

# PG\_CONNECT

## PG\_CONNECT action

Function

Declaration

Parameters

Return code

Description

The function establishes the connection to a table.

```
PG_CONNECT dbObjIdent, accessIdent_Int, handleIdent_Int, pgSizeIdent_Int,
whereIdent_Str, orderByIdent_Str, rowCountIdent_Int, retCodeIdent_Int
[TRANS transHandle_Int] [MODE mode_Int] [ORAHINT hintIdent_Str]
```

|                   |        |   |
|-------------------|--------|---|
| dbObjIdent        | in     | Reference to an object of Database table type.  |
| accessIdent_Int   | in     | Value identifier: required access to the table (_DB_READ, _DB_MODIFY).  |
| handleIdent_Int   | output | Identifier for an unique number (handle) of the connection to the table.  |
| pgSizeIdent_Int   | in     | Identifier to determine maximal page size.  |
| whereIdent_Str    | in     | Identifier of Text type. Condition to select from the table.  |
| orderByIdent_Str  | in     | Identifier of Text type. Sorting in the final selection.  |
| rowCountIdent_Int | output | Identifier of Int type. The number of rows in the selection.  |
| retCodeIdent_Int  | output | Return code identifier.   |
| transHandle_Int   | in     | Identifier for an unique number (handle) for the connection to the database.  |
| mode_Int          | in     | Identifier of Int type - 0/1.   |
| hintIdent_Str     | in     | Expression of String type that defines Oracle SQL hint. It is used as an instruction for the performance optimizer of SQL command. The value is used without the opening and terminating characters /*+ <orahint> */.<br>The example is mentioned <a href="#">here</a> .<br><br><b>Note:</b> In this situation, Oracle SQL hint is used for ESL action <a href="#">PG_READ</a> , because in ESL action <a href="#">SQL_CONNECT</a> SELECT expression is compounded for next called ESL action <a href="#">PG_READ</a> . |

The value of the parameter *transHandle\_Int*. See the table of [error codes](#). It is possible to get [extended error information](#).

Page access is initialized by the action **PG\_CONNECT**. It performs the table opening, during which it is possible to specify conditions for the final selection (reduction of visible rows in the table). Selection is limited by the condition with the value *whereIdent\_Str*. This text (a value of the identifier *whereIdent\_Str*) is used in the clause **WHERE** in the SQL command.

The action allows to define a way to sort rows in the final selection by means of a value of the identifier *orderByIdent\_Str*. The value of the identifier is used in the clause ORDER BY in the SQL command.

The required page size is defined by a value of the identifier *pgSizeIdent\_Int*. This page size is to be used during reading from the table (whole page is read at the same time). A larger page may cause the big memory block allocation, and thereby to cause the malfunction of the operating system. Maximum number of returned rows is limited by a Database configuration parameter [Maximum returned rows](#).

After the successful opening a table (*retCodeIdent\_Int* = \_ERR\_NO\_ERROR), the identifier *rowCountIdent\_Int* contains the number of rows in the final selection and the identifier *handleIdent\_Int* contains the number of the connection (handle).

If the parameter *mode\_Int* = 1, the pages can be read only sequentially from 1 to maximal page. To move back is disabled. In this case, DbManager will use optimal way of reading from database. If parameter is not use, the value = 0.

Having finished a work with a table, it is necessary to close it using the action [PG\\_DISCONNECT](#).

**Example**

Work with a database table (actions PG\_ ...).

**Related topics**

[DB\\_TRANS\\_OPEN](#)  
[DB\\_TRANS\\_COMMIT](#)  
[DB\\_TRANS\\_ROLLBACK](#)  
[DB\\_TRANS\\_CLOSE](#)

[PG\\_DISCONNECT](#)  
[PG\\_READ](#)  
[PG\\_INSERT](#)  
[PG\\_DELETE](#)  
[PG\\_UPDATE](#)

[All database related actions](#)

---

**Related pages:**

[Script actions](#)