

# 4D For ADO

---

*Reference Guide*  
*Windows® and Mac OS® Versions*



---

## **4D For ADO**

**Version 2003 and 2004 for Windows® and Mac OS®**

*Copyright © 2006 4D SA / 4D, Inc.  
All rights reserved*

---

The Software described in this manual is governed by the grant of license in the 4D Product Line License Agreement provided with the Software in this package. The Software, this manual, and all documentation included with the Software are copyrighted and may not be reproduced in whole or in part except for in accordance with the 4D Product Line License Agreement.

4D Write, 4D Draw, 4D View, 4th Dimension, 4D, the 4D logo and 4D Server are registered trademarks of 4D SA.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Apple, Macintosh, Mac OS and QuickTime are trademarks or registered trademarks of Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2006, is a product of Altura Software, Inc. This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

4th Dimension includes cryptographic software written by Eric Young (eay@cryptsoft.com)

4th Dimension includes software written by Tim Hudson (tjh@cryptsoft.com).

ACROBAT © Copyright 1987-2006, Secret Commercial Adobe Systems Inc. All rights reserved. ACROBAT is a registered trademark of Adobe Systems Inc.

All other referenced trade names are trademarks or registered trademarks of their respective holders.

# Contents

## 1. 4D For ADO.....5

Preface.....	7
DBGateway_Connect.....	9
DBGateway_Close.....	10
DBGateway_ErrorCode.....	11
DBGateway_ErrorString.....	12
DBGateway_Select.....	13
DBGateway_Execute.....	14
DBGateway_FieldCount.....	15
DBGateway_FieldName.....	16
DBGateway_NextRow.....	17
DBGateway_GetStringField.....	18
DBGateway_GetTextField.....	19
DBGateway_GetIntegerField.....	20
DBGateway_GetRealField.....	21
DBGateway_GetLongIntField.....	22
DBGateway_GetPictureField.....	23
DBGateway_GetBlobField.....	24
DBGateway_GetDateField.....	25
DBGateway_GetTimeField.....	26
DBGateway_CloseSelect.....	27
DBGateway_AddStringParameter.....	28
DBGateway_AddIntegerParameter.....	29
DBGateway_AddTextParameter.....	30
DBGateway_AddLongIntParameter.....	31
DBGateway_AddRealParameter.....	32
DBGateway_AddDateParameter.....	33
DBGateway_AddPictureParameter.....	34
DBGateway_AddBlobParameter.....	35

## Command Index.....37



# 1

---

## 4D For ADO



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

**4D For ADO 2003** requires the following minimum configurations:

- 4D/4D Server 2003.X
- Mac OS 9.2 or 10.1 and higher
- Windows 98 SE or higher (DBGateway.exe service)
- MDAC 2.6 or higher (Windows that hosts DBGateway.exe service) (<http://msdn.microsoft.com/data/mdac/default.aspx>)

**4D For ADO 2004** requires the following minimum configurations:

- 4D/4D Server 2004.X
- Mac OS X 10.2.X or higher
- Windows 2000 or higher (DBGateway.exe service)
- MDAC 2.6 or higher (Windows that hosts DBGateway.exe service) (<http://msdn.microsoft.com/data/mdac/default.aspx>)

4D for ADO organizes an original technical architecture to open 4D applications (Windows and Mac OS) to the largest choice of connectivity options:

- Through OLEDB:

Active Directory Service  
Advantage  
AS/400 (from IBM)  
AS/400 and VSAM (from Microsoft)  
Commerce Server  
DB2  
DTS Packages  
Exchange  
Excel  
Internet Publishing  
Index Server  
Microsoft Jet  
Microsoft Project  
MySQL  
ODBC Databases  
OLAP Services  
Oracle (from Microsoft)  
Oracle (from Oracle)  
Pervasive  
Simple Provider

SQLBase  
SQL Server  
SQL Server via SQLXMLEDB  
Sybase Adaptive Server Anywhere  
Sybase Adaptive Server Enterprise  
Text Files  
UniData and UniVerse  
Visual FoxPro

- Through ODBC drivers:

Access  
AS/400  
dBASE  
Excel  
Informix  
Interbase (from Easysoft)  
Interbase (from InterSolv)  
Lotus Notes  
Mimer  
MySQL  
Oracle (from Microsoft)  
Oracle (from Oracle)  
Paradox  
SQL Server  
Sybase  
Sybase SQL Anywhere  
Text  
Teradata  
Visual FoxPro

#### **4D for ADO data transport model:**

4D for ADO uses an efficient, internal TCP/IP based protocol to provide fast communication between the 4D for ADO plugin and the DBGateway.exe service which can be placed server side or on any Windows box in the network. DBGateway handles fast communication from your 4D application (in the local network or via Internet), then takes care of ADO communication with the target data sources.



---

DBGateway\_Connect (host; connStr) → Longint

Parameter	Type		Description
host	String	→	DNS Server name or IP Number
connStr	String	→	ADO connection string
Function result	Longint	←	Connection ID that is used by other calls

### Description

The DBGateway\_Connect command returns a Longint containing the connection ID used by other calls. The address passed to the function for the server location supports both DNS names, as well as numeric (x.x.x.x) IP addresses.

host is a String containing the DNS server name or IP number.

connStr is an ADO connection string that contains the details necessary to select and connect to the database. If you're accessing your database via ODBC, you can set up a system DSN in the ODBC control panel, and use a connection string of:

```
"Provider=MSDASQL;DSN=DSNName".
```

If you want to connect to a specific Jet database file, and you don't want to set up a DSN, then use:

```
"Provider=MSDASQL;Driver={Microsoft Access Driver (*.mdb)};
DBQ=c:\home\biblio.mdb".
```

Using ADO means many ways to connect to many data sources (including Windows services, RDBMS, etc.) but determining the right "Connect String" (depending on the target data source and the driver used) can be tricky. Here is a link to a website that centralizes ADO compliants concerning "Connect Strings":

<http://www.carlprothman.net/Default.aspx?tabid=81>

DBGateway\_Close (connID)

<b>Parameter</b>	<b>Type</b>		<b>Description</b>
connID	Longint	→	Connection ID returned by DBGateway_Connect

### **Description**

The DBGateway\_Close command closes the connection session referenced by the connID parameter.

connID is a Longint returned by DBGateway\_Connect.

DBGateway\_ErrorCode (connID) → Longint

<b>Parameter</b>	<b>Type</b>		<b>Description</b>
connID	Longint	→	Connection ID returned by DBGateway_Connect
Function result	Longint	←	Error code (where 0 means no error)

**Description**

The DBGateway\_ErrorCode command returns the code of the last error that occurred in the connection session referenced by the connID parameter. If no error occurred, the command returns 0.

connID is a Longint returned by DBGateway\_Connect.

DBGateway\_ErrorString (connID) → String

Parameter	Type		Description
connID	Longint	→	Connection ID returned by DBGateway_Connect
Function result	String	←	Error message returned by DB Engine

### Description

The DBGateway\_ErrorString command returns the last error message that occurred in the connection session referenced by the connID parameter. The String returned represents the ADO error message returned by the DB engine (where an empty string means no error has occurred).

connID is a Longint returned by DBGateway\_Connect.

DBGateway\_Select (connID; sqlQuery) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by DBGateway_Connect
sqlQuery	Text	→	SQL Query (SELECT..., SHOW..., DESCRIBE...)
Function result	Longint	←	Select ID that is used by selection-related methods

### Description

The DBGateway\_Select command handles SQL statements that return data (such as "SELECT \* FROM myTable"). It returns a Longint representing the resultant data set.

connID is a Longint returned by DB\_Gateway\_Connect.

sqlQuery is a Text variable that contains the SQL statement.

### Example

```

$connectStr:="Provider=MSDASQL;Driver={Microsoft Access Driver (*.mdb)};
DBQ=c:\home\biblio.mdb"
connID:=DBGateway_Connect("192.168.0.12" ; $connectStr )
If(connID#0)
⇒   sellID:=DBGateway_Select("SELECT * FROM myTable ")
    If(sellID#0)
        `Successful, handle your resultant data set here
        ...
        DBGateway_CloseSelect(sellID)
    Else
        ALERT(DBGateway_ErrorString(connID))
    End if

    DBGateway_Close(connID)
Else
    ALERT("Connection failed")
End if

```

---

DBGateway\_Execute (connID; sqlQuery)

Parameter	Type		Description
connID	Longint	→	Connection ID returned by DBGateway_Connect
sqlQuery	Text	→	SQL Query (INSERT..., UPDATE..., CREATE TABLE...)

### Description

The DBGateway\_Execute command executes a SQL statement that is not supposed to return a data set (inserting records, updating records, deleting records, etc.).

connID is a Longint returned by DBGateway\_Connect.

sqlQuery is a Text variable that contains the SQL statement.

### Example

```
⇒   DBGateway_Execute(connID; "INSERT INTO mydata (id, name, amount) VALUES
                                     (1275, 'SOFT CIE', 990.99)")
    If(DBGateway_ErrorCode(connID)#0)
      ALERT(DBGateway_ErrorString(connID))
    Else
      `Record inserted
    ...
    End if
```

DBGateway\_FieldCount (selectID) → Integer

Parameter	Type		Description
selectID	Longint	→	Select ID returned by DBGateway_Select
Function result	Integer	←	Number of fields in selection cursor

### Description

The DBGateway\_FieldCount command retrieves the number of fields contained in the result set referenced by selectID.

selectID is a Longint returned by DBGateway\_Select.

DBGateway\_FieldName (selectID; fieldIndex; fieldName)

<b>Parameter</b>	<b>Type</b>		<b>Description</b>
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
fieldName	String	←	Name of the field

**Description**

The DBGateway\_FieldName command retrieves the column name for a particular field.

selectID is a Longint returned by DBGateway\_Select.

fieldIndex is an Integer that specifies the index position of the field in the table.

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



DBGateway\_NextRow (selectID) → Integer

Parameter	Type	Description
selectID	Longint	→ Select ID returned by DBGateway_Select
Function result	Integer	← 1 if moved to next row; 0 if there is no next row

### Description

The DBGateway\_NextRow command advances through the rows of the result. It also needs to be called to move onto the first row of the result (if it exists).

The function returns 1 if it successfully moved to the next row and 0 if no more rows exist in the result.

selectID is a Longint returned from DBGateway\_Select.

Normally, you would call the method with a while loop, like:

```
While (0# DBGateway_NextRow(SelectID))
  ...do stuff with content of the row...
End While
```

### Example

```
selID:= DBGateway_Select(connID;"SELECT num, name, amount FROM clients WHERE
                                             last_order_date > '2005-05-31'")
If(selID#0)
⇒  While(0# DBGateway_NextRow(selID))
      DBGateway_GetIntegerField(selID;0;$clientNum)
      DBGateway_GetStringField(selID;1;$clientName)
      DBGateway_GetRealField(selID;2;$clientAmount)
    End While
Else
  `handle error here
End if
```

DBGateway\_GetStringField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
fieldValue	String	←	String value of the field

### Description

The DBGateway\_GetStringField command retrieves the String value of the field passed in the fieldIndex parameter.

selectID is a Longint returned from DBGateway\_Select.

fieldIndex is an Integer that specifies the index position of the field in the select statement.

fieldValue is a String variable where the value of the field is returned.

DBGateway\_GetTextField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
fieldValue	Text	←	Text value of the field

### Description

The DBGateway\_GetTextField command retrieves the Text value of the field passed in the fieldIndex parameter.

selectID is a Longint returned from DBGateway\_Select.

fieldIndex is an Integer that specifies the index position of the field in the select statement.

fieldValue is a Text variable where the value of the field is returned.

DBGateway\_GetIntegerField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
fieldValue	Integer	←	Integer value of the field

## Description

The DBGateway\_GetIntegerField command retrieves the Integer value of the field passed in the fieldIndex parameter.

selectID is a Longint returned from DBGateway\_Select.

fieldIndex is an Integer that specifies the index position of the field in the select statement.

fieldValue is an Integer variable where the value of the field is returned.

DBGateway\_GetRealField (selectID; fieldIndex; fieldValue)

<b>Parameter</b>	<b>Type</b>		<b>Description</b>
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
fieldValue	Real	←	Real value of the field

**Description**

The DBGateway\_GetRealField command retrieves the Real value of the field passed in the fieldIndex parameter.

selectID is a Longint returned from DBGateway\_Select.

fieldIndex is an Integer that specifies the index position of the field in the select statement.

fieldValue is a Real variable where the value of the field is returned.

DBGateway\_GetLongIntField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
fieldValue	Longint	←	Longint value of the field

### Description

The DBGateway\_GetLongIntField command retrieves the Longint value of the field passed in the fieldIndex parameter.

selectID is a Longint returned from DBGateway\_Select.

fieldIndex is an Integer that specifies the index position of the field in the select statement.

fieldValue is a Longint variable where the value of the field is returned.

DBGateway\_GetPictureField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
fieldValue	Picture	←	Picture value of the field

### Description

The DBGateway\_GetPictureField command retrieves the Picture value of the field passed in the fieldIndex parameter.

selectID is a Longint returned from DBGateway\_Select.

fieldIndex is an Integer that specifies the index position of the field in the select statement.

fieldValue is a Picture variable where the value of the field is returned.

DBGateway\_GetBlobField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
fieldValue	Blob	←	Blob value of the field

### Description

The DBGateway\_GetBlobField command retrieves the Blob value of the field passed in the fieldIndex parameter.

selectID is a Longint returned from DBGateway\_Select.

fieldIndex is an Integer that specifies the index position of the field in the select statement.

fieldValue is a Blob variable where the value of the field is returned.



DBGateway\_GetDateField (selectID; fieldIndex; day; month; year)

<b>Parameter</b>	<b>Type</b>		<b>Description</b>
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
day	Integer	←	Day value of the date field
month	Integer	←	Month value of the date field
year	Integer	←	Year value of the date field

**Description**

The DBGateway\_GetDateField command retrieves the day, month and year values of the field passed in the fieldIndex parameter.

selectID is a Longint returned from DBGateway\_Select.

fieldIndex is an Integer that specifies the index position of the field in the select statement.

day is an Integer containing the value of the day component of the date.

month is an Integer containing value of the month component of the date.

year is an Integer containing the value of the year component of the date.

DBGateway\_GetTimeField (selectID; fieldIndex; hour; minutes; seconds)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by DBGateway_Select
fieldIndex	Integer	→	Index number of the field
hour	Integer	←	Hour value of the time field
minutes	Integer	←	Minutes value of the time field
seconds	Integer	←	Seconds value of the time field

### Description

The DBGateway\_GetTimeField command retrieves the hour, minutes and seconds values of the field passed in the fieldIndex parameter.

selectID is a Longint returned from DBGateway\_Select.

fieldIndex is a Integer that specifies the index position of the field in the select statement.

hour is an Integer containing value of the hour component of the time.

minutes is an Integer containing the value of the minutes component of the time.

seconds is an Integer containing the value of the seconds component of the time.

DBGateway\_CloseSelect (selectID)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by DBGateway_Select

### Description

The DBGateway\_CloseSelect command closes the result set (DB cursor) opened by DBGateway\_Select.

selectID is a longint returned by DBGateway\_Select.

### Example

```

selectID := DBGateway_Select(connID;"SELECT company_client FROM clients WHERE
                                                    ref_client = 725")
If(selectID # 0)
    DBGateway_GetStringField(selectID;0;$compClient)
⇒    DBGateway_CloseSelect(selectID)
Else
    ALERT (DBGateway_ErrorString)
End if
    
```

DBGateway\_AddStringParameter (connID; tagIndex; paramValue)

Parameter	Type	Description
connID	Longint	→ Connection ID returned by DBGateway_Connect
tagIndex	String	→ Tag which will be replaced in the SQL query
paramValue	String	→ String value to be substituted for the tag

### Description

The DBGateway\_AddStringParameter command replaces a tag (tagIndex) with a String (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by DBGateway\_Connect.

tagIndex is a String in the Select or Execute statement that will be replaced by paramValue at the time of execution.

paramValue is the String value that will replace tagIndex at the time of execution.

### Example

```
⇒ DBGateway_AddStringParameter(connID;"%1";"My string")  
DBGateway_Execute(connID,"INSERT INTO test(field1, field2, field3) VALUES  
(%1, %2, %3))
```

Note that this call to DBGateway\_AddStringParameter will also add quotes around the string and add SQL escape sequences for handling values with quotes.

---

DBGateway\_AddIntegerParameter (connID; tagIndex; paramValue)

Parameter	Type	Description
connID	Longint	→ Connection ID returned by DBGateway_Connect
tagIndex	String	→ Tag which will be replaced in the SQL query
paramValue	Integer	→ Integer value to substituted for the tag

### Description

The DBGateway\_AddIntegerParameter command replaces a tag (tagIndex) with an Integer (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by DBGateway\_Connect.

tagIndex is a String in the Select or Execute statement that will be replaced by paramValue at the time of execution.

paramValue is the Integer value that will replace the tagIndex at the time of execution.

### Example

⇒ *DBGateway\_AddIntegerParameter*(connID;"%2";5)  
*DBGateway\_Execute*(connID,"INSERT INTO test(field1, field2, field3) VALUES  
(%1, %2, %3))

DBGateway\_AddTextParameter (connID; tagIndex; paramValue)

<b>Parameter</b>	<b>Type</b>		<b>Description</b>
connID	Longint	→	Connection ID returned by DBGateway_Connect
tagIndex	String	→	Tag which will be replaced in the SQL query
paramValue	Text	→	Text value to be substituted for the tag

**Description**

The DBGateway\_AddTextParameter command replaces a tag (tagIndex) with a Text (paramValue) that you can use from your SQL query instead of the value itself.

connID is a longint returned by DBGateway\_Connect.

tagIndex is a String in the Select or Execute statement that will be replaced by paramValue at the time of execution.

paramValue is the Text value that will replace tagIndex at the time of execution.

Note that this call to DBGateway\_AddTextParameter will also add quotes around the text and add SQL escape sequences for handling values with quotes.

---

DBGateway\_AddLongIntParameter (connID; tagIndex; paramValue)

Parameter	Type	Description
connID	Longint	→ Connection ID returned by DBGateway_Connect
tagIndex	String	→ Tag which will be replaced in the SQL query
paramValue	Longint	→ Longint value to substituted for the tag

### Description

The DBGateway\_AddLongIntParameter command replaces a tag (tagIndex) with a Longint (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by DBGateway\_Connect.

tagIndex is a String in the Select or Execute statement that will be replaced by paramValue at the time of execution.

paramValue is the Longint value that will replace tagIndex at the time of execution.

### Example

```
⇒ DBGateway_AddLongIntParameter(connID;"%1";128225)
   DBGateway_Execute(connID,"INSERT INTO test(field1, field2, field3) VALUES
                                                                (%1, %2, %3))
```

---

DBGateway\_AddRealParameter (connID; tagIndex; paramValue)

Parameter	Type		Description
connID	Longint	→	Connection ID returned by DBGateway_Connect
tagIndex	String	→	Tag which will be replaced in the SQL query
paramValue	Real	→	Real value to substituted for the tag

### Description

The DBGateway\_AddRealParameter command replaces a tag (tagIndex) with a Real (paramValue) that you can use from your SQL query instead of the value itself itself.

connID is a Longint returned by DBGateway\_Connect.

tagIndex is a String in the Select or Execute statement that will be replaced by paramValue at the time of execution.

paramValue is the Real value that will replace tagIndex at the time of execution.

### Example

```
⇒ DBGateway_AddRealParameter(connID;"%3";5.95)
   DBGateway_Execute(connID,"INSERT INTO test(field1, field2, field3) VALUES
                                                                (%1, %2, %3))
```



DBGateway\_AddDateParameter (connID; tagIndex; paramValue)

<b>Parameter</b>	<b>Type</b>		<b>Description</b>
connID	Longint	→	Connection ID returned by
DBGateway_Connect			
tagIndex	String	→	Tag which will be replaced in the SQL query
paramValue	Date	→	Date value to substituted for the tag

**Description**

The DBGateway\_AddDateParameter command replaces a tag (tagIndex) with a Date (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by DBGateway\_Connect.

tagIndex s a String in the Select or Execute statement that will be replaced by paramValue at the time of execution.

paramValue is the Date value that will replace tagIndex at the time of execution.

Note that this call to DBGateway\_AddDateParameter takes care of formatting your 4D Date value to a SQL Date value.

---

DBGateway\_AddPictureParameter (connID; tagIndex; paramValue)

Parameter	Type	Description
connID	Longint	→ Connection ID returned by DBGateway_Connect
tagIndex	String	→ Tag which will be replaced in the SQL query
paramValue	Picture	→ Picture value to substituted for the tag

### Description

The DBGateway\_AddPictureParameter command replaces a tag (tagIndex) with a Picture (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by DBGateway\_Connect.

tagIndex is a String in the Select or Execute statement that will be replaced by paramValue at the time of execution.

paramValue is the Picture value that will replace tagIndex at the time of execution.

### Example

```
⇒ DBGateway_AddPictureParameter(connID;"%3";$myPict)
   DBGateway_Execute(connID,"INSERT INTO test(field1, field2, field3) VALUES
                                     (%1, %2, %3))
```

DBGateway\_AddBlobParameter (connID; tagIndex; paramValue)

<b>Parameter</b>	<b>Type</b>		<b>Description</b>
connID	Longint	→	Connection ID returned by DBGateway_Connect
tagIndex	String	→	Tag which will be replaced in the SQL query
paramValue	Blob	→	Blob value to substituted for the tag

**Description**

The DBGateway\_AddBlobParameter command replaces a tag (tagIndex) with a Blob (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by DBGateway\_Connect.

tagIndex is a String in the Select or Execute statement that will be replaced by paramValue at the time of execution.

paramValue is the Blob value that will replace tagIndex at the time of execution.



# Command Index

## A

DBGateway_AddBlobParameter.....	35
DBGateway_AddDateParameter.....	33
DBGateway_AddIntegerParameter.....	29
DBGateway_AddLongIntParameter.....	31
DBGateway_AddPictureParameter.....	34
DBGateway_AddRealParameter.....	32
DBGateway_AddStringParameter.....	28
DBGateway_AddTextParameter.....	30

## C

DBGateway_Close.....	10
DBGateway_CloseSelect.....	27
DBGateway_Connect.....	9

## E

DBGateway_ErrorCode.....	11
DBGateway_ErrorString.....	12
DBGateway_Execute.....	14

## F

DBGateway_FieldCount.....	15
DBGateway_FieldName.....	16

## G

DBGateway_GetBlobField.....	24
DBGateway_GetDateField.....	25
DBGateway_GetIntegerField.....	20
DBGateway_GetLongIntField.....	22
DBGateway_GetPictureField.....	23

DBGateway_GetRealField.....	21
DBGateway_GetStringField.....	18
DBGateway_GetTextField.....	19
DBGateway_GetTimeField.....	26

## N

DBGateway_NextRow.....	17
------------------------	----

## S

DBGateway_Select.....	13
-----------------------	----