

## Clone

Creates a duplicate **Recordset object** that refers to the original *Recordset* object..

tags:  
Methods

### Applies to ...

| Object           | Description   |
|------------------|---|
| <b>Recordset</b> | A set of records derived from a table, a query or an SQL statement. |

### Syntax

```
recordset.Clone( )
```

### Returned value

A **recordset** object.

### Remarks

- Use the *Clone* method to create multiple, duplicate *Recordset* objects. Each *Recordset* can have its own current record. Using *Clone* by itself doesn't change the data in the objects or in their underlying structures. When you use the *Clone* method, you can share bookmarks between two or more *Recordset* objects because their bookmarks are interchangeable.
- You can use the *Clone* method when you want to perform an operation on a *Recordset* that requires multiple current records. When created the current record of the clone is the first record if it exists.
- A cloned *Recordset* is always read-only.
- It is forbidden to use the *Clone* method on a cloned *Recordset*.
- The cloned *Recordset(s)* must be closed before the original *Recordset*.

### Error Messages

|   |
|---|
| Cloning a cloned Recordset is forbidden |
|---|

### See also

#### Close

#### OpenRecordset

### Example

Use a cloned recordset to move to a bookmark without changing the current record

```
Dim oRecordset1 As Object, oRecordset2 As Object, vBookmark As Variant
    Set oRecordset1 = Application.CurrentDb().OpenRecordset("SELECT * FROM [PRODUCTS]")
    oRecordset1.Move(300)
    DebugPrint oRecordset1.Fields("DENOMINATION").Value
    vBookmark = oRecordset1.Bookmark
    Set oRecordset2 = oRecordset1.Clone() ' Bookmarks become inter
    oRecordset1.MoveLast ' Continue navigation
    oRecordset2.Bookmark = vBookmark
    DebugPrint oRecordset2.Fields("DENOMINATION").Value
```

```
oRecordset2.mClose()  
oRecordset1.mClose()
```

```
' First close clone
```

Bookmark this page » » [Clone](#)