

Recordset: Locking Records (ODBC)

Article • 08/02/2021

This topic applies to the MFC ODBC classes.

This topic explains:

- [The kinds of record locking available.](#)
- [How to lock records in your recordset during updates.](#)

When you use a recordset to update a record on the data source, your application can lock the record so no other user can update the record at the same time. The state of a record updated by two users at the same time is undefined unless the system can guarantee that two users cannot update a record simultaneously.

ⓘ Note

This topic applies to objects derived from `CRecordset` in which bulk row fetching has not been implemented. If you have implemented bulk row fetching, some of the information does not apply. For example, you cannot call the `Edit` and `Update` member functions. For more information about bulk row fetching, see [Recordset: Fetching Records in Bulk \(ODBC\)](#).

Record-Locking Modes

The database classes provide two [record-locking modes](#):

- Optimistic locking (the default)
- Pessimistic locking

Updating a record occurs in three steps:

1. You begin the operation by calling the [Edit](#) member function.
2. You change the appropriate fields of the current record.
3. You end the operation — and normally commit the update — by calling the

[Update](#) member function.

Optimistic locking locks the record on the data source only during the `update` call. If you use optimistic locking in a multiuser environment, the application should handle an `update` failure condition. Pessimistic locking locks the record as soon as you call `Edit` and does not release it until you call `update` (failures are indicated through the `CDBException` mechanism, not by a value of `FALSE` returned by `update`). Pessimistic locking has a potential performance penalty for other users, because concurrent access to the same record might have to wait until completion of your application's `update` process.

Locking Records in Your Recordset

If you want to change a recordset object's [locking mode](#) from the default, you must change the mode before you call `Edit`.

To change the current locking mode for your recordset

1. Call the [SetLockingMode](#) member function, specifying either `CRecordset::pessimistic` or `CRecordset::optimistic`.

The new locking mode remains in effect until you change it again or the recordset is closed.

Note

Relatively few ODBC drivers currently support pessimistic locking.

See also

[Recordset \(ODBC\)](#)

[Recordset: Performing a Join \(ODBC\)](#)

[Recordset: Adding, Updating, and Deleting Records \(ODBC\)](#)