4D For MySQL

Reference Guide
Windows® and Mac OS® Versions

4th Dimension®

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

Contents

1

4D For MySQL

Preface

4D For MySQL

version 2003/2004

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

4D For MySQL 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
- MySQL 3.23.XX or higher (Unix, Linux, Windows, Mac OS X)

4D For MySQL 2004 requires the following minimum configurations:

- 4D/4D Server 2004.X
- Mac OS 10.2.X or higher / Windows 2000 or higher
- MySQL 3.23.XX or higher (Unix, Linux, Windows, Mac OS X)

MySQL_Connect (server; database; user; password) → Longint

Parameter	Type		Description
server	String	→	IP address of MySQL server
database	String	→	Individual MySQL database to connect to
user	String	→	User name
password	String	→	Password
Function result	Longint	←	Connection ID

Description

The MySQL_Connect command returns a Longint that can be passed to any MySQLConnect for 4D methods that expect a connection ID. MySQL_Connect returns 0 when the connection could not be established. Use MySQL_LastConnectFailure to get the reason for this failure.

Once you are done with the connection to the MySQL server, you can call MySQL_Close to indicate you are done with the connection.

server is a String that specifies the server to connect to either using a DNS address or the x.x.x.x form.

database is a String containing the name of the database you want to connect to.

user is a String containing the user name.

password is a String containing the password.

Example

```
⇒	connID:=MySQL_Connect("192.168.0.21";"MyDB";"MyLogin";"MyPass")
⇒	connID:=MySQL_Connect("mydomain:3308";"MyDB";"MyLogin";"MyPass")
```

Note: When connecting to 4.1.X and 5.0.X MySQL databases servers:
Execute the following SQL statement from MySQL tool or another MySQL Client:
“SET PASSWORD FOR ‘some_user’@’some_host’ = OLD_PASSWORD(‘mypassw’)”.

MySQL_LastConnectFailure

4D For MySQL

version 2004

MySQL_LastConnectFailure → String

Parameter	Type		Description
This command does not require any parameters			
Function result	String	←	Reason for connection failure

Description

The MySQL_LastConnectFailure command returns a String corresponding to the reason for the connection failure.

MySQL_Close

4D For MySQL

version 2004

MySQL_Close (connID)

Parameter	Type		Description
connID	Longint	→	Connection ID returned by MySQL_Connect

Description

The MySQL_Close command receives the connection ID (Longint) returned from MySQL_Connect.

MySQL_AddIntegerParameter

MySQL_AddIntegerParameter (connID; key; paramValue)

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

Description

The MySQL_AddIntegerParameter command replaces a key with an Integer (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by MySQL_Connect.

key 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 key at the time of execution.

Example

```
⇒   MySQL_AddIntegerParameter(connID;"%2";5)
    MySQL_Execute(connID,"INSERT INTO test(field1, field2, field3) VALUES (%1, %2, %3))
```

MySQL_AddLongint Parameter (connID; key; paramValue)

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

Description

The MySQL_AddLongint Parameter command replaces a key with a Longint (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by MySQL_Connect.

key 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 key at the time of execution.

Example

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

MySQL_AddRealParameter

MySQL_AddRealParameter (connID; key; paramValue)

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

Description

The MySQL_AddRealParameter command replaces a key with a Real (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by MySQL_Connect.

key 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 key at the time of execution.

Example

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

MySQL_AddStringParameter

4D For MySQL

version 2004

MySQL_AddStringParameter (connID; key; paramValue)

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

Description

The MySQL_AddStringParameter command replaces a key with a String (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by MySQL_Connect.

key 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 key at the time of execution.

Example

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

Note that this call to MySQL_AddStringParameter will also add quotes around the string, and add MySQL escape sequences for handling values with quotes.

MySQL_AddTextParameter

MySQL_AddTextParameter (connID; key; paramValue)

Parameter	Type		Description
connID	Longint	→	Connection ID returned by MySQL_Connect
key	String	→	String which will be replaced in the SQL query
paramValue	Text	→	Text value to be substituted for the key

Description

The MySQL_AddTextParameter command replaces a key with a Text (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by MySQL_Connect.

key 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 key at the time of execution.

Note that this call to MySQL_AddTextParameter will also add quotes around the text, and add MySQL escape sequences for handling values with quotes.

MySQL_AddDateParameter

MySQL_AddDateParameter (connID; key; paramValue)

Parameter	Type		Description
connID	Longint	→	Connection ID returned by MySQL_Connect
key	String	→	String which will be replaced in the SQL query
paramValue	Date	→	Date value to be substituted for the key

Description

The MySQL_AddDateParameter command replaces a key with a Date (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by MySQL_Connect.

key is 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 key at the time of execution.

Note that this call to MySQL_AddDateParameter takes care of formatting your 4D Date value to an SQL Date value.

MySQL_AddPictureParameter

MySQL_AddPictureParameter (connID; key; paramValue)

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

Description

The MySQL_AddPictureParameter command replaces a key with a Picture (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by MySQL_Connect.

key 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 key at the time of execution.

Example

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

MySQL_AddBlobParameter

MySQL_AddBlobParameter (connID; key; paramValue)

Parameter	Type		Description
connID	Longint	→	Connection ID returned by MySQL_Connect
key	String	→	String which will be replaced in the SQL query
paramValue	Blob	→	Blob value to be substituted for the key

Description

The MySQL_AddBlobParameter command replaces a key with a Blob (paramValue) that you can use from your SQL query instead of the value itself.

connID is a Longint returned by MySQL_Connect.

key 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 key at the time of execution.

MySQL_Select

MySQL_Select (connID; query) → Longint

Parameter	Type		Description
connID	Longint	→	Connection ID returned by MySQL_Connect
query	Text	→	SQL statement to execute
Function result	Longint	←	Select ID

Description

The MySQL_Select command executes every SQL statement that is supposed to return a set of data, such as “SELECT * FROM mytable”, “SHOW DATABASES”, “SHOW TABLES”, etc. The returned Longint identifies the cursor that contains the results of the MySQL_Select call. This value is used to retrieve the contents of the Select query.

If the method returns 0, then an error occurred and the error details can be retrieved with MySQL_ErrorCode/MySQL_ErrorString.

If you want to execute an SQL statement that doesn't return a cursor, you must use the MySQL_Execute method.

connID is a Longint returned from MySQL_Connect.

query is a text introducing the SQL statement (e.g. “SHOW TABLES FROM mydb”).

Examples

(1) This is an example of the basic use of MySQL_Select:

```
⇒	selID:=MySQL_Select(connID;"SELECT * FROM clients WHERE zip_code = '90210'")
```

(2) This is an example of a complete Select session:

```
	`Start Connection
connID:=MySQL_Connect($Host;$Database;$User;$Password)
If(connID#0)
		`Introduce your query here
	$Query:="SELECT id_clt, name_clt, email_clt, country_clt FROM clients"
```

```
            `Execute query
⇒       selectID:=MySQL_Select(connID;$Query)
        If(SelectID#0)
                `Get query result
            While(0#MySQL_NextRow(selectID))
                $NewPos:=Size of array(arrLON_ClientID)
                MySQL_GetLongIntField(selectID;0;$id_clt)
                INSERT ELEMENT(arrLON_ClientID;$NewPos;1)
                arrLON_ClientID{$NewPos}:=$id_clt
                MySQL_GetStringField(selectID;1;$name_clt)
                INSERT ELEMENT(arrSTR_Name;$NewPos;1)
                arrSTR_Name{$NewPos}:=$name_clt
                MySQL_GetStringField(selectID;1;$email_clt)
                INSERT ELEMENT(arrSTR_Email;$NewPos;1)
                arrSTR_Email{$NewPos}:=$email_clt
            End while
                `Close Select
            MySQL_CloseSelect(selectID)
        Else
            ALERT(MySQL_ErrorString(connID))
        End if
            `Connection close
        MySQL_Close(connID)
    Else
            `Getting the last Connect Failure
        ALERT(MySQL_LastConnectFailure)
    End if
```

MySQL_Execute (connID; statement)

Parameter	Type		Description
connID	Longint	→	Connection ID returned by MySQL_Connect
statement	Text	→	SQL statement to execute

Description

The MySQL_Execute command executes SQL statements.

connID is a Longint returned from MySQL_Connect.

statement is a text introducing the SQL statement (e.g. “INSERT INTO mytable (name, date) VALUES (‘Doe’, ‘2005-06-10’)”).

Call the MySQL_ErrorCode method to determine whether the method succeeded or failed. If it returns 0, then the SQL statement executed successfully.

Examples

(1) This example shows a basic use of this command:

```
⇒    MySQL_Execute(connID;"UPDATE clients SET good_one='Yes' WHERE zip_code='90210'")
```

(2) This example inserts variable values into the Clients column:

```
     MySQL_AddIntegerParameter(connID;"%1";4578)
     MySQL_AddStringParameter(connID;"%2";"Boeing Corp")
     MySQL_AddRealParameter(connID;"%3";95798.90)
     MySQL_AddTextParameter(connID;"%4";"It’s a test")

⇒    MySQL_Execute(connID;"INSERT INTO clients (num, name, amount, comment) VALUES
                                                              (%1, %2, %3, %4)")
     If(MySQL_ErrorCode(connID)#0)
        ALERT(MySQL_ErrorString(connID))
     Else
        `Statement executed
     End if
```

MySQL_ErrorCode

MySQL_ErrorCode (connID) → Longint

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

Description

The MySQL_ErrorCode command returns the MySQL error code for the last call to MySQL_Select/MySQL_Execute. It returns 0 if the call executed successfully.

connID is a Longint returned from MySQL_Connect.

Example

```
   MySQL_Execute(connID;"UPDATE clients SET good_one='Yes' WHERE zip_code='90210'")
⇒  $err:=MySQL_ErrorCode(connID)
   If($err#0)
      ALERT("Error # "+String($err))
   End if
```

MySQL_ErrorString

MySQL_ErrorString (connID) → String

Parameter	Type		Description
connID	Longint	→	Connection ID returned by MySQL_Connect
Function result	String	←	Error message string

Description

The MySQL_ErrorString command returns the MySQL error string for the last call to MySQL_Select/MySQL_Execute. It returns an empty string if the call executed successfully.

connID is a Longint returned from MySQL_Connect.

Example

```
	MySQL_Execute(connID;"UPDATE clients SET good_one='Yes' WHERE zip_code='90210'")
	$err:=MySQL_ErrorCode(connID)
	If($err#0)
⇒		$errStr:=MySQL_ErrorString(connID)
		ALERT("Error:  "+$errStr)
	End if
```

MySQL_FieldCount

MySQL_FieldCount (selectID) → Integer

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
Function result	Integer	←	Number of fields

Description

The MySQL_FieldCount command retrieves the number of fields contained in the result set.

selectID is a Longint returned by MySQL_Select.

The Integer returned indicates the number of fields.

MySQL_GetFieldName

MySQL_GetFieldName (selectID; fieldIndex; fieldName)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
fieldName	String	←	Field name

Description

The MySQL_GetFieldName command retrieves the column name for a particular field.

selectID is a Longint returned by MySQL_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.

MySQL_NextRow

MySQL_NextRow (selectID) → Integer

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
Function result	Integer	←	1 if moved to next row; 0 if no more rows exist

Description

The MySQL_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; 0 if no more rows exist in the result.

selectID is a Longint returned by MySQL_Select.

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

```
While (0#MySQL_NextRow(selectID))
   `...do stuff with content of the row...
End while
```

Example

```
 selID:=MySQL_Select(sonnID;"SELECT num, name, amount FROM clients WHERE
                                                    last_order_date > '2005-05-31'"
If(selID#0)
⇒    While(0#MySQL_NextRow(selID))
        MySQL_GetIntegerField(selID;0;$clientNum)
        MySQL_GetStringField(selID;1;$clientName)
        MySQL_GetRealField(selID;2;$clientAmount)
     End while
Else
   `handle error here
End if
```

MySQL_GetStringField

MySQL_GetStringField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
fieldValue	String	←	Value of the field

Description

The MySQL_GetStringField command returns the value of the field referenced by fieldIndex.

selectID is a Longint returned by MySQL_Select.

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

fieldValue returns a String variable containing the value of the field.

MySQL_GetDateField (selectID; fieldIndex; day; month; year)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
day	Integer	←	Value of the day component of the date
month	Integer	←	Value of the month component of the date
year	Integer	←	Value of the year component of the date

Description

The MySQL_GetDateField command returns the day, month and year values of the field referenced by fieldIndex.

selectID is a Longint returned by MySQL_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 the value of the month component of the date.

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

MySQL_GetIntegerField

MySQL_GetIntegerField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
fieldValue	Integer	←	Value of the field

Description

The MySQL_GetIntegerField command returns the value of the field referenced by fieldIndex.

selectID is a Longint returned by MySQL_Select.

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

fieldValue returns an Integer variable containing the value of the field.

MySQL_GetLongIntField

MySQL_GetLongIntField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
fieldValue	Integer	←	Value of the field

Description

The MySQL_GetLongIntField command returns the value of the field referenced by fieldIndex.

selectID is a Longint returned by MySQL_Select.

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

fieldValue returns a Longint variable containing the value of the field.

MySQL_GetTextField

4D For MySQL

version 2004

MySQL_GetTextField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
fieldValue	Text	←	Value of the field

Description

The MySQL_GetTextField command returns the value of the field referenced by fieldIndex.

selectID is a Longint returned by MySQL_Select.

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

fieldValue returns a Text variable containing the value of the field.

MySQL_GetRealField

MySQL_GetRealField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
fieldValue	Real	←	Value of the field

Description

The MySQL_GetRealField command returns the value of the field referenced by fieldIndex.

selectID is a Longint returned by MySQL_Select.

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

fieldValue returns a Real variable containing the value of the field.

MySQL_GetTimeField

MySQL_GetTimeField (selectID; fieldIndex; hour; minutes; seconds)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
hour	Integer	←	Value of the hour component of the time
minutes	Integer	←	Value of the minutes component of the time
seconds	Integer	←	Value of the seconds component of the time

Description

The MySQL_GetTimeField command returns the hour, minutes and seconds values of the field referenced by fieldIndex.

selectID is a Longint returned by MySQL_Select.

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

hour is an Integer containing the 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.

MySQL_GetPictureField

MySQL_GetPictureField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
fieldValue	Picture	←	Value of the field

Description

The MySQL_GetPictureField command returns the value of the field referenced by fieldIndex.

selectID is a Longint returned by MySQL_Select.

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

fieldValue returns a Picture variable containing the value of the field.

MySQL_GetBlobField

MySQL_GetBlobField (selectID; fieldIndex; fieldValue)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select
fieldIndex	Integer	→	Index of the field - 0...FieldCount -1
fieldValue	Blob	←	Value of the field

Description

The MySQL_GetBlobField command returns the value of the field referenced by fieldIndex.

selectID is a Longint returned by MySQL_Select.

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

fieldValue returns a Blob variable containing the value of the field.

MySQL_CloseSelect

MySQL_CloseSelect (selectID)

Parameter	Type		Description
selectID	Longint	→	Select ID returned by MySQL_Select

Description

The MySQL_CloseSelect command closes the MySQL_Select call referenced in selectID.

Once you're done with the results from the MySQL_Select call, then call MySQL_CloseSelect.

selectID is a Longint returned by MySQL_Select.

See Also

MySQL_Select.

Command Index