# CreateField

Appends a new **Field object** to a **table**.
This method offers the capacity to create from the Basic code a new table field independently from the underlying RDBMS. Thus without using SQL. However only a limited set of options are available for field descriptions.

tags:
Methods

## Applies to ...

| Object | Description |
|---|---|
| **TableDef** | The representation of a table in the currently connected database. |

## Syntax

Set *field* = *tabledef*.`CreateField`(*name, type, size, attributes*)

## Arguments - Returned value

| Argument | Type | Optional | Description | Returned value |
|---|---|---|---|---|
| name | String | | The name of the new field | A **field** object |
| type | dbInteger<br>dbLong<br>dbBigInt<br>dbFloat<br>vbSingle<br>dbDouble<br>dbNumeric<br>dbDecimal<br>dbText<br>dbChar<br>dbMemo<br>dbDate<br>dbTime<br>dbTimeStamp<br>dbBinary<br>dbVarBinary<br>dbLongBinary<br>dbBoolean | Y | The **database type ("DbType")** of the new field | |
| size | numeric | Y | The length of the field. It is ignored when not relevant.<br>If *size* has a non-integer value, the first decimal digit at the right of the decimal point determines the number of decimal digits. | |
| attributes | dbAutoIncrField | Y | Indicates if present that the field is a primary key and is incremented by the RDBMS at each new record insertion. | |

## List of available field types

| DataType | DbType | TypeName |
|---|---|---|
| com.sun.star.sdbc.DataType.BIT | *dbUndefined* | BIT |
| com.sun.star.sdbc.DataType.BOOLEAN | dbBoolean | BOOLEAN |
| com.sun.star.sdbc.DataType.TINYINT | dbInteger | TINYINT |
| com.sun.star.sdbc.DataType.SMALLINT | dbLong | SMALLINT |
| com.sun.star.sdbc.DataType.INTEGER | dbLong | INTEGER |
| com.sun.star.sdbc.DataType.BIGINT | dbBigInt | BIGINT |
| com.sun.star.sdbc.DataType.FLOAT | dbFloat | FLOAT |
| com.sun.star.sdbc.DataType.REAL | dbSingle | REAL |
| com.sun.star.sdbc.DataType.DOUBLE | dbDouble | DOUBLE |
| com.sun.star.sdbc.DataType.NUMERIC | dbNumeric | NUMERIC |
| com.sun.star.sdbc.DataType.DECIMAL | dbDecimal | DECIMAL |
| com.sun.star.sdbc.DataType.CHAR | dbtext | CHAR |
| com.sun.star.sdbc.DataType.VARCHAR | dbChar | VARCHAR |
| com.sun.star.sdbc.DataType.LONGVARCHAR | dbMemo | LONGVARCHAR |
| com.sun.star.sdbc.DataType.DATE | dbDate | DATE |
| com.sun.star.sdbc.DataType.TIME | dbTime | TIME |
| com.sun.star.sdbc.DataType.TIMESTAMP | dbTimeStamp | TIMESTAMP |
| com.sun.star.sdbc.DataType.BINARY | dbBinary | BINARY |
| com.sun.star.sdbc.DataType.VARBINARY | dbVarBinary | VARBINARY |
| com.sun.star.sdbc.DataType.LONGVARBINARY | dbLongBinary | LONGVARBINARY |
| com.sun.star.sdbc.DataType.CLOB | *dbUndefined* | CLOB |
| com.sun.star.sdbc.DataType.BLOB | *dbUndefined* | BLOB |

### Remarks

- The *dbAutoIncrField* attribute MUST NOT be set when the *CreateField* method is applied on an existing table. It is valid only between the **CreateTableDef** and **Add** methods.
- When the *dbAutoIncrField* attribute is present (for a new table only ...) the field is created and becomes the primary key of the table. The auto-increment attribute of the newly created primary key will be applied only if the RDBMS and the currently used driver allow it. As an example, for the embedded HSQLDB database, the attribute will be applied, for a "splitted" HSQLDB database, it might not be applied.
- When the *dbAutoIncrField* attribute is not present the newly created field is "nullable", i.e. when not initialized it gets the `Null` value as default.
- If additional attributes are needed to specify the new field more accurately, use an SQL statement given as argument to the **RunSQL action**.
- The *CreateField* method must not be invoked from a **standalone form** defined in a non-Base (Writer, Calc, ...) document.
- Instead of using the numeric values for DbType, one may copy next code and paste it in his/her own code. This allows to use symbolic names, close to or identical with the field types existing in MSAccess.

```
REM Types
REM ----------------------------------------------------------------
Const dbBigInt = 16
Const dbBinary = 9
Const dbBoolean = 1
Const dbByte = 2
Const dbChar = 18
Const dbCurrency = 5
Const dbDate = 8
```

```
Const dbDecimal = 20
Const dbDouble = 7
Const dbFloat = 21
Const dbGUID = 15
Const dbInteger = 3
Const dbLong = 4
Const dbLongBinary = 11 '        (OLE Object)
Const dbMemo= 12
Const dbNumeric = 19
Const dbSingle = 6
Const dbText = 10
Const dbTime = 22
Const dbTimeStamp = 23
Const dbVarBinary = 17
Const dbUndefined = -1
REM Attributes
REM ----------------------------------------------------------------
Const dbAutoIncrField = 16
```

## Error messages

| |
|---|
| Argument nr. ... [Value = '...'] is invalid |
| Field '...' could not be created |
| Method 'TableDef.CreateField' not applicable in this context |

## See also

**Add**
**CreateQueryDef**
**CreateTableDef**
**TableDefs**
**TableDef**
**QueryDefs**
**QueryDef**

## Example

Build a new table in the database without using SQL

```
Dim oDatabase As Object, oTable As Object
        Set oDatabase = Application.CurrentDb()
        Set oTable = oDatabase.CreateTableDef("myNewTable")                    '        Create
        With oTable
                .CreateField("myAutoIncrFld", dbLong, , dbAutoIncrField)        '        Create
                .CreateField("myDecimalFld", dbDecimal, 10.4)
        End With
        oDatabase.TableDefs().Add(oTable)                                       '        Final
```

Add a field to an existing table

```
        Set oTable = oDatabase.TableDefs("myExistingTable")
        oTable.CreateField("myBinaryFld", dbBinary)
```

Bookmark this page » » CreateField