpdate...
42800      D                                PR                                ExtProc(*JAVA
42900      D                                : 'java.sql.PreparedStatement'
43000      D                                : 'executeUpdate')
43100      D                                like(jInt)
43200
43300      D  callExecute...
43400      D                                PR                                1N  ExtProc(*JAVA
43500      D                                : 'java.sql.CallableStatement'
43600      D                                : 'execute')
43700
43800      D  prepExecute...
43900      D                                PR                                1N  ExtProc(*JAVA
44000      D                                : 'java.sql.PreparedStatement'
44100      D                                : 'execute')
44200
44300      D  Exception_getMessage...
44400      D                                PR                                ExtProc(*JAVA
44500      D                                : 'java.lang.Throwable'
44600      D                                : 'getMessage')
44700      D                                like(jstring)
44800
44900      D  rs_getStatement...
45000      D                                PR                                ExtProc(*JAVA
45100      D                                : 'java.sql.ResultSet'
45200      D                                : 'getStatement')
45300      D                                like(Statement)
45400
45500      D  stmt_Close          PR                                ExtProc(*JAVA
45600      D                                : 'java.sql.Statement'
45700      D                                : 'close')
45800
45900      *+++++
46000      * MySQL_Connect(): Create a connection to a MYSQL server
46100      *
46200      *   server = (input) mysql server to connect to
46300      *   database = (input) database to use on server
46400      *   userid = (input) userid to log in with
46500      *   password = (input) password to log in with
46600      *
46700      * Returns a connection handle or *NULL upon failure
46800      *+++++
46900      P MySQL_Connect      B                                export
47000      D MySQL_Connect      PI                                like(Connection)
47100      D  server              256A  varying const options(*varsize)
47200      D  database            256A  varying const options(*varsize)
47300      D  userid              50A   varying const options(*varsize)
47400      D  password            50A   varying const options(*varsize)
47500
47600      D url                  s              256A  varying
47700      D conn                 s              like(Connection)

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
47800
47900      /free
48000          url = 'jdbc:mysql://' + %trim(server) + '/'
48100              + %trim(database);
48200
48300          conn = JDBC_connect('com.mysql.jdbc.Driver'
48400                          : url
48500                          : userid
48600                          : password );
48700
48800          return conn;
48900      /end-free
49000 P                      E
49100
49200
49300      ++++++
49400      * JDBC_Connect(): Create a connection using JDBC driver
49500      *
49600      *     driver = (input) Java class name of JDBC driver to use
49700      *     url = (input) JDBC URL of database to connect to
49800      *     userid = (input) userid to log in with
49900      *     password = (input) password to log in with
50000      *
50100      * Returns a connection handle or *NULL upon failure
50200      ++++++
50300 P JDBC_Connect      B      export
50400 D JDBC_Connect      PI      like(Connection)
50500 D   drivname                256A   varying const options(*varsize)
50600 D   url                    256A   varying const options(*varsize)
50700 D   userid                  50A   varying const options(*varsize)
50800 D   password                50A   varying const options(*varsize)
50900
51000 D drv                      s      like(Driver)
51100 D temp                     s      like(Connection)
51200 D conn                     s      like(Connection)
51300 D prop                     s      like(Properties)
51400
51500      /free
51600          prop = JDBC_Properties();
51700          JDBC_setProp(prop: 'user': userid);
51800          JDBC_setProp(prop: 'password': password);
51900          conn = JDBC_ConnProp(drivname: url: prop);
52000          JDBC_freeProp(prop);
52100          return conn;
52200
52300      // -----
52400      // The following code currently does not work
52500      // because there is no 'connect(url,user,pass)'
52600      // method in java.sql.Driver.
52700      // -----
52800
52900      jdbc_begin_object_group(50);
53000

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
53100      monitor;
53200      // Find & Instantiate Driver
53300
53400      drv = new_Driver(drvname);
53500      if (drv = *NULL);
53600          jdbc_end_object_group();
53700      return *NULL;
53800      endif;
53900
54000      // Register with DriverManager
54100      //   and connect.
54200
54300      registerDriver(drv);
54400      temp = Driver_Connect( drv : url: userid: password);
54500      if (temp = *NULL);
54600          jdbc_end_object_group();
54700      return *NULL;
54800      endif;
54900
55000      jdbc_end_object_group(temp: conn);
55100      return conn;
55200  on-error;
55300      jdbc_end_object_group();
55400      return *NULL;
55500  endmon;
55600
55700  /end-free
55800  P          E
55900
56000
56100  ++++++
56200  * JDBC_ConnProp(): Connect to database w/properties object
56300  *
56400  *   driver = (input) Java class name of JDBC driver to use
56500  *   url = (input) JDBC URL to connect to
56600  *   prop = (input) properties to use when connecting
56700  *
56800  * Returns a connection handle or *NULL upon failure
56900  ++++++
57000  P JDBC_ConnProp  B          export
57100  D JDBC_ConnProp  PI          like(Connection)
57200  D   drvname          256A    varying const options(*varsize)
57300  D   url          256A    varying const options(*varsize)
57400  D   prop          like(Properties)
57500  D   drv          s          like(Driver)
57600  D   temp          s          like(Connection)
57700  D   conn          s          like(Connection)
57800  /free
57900      jdbc_begin_object_group(50);
58000
58100      monitor;
58200      // Find & Instantiate Driver
58300

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
58400      drv = new_Driver(drvname);
58500      if (drv = *NULL);
58600          jdbc_end_object_group();
58700      return *NULL;
58800      endif;
58900
59000      // Register with DriverManager
59100      //   and connect.
59200
59300      registerDriver(drv);
59400      temp = Driver_ConnProp( drv : url: prop );
59500      if (temp = *NULL);
59600          jdbc_end_object_group();
59700      return *NULL;
59800      endif;
59900
60000      jdbc_end_object_group(temp: conn);
60100      return conn;
60200  on-error;
60300      jdbc_end_object_group();
60400      return *NULL;
60500  endmon;
60600
60700  /end-free
60800  P          E
60900
61000
61100  ++++++
61200  *   JDBC_Properties(): Create new properties object
61300  *
61400  *   returns the object.
61500  ++++++
61600  P JDBC_Properties...
61700  P          B          export
61800  D JDBC_Properties...
61900  D          PI          like(Properties)
62000  D prop          s          like(Properties)
62100  /free
62200      jdbc_get_jni_env();
62300      prop = new_Properties();
62400      return prop;
62500  /end-free
62600  P          E
62700
62800
62900  ++++++
63000  *   JDBC_setProp(): Set one property in Properties object
63100  *
63200  *   prop = (i/o) Properties object to set property in
63300  *   key = (input) property to set
63400  *   value = (input) value to assign to property
63500  ++++++
63600  P JDBC_setProp      B          export
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
63700      D JDBC_setProp      PI
63800      D   prop
63900      D   key              256A   like(Properties)
64000      D   value           256A   varying const options(*varsize)
64100      D   keystr          s       like(jString)
64200      D   valstr          s       like(jString)
64300      /free
64400      D   keystr = s(key);
64500      D   valstr = s(value);
64600      D   setProperty(prop: keystr: valstr);
64700      D   DeleteLocalRef(JNIENV_P: keystr);
64800      D   DeleteLocalRef(JNIENV_P: valstr);
64900      /end-free
65000      P                      E
65100
65200
65300      *+++++
65400      *   JDBC_freeProp(): Release properties object
65500      *
65600      *   prop = (i/o) Properties object to release
65700      *+++++
65800      P JDBC_freeProp      B       export
65900      D JDBC_freeProp      PI
66000      D   prop              like(Properties)
66100      /free
66200      D   DeleteLocalRef(JNIENV_P: prop);
66300      D   prop = *NULL;
66400      /end-free
66500      P                      E
66600
66700
66800      *+++++
66900      * This calls the JNI functions to retrieve a JDBC driver
67000      * object based on the class name.
67100      *
67200      *   driver_name = (input) Java class name of JDBC driver
67300      *
67400      * returns new java.sql.Driver object
67500      *+++++
67600      P new_Driver          B
67700      D new_Driver          PI       like(Driver)
67800      D   driver_name       256A   varying const
67900
68000      D QDCXLATE            PR       ExtPgm('QDCXLATE')
68100      D   len               5P 0    const
68200      D   data              32702A  options(*varsize)
68300      D   table             10A     const
68400
68500      D                      ds
68600      D   drv_name          256A   varying
68700      D   drv_fixed         256A   overlay(drv_name:3)
68800
68900      D CONSTRUCTOR        C       x'3c696e69743e'

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
69000      D NPARMVOID          C          x'282956'
69100
69200      D env                s          *      static inz(*NULL)
69300      D msg                s          256A    varying
69400      D cls                s          like(jclass)
69500      D mid                s          like(jmethodid)
69600      D obj                s          like(Driver)
69700      /free
69800          if (env = *NULL);
69900              env = jdbc_get_jni_env();
70000          endif;
70100
70200          drv_name = %xlate('.' : '/' : driver_name);
70300          QDCXLATE(%len(drv_name): drv_fixed: 'QTCASC');
70400
70500          cls = FindClass(env: drv_name);
70600          if (jni_CheckError(*omit));
70700              return *NULL;
70800          endif;
70900
71000          if ( cls = *NULL );
71100              msg = 'Class ' + driver_name + ' not found!';
71200              exsr DiagMsg;
71300              return *NULL;
71400          endif;
71500
71600          mid = GetMethodID(env: cls: CONSTRUCTOR: NPARMVOID);
71700          if (jni_CheckError(*omit));
71800              return *NULL;
71900          endif;
72000
72100          if ( mid = *NULL );
72200              msg = 'Constructor method for ' + driver_name
72300                  + ' not found!';
72400              exsr DiagMsg;
72500              return *NULL;
72600          endif;
72700
72800          obj = NewObject(env: cls: mid);
72900          if (jni_CheckError(*omit));
73000              return *NULL;
73100          endif;
73200
73300          if ( obj = *NULL );
73400              msg = 'Null returned when attempting to create '
73500                  + driver_name + ' object';
73600              exsr DiagMsg;
73700              return *NULL;
73800          endif;
73900
74000          return obj;
74100
74200          begsr DiagMsg;
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

SEQNBR\*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0

```

74300      SndPgmMsg( 'CPF9897'
74400          : 'QCPFMSG *LIBL'
74500          : msg
74600          : %len(msg)
74700          : '*DIAG'
74800          : '*'
74900          : 0
75000          : MsgKey
75100          : ErrCode );
75200      endsr;
75300  /end-free
75400  P          E
75500
75600
75700  ++++++
75800  * Driver_ConnProp(): This calls the JNI routines (directly)
75900  *   to connect a driver to a data source.
76000  *
76100  * NOTE: I used the JNI routines instead of RPG's SQL support
76200  *       because I have more control over the error checking.
76300  *
76400  *       drv = (input) driver object to connect with
76500  *       url = (input) JDBC url to server
76600  *       prop = (input) connection properties
76700  *
76800  * Returns a Connection object, or *NULL upon failure
76900  ++++++
77000  P Driver_ConnProp...
77100  P          B
77200  D Driver_ConnProp...
77300  D          PI          like(Connection)
77400  D   drv          like(Driver) value
77500  D   url          256A   varying const options(*varsize)
77600  D   prop          like(Properties) value
77700
77800  D QDCXLATE          PR          ExtPgm('QDCXLATE')
77900  D   len          5P 0   const
78000  D   data          32702A options(*varsize)
78100  D   table          10A   const
78200
78300  D clsDriver          s          15A   inz('java/sql/Driver')
78400  D methConn          s          7A   inz('connect')
78500  D sigConn          s          63A   inz('(Ljava/lang/String;-
78600  D                               Ljava/util/Properties;)-
78700  D                               Ljava/sql/Connection;')
78800  D CallConnectMethod...
78900  D          PR          LIKE(jobject)
79000  D          EXTPROC(*CWIDEN
79100  D          : JNINativeInterface.
79200  D          CallObjectMethod_P)
79300  D env          LIKE(JNIEnv_P) VALUE
79400  D obj          LIKE(jobject) VALUE
79500  D methodID          LIKE(jmethodID) VALUE

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
79600      D drv                                LIKE(Driver) VALUE
79700      D                                options(*nopass)
79800      D url                                LIKE(jString) VALUE
79900      D                                options(*nopass)
80000      D prop                              LIKE(Properties) VALUE
80100      D                                options(*nopass)
80200
80300      D msg                                s                256A    varying
80400      D cls                                s                like(jclass)
80500      D mid                                s                like(jmethodid)
80600      D conn                              s                like(Connection)
80700      D str                                s                like(jString)
80800
80900      /free
81000
81100          QDCXLATE( %size(clsDriver)
81200              : clsDriver
81300              : 'QTCPASC');
81400          QDCXLATE( %size(methConn)
81500              : methConn
81600              : 'QTCPASC');
81700          QDCXLATE( %size(sigConn)
81800              : sigConn
81900              : 'QTCPASC');
82000
82100          cls = FindClass(JNIENV_P: clsDriver);
82200          if (jni_CheckError(*omit));
82300              return *NULL;
82400          endif;
82500
82600          if ( cls = *NULL );
82700              msg = 'Class java.sql.Driver not found!';
82800              exsr DiagMsg;
82900              return *NULL;
83000          endif;
83100
83200          mid = GetMethodID(JNIENV_P: cls: methConn: sigConn);
83300          if (jni_CheckError(*omit));
83400              return *NULL;
83500          endif;
83600
83700          if ( mid = *NULL );
83800              msg = 'Connect method not found in java.sql.Driver'
83900                  + ' class!';
84000              exsr DiagMsg;
84100              return *NULL;
84200          endif;
84300
84400          status('Connecting to database...');
84500
84600          str = s(url);
84700          conn = CallConnectMethod(JNIENV_P: drv: mid: str: prop);
84800          status('.');

```



SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
84900      if (jni_CheckError(*omit));
85000          return *NULL;
85100      endif;
85200      DeleteLocalRef(JNIENV_P: str);
85300
85400      if ( conn = *NULL );
85500          msg = 'Unable to connect.';
85600          exsr DiagMsg;
85700          return *NULL;
85800      endif;
85900
86000      return conn;
86100
86200      begsr DiagMsg;
86300          SndPgmMsg( 'CPF9897'
86400                  : 'QCPFMSG *LIBL'
86500                  : msg
86600                  : %len(msg)
86700                  : '*DIAG'
86800                  : '*'
86900                  : 0
87000                  : MsgKey
87100                  : ErrCode );
87200      endsr;
87300  /end-free
87400  P          E
87500
87600
87700  *+++++
87800  * Driver_connect(): This calls the JNI routines (directly)
87900  *   to connect a driver to a data source.
88000  *
88100  * NOTE: I used the JNI routines instead of RPG's SQL support
88200  *   because I have more control over the error checking.
88300  *
88400  *   drv = (input) driver object to connect with
88500  *   url = (input) JDBC url to server
88600  *   user = (input) userid to connect with
88700  *   pass = (input) password to connect with
88800  *
88900  * Returns a Connection object, or *NULL upon failure
89000  *+++++
89100  P Driver_connect  B
89200  D Driver_connect  PI          like(Connection)
89300  D   drv          like(Driver) value
89400  D   url          256A  varying const options(*varsize)
89500  D   user         50a   varying const options(*varsize)
89600  D   pass         50a   varying const options(*varsize)
89700
89800  D QDCXLATE        PR          ExtPgm('QDCXLATE')
89900  D   len           5P 0  const
90000  D   data         32702A  options(*varsize)
90100  D   table        10A   const

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
90200
90300      D clsDriver          s          15A      inz('java/sql/Driver')
90400      D methConn          s          7A        inz('connect')
90500      D sigConn           s          77A        inz('(Ljava/lang/String;-
90600      D                               Ljava/lang/String;-
90700      D                               Ljava/lang/String;)-
90800      D                               Ljava/sql/Connection;')
90900      D CallConnectMethod...
91000      D                               PR
91100      D                               LIKE(jobject)
91200      D                               EXTPROC(*CWIDEN
91300      D                               : JNINativeInterface.
91400      D                               CallObjectMethod_P)
91500      D env                 LIKE(JNIEnv_P) VALUE
91600      D obj                 LIKE(jobject) VALUE
91700      D methodID            LIKE(jmethodID) VALUE
91800      D drv                 LIKE(Driver) VALUE
91900      D url                 LIKE(jString) VALUE
92000      D                               options(*nopass)
92100      D userid             LIKE(jString) VALUE
92200      D                               options(*nopass)
92300      D passwd            LIKE(jString) VALUE
92400      D                               options(*nopass)
92500
92600      D msg                 s          256A      varying
92700      D cls                 s          like(jclass)
92800      D mid                 s          like(jmethodid)
92900      D conn                s          like(Connection)
93000      D urlstr             s          like(jString)
93100      D userstr            s          like(jString)
93200      D passsstr           s          like(jString)
93300
93400      /free
93500
93600          QDCXLATE( %size(clsDriver)
93700              : clsDriver
93800              : 'QTCPASC');
93900          QDCXLATE( %size(methConn)
94000              : methConn
94100              : 'QTCPASC');
94200          QDCXLATE( %size(sigConn)
94300              : sigConn
94400              : 'QTCPASC');
94500
94600          cls = FindClass(JNIENV_P: clsDriver);
94700          if (jni_CheckError(*omit));
94800              return *NULL;
94900          endif;
95000
95100          if ( cls = *NULL );
95200              msg = 'Class java.sql.Driver not found!';
95300              exsr DiagMsg;
95400              return *NULL;

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
95500      endif;
95600
95700      mid = GetMethodID(JNIENV_P: cls: methConn: sigConn);
95800      if (jni_CheckError(*omit));
95900          return *NULL;
96000      endif;
96100
96200      if ( mid = *NULL );
96300          msg = 'Connect method not found in java.sql.Driver'
96400              + ' class!';
96500          exsr DiagMsg;
96600          return *NULL;
96700      endif;
96800
96900      status('Connecting to database...');
97000
97100      urlstr = s(url);
97200      userstr = s(url);
97300      passstr = s(url);
97400
97500      conn = CallConnectMethod( JNIENV_P
97600                              : drv
97700                              : mid
97800                              : urlstr
97900                              : userstr
98000                              : passstr );
98100
98200      status('.');
98300      if (jni_CheckError(*omit));
98400          return *NULL;
98500      endif;
98600      DeleteLocalRef(JNIENV_P: urlstr);
98700      DeleteLocalRef(JNIENV_P: userstr);
98800      DeleteLocalRef(JNIENV_P: passstr);
98900
99000      if ( conn = *NULL );
99100          msg = 'Unable to connect.';
99200          exsr DiagMsg;
99300          return *NULL;
99400      endif;
99500
99600      return conn;
99700
99800      begsr DiagMsg;
99900          SndPgmMsg( 'CPF9897'
100000              : 'QCPFMSG *LIBL'
100100              : msg
100200              : %len(msg)
100300              : '*DIAG'
100400              : '*'
100500              : 0
100600              : MsgKey
100700              : ErrCode );
100800      endsr;
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
100800      /end-free
100900      P                      E
101000
101100
101200      *+++++
101300      *   JDBC_ExecUpd(): Execute a Query that doesn't return results
101400      *
101500      *       conn = (input) database connection
101600      *       sql  = (input) SQL Code to execute
101700      *
101800      * returns a row count, or 0 where a row count is not applicable
101900      *       or -1 upon failure
102000      *+++++
102100      P JDBC_ExecUpd      B                      export
102200      D JDBC_ExecUpd      PI                      10I 0
102300      D   conn                      like(Connection) const
102400      D   sql                      32767A      varying const options(*varsize)
102500      D rc                      s                      10I 0
102600      D stmt                      s                      like(Statement)
102700      /free
102800          jdbc_begin_object_group(50);
102900
103000      monitor;
103100          stmt = createStatement( conn );
103200          rc = executeUpdate( stmt : s(sql));
103300          stmt_close(stmt);
103400          jdbc_end_object_group();
103500      on-error;
103600          jdbc_end_object_group();
103700          return -1;
103800      endmon;
103900      return rc;
104000      /end-free
104100      P                      E
104200
104300
104400      *+++++
104500      *   JDBC_ExecQry(): Execute a Query that returns a result
104600      *
104700      *       conn = (input) database connection
104800      *       sql  = (input) SQL code to execute
104900      *
105000      * Returns a result handle, or *NULL upon failure
105100      *+++++
105200      P JDBC_ExecQry      B                      export
105300      D JDBC_ExecQry      PI                      like(ResultSet)
105400      D   conn                      like(Connection) const
105500      D   sql                      32767A      varying const options(*varsize)
105600      D stmt                      s                      like(Statement)
105700      D temp                      s                      like(ResultSet)
105800      D rs                      s                      like(ResultSet)
105900      /free
106000          jdbc_begin_object_group(50);

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
106100
106200      monitor;
106300          stmt = createStatement( conn );
106400          temp = executeQuery( stmt : s(sql));
106500          jdbc_end_object_group(temp: rs);
106600      on-error;
106700          jdbc_end_object_group();
106800      return *NULL;
106900  endmon;
107000
107100      return rs;
107200  /end-free
107300 P              E
107400
107500
107600      ++++++
107700      * JDBC_PrepStmt(): Create a Prepared Statement
107800      *
107900      *      conn = (input) Connection to prepare statement for
108000      *      sql  = (input) SQL statement to prepare
108100      *
108200      * Returns a prepared statement object, or *NULL upon error
108300      ++++++
108400 P JDBC_PrepStmt      B      export
108500 D JDBC_PrepStmt      PI      like(PreparedStatement)
108600 D      conn           like(Connection) const
108700 D      sql            32767A  varying const options(*varsize)
108800 D temp               s      like(PreparedStatement)
108900 D stmt              s      like(PreparedStatement)
109000  /free
109100      jdbc_begin_object_group(50);
109200
109300      monitor;
109400          temp = prepareStatement( conn : s(sql) );
109500          jdbc_end_object_group(temp: stmt);
109600      on-error;
109700          jdbc_end_object_group();
109800      return *NULL;
109900  endmon;
110000
110100      return stmt;
110200  /end-free
110300 P              E
110400
110500
110600      ++++++
110700      * JDBC_PrepStmtC(): Create a Prepared Statement
110800      *      from a Unicode String
110900      *
111000      *      conn = (input) Connection to prepare statement for
111100      *      sql  = (input) SQL statement to prepare
111200      *
111300      * Returns a prepared statement object, or *NULL upon error

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
111400      ++++++
111500      P JDBC_PrepStmtC  B      export
111600      D JDBC_PrepStmtC  PI      like(PreparedStatement)
111700      D   conn      like(Connection) const
111800      D   sql      16383c   varying const options(*varsize)
111900      D temp      s      like(PreparedStatement)
112000      D stmt      s      like(PreparedStatement)
112100      /free
112200      jdbc_begin_object_group(50);
112300
112400      monitor;
112500      temp = prepareStatement( conn : new_StringC(sql) );
112600      jdbc_end_object_group(temp: stmt);
112700      on-error;
112800      jdbc_end_object_group();
112900      return *NULL;
113000      endmon;
113100
113200      return stmt;
113300      /end-free
113400      P      E
113500
113600
113700      ++++++
113800      * JDBC_PrepCall(): Create a Prepared Call
113900      *
114000      *   conn = (input) Connection to prepare call for
114100      *   sql = (input) SQL call to prepare
114200      *
114300      * Returns a prepared call object, or *NULL upon error
114400      ++++++
114500      P JDBC_PrepCall  B      export
114600      D JDBC_PrepCall  PI      like(CallableStatement)
114700      D   conn      like(Connection) const
114800      D   sql      32767A   varying const options(*varsize)
114900      D temp      s      like(CallableStatement)
115000      D stmt      s      like(CallableStatement)
115100      /free
115200      jdbc_begin_object_group(50);
115300
115400      monitor;
115500      temp = prepareCall( conn : s(sql) );
115600      jdbc_end_object_group(temp: stmt);
115700      on-error;
115800      jdbc_end_object_group();
115900      return *NULL;
116000      endmon;
116100      return stmt;
116200      /end-free
116300      P      E
116400
116500
116600      ++++++
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
116700 * JDBC_PrepCallC(): Create a Prepared Call
116800 *          using a Unicode string
116900 *
117000 *      conn = (input) Connection to prepare call for
117100 *      sql = (input) SQL call to prepare
117200 *
117300 * Returns a prepared call object, or *NULL upon error
117400 ++++++
117500 P JDBC_PrepCallC  B          export
117600 D JDBC_PrepCallC  PI          like(CallableStatement)
117700 D      conn          like(Connection) const
117800 D      sql          16383C    varying const options(*varsize)
117900 D temp              s          like(CallableStatement)
118000 D stmt              s          like(CallableStatement)
118100 /free
118200     jdbc_begin_object_group(50);
118300
118400     monitor;
118500         temp = prepareCall( conn : new_StringC(sql) );
118600         jdbc_end_object_group(temp: stmt);
118700     on-error;
118800         jdbc_end_object_group();
118900         return *NULL;
119000     endmon;
119100     return stmt;
119200 /end-free
119300 P                      E
119400
119500
119600 ++++++
119700 * JDBC_GetMetaData(): Get ResultSet MetaData
119800 *
119900 *      rs = (input) ResultSet handle
120000 *
120100 * Returns a ResultSet MetaData object, or *NULL upon failure
120200 ++++++
120300 P JDBC_GetMetaData...
120400 P      B          export
120500 D JDBC_GetMetaData...
120600 D      PI          like(ResultSetMetaData)
120700 D      rs          like(ResultSet)
120800 D temp              s          like(ResultSetMetaData)
120900 D rsmd              s          like(ResultSetMetaData)
121000 /free
121100     jdbc_begin_object_group(50);
121200     monitor;
121300         temp = getMetaData( rs );
121400         jdbc_end_object_group(temp: rsmd);
121500     on-error;
121600         jdbc_end_object_group();
121700         return *NULL;
121800     endmon;
121900     return rsmd;

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
122000      /end-free
122100      P                      E
122200
122300
122400      *+++++
122500      *   JDBC_GetColCount():  Get ResultSet Column Count
122600      *
122700      *       rsmd = (input) ResultSet MetaData handle
122800      *
122900      *   Returns column count
123000      *+++++
123100      P JDBC_GetColCount...
123200      P                      B                      export
123300      D JDBC_GetColCount...
123400      D                      PI                      10I 0
123500      D   rsmd                      like(ResultSetMetaData)
123600      D colcnt                      s                      like(jInt)
123700      /free
123800          jdbc_begin_object_group(50);
123900          monitor;
124000              colcnt = getColumnCount( rsmd );
124100          on-error;
124200              colcnt = 0;
124300          endmon;
124400          jdbc_end_object_group();
124500          return colcnt;
124600      /end-free
124700      P                      E
124800
124900
125000      *+++++
125100      *   JDBC_GetColName():  Get column name from ResultSet MetaData
125200      *
125300      *       rsmd = (input) ResultSet MetaData handle to use
125400      *       col = (input) column number to retrieve
125500      *
125600      *   Returns column name if successful, or '' otherwise
125700      *+++++
125800      P JDBC_GetColName...
125900      P                      B                      export
126000      D JDBC_GetColName...
126100      D                      PI                      256A   varying
126200      D   rsmd                      like(ResultSetMetaData)
126300      D   col                      10I 0   value
126400      D result                      s                      256A   varying
126500      D str                        s                      like(jString)
126600      /free
126700          jdbc_begin_object_group(10);
126800          monitor;
126900              str = getColumnName(rsmd: col);
127000              if (str = *NULL);
127100                  result = '';
127200              else;

```



SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
127300      result = r(str);
127400      endif;
127500      on-error;
127600      result = '';
127700      endmon;
127800      jdbc_end_object_group();
127900      return result;
128000  /end-free
128100  P          E
128200
128300
128400  ++++++
128500  *   JDBC_GetColDspSize():  Get column display size from ResultSet MetaData
128600  *
128700  *       rsmd = (input) ResultSet MetaData handle to use
128800  *       col = (input) column number to retrieve
128900  *
129000  *   Returns column display size
129100  ++++++
129200  P   JDBC_GetColDspSize...
129300  P           B                      export
129400  D   JDBC_GetColDspSize...
129500  D           PI                      10I 0
129600  D   rsmd                      like(ResultSetMetaData)
129700  D   col                       10I 0 value
129800  D   colsize                   s      like(jInt)
129900  /free
130000      jdbc_begin_object_group(10);
130100      monitor;
130200      colsize = getColumnDisplaySize(rsmd: col);
130300      on-error;
130400      colsize = 0;
130500      endmon;
130600      jdbc_end_object_group();
130700      return colsize;
130800  /end-free
130900  P          E
131000
131100
131200  ++++++
131300  *   JDBC_GetColTypeName():  Get column SQL type name from ResultSet MetaData
131400  *
131500  *       rsmd = (input) ResultSet MetaData handle to use
131600  *       col = (input) column number to retrieve
131700  *
131800  *   Returns column SQL type name if successful, or '' otherwise
131900  ++++++
132000  P   JDBC_GetColTypNme...
132100  P           B                      export
132200  D   JDBC_GetColTypNme...
132300  D           PI                      256A  varying
132400  D   rsmd                      like(ResultSetMetaData)
132500  D   col                       10I 0 value
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
132600      D result          s          256A   varying
132700      D str            s          like(jString)
132800      /free
132900          jdbc_begin_object_group(10);
133000      monitor;
133100          str = getColumnTypeName(rsmd: col);
133200          if (str = *NULL);
133300              result = '';
133400          else;
133500              result = r(str);
133600          endif;
133700      on-error;
133800          result = '';
133900      endmon;
134000      jdbc_end_object_group();
134100      return result;
134200  /end-free
134300  P          E
134400
134500
134600      ++++++
134700      * JDBC_SetString(): Set a string in a prepared statement
134800      *
134900      *     prep = (input) Prepared statement to set string in
135000      *     idx = (input) Parameter index to set
135100      *     str = (input) String to set
135200      *     null = (input/optional) set field to NULL in database
135300      ++++++
135400  P JDBC_SetString...
135500  P          B          export
135600  D JDBC_SetString...
135700  D          PI          1N
135800  D     prep
135900  D     idx          10I 0 value
136000  D     str          32767A   varying const options(*varsize)
136100  D     null          1N     const options(*nopass)
136200  D result          s          1n
136300  /free
136400      jdbc_begin_object_group(10);
136500
136600      monitor;
136700          if (%parms>=4 and null=*ON);
136800              setString( prep: idx: *NULL);
136900          else;
137000              setString( prep: idx: s(str));
137100          endif;
137200          result = *ON;
137300      on-error;
137400          result = *OFF;
137500      endmon;
137600
137700      jdbc_end_object_group();
137800      return result;

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
137900      /end-free
138000      P                      E
138100
138200
138300      *+++++
138400      * JDBC_SetStringC(): Set a string in a prepared statement
138500      *                      using Unicode input
138600      *
138700      *      prep = (input) Prepared statement to set string in
138800      *      idx  = (input) Parameter index to set
138900      *      str  = (input) String to set
139000      *      null = (input/optional) set field to NULL in database
139100      *+++++
139200      P JDBC_SetStringC...
139300      P                      B                      export
139400      D JDBC_SetStringC...
139500      D                      PI                      1N
139600      D      prep                      like(PreparedStatement) const
139700      D      idx                      10I 0 value
139800      D      str                      16383C varying const options(*varsize)
139900      D      null                      1N      const options(*nopass)
140000      D result                      s                      1n
140100      /free
140200      jdbc_begin_object_group(10);
140300
140400      monitor;
140500      if (%parms>=4 and null=*ON);
140600      setString( prep: idx: *NULL);
140700      else;
140800      setString( prep: idx: new_StringC(str));
140900      endif;
141000      result = *ON;
141100      on-error;
141200      result = *OFF;
141300      endmon;
141400
141500      jdbc_end_object_group();
141600      return result;
141700      /end-free
141800      P                      E
141900
142000
142100      *+++++
142200      * JDBC_SetInt(): Set an integer in a prepared statement
142300      *
142400      *      prep = (input) Prepared statement to set value in
142500      *      idx  = (input) Parameter index to set
142600      *      int  = (input) value to set
142700      *      null = (input/optional) set field to NULL in database
142800      *+++++
142900      P JDBC_SetInt      B                      export
143000      D JDBC_SetInt      PI                      1N
143100      D      prep                      like(PreparedStatement) const

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC  
 MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
143200      D   idx                      10I 0 value
143300      D   int                      10I 0 value
143400      D   null                    1N   const options(*nopass)
143500      D result                      s      1n
143600      /free
143700          jdbc_begin_object_group(10);
143800          monitor;
143900              if (%parms>=4 and null=*ON);
144000                  setString( prep: idx: *NULL);
144100              else;
144200                  setInt( prep: idx: int);
144300              endif;
144400              result = *ON;
144500          on-error;
144600              result = *OFF;
144700          endmon;
144800          jdbc_end_object_group();
144900          return result;
145000      /end-free
145100      P                      E
145200
145300
145400      ++++++
145500      * JDBC_SetDouble(): Set a floating point value in a prepared
145600      *                      statement
145700      *
145800      *      prep = (input) Prepared statement to set value in
145900      *      idx = (input) Parameter index to set
146000      *      double = (input) value to set
146100      *      null = (input/optional) set field to NULL in database
146200      ++++++
146300      P JDBC_SetDouble...
146400      P                      B                      export
146500      D JDBC_SetDouble...
146600      D                      PI                      1N
146700      D      prep                      like(PreparedStatement) const
146800      D      idx                      10I 0 value
146900      D      double                    8F   value
147000      D      null                    1N   const options(*nopass)
147100      D result                      s      1n
147200      /free
147300          jdbc_begin_object_group(10);
147400          monitor;
147500              if (%parms>=4 and null=*ON);
147600                  setString( prep: idx: *NULL);
147700              else;
147800                  setDouble( prep: idx: double);
147900              endif;
148000              result = *ON;
148100          on-error;
148200              result = *OFF;
148300          endmon;
148400          jdbc_end_object_group();

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
148500      return result;
148600      /end-free
148700      P          E
148800
148900
149000      ++++++
149100      * JDBC_SetDecimal(): Set a decimal value in a prepared stmt
149200      *
149300      *     prep = (input) Prepared statement to set value in
149400      *     idx = (input) Parameter index to set
149500      *     dec = (input) value to set
149600      *     null = (input/optional) set field to NULL in database
149700      ++++++
149800      P JDBC_SetDecimal...
149900      P          B          export
150000      D JDBC_SetDecimal...
150100      D          PI          1N
150200      D     prep          like(PreparedStatement) const
150300      D     idx          10I 0 value
150400      D     dec          30P 9 value
150500      D     null          1N     const options(*nopass)
150600      D result          s          1n
150700      /free
150800      jdbc_begin_object_group(10);
150900      monitor;
151000      if (%parms>=4 and null=*ON);
151100      setBigDecimal(prepare: idx: *NULL);
151200      else;
151300      setBigDecimal(prepare: idx: new_BigDecimal(s(%char(dec))));
151400      endif;
151500      result = *ON;
151600      on-error;
151700      result = *OFF;
151800      endmon;
151900      jdbc_end_object_group();
152000      return result;
152100      /end-free
152200      P          E
152300
152400
152500      ++++++
152600      * JDBC_SetDate(): Set a date value in a prepared stmt
152700      *
152800      *     prep = (input) Prepared statement to set value in
152900      *     idx = (input) Parameter index to set
153000      *     date = (input) value to set
153100      *     null = (input/optional) set field to NULL in database
153200      ++++++
153300      P JDBC_SetDate...
153400      P          B          export
153500      D JDBC_SetDate...
153600      D          PI          1N
153700      D     prep          like(PreparedStatement) const
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
153800      D      idx                                10I 0 value
153900      D      date                                D      datfmt(*iso) const
154000      D      null                                1N      const options(*nopass)
154100      D      jdt                                s      like(jSqlDate)
154200      D      str                                s      like(jString)
154300      D      rc                                s      1N      inz(*ON)
154400      D      Date_valueOf      PR      like(jSqlDate)
154500      D      extproc(*java
154600      D      : 'java.sql.Date'
154700      D      : 'valueOf')
154800      D      static
154900      D      theDate                                like(jString)
155000      /free
155100      jdbc_begin_object_group(10);
155200
155300      monitor;
155400      str = s(%char(date:*iso));
155500      jdt = Date_valueOf(str);
155600      if (jdt = *NULL);
155700      rc = *OFF;
155800      else;
155900      if (%parms>=4 and null=*ON);
156000      setDate(prep: idx: *NULL);
156100      else;
156200      setDate(prep: idx: jdt);
156300      endif;
156400      rc = *ON;
156500      endif;
156600      on-error;
156700      rc = *OFF;
156800      endmon;
156900
157000      jdbc_end_object_group();
157100      return rc;
157200      /end-free
157300      P      E
157400
157500      ++++++
157600      * JDBC_SetTime(): Set a time value in a prepared stmt
157700      *
157800      *      prep = (input) Prepared statement to set value in
157900      *      idx = (input) Parameter index to set
158000      *      time = (input) value to set
158100      *      null = (input/optional) set field to NULL in database
158200      ++++++
158300      P JDBC_SetTime...
158400      P      B      export
158500      D JDBC_SetTime...
158600      D      PI      1N
158700      D      prep      like(PreparedStatement) const
158800      D      idx      10I 0 value
158900      D      time      T      timfmt(*hms) const
159000      D      null      1N      const options(*nopass)

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
159100      D jtm              s              like(jSqlTime)
159200      D str              s              like(jString)
159300      D rc              s              1N      inz(*ON)
159400      D Time_valueOf    PR              like(jSqlTime)
159500      D                  extproc(*java
159600      D                  : 'java.sql.Time'
159700      D                  : 'valueOf')
159800      D                  static
159900      D theTime          like(jString)
160000      /free
160100          if (JniVersion() < 1.00004);
160200              SndPgmMsg( 'CPF9897'
160300                  : 'QCPFMSG *LIBL'
160400                  : 'JDK Version 1.4 or later required!'
160500                  : 80
160600                  : '*ESCAPE'
160700                  : '*PGMBDY'
160800                  : 1
160900                  : MsgKey
161000                  : ErrCode );
161100          return *OFF;
161200      endif;
161300
161400      jdbc_begin_object_group(10);
161500
161600      monitor;
161700          str = s(%char(time:*hms:));
161800          jtm = Time_valueOf(str);
161900          if (jtm = *NULL);
162000              rc = *OFF;
162100          else;
162200              if (%parms>=4 and null=*ON);
162300                  setTime(prepare: idx: *NULL);
162400              else;
162500                  setTime(prepare: idx: jtm);
162600              endif;
162700              rc = *ON;
162800          endif;
162900      on-error;
163000          rc = *OFF;
163100      endmon;
163200
163300      jdbc_end_object_group();
163400      return rc;
163500  /end-free
163600  P              E
163700
163800
163900      ++++++
164000      * JDBC_SetTimestamp(): Set a timestamp value in a prepared stmt
164100      *
164200      *      prep = (input) Prepared statement to set value in
164300      *      idx = (input) Parameter index to set

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC  
 MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
164400      *      ts = (input) value to set
164500      *      null = (input/optional) set field to NULL in database
164600      *+++++
164700      P JDBC_SetTimestamp...
164800      P          B                      export
164900      D JDBC_SetTimestamp...
165000      D          PI                      1N
165100      D      prep                      like(PreparedStatement) const
165200      D      idx                      10I 0 value
165300      D      ts                      Z      const
165400      D      null                      1N      const options(*nopass)
165500      D      jts                      s          like(jSqlTimestamp)
165600      D      charTS                  s          29A      varying
165700      D      str                      s          like(jString)
165800      D      rc                      s          1N      inz(*ON)
165900      D      TS_valueOf              PR          like(jSqlTimestamp)
166000      D                              extproc(*java
166100      D                              : 'java.sql.Timestamp'
166200      D                              : 'valueOf')
166300      D                              static
166400      D      theTS                      like(jString)
166500      /free
166600          if (JniVersion() < 1.00004);
166700              SndPgmMsg( 'CPF9897'
166800                      : 'QCPFMSG *LIBL'
166900                      : 'JDK Version 1.4 or later required!'
167000                      : 80
167100                      : '*ESCAPE'
167200                      : '*PGMBDY'
167300                      : 1
167400                      : MsgKey
167500                      : ErrCode );
167600          return *OFF;
167700      endif;
167800
167900      jdbc_begin_object_group(10);
168000
168100      monitor;
168200          charTS = %char(ts:iso) + '000';
168300          %subst(charTS:11:1) = ' ';
168400          %subst(charTS:14:1) = ':';
168500          %subst(charTS:17:1) = ':';
168600          str = s(charTS);
168700          jts = TS_valueOf(str);
168800          if (jts = *NULL);
168900              rc = *OFF;
169000          else;
169100              if (%parms>=4 and null=*ON);
169200                  setTimestamp(prepare: idx: *NULL);
169300              else;
169400                  setTimestamp(prepare: idx: jts);
169500              endif;
169600              rc = *ON;

```



SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
169700          endif;
169800          on-error;
169900          rc = *OFF;
170000          endmon;
170100
170200          jdbc_end_object_group();
170300          return rc;
170400      /end-free
170500  P          E
170600
170700
170800      ++++++
170900      * JDBC_RegisterOutParameter(): Register an output parameter
171000      *                                returned from a stored-proc call
171100      *
171200      *      call = (input) Callable statement to register output for
171300      *      idx = (input) Parameter index to register
171400      *      type = (input) Data type to register
171500      ++++++
171600  P JDBC_RegisterOutParameter...
171700  P          B          export
171800  D JDBC_RegisterOutParameter...
171900  D          PI
172000  D      call          like(CallableStatement) const
172100  D      idx          10I 0 value
172200  D      type          10I 0 value
172300  /free
172400      jdbc_begin_object_group(10);
172500      monitor;
172600      registerOutParameter( call: idx: type);
172700      on-error;
172800      // ignore error
172900      endmon;
173000      jdbc_end_object_group();
173100      return;
173200  /end-free
173300  P          E
173400
173500
173600      ++++++
173700      * JDBC_GetString(): Get a string from a called statement
173800      *
173900      *      call = (input) Callable statement to get string from
174000      *      idx = (input) Parameter index to get
174100      ++++++
174200  P JDBC_GetString...
174300  P          B          export
174400  D JDBC_GetString...
174500  D          PI          32767A      varying
174600  D      call          like(CallableStatement) const
174700  D      idx          10I 0 value
174800  D result          s          32767A      varying
174900  D str              s          like(jString)

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
175000      /free
175100      jdbc_begin_object_group(10);
175200
175300      monitor;
175400          str = getString( call: idx);
175500          if (str = *NULL);
175600              result = '';
175700          else;
175800              result = r(str);
175900          endif;
176000      on-error;
176100          result = '';
176200      endmon;
176300
176400      jdbc_end_object_group();
176500      return result;
176600  /end-free
176700  P          E
176800
176900
177000      ++++++
177100      * JDBC_GetInt(): Get an integer from a called statement
177200      *
177300      *      call = (input) Callable statement to get string from
177400      *      idx = (input) Parameter index to get
177500      ++++++
177600  P JDBC_GetInt...
177700  P          B          export
177800  D JDBC_GetInt...
177900  D          PI          10I 0
178000  D      call          like(CallableStatement) const
178100  D      idx          10I 0 value
178200  D result          s          10I 0
178300      /free
178400      jdbc_begin_object_group(10);
178500
178600      monitor;
178700          result = getInt( call: idx);
178800      on-error;
178900          result = 0;
179000      endmon;
179100
179200      jdbc_end_object_group();
179300      return result;
179400  /end-free
179500  P          E
179600
179700
179800      ++++++
179900      * JDBC_GetShort(): Get a short from a called statement
180000      *
180100      *      call = (input) Callable statement to get string from
180200      *      idx = (input) Parameter index to get

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
180300      ++++++
180400      P JDBC_GetShort...
180500      P          B                      export
180600      D JDBC_GetShort...
180700      D          PI                      10I 0
180800      D call                      like(CallableStatement) const
180900      D idx                      10I 0 value
181000      D result                      s          10I 0
181100      D short                      s          like(jShort)
181200      /free
181300          jdbc_begin_object_group(10);
181400
181500          monitor;
181600              short = getShort( call: idx);
181700              if (short = 0);
181800                  result = 0;
181900              else;
182000                  result = short;
182100              endif;
182200          on-error;
182300              result = 0;
182400          endmon;
182500
182600          jdbc_end_object_group();
182700          return result;
182800      /end-free
182900      P          E
183000
183100
183200      ++++++
183300      * JDBC_GetBoolean(): Get a boolean from a called statement
183400      *
183500      * call = (input) Callable statement to get string from
183600      * idx = (input) Parameter index to get
183700      ++++++
183800      P JDBC_GetBoolean...
183900      P          B                      export
184000      D JDBC_GetBoolean...
184100      D          PI                      1N
184200      D call                      like(CallableStatement) const
184300      D idx                      10I 0 value
184400      D result                      s          1N
184500      /free
184600          jdbc_begin_object_group(10);
184700          monitor;
184800              result = getBoolean( call: idx);
184900          on-error;
185000              result = *OFF;
185100          endmon;
185200          jdbc_end_object_group();
185300          return result;
185400      /end-free
185500      P          E

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
185600
185700
185800      ++++++
185900      * JDBC_ExecPrepQry(): Execute a query from a prepared statement
186000      *
186100      *      prep = (input) prepared statement to execute
186200      *
186300      * Returns ResultSet object or *NULL upon failure
186400      ++++++
186500 P JDBC_ExecPrepQry...
186600 P          B                      export
186700 D JDBC_ExecPrepQry...
186800 D          PI                      like(ResultSet)
186900 D      prep                      like(PreparedStatement) const
187000 D      temp                      s                      like(ResultSet)
187100 D      rs                        s                      like(ResultSet)
187200 /free
187300      jdbc_begin_object_group(50);
187400      monitor;
187500          temp = PrepExecuteQuery(prepare);
187600          jdbc_end_object_group(temp: rs);
187700      on-error;
187800          jdbc_end_object_group();
187900          return *null;
188000      endmon;
188100      return rs;
188200 /end-free
188300 P          E
188400
188500
188600      ++++++
188700      * JDBC_ExecPrepUpd(): Execute SQL that doesn't return data
188800      *      from a prepared statement
188900      *
189000      *      prep = (input) prepared statement to execute
189100      *
189200      * Returns 0 or a row count if successful
189300      *      or -1 upon failure
189400      ++++++
189500 P JDBC_ExecPrepUpd...
189600 P          B                      export
189700 D JDBC_ExecPrepUpd...
189800 D          PI                      10I 0
189900 D      prep                      like(PreparedStatement) const
190000 D      rc                        s                      10i 0
190100 /free
190200      jdbc_begin_object_group(50);
190300      monitor;
190400          rc = PrepExecuteUpdate(prepare);
190500      on-error;
190600          rc = -1;
190700      endmon;
190800      jdbc_end_object_group();
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
190900      return rc;
191000      /end-free
191100      P              E
191200
191300
191400      ++++++
191500      * JDBC_ExecCall(): Execute SQL statement that calls a stored
191600      *                  procedure
191700      *
191800      *      call = (input) callable statement to execute
191900      *
192000      * Returns *ON if first result is a ResultSet
192100      *      or *OFF if first result is an update count
192200      *
192300      * Use JDBC_moreResults(), JDBC_getUpdateCount() and
192400      *      JDBC_getResultSet() to get results of this function
192500      ++++++
192600      P JDBC_ExecCall...
192700      P              B                  export
192800      D JDBC_ExecCall...
192900      D              PI                  1N
193000      D      call                  like(CallableStatement) const
193100      D rc              s                  1n
193200      /free
193300      jdbc_begin_object_group(50);
193400      monitor;
193500      rc = CallExecute(call);
193600      on-error;
193700      rc = *OFF;
193800      endmon;
193900      jdbc_end_object_group();
194000      return rc;
194100      /end-free
194200      P              E
194300
194400
194500      ++++++
194600      * JDBC_Execute(): Execute Prepared SQL Statement
194700      *
194800      *      stmt = (input) prepared statement to execute
194900      *
195000      * Returns *ON if first result is a ResultSet
195100      *      or *OFF if first result is an update count
195200      *
195300      * Use JDBC_moreResults(), JDBC_getUpdateCount() and
195400      *      JDBC_getResultSet() to get results of this function
195500      ++++++
195600      P JDBC_Execute...
195700      P              B                  export
195800      D JDBC_Execute...
195900      D              PI                  1N
196000      D      stmt                  like(PreparedStatement) const
196100      D rc              s                  1n
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
196200      /free
196300          jdbc_begin_object_group(50);
196400          monitor;
196500              rc = PrepExecute(stmt);
196600          on-error;
196700              rc = *OFF;
196800          endmon;
196900          return rc;
197000      /end-free
197100  P                      E
197200
197300
197400      ++++++
197500      * JDBC_FreePrepStmt(): Free prepared statement
197600      *
197700      *      prep = (input) Prepared Statement to Free
197800      ++++++
197900  P JDBC_FreePrepStmt...
198000  P          B                      export
198100  D JDBC_FreePrepStmt...
198200  D          PI
198300  D      prep                      like(PreparedStatement)
198400      /free
198500          stmt_close(prepare);
198600          DeleteLocalRef(JNIENV_P: prepare);
198700          prepare = *NULL;
198800      /end-free
198900  P                      E
199000
199100
199200      ++++++
199300      * JDBC_FreeCallStmt(): Free callable statement
199400      *
199500      *      call = (input) Callable Statement to Free
199600      ++++++
199700  P JDBC_FreeCallStmt...
199800  P          B                      export
199900  D JDBC_FreeCallStmt...
200000  D          PI
200100  D      call                      like(CallableStatement)
200200      /free
200300          DeleteLocalRef(JNIENV_P: call);
200400          call = *NULL;
200500      /end-free
200600  P                      E
200700
200800
200900      ++++++
201000      * JDBC_NextRow(): Position result to next row
201100      *
201200      *      rs = (input) Result handle to move
201300      *
201400      * Returns *ON if successful, *OFF otherwise
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
201500      ++++++
201600      P JDBC_NextRow      B      export
201700      D JDBC_NextRow      PI      1N
201800      D      rs      like(ResultSet)
201900      /free
202000      monitor;
202100      return nextRec(rs);
202200      on-error;
202300      return *OFF;
202400      endmon;
202500      /end-free
202600      P      E
202700
202800
202900      ++++++
203000      * JDBC_GetCol(): Get column from current row
203100      *
203200      *      rs = (input) Result handle to use
203300      *      col = (input) column number to retrieve
203400      *      nullInd = (output/optional) will be set to *ON if the
203500      *                  field is null, or *OFF otherwise.
203600      *
203700      * Returns column value if successful, or '' otherwise
203800      ++++++
203900      P JDBC_GetCol      B      export
204000      D JDBC_GetCol      PI      32767A      varying
204100      D      rs      like(ResultSet)
204200      D      col      10I 0      value
204300      D      nullInd      1N      options(*nopass:*omit)
204400      D result      s      32767A      varying
204500      D str      s      like(jstring)
204600      D null      s      1N      inz(*OFF)
204700      /free
204800      jdbc_begin_object_group(5);
204900      monitor;
205000      str = getColString(rs: col);
205100      if (str = *NULL);
205200      result = '';
205300      null = *ON;
205400      else;
205500      result = r(str);
205600      endif;
205700      on-error;
205800      null = *ON;
205900      result = '';
206000      endmon;
206100      jdbc_end_object_group();
206200      if (%parms >= 3 and %addr(nullInd)<>*NULL);
206300      nullInd = Null;
206400      endif;
206500      return result;
206600      /end-free
206700      P      E

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
206800
206900
207000      ++++++
207100      *   JDBC_GetColC():  Get column from current row in Unicode
207200      *
207300      *       rs = (input) Result handle to use
207400      *       col = (input) column number to retrieve
207500      *   nullInd = (output/optional) will be set to *ON if the
207600      *               field is null, or *OFF otherwise.
207700      *
207800      *   Returns column value if successful, or '' otherwise
207900      ++++++
208000      P   JDBC_GetColC      B               export
208100      D   JDBC_GetColC      PI              16383C   varying
208200      D       rs                                like(ResultSet)
208300      D       col                                10I 0   value
208400      D   nullInd                                1N   options(*nopass:*omit)
208500      D result                s              16383c   varying
208600      D str                    s                                like(jstring)
208700      D null                  s              1N       inz(*OFF)
208800      /free
208900          jdbc_begin_object_group(5);
209000          monitor;
209100              str = getColString(rs: col);
209200              if (str = *NULL);
209300                  result = %ucs2('');
209400                  null = *ON;
209500              else;
209600                  result = c(str);
209700              endif;
209800          on-error;
209900              null = *ON;
210000              result = %ucs2('');
210100          endmon;
210200          jdbc_end_object_group();
210300          if (%parms >= 3 and %addr(nullInd)<>*NULL);
210400              nullInd = Null;
210500          endif;
210600          return result;
210700      /end-free
210800      P               E
210900
211000
211100      ++++++
211200      *   JDBC_FreeResult(): Free result handle
211300      *
211400      *       rs = (input) Result handle to free
211500      ++++++
211600      P   JDBC_FreeResult...
211700      P       B               export
211800      D   JDBC_FreeResult...
211900      D       PI
212000      D       rs                                like(ResultSet)

```



SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
212100      D stmt                      s                      like(Statement)
212200      /free
212300          stmt = rs_getStatement(rs);
212400          stmt_close(stmt);
212500          DeleteLocalRef(JNIENV_P: rs);
212600          DeleteLocalRef(JNIENV_P: stmt);
212700          rs = *NULL;
212800      /end-free
212900      P                      E
213000
213100
213200      *+++++
213300      *   JDBC_Commit(): Commit transaction
213400      *
213500      *       conn = (input) Connection to commit on
213600      *+++++
213700      P JDBC_Commit      B                      export
213800      D JDBC_Commit      PI
213900      D   conn                      like(Connection)
214000      /free
214100          commitTrn(conn);
214200      /end-free
214300      P                      E
214400
214500
214600      *+++++
214700      *   JDBC_Rollback(): Rollback transaction
214800      *
214900      *       conn = (input) Connection to rollback on
215000      *+++++
215100      P JDBC_Rollback    B                      export
215200      D JDBC_Rollback    PI
215300      D   conn                      like(Connection)
215400      /free
215500          rollbackTrn(conn);
215600      /end-free
215700      P                      E
215800
215900
216000      *+++++
216100      *   JDBC_Close(): Close connection to server
216200      *                   and free connection handle
216300      *
216400      *       conn = (input) Connection to close
216500      *+++++
216600      P JDBC_Close      B                      export
216700      D JDBC_Close      PI
216800      D   conn                      like(Connection)
216900      /free
217000          closeConn(conn);
217100          DeleteLocalRef(JNIENV_P: conn);
217200          conn = *NULL;
217300      /end-free
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
217400      P                      E
217500
217600
217700      *-----
217800      *  jdbc_get_jni_env():
217900      *
218000      *  Service program utility to get a pointer to the JNI environment
218100      *  you'll need this pointer in order to call many of the JNI
218200      *  routines.
218300      *
218400      *  returns the pointer, or *NULL upon error
218500      *-----
218600      P  jdbc_get_jni_env...
218700      P                      B
218800      D  jdbc_get_jni_env...
218900      D                      PI                      *
219000
219100      D  wwEnv              s                      *
219200
219300      /free
219400          wwEnv = attach_jvm();
219500          if (wwEnv = *NULL);
219600              wwEnv = start_jvm();
219700          endif;
219800
219900          JNIENV_P = wwEnv;
220000          return wwEnv;
220100      /end-free
220200      P                      E
220300
220400
220500      *-----
220600      *  jni_checkError():  Check for an error in JNI routines
220700      *-----
220800      P  jni_checkError  B
220900      D  jni_checkError  PI                      1N
221000      D      errString              256A      varying options(*omit)
221100
221200      D  sleep              pr                      10I 0  extproc('sleep')
221300      D      intv              10I 0  value
221400
221500      D  exc              s                      like(jthrowable)
221600      D  str              s                      like(jstring)
221700      d  msg              s                      256A      varying
221800      /free
221900          if (%addr(errString) <> *NULL);
222000              %len(errString) = 0;
222100          endif;
222200
222300          exc = ExceptionOccurred(JNIENV_P);
222400          if (exc = *NULL);
222500              return *OFF;
222600          endif;

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

SEQNBR\*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0

```

222700
222800      ExceptionDescribe(JNIENV_P);
222900      sleep(10);
223000
223100      str = Exception_getMessage(exc);
223200      msg = r(str);
223300      DeleteLocalRef(JNIENV_P: str);
223400
223500      SndPgmMsg( 'CPF9897'
223600                : 'QCPFMSG    *LIBL'
223700                : msg
223800                : %len(msg)
223900                : '*DIAG'
224000                : '*'
224100                : 0
224200                : MsgKey
224300                : ErrCode );
224400
224500      if (%addr(errString) <> *NULL);
224600          errString = msg;
224700      endif;
224800
224900      ExceptionClear(JNIENV_P);
225000      DeleteLocalRef(JNIENV_P: exc);
225100
225200      return *ON;
225300 /end-free
225400 P          E
225500
225600
225700 *-----
225800 * jdbc_begin_object_group(): Start a new group of objects
225900 *   which will all be freed when jdbc_end_object_group()
226000 *   gets called.
226100 *
226200 *   peCapacity = maximum number of objects that can be
226300 *   referenced within this object group.
226400 *
226500 * NOTE: According to the 1.2 JNI Spec, you can create more
226600 *   objects in the new frame than peCapacity allows. The
226700 *   peCapacity is the guaranteed number. When no object
226800 *   groups are used, 16 references are guaranteed, so if
226900 *   you specify 16 here, that would be comparable to a
227000 *   "default value".
227100 *
227200 * Returns 0 if successful, or -1 upon error
227300 *-----
227400 P jdbc_begin_object_group...
227500 P          B
227600 D jdbc_begin_object_group...
227700 D          PI          10I 0
227800 D   peCapacity          10I 0 value
227900

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
228000      D wwEnv          s          *
228100      D wwRC          s          10I 0
228200
228300      /free
228400
228500      wwEnv = jdbc_get_jni_env();
228600      if (wwEnv = *NULL);
228700          return -1;
228800      endif;
228900
229000      if ( PushLocalFrame (wwEnv: peCapacity) <> JNI_OK );
229100          return -1;
229200      else;
229300          return 0;
229400      endif;
229500
229600      /end-free
229700      P          E
229800
229900
230000      *-----
230100      * jdbc_end_object_group(): Frees all Java objects that
230200      *   have been created since calling jdbc_begin_object_group()
230300      *
230400      *   peOldObj = (see below)
230500      *   peNewObj = Sometimes it's desirable to preserve one
230600      *   object by moving it from the current object group
230700      *   to the parent group. These parameters allow you
230800      *   to make that move.
230900      *
231000      * Returns 0 if successful, or -1 upon error
231100      *-----
231200      P jdbc_end_object_group...
231300      P          B
231400      D jdbc_end_object_group...
231500      D          PI          10I 0
231600      D   peOldObj          like(jObject) const
231700      D          options(*nopass)
231800      D   peNewObj          like(jObject)
231900      D          options(*nopass)
232000
232100      D wwOld          s          like(jObject) inz(*NULL)
232200      D wwNew          s          like(jObject)
232300
232400      /free
232500
232600      jdbc_get_jni_env();
232700      if (JNIENV_p = *NULL);
232800          return -1;
232900      endif;
233000
233100      if %parms >= 2;
233200          wwOld = peOldObj;

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
233300         endif;
233400
233500         wwNew = PopLocalFrame (JNIENV_p: wwOld);
233600
233700         if %parms >= 2;
233800             peNewObj = wwNew;
233900         endif;
234000
234100         return 0;
234200
234300     /end-free
234400 P             E
234500
234600
234700 *-----
234800 *   start_jvm():   Start the Java Virtual Machine (JVM)
234900 *
235000 *   NOTE: Originally, this called JNI routines to start a new JVM,
235100 *           but that meant that a classpath and other options needed
235200 *           to be set manually in the JNI invocation.
235300 *
235400 *           I decided that it would be better to reduce the complexity
235500 *           and let RPG start the JVM, so I merely create & destroy
235600 *           a string here so that RPG will automatically start the
235700 *           JVM for me.
235800 *
235900 *   returns a pointer to the JNI environment
236000 *           or *NULL upon failure.
236100 *-----
236200 P start_jvm      B
236300 D start_jvm      PI          *
236400
236500 D wwStr          s              like(jString)
236600
236700     /free
236800         status('Starting Java Virtual Machine...');
236900         wwStr = s('Temp String');
237000         status('.');
237100         JNIENV_P = attach_jvm();
237200         DeleteLocalRef(JNIENV_P: wwStr);
237300         return JNIENV_P;
237400     /end-free
237500 P             E
237600
237700
237800 *-----
237900 *   attach_jvm():  Attach to JVM if it's running
238000 *
238100 *   Returns a pointer to the JNI environment, or *NULL upon error
238200 *-----
238300 P attach_jvm      B
238400 D attach_jvm      PI          *
238500

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
238600      D dsAtt          ds          likeds(JavaVMAttachArgs)
238700      D wwJVM          s           like(JavaVM_p) dim(1)
238800      D wwJVMc         s           like(jSize)
238900      D wwEnv          s           * inz(*null) static
239000      D wwRC           s           10I 0
239100      /free
239200
239300          if (wwEnv <> *NULL);
239400              return wwEnv;
239500      endif;
239600
239700      status('Attaching RPG program to Java Virtual Machine...');
239800
239900      monitor;
240000          wwRC = JNI_GetCreatedJavaVMs(wwJVM: 1: wwJVMc);
240100
240200          if (wwRC <> JNI_OK or wwJVMc = 0);
240300              return *NULL;
240400          endif;
240500
240600          JVM_P = wwJVM(1);
240700          dsAtt = *ALLx'00';
240800          dsAtt.version = JNI_VERSION_1_2;
240900
241000          wwRC = AttachCurrentThread (wwJVM(1): wwEnv: %addr(dsAtt));
241100          if (wwRC <> JNI_OK);
241200              wwEnv = *NULL;
241300          endif;
241400
241500      on-error;
241600          wwEnv = *NULL;
241700      endmon;
241800
241900      status('.');
242000      return wwEnv;
242100
242200      /end-free
242300      P          E
242400
242500
242600      ++++++
242700      * JNIVersion(): Get version of current JNI environment
242800      ++++++
242900      P JNIVersion      B
243000      D JNIVersion      PI          10P 5
243100      D                  ds
243200      D ver              10I 0
243300      D high              5I 0 overlay(ver:1)
243400      D low               5I 0 overlay(ver:*NEXT)
243500      D result            s          10P 5 static
243600      /free
243700          if (result = 0);
243800              ver = GetVersion(jdbc_get_jni_env());

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
243900      result = high + (low / 100000);
244000      endif;
244100      return result;
244200  /end-free
244300  P          E
244400
244500
244600      ++++++
244700      * Get UTC offset (in seconds) for timezone.
244800      ++++++
244900  P Timezone      B
245000  D Timezone      PI          10I 0
245100
245200  D CEEUTC0      PR
245300  D  hours      10I 0
245400  D  mins      10I 0
245500  D  secs      8F
245600  D  fc        12A  options(*omit)
245700
245800  D utc_offset_set  s          1N  inz(*OFF) static
245900  D utc_offset      s          10I 0          static
246000
246100  D junk1          s          10I 0 static
246200  D junk2          s          10I 0 static
246300  D junk3          s          8F  static
246400
246500  /free
246600
246700      if (utc_offset_set = *OFF);
246800          CEEUTC0(junk1:junk2:junk3:*omit);
246900          utc_offset = junk3;
247000          utc_offset_set = *ON;
247100      endif;
247200      return utc_offset;
247300
247400  /end-free
247500  P          E
247600
247700
247800  P status          B          export
247900  D status          PI          1N
248000  D peMsgTxt      256A  const
248100
248200  D SndPgmMsg      PR          ExtPgm('QMHSNDPM')
248300  D  MessageID      7A  Const
248400  D  QualMsgF      20A  Const
248500  D  MsgData      256A  Const
248600  D  MsgDtaLen      10I 0 Const
248700  D  MsgType      10A  Const
248800  D  CallStkEnt      10A  Const
248900  D  CallStkCnt      10I 0 Const
249000  D  MessageKey      4A
249100  D  ErrorCode      1A

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
249200
249300      D dsEC                DS
249400      D dsECBytesP          1      4I 0 inz(256)
249500      D dsECBytesA          5      8I 0 inz(0)
249600      D dsECMsgID           9      15
249700      D dsECReserv          16     16
249800      D dsECMsgDta          17    256
249900
250000      D wwMsgLen             S      10I 0
250100      D wwTheKey            S      4A
250200
250300      c      ' '            checkr  peMsgTxt      wwMsgLen
250400      c                        if      wwMsgLen<1
250500      c                        return  *OFF
250600      c                        endif
250700
250800      c                        callp    SndPgmMsg('CPF9897': 'QCPFMSG *LIBL':
250900      c                        peMsgTxt: wwMsgLen: '*STATUS':
251000      c                        '*EXT': 0: wwTheKey: dsEC)
251100
251200      c                        if      dsECBytesA>0
251300      c                        return  *off
251400      c                        else
251500      c                        return  *on
251600      c                        endif
251700      P                      E
251800
251900
252000      *+++++
252100      *   JDBC_GetColByName():  Get column from current row by
252200      *                               column name
252300      *
252400      *           rs = (input) Result handle to use
252500      *   columnName = (input) column name to retrieve
252600      *           nullInd = (output/optional) will be set to *ON if the
252700      *                               field is null, or *OFF otherwise.
252800      *
252900      *   Returns column value if successful, or '' otherwise
253000      *+++++
253100      P   JDBC_GetColByName...
253200      P           b                      Export
253300      D   JDBC_GetColByName...
253400      D           pi          32767a    varying
253500      D   rs                      Like(ResultSet)
253600      D   ColumnName          32767a    varying Const options(*varsize)
253700      D   nullInd              1N      options(*nopass:*omit)
253800
253900      D null                    s          1n    inz(*off)
254000      D parm                    s          Like(jstring)
254100      D str                      s          Like(jstring)
254200      D retField                s          32767a
254300
254400      /free

```



SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
254500      if (%len(ColumnNme)=0 or ColumnNme=*blank);
254600          null = *on;
254700          retField = '';
254800      else;
254900          jdbc_begin_object_group(100);
255000          Monitor;
255100              parm = s(ColumnNme);
255200              str = getColStrByName(rs: parm);
255300              if (str = *null);
255400                  null=*on;
255500                  retField = '';
255600              else;
255700                  null=*off;
255800                  retField = r(str);
255900              endif;
256000          On-Error;
256100              retField = '';
256200              null = *on;
256300          Endmon;
256400          jdbc_end_object_group();
256500      Endif;
256600
256700      if (%parms >= 3 and %addr(nullInd)<>*NULL);
256800          nullInd = Null;
256900      endif;
257000
257100      Return retField;
257200  /end-free
257300  P          e
257400
257500
257600  ++++++
257700  *   JDBC_GetColByNameC():  Get column data in Unicode from the
257800  *                           current row by column name.
257900  *
258000  *           rs = (input) Result handle to use
258100  *   columnName = (input) column name to retrieve
258200  *           nullInd = (output/optional) will be set to *ON if the
258300  *                           field is null, or *OFF otherwise.
258400  *
258500  *   Returns column value if successful, or '' otherwise
258600  ++++++
258700  P   JDBC_GetColByNameC...
258800  P           b                      Export
258900  D   JDBC_GetColByNameC...
259000  D           pi          16383c    varying
259100  D   rs
259200  D   columnName          16383c    varying Const options(*varsize)
259300  D   nullInd              1N      options(*nopass:*omit)
259400
259500  D   null                s          1n    inz(*off)
259600  D   parm                s          Like(jstring)
259700  D   str                  s          Like(jstring)

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
259800      D retField          s          16383c
259900
260000      /free
260100          if (%len(ColumnNme)=0 or ColumnNme=*blank);
260200              null = *on;
260300              retField = %ucs2('');
260400          else;
260500              jdbc_begin_object_group(100);
260600              Monitor;
260700                  parm = new_StringC(ColumnNme);
260800                  str = getColStrByName(rs: parm);
260900                  if (str = *null);
261000                      null=*on;
261100                      retField = %ucs2('');
261200                  else;
261300                      null=*off;
261400                      retField = c(str);
261500                  endif;
261600              On-Error;
261700                  retField = %ucs2('');
261800                  null = *on;
261900              Endmon;
262000              jdbc_end_object_group();
262100          Endif;
262200
262300          if (%parms >= 3 and %addr(nullInd)<>*NULL);
262400              nullInd = Null;
262500          endif;
262600
262700          Return retField;
262800      /end-free
262900      P                      e
263000
263100
263200      ++++++
263300      *   JDBC_getUpdateCount(): Get the number of rows modified
263400      *       by the last call to JDBC_ExecCall()
263500      *
263600      *       stmt = (input) PreparedStatement or CallableStatement
263700      *
263800      *   Returns number of modified rows, or -1 if the current
263900      *       result is a ResultSet or if there are no results
264000      *+++++
264100      P JDBC_getUpdateCount...
264200      P          b                      Export
264300      D JDBC_getUpdateCount...
264400      D          pi                      10i 0
264500      D  stmt                      Like(Statement) const
264600      D  getUpdateCount  pr          10i 0  extproc(*JAVA
264700      D                                  : 'java.sql.Statement'
264800      D                                  : 'getUpdatecount')
264900      c          return  getUpdateCount(stmt)
265000      P                      E

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
265100
265200
265300      ++++++
265400      *   JDBC_getResultSet(): Get the first/next ResultSet
265500      *           returned by JDBC_ExecCall()
265600      *
265700      *           stmt = (input) PreparedStatement or CallableStatement
265800      *
265900      *   Returns the ResultSet or *NULL if the current result is
266000      *           not a ResultSet or if there are no more results
266100      ++++++
266200 P   JDBC_getResultSet...
266300 P           b                               Export
266400 D   JDBC_getResultSet...
266500 D           pi                               like(ResultSet)
266600 D   stmt                               Like(Statement) const
266700 D   getResultSet      pr               like(ResultSet)
266800 D                                           extproc(*JAVA
266900 D                                           : 'java.sql.Statement'
267000 D                                           : 'getResultSet')
267100 c           return      getResultSet(stmt)
267200 P           E
267300
267400
267500      ++++++
267600      *   JDBC_getMoreResults(): Move the results of this statement
267700      *           to the next result returned.
267800      *
267900      *           stmt = (input) PreparedStatement or CallableStatement
268000      *
268100      *   Returns *ON if the next result is a ResultSet object,
268200      *           or *OFF if it's an update count or there are no
268300      *           more results
268400      ++++++
268500 P   JDBC_getMoreResults...
268600 P           b                               Export
268700 D   JDBC_getMoreResults...
268800 D           pi                               ln
268900 D   stmt                               Like(Statement) const
269000 D   getMoreResults    pr               ln       extproc(*JAVA
269100 D                                           : 'java.sql.Statement'
269200 D                                           : 'getMoreResults')
269300 c           return      getMoreResults(stmt)
269400 P           E
269500
269600
269700      /if defined(BLOB_SUPPORT)
269800
269900      ++++++
270000      *   JDBC_getBlob(): Get BLOB column from current row
270100      *
270200      *           rs = (input) Result handle to use
270300      *           col = (input) column number to retrieve
```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
270400      * nullInd = (output/optional) will be set to *ON if the
270500      *           field is null, or *OFF otherwise.
270600      *
270700      * Returns column value if successful, or '' otherwise
270800      *+++++
270900      P JDBC_getBlob      B      export
271000      D JDBC_getBlob      PI      32767A  varying
271100      D rs                  like(ResultSet)
271200      D col                  10I 0  value
271300      D nullInd            1N      options(*nopass:*omit)
271400
271500      D blob_length          PR      20i 0  extproc(*java
271600      D                      : 'java.sql.Blob'
271700      D                      : 'length')
271800
271900
272000      D result                s      32767A  varying
272100      D str                   s      like(jstring)
272200      D null                  s      1N      inz(*OFF)
272300      /free
272400          jdbc_begin_object_group(5);
272500          monitor;
272600              str = getColString(rs: col);
272700              if (str = *NULL);
272800                  result = '';
272900                  null = *ON;
273000              else;
273100                  result = r(str);
273200              endif;
273300          on-error;
273400              null = *ON;
273500              result = '';
273600          endmon;
273700          jdbc_end_object_group();
273800          if (%parms >= 3 and %addr(nullInd)<>*NULL);
273900              nullInd = Null;
274000          endif;
274100          return result;
274200      /end-free
274300      P                      E
274400
274500
274600      *+++++
274700      * blob_getBytes():  Calls the JNI functions needed to
274800      * retrieve a raw string of bytes containing the BLOB
274900      * data.
275000      *
275100      * obj = (input) Blob object to get contents of
275200      * pos = (input) starting position to retrieve from blob
275300      * len = (input) number of bytes to extract
275400      *
275500      * returns pointer to teraspace memory containing the blob
275600      * (freshly allocated -- must be freed separately.)

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```

SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
275700      *          or *null upon failure
275800      *+++++
275900      P blob_getBytes      B
276000      D blob_getBytes      PI          *
276100      D obj                like(Blob)
276200      D pos                20i 0 value
276300      D len                10i 0 value
276400
276500      D QDCXLATE            PR          ExtPgm('QDCXLATE')
276600      D len                5P 0 const
276700      D data                32702A options(*varsize)
276800      D table               10A const
276900
277000      D jni_getBytes        PR          LIKE(jobject)
277100      D                    EXTPROC(*CWIDEN
277200      D                    : JNINativeInterface.
277300      D                    CallObjectMethod_P)
277400      D env                 LIKE(JNIEnv_P) VALUE
277500      D obj                 LIKE(jobject) VALUE
277600      D methodID            LIKE(jmethodID) VALUE
277700      D pos                 LIKE(jlong) VALUE
277800      D                    options(*nopass)
277900      D length              LIKE(jint) VALUE
278000      D                    options(*nopass)
278100
278200      D BLOB_CLASS            C          x'6a6176612f73716c2f426c6f62'
278300      D NAME_GETBYTES        C          x'6765744279746573'
278400      D SIG_GETBYTES          C          x'284a49295b42'
278500
278600      D env                  s          * static inz(*NULL)
278700      D msg                   s          256A varying
278800      D cls                   s          like(jclass)
278900      D mid                   s          like(jmethodid)
279000      /free
279100          if (env = *NULL);
279200              env = jdbc_get_jni_env();
279300          endif;
279400
279500          if (cls = *null);
279600              cls = FindClass(env: BLOB_CLASS);
279700              if (jni_CheckError(*omit));
279800                  return *NULL;
279900              endif;
280000          endif;
280100
280200          if ( cls = *NULL );
280300              msg = 'Class java.sql.Blob not found!';
280400              exsr DiagMsg;
280500              return *NULL;
280600          endif;
280700
280800          if (mid = *null);
280900              mid = GetMethodID(env: cls: NAME_GETBYTES: SIG_GETBYTES);

```

SOURCE FILE . . . . . CDJDBC/QRPGLESRC

MEMBER . . . . . JDBCR4

```
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
281000      if (jni_CheckError(*omit));
281100          return *NULL;
281200      endif;
281300  endif;
281400
281500      if ( mid = *NULL );
281600          msg = 'getBytes() method for java.sql.Blob'
281700              + ' not found!';
281800          exsr DiagMsg;
281900          return *NULL;
282000      endif;
282100
282200      arr = jni_getBytes( env: obj: mid: pos: len);
282300      if (jni_CheckError(*omit));
282400          msg = 'error calling jni_getBytes routine';
282500          exsr DiagMsg;
282600          return *NULL;
282700      endif;
282800
282900      buf = getByteArrayElements( env: arr: isCopy );
283000      if (jni_CheckError(*omit));
283100          msg = 'error calling getByteArrayElements()';
283200          return *NULL;
283300      endif;
283400
283500      ret = %alloc(len);
283600      memcpy(ret: buf: len);
283700
283800      ReleaseByteArrayElements( JNIENV_P: arr: buf: 0 );
283900      DeleteLocalRef(JNIENV_P: arr);
284000
284100      return ret;
284200
284300      begsr DiagMsg;
284400          SndPgmMsg( 'CPF9897'
284500                  : 'QCPFMSG *LIBL'
284600                  : msg
284700                  : %len(msg)
284800                  : '*DIAG'
284900                  : '*'
285000                  : 0
285100                  : MsgKey
285200                  : ErrCode );
285300      endsr;
285400  /end-free
285500  P      E
285600  /endif
```

\* \* \* \* E N D O F S O U R C E \* \* \* \*