

# Recordset: Bookmarks and Absolute Positions (ODBC)

Article • 08/02/2021

This topic applies to the MFC ODBC classes.

When navigating through a recordset, you often need a way of returning to a particular record. A record's bookmark and absolute position provide two such methods.

This topic explains:

- [How to use bookmarks.](#)
- [How to set the current record using absolute positions.](#)

## Bookmarks in MFC ODBC

A bookmark uniquely identifies a record. When you navigate through a recordset, you cannot always rely on the absolute position of a record because records can be deleted from the recordset. The reliable way to keep track of the position of a record is to use its bookmark. Class `CRecordset` supplies member functions for:

- Getting the bookmark of the current record, so you can save it in a variable ([GetBookmark](#)).
- Moving quickly to a given record by specifying its bookmark, which you saved earlier in a variable ([SetBookmark](#)).

The following example illustrates how to use these member functions to mark the current record and later return to it:

C++

```
// rs is a CRecordset or
// CRecordset-derived object

CDBVariant varRecordToReturnTo;
rs.GetBookmark( varRecordToReturnTo );

// More code in which you
```

```
// move to other records  
  
rs.SetBookmark( varRecordToReturnTo );
```

You do not need to extract the underlying data type from the [CDBVariant Class](#) object. Assign the value with `GetBookmark` and return to that bookmark with `SetBookmark`.

#### ⓘ Note

Depending on your ODBC driver and recordset type, bookmarks might not be supported. You can easily determine whether bookmarks are supported by calling **`CRecordset::CanBookmark`**. Furthermore, if bookmarks are supported, you must explicitly choose to implement them by specifying the `CRecordset::useBookmarks` option in the **`CRecordset::Open`** member function. You should also check the persistence of bookmarks after certain recordset operations. For example, if you Requery a recordset, bookmarks might no longer be valid. Call **`CDatabase::GetBookmarkPersistence`** to check whether you can safely call `SetBookmark`.

## Absolute Positions in MFC ODBC

Besides bookmarks, class `CRecordset` allows you to set the current record by specifying an ordinal position. This is called absolute positioning.

#### ⓘ Note

Absolute positioning is not available on forward-only recordsets. For more information about forward-only recordsets, see **`Recordset (ODBC)`**.

To move the current record pointer using absolute position, call **`CRecordset::SetAbsolutePosition`**. When you pass a value to `SetAbsolutePosition`, the record corresponding to that ordinal position becomes the current record.

#### ⓘ Note

The absolute position of a record is potentially unreliable. If the user deletes

records from the recordset, the ordinal position of any subsequent record changes. Bookmarks are the recommended method for moving the current record. For more information, see **Bookmarks in MFC ODBC**.

For more information about recordset navigation, see [Recordset: Scrolling \(ODBC\)](#).

## See also

[Recordset \(ODBC\)](#)