4D For PostgreSQL

Reference Guide
Windows® and Mac OS® Versions

4th Dimension®

4D For PostgreSQL
Version 2003 and 2004 for Windows® and Mac OS®

Contents

1

4D For PostgreSQL

This documentation covers 4D For PostgreSQL version 2003 and version 2004.

4D For PostgreSQL 2003 requires the following minimum configurations:

- 4D/4D Server 2003.X
- Mac OS 9.2 or 10.1 and higher / Windows 98 SE and higher
- PostgreSQL 7.2.X and higher (including native Windows 8.0.X PostgreSQL server)

4D For PostgreSQL 2004 requires the following minimum configurations:

- 4D/4D Server 2004.X
- Mac OS X 10.3.X (Panther) and higher / Windows 2000 and higher
- PostgreSQL 7.2.X and higher (including native Windows 8.0.X PostgreSQL server)

PGSQL_Connect (host ; port; options; tty; database; user; password) → Longint

Parameter	Type		Description
host	String	→	Host address
port	String	→	Port number
options	String	→	Connection options
tty	String	→	File or tty to get debug output from back end
database	String	→	Database name
user	String	→	User name
password	String	→	Password
Function result	Longint	←	Connection ID

Description

The PGSQL_Connect command connects to the specified database and returns a connection ID (Longint) that is used during the connection session.

host is a String that corresponds to the PostgreSQL server's address (IP or DNS).

port is a String that corresponds to the port used by the PostgreSQL server to listen at (default is 5432).

options is a String that corresponds to the PostgreSQL connection options (please see the PostgreSQL documentation).

tty is a String that corresponds to the tty optional output options (please see the PostgreSQL documentation).

database is a String that corresponds to the PostgreSQL database name you want to connect to.

user is a String that corresponds to the PostgreSQL user.

password is a String that corresponds to the password used by the PostgreSQL user.

Example

```
    `Init
$Host:="192.168.0.1"
$Tty:=" "
$Options:=" "
$Database:="Manager"
$User:="Doug"
$Password:="r74d96gRfx"
    `Connection
⇒   PGConnID:=PGSQL_Connect($Host;$Port;$Options;$Tty;$Database;$User;$Password)
    If(PGConnID#0)
        `Successful connection
        ...
    Else
        `PGConnID is null, we have a connection error
        ...
    End if
```

PGSQL_Close (connID)

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect

Description

The PGSQL_Close command closes a regular connection session with the connection ID that was returned by the PGSQL_Connect command.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

PGSQL_Execute (connID; sqlQuery) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
sqlQuery	Text	→	SQL statement
Function result	Longint	←	Result ID

Description

The PGSQL_Execute command executes the SQL statement and returns a result ID (Longint) that is used with all commands that works on the query result.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

sqlQuery is a Text parameter that contains the SQL statement you want to execute.

Examples

(1) Inserting a record

```
      `Connection
   PGConnID:=PGSQL_Connect($Host;$Port;$Options;$Tty;$Database;$User;$Password)
   If(PGConnID#0)
      $Query:="INSERT INTO clients (id_clt, name_clt, email_clt, country_clt) VALUES "
⇒     $resID:=PGSQL_Execute(PGConnID;$Query)
      If($resID#0)
            `Get our Execute result status
         $rsError:=PGSQL_GetResultStatus($resID;$resultStatus)
            `We are performing an INSERT, so we are waiting for a
                                                PGRES_COMMAND_OK result status
         If($resultStatus="PGRES_COMMAND_OK")
            `Our INSERT has been executed
         Else
               `We didn't get the expected result status
            ALERT($resultStatus)
         End if
         PGSQL_CloseResult($resID)
```

```
    Else
        `$resID is null, we have a result error
        ALERT ("null result pointer")
    End if
    PGSQL_Close(PGConnID)
Else
    ALERT(PGSQL_GetLastConnError(PGConnID))
End if
```

(2) Executing a SELECT

```
    ` Connection
PGConnID:=PGSQL_Connect($Host;$Port;$Options;$Tty;$Database;$User;$Password)
If(PGConnID#0)
    $Query:="SELECT id_clt, name_clt, email_clt, country_clt FROM clients"
⇒   $resID:=PGSQL_Execute(PGConnID;$Query)
    If($resID#0)
        ` Get our Execute result status
        $rsError:=PGSQL_GetResultStatus($resID;$resultStatus)
        ` We are performing a SELECT, so we are waiting for a PGRES_TUPLES_OK
                                                                    result status
        If($resultStatus="PGRES_TUPLES_OK")
            ` Our SELECT has been executed
            ` Populate Results
            $err:=PGSQL_GetRowCount($resID;$nbRows)
            For($r;0;$nbRows-1)
                $NewPos:=Size of array(ArrLON_ClientID)+1
                $err:=PGSQL_GetLongintValue($resID;0;$r;$id_clt)
                INSERT ELEMENT(arrLON_ClientID;$NewPos;1)
                arrLON_ClientID{$NewPos}:=$id_clt
                $err:=PGSQL_GetStringValue($resID;1;$r;$name_clt)
                INSERT ELEMENT(arrSTR_Name;$NewPos;1)
                arrSTR_Name{$NewPos}:=$name_clt
                $err:=PGSQL_GetStringValue($resID;2;$r;$email_clt)
                INSERT ELEMENT (arrSTR_Email;$NewPos;1)
                arrSTR_Email{$NewPos}:=$email_clt
                $err:=PGSQL_GetStringValue($resID;3;$r;$country_clt)
                INSERT ELEMENT (arrSTR_Country;$NewPos;1)
                arrSTR_Country{$NewPos}:=$country_clt
            End for
        Else
            `We didn't get the expected result status
            `Catch error here
        End if
        PGSQL_CloseResult($resID)
```

```
   Else
         `$resID is null, we have a result error
         `Catch error here
   End if
   PGSQL_Close(PGConnID)
Else
      `PGConnID is null, we have a connection error
   ALERT(PGSQL_GetLastConnError(PGConnID))
End if
```

PGSQL_GetFieldCount (resID; fieldCount) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
fieldCount	Integer	←	Number of fields in the result
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetFieldCount command gives the number of fields that are found in the result set returned by PGSQL_Execute. It returns an error code (Longint) where 0 means no error.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

fieldCount is an integer that corresponds to the number of fields in the result.

PGSQL_GetFieldName (resID; fieldIndex; fieldName) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
fieldIndex	Integer	→	Field index in the result set (0 based)
fieldName	String	←	Field name
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetFieldName command gives the name of the field specified by the index of this field in the result set returned by PGSQL_Execute. It returns an error code (Longint) where 0 means no error.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

fieldIndex is an Integer that corresponds to the field index in the result set (0 based).

fieldName is a String that is returned which contains the field name.

PGSQL_GetRowCount (resID; rowCount) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowCount	Longint	←	Number of rows in the result set
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetRowCount command gives the number of rows present in the result set (for example, following a SQL SELECT statement) returned by PGSQL_Execute. It returns an error code (Longint) where 0 means no error.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowCount is a Longint that is returned which corresponds to the number of rows in the result set.

PGSQL_GetFieldIndex (resID; fieldName; fieldIndex) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
fieldName	String	→	Field name
fieldIndex	Integer	←	Field index in the result set
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetFieldIndex command gives the index of the fieldName field in the result set returned by PGSQL_Execute. It returns an error code (Longint) where 0 means no error. fieldIndex is 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

fieldName is a String that corresponds to the field name.

fieldIndex is an Integer that is returned which corresponds to the field index in the result set.

PGSQL_GetFieldType (resID; fieldIndex; fieldType) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
fieldIndex	Integer	→	Field index in the result set
fieldType	Integer	←	Field type
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetFieldType command gives the type of the field at the fieldIndex index in the result set returned by PGSQL_Execute. It returns an error code (Longint) where 0 means no error. fieldIndex is 0 based.

resID is a longint that corresponds to the result ID returned by PGSQL_Execute.

fieldIndex is an Integer that corresponds to the field index in the result set (0 based).

fieldType is an integer that is returned which corresponds to the field type. See the PostgreSQL documentation to find out which data type corresponds to the integer returned.

PGSQL_GetFieldMod (resID; fieldIndex; fieldMod) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
fieldIndex	Integer	→	Field index in the result set
fieldMod	Integer	←	Type-specific modification data
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetFieldMod command gives the type-specific modification data of the field associated with the given field index. It returns an error code (Longint) where 0 means no error. fieldIndex is 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

fieldIndex is an Integer that corresponds to the field index in the result set (0 based).

fieldMod is an integer that is returned which corresponds to the type-specific modification data.

PGSQL_GetFieldSize (resID; fieldIndex; fieldSize) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
fieldIndex	Integer	→	Field index in the result set
fieldSize	Integer	←	Field size
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetFieldSize command gives the space allocated for this field in a database tuple; in other words, the size of the server's binary representation of the data type. -1 is returned if the field is variable size. It returns an error code (Longint) where 0 means no error. fieldIndex is 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

fieldIndex is an Integer that corresponds to the field index in the result set (0 based).

fieldSize is an integer that is returned which corresponds to the field size.

PGSQL_IsBinaryTuples (resID; isBinary) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
isBinary	Integer	←	Binary or ASCII data
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_IsBinaryTuples command returns 1 in the isBinary parameter if the result set contains binary tuple data and 0 if it contains ASCII data. It returns an error code (Longint) where 0 means no error.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

isBinary is an Integer that is returned which corresponds to the Binary or ASCII data (binary = 1, Ascii = 0).

PGSQL_IsFieldDataNull (resID; rowIndex; fieldIndex; isNull) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
isNull	Integer	←	Contains null
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_IsFieldDataNull command gives 1 if the field contains a NULL; 0 if it contains a non-null value. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the row index (0 based).

fieldIndex is an Integer that corresponds to the Field index (0 based).

isNull is an Integer that is returned which corresponds to the null value (1 = contains null or 0 = contains a non-null value).

PGSQL_GetFieldLength (resID; rowIndex; fieldIndex; fieldLength) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
fieldLength	Integer	←	Field length
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetFieldLength command gives the actual data length for the particular data value; that is, the size of the object pointed to by the PGSQL_Get[TYPE]Value command . Note that for character-represented values, this size has little to do with the binary size reported by PGSQL_GetFieldSize. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the row index (0 based).

fieldIndex is an Integer that corresponds to the field index (0 based).

fieldLength is an Integer that is returned which corresponds to the field length in bytes.

PGSQL_GetStringValue (resID; rowIndex; fieldIndex; stringValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
stringValue	String	←	Value returned (String)
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetStringValue command gives the field value of the string type for a field index, at a row index. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the row index (0 based).

fieldIndex is an Integer that corresponds to the field index (0 based).

stringValue is a String that is returned which corresponds to the String value.

Example

```
C_STRING($myString)

$err:=PGSQL_GetStringValue(resID;3;$i;$myString)
```

PGSQL_GetIntegerValue (resID; rowIndex; fieldIndex; integerValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
integerValue	Integer	←	Value returned (Integer)
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetIntegerValue command gives the field value of the integer type for a field index, at a row index. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the row index (0 based).

fieldIndex is an Integer that corresponds to the field index (0 based).

integerValue is an Integer that is returned which corresponds to the Integer value.

PGSQL_GetLongintValue (resID; rowIndex; fieldIndex; longintValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
longintValue	Longint	←	Value returned (Longint)
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetLongintValue command gives the field value of the Longint type for a field index, at a row index. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the row index (0 based).

fieldIndex is an Integer that corresponds to the field index (0 based).

longintValue is a Longint that is returned which corresponds to the Longint value.

PGSQL_GetRealValue (resID; rowIndex; fieldIndex; realValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
realValue	Real	←	Value returned (Real)
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetRealValue command gives the field value of the Real type for a field index, at a row index. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the row index (0 based).

fieldIndex is an Integer that corresponds to the field index (0 based).

realValue is a Real that is returned which corresponds to the Real value.

PGSQL_GetDateValue (resID; rowIndex; fieldIndex; dateValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
dateValue	Date	←	Value returned (Date)
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetDateValue command gives the field value of the Date type for a field index, at a row index. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the row index (0 based).

fieldIndex is an Integer that corresponds to the field index (0 based).

dateValue is a Date that is returned which corresponds to the Date value.

PGSQL_GetTimeValue (resID; rowIndex; fieldIndex; hourValue; minValue; secValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
hourValue	Integer	←	Hour part of the time
minValue	Integer	←	Minutes part of the time
secValue	Integer	←	Seconds part of the time
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetTimeValue command gives the field value (hour, minutes, seconds) for a field index, at a row index. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the row index (0 based).

fieldIndex is an Integer that corresponds to the field index (0 based).

hourValue is an Integer that corresponds to the hour part of the time.

minValue is an Integer that corresponds to the minutes part of the time.

secValue is an Integer that corresponds to the seconds part of the time.

PGSQL_GetTextValue (resID; rowIndex; fieldIndex; textValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
textValue	Text	←	Value returned (Text)
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetTextValue command gives the field value of the Text type for a field index, at a row index. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the Row index (0 based).

fieldIndex is an Integer that corresponds to the Field index (0 based).

textValue is a Text that is returned which corresponds to the Text value.

PGSQL_GetBooleanValue (resID; rowIndex; fieldIndex; booleanValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
rowIndex	Longint	→	Row index
fieldIndex	Integer	→	Field index
booleanValue	Integer	←	Value returned (0 is False/1 is True)
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetBooleanValue command gives the field value of the Boolean type for a field index, at a row index. It returns an error code (Longint) where 0 means no error. fieldIndex and rowIndex are 0 based.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

rowIndex is a Longint that corresponds to the Row index (0 based).

fieldIndex is an Integer that corresponds to the Field index (0 based).

booleanValue is an Integer that corresponds to the value returned (0 is False/ 1 is True).

PGSQL_GetAffectedRows (resID; affectedRows) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
affectedRows	Longint	←	Number of rows affected by query execution
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetAffectedRows command gives the number of rows which have been affected by the last query execution (for example, DELETE sql statement). It returns an error code (Longint) where 0 means no error.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

affectedRows is a Longint that is returned which corresponds to the number of rows affected by the query execution.

PGSQL_GetLastCmdStatus (resID; lastCmdStatus) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
lastCmdStatus	String	←	Status of the last command executed
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetLastCmdStatus command gives the status (String) of the last command executed. It returns an error code (Longint) where 0 means no error.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

lastCmdStatus is a String that is returned which corresponds to the status of the last command executed.

PGSQL_GetResultStatus (resID; resultStatus) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
resultStatus	String	←	Status of the result
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetResultStatus command gives the result status of the command. It returns an error code (Longint) where 0 means no error.

resultStatus can return one of the following values:
PGRES_EMPTY_QUERY -- The string sent to the backend was empty.
PGRES_COMMAND_OK -- Successful completion of a command returning no data.
PGRES_TUPLES_OK -- The query successfully executed.
PGRES_COPY_OUT -- Copy Out (from server) data transfer started.
PGRES_COPY_IN -- Copy In (to server) data transfer started.
PGRES_BAD_RESPONSE -- The server's response was not understood.
PGRES_NONFATAL_ERROR
PGRES_FATAL_ERROR

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

resultStatus is a String that is returned which corresponds to the status of the result.

PGSQL_GetResultErrorMess (resID; errorMess) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
errorMess	String	←	Error message
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetResultErrorMess command gives the error message of a command that fails to execute. It returns an error code (Longint) where 0 means no error.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

errorMess is a String that is returned which corresponds to the error message.

PGSQL_CloseResult (resID)

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute

Description

The PGSQL_CloseResult command closes the result set returned by PGSQL_Execute. You must use it once you no longer need to deal with a result set.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

PGSQL_ResetConn (connID)

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect

Description

The PGSQL_ResetConn command performs a new connection with the same connection parameters used in the last PGSQL_Connect.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

PGSQL_GetDbName (connID; dbName) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
dbName	String	←	Current database name
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetDbName command gives the name of the current database. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

dbName is a String that is returned which corresponds to the current database name.

PGSQL_GetUserName (connID; userName) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
userName	String	←	Current user name
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetUserName command gives the name of the current user. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

userName is a String that is returned which corresponds to the current user name.

PGSQL_GetPassword (connID; password) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
password	String	→	Current password
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetPassword command gives the name of the current password. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

password is a String that is returned which contains the current password.

PGSQL_GetHostname (connID; hostName) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
hostName	String	←	Current host name
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetHostname command gives the name of the host. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

hostName is a String that is returned which contains the current host name.

PGSQL_GetPortNum (connID; portNum) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
portNum	Integer	←	Current port number
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetPortNum command gives the current port number used. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

portNum is a String that is returned which contains the current port number.

PGSQL_GetOptions (connID; options) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
options	String	←	Current connection options
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetOptions command gives the current connection options. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

options is a String that is returned which contains the current connection options.

PGSQL_GetConnStatus (connID) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
Function result	Longint	←	Current connection status

Description

The PGSQL_GetConnStatus command returns the current connection status (Longint):
- 0 = CONNECTION_OK.
- 1 = CONNECTION_BAD.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

PGSQL_GetLastConnError (connID) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
Function result	Longint	←	Last connection error message

Description

The PGSQL_GetLastConnError command returns the last connection error message (String).

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

PGSQL_GetOIDValue (resID; fieldIndex; rowIndex; objectValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
fieldIndex	Integer	→	Field index
rowIndex	Longint	→	Row index
objectValue	Longint	←	Object ID value
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetOIDValue command gets the Large Object Id corresponding to the field index and row index passed as parameters. The OID value (basically a Longint) is returned in the value parameter and can be used in other Large Object routines. It returns an error code (Longint) where 0 means no error.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

fieldIndex is an Integer that corresponds to the field index (0 based).

rowIndex is a Longint that corresponds to the row index (0 based).

objectValue is a Longint that is returned which corresponds to the Object ID value.

PGSQL_GetByteaValue (resID; fieldIndex; rowIndex; byteaValue) → Longint

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
fieldIndex	Integer	→	Field index
rowIndex	Longint	→	Row index
byteaValue	Blob	←	Bytea value as a 4D Blob
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetByteaValue command gets the Bytea value corresponding to the field index and row index passed as parameters. The Bytea value (basically a binary data) is returned as a 4D Blob datatype. It returns an error code (Longint) where 0 means no error.

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

fieldIndex is an Integer that corresponds to the field index.

rowIndex is a Longint that corresponds to the row index.

byteaValue is a Blob that is returned which contains the Bytea value as a 4D Blob.

PGSQL_EscapeBytea (byteaToEscape; escapedBytea)

Parameter	Type		Description
byteaToEscape	Blob	→	Pure Bytea value
escapedBytea	Blob	←	Escaped Bytea value

Description

The PGSQL_EscapeBytea command passes a 4D Blob that contains pure Bytea value and returns another 4D Blob containing an escaped version of the Bytea value. This way the returned value can be used in a PGSQL_Execute function.

byteaToEscape is a Blob that corresponds to the pure Bytea value.

escapedBytea is a Blob that is returned which contains the escaped Bytea value.

PGSQL_UnescapeBytea (byteaToUnescape; unescapedBytea)

Parameter	Type		Description
byteaToUnescape	Blob	→	Escaped Bytea value
unescapedBytea	Blob	←	Pure Bytea value

Description

The PGSQL_UnescapeBytea command passes a 4D Blob that contains an unescaped Bytea value and returns another 4D Blob containing a pure Bytea value.

byteaToUnescape is a Blob that corresponds to the escaped Bytea value.

unescapedBytea is a Blob that is returned which contains the pure Bytea value.

PGSQL_CreateLOB (connID; mode) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
mode	Integer	→	Rights mode
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_CreateLOB command creates an empty large object and returns its objectID. Codes for modes are as follows:
0 – Read and Write
1 – Read only
2 – Write only

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

mode is an Integer that corresponds to the rights mode.

PGSQL_ImportLOB (connID; filePath; objectID) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
filePath	String	→	File path of the doc you want to import
objectID	Longint	←	Large object reference
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_ImportLOB command imports the document data referenced by filePath and returns the Object ID that references the LOB in the Large Object System table. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

filePath is a String that corresponds to the file path of the doc you want to import.

objectID is a Longint that is returned which corresponds to the large object reference.

PGSQL_ExportLOB (connID; filePath; objectID) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
filePath	String	→	File path of the doc you want to import
objectID	Longint	←	Large object reference
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_ExportLOB command exports the binary data referenced by objectID in the file indicated by filePath. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

filePath is a String that corresponds to the file path of the doc you want to import.

objectID is a Longint that corresponds to the large object reference.

PGSQL_GetLOBData (connID; objectID; lobData) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
objectID	Longint	→	Large object reference
lobData	Blob	←	Blob returned
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_GetLOBData command gets the binary data referenced by objectID in the lobData (4D Blob) parameter. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

objectID is a Longint that corresponds to the Large object reference.

lobData contains the Blob that is returned.

PGSQL_SetLOBData (connID; objectID; lobData) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
objectID	Longint	→	Large object reference
lobData	Blob	→	Inserted Blob
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_SetLOBData command sets the lobData (4D Blob) into the Large Object referenced by objectID. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

objectID is a Longint that corresponds to the Large object reference.

lobData is a Blob that contains the Blob to set.

PGSQL_RemoveLOB (connID; objectID) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
objectID	Longint	→	Large object reference
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_RemoveLOB command removes the Large Object referenced by objectID from the Large Objects System Table. It returns an error code (Longint) where 0 means no error.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

objectID is a Longint that corresponds to the large object reference.

PGSQL_Register (serialNum) → Longint

Parameter	Type		Description
serialNum	String	→	Serial number
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_Register command validates your serial number. It returns 0 for a valid number and 10999 for an invalid number (in which case you switch to demo mode). This routine will be changed at a later date in order to take Windows serial numbers into account. It returns an error code (Longint) where 0 means no error.

serialNum is a String that contains the Serial number.

PGSQL_ExecuteWithBlob (connID; blobedQuery) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by PGSQL_Connect
blobedQuery	Blob	→	SQL or PL/pgSQL Query As blob
Function result	Longint	←	Error code (where 0 means no error)

Description

The PGSQL_ExecuteWithBlob command executes a SQL query passed in a Blob parameter. The main interest of this routine compared to PGSQLExecute is that using a Blob, you can pass queries that contain more than 32 KB of data. If you don't expect to use queries longer than 32 KB, use PGSQL_Execute instead. Returns an error code (Longint) that is used with all commands that work on the query result.

connID is a Longint that corresponds to the connection ID returned by PGSQL_Connect.

blobedQuery is a Blob that corresponds to the SQL or PL/pgSQL Query As Blob.

PGSQL_GetValues (resID; fieldIndex; columnArray; startRow; rowCount)

Parameter	Type		Description
resID	Longint	→	Result ID returned by PGSQL_Execute
fieldIndex	Integer	→	Field index
columnArray	Array	←	Array
startRow	Longint	→	Row number from which column data will be returned (default is 0)
		←	Actual row number from which data is returned
rowCount	Longint	→	Maximum number of rows expected
		←	Actual number of rows returned

Description

The PGSQL_GetValues command executes a bind between a PostgreSQL column data in a database cursor (selection) and a 4D array. You can detemine the maximum returned rows or from which row number the cursor will start to return data. The PGSQL_GetValues mechanism is 40 times faster than the usual process (in a 4D loop).

resID is a Longint that corresponds to the result ID returned by PGSQL_Execute.

fieldIndex the field index in the database cursor.

columnArray is a 4D Array (Integer, Longint, Real, Date, String, Text, Boolean) that will be bound with the corresponding column data.

In the startRow parameter, you pass the starting row number from which column data will be returned (default is 0). This parameter then returns the actual starting row number from which column data has been returned.

In the rowCount parameter, you pass the maximum number of rows you expect to be returned in the column array. This parameter then returns the actual number of rows returned in the column array.

Command Index

C

E

G

I

R

S

U