**Locking Hints**

**SQL Server 2012**

You can specify a range of table-level locking hints by using the SELECT, INSERT, UPDATE, and DELETE statements to modify the default locking behavior of Microsoft SQL Server Compact 4.0. Use locking hints only when absolutely necessary. They can adversely affect concurrency.

|  |
| --- |
| **Important** |
| SQL Server Compact 4.0 automatically acquires the locks required for an operation. If you use the locking hints listed in the following table, SQL Server Compact 4.0 increases the amount of locking that occurs. You cannot use locking hints to avoid locking resources. |

The following table describes the locking hints that you can use in SQL Server Compact 4.0.

|  |  |
| --- | --- |
| Locking hint name | **Hint description** |
| GRANULARITY |  |
| ROWLOCK | Use row-level locks when reading or modifying data. These are acquired and released as appropriate.  SELECT operations take S locks on rows. |
| PAGLOCK | Use page-level locks when reading or modifying data. These are acquired and released as appropriate.  SELECT operations take S locks on pages. |
| TABLOCK | Use a table lock when reading or modifying data. This lock is held until the end of the statement.  SELECT operations take S locks on tables. |
| DBLOCK | Use a database lock when reading or modifying data. This lock is held until the end of the statement.  SELECT operations take S locks on databases. |
| LOCKMODES |  |
| UPDLOCK | Use update locks instead of shared locks while reading a table, and use hold locks until the end of the statement or transaction. UPDLOCK lets you read data without blocking other readers, and to update it later with the assurance that the data has not changed since you last read it.  SELECT operations take U locks. The default granularity is ROWLOCK. |
| XLOCK | Use exclusive locks instead of shared locks while reading a table, and use hold locks until the end of the statement or transaction.  SELECT operations take X locks. The default granularity is ROWLOCK. |
| DURATION |  |
| HOLDLOCK | Use a hold lock to hold a lock until completion of the transaction, instead of releasing the lock as soon as the required table, row, or data page is no longer required.  If no granularity is specified, ROWLOCK is applied. |
| NOLOCK | Does not issue any locks. This is the default for SELECT operations. It does not apply to INSERT, UPDATE, and DELETE statements.  **Note**  With SQL Server, a NOLOCK hint enables Read Uncommitted behavior. With SQL Server Compact 4.0, using a NOLOCK hint still gives a Read Committed isolation level. SQL Server Compact 4.0 maintains copies of data to ensure that data can be read without needing share locks to help protect the data. |

For more information about using locking hints, see "Locking Hints" in SQL Server Books Online.

[NOLOCK Hint](javascript:void(0))

Using locking hints in SQL Server Compact 4.0 is similar to that of SQL Server. However, for SQL Server Compact 4.0, the NOLOCK hint has a very different behavior than it does for SQL Server. In SQL Server Compact 4.0, the NOLOCK hint is the default for SELECT statements, but this still enforces Read Committed behavior.

In SQL Server, a SELECT statement with the default isolation level of Read Committed causes S locks being taken and released on rows as they are read. Although this enforces the isolation level, it means that a SELECT statement waits if an incompatible lock exists on a row for which an S lock is required. When the NOLOCK hint is specified, the SELECT operation does not try to take the S lock and the data is read. Although this lets the operation succeed, it also means that the SELECT statement can read uncommitted data.

SQL Server Compact 4.0 does not use S locks to ensure that data is Read Committed. Because SQL Server Compact 4.0 uses a page versioning mechanism when changing data, the data that is required by a SELECT statement can be read from the appropriate copy of the page. It is not required to take S locks to ensure Read Committed. Therefore, although SQL Server Compact 4.0 is using NOLOCK for a SELECT statement, the data is read at the isolation level of Read Committed. You cannot have a dirty read (Read Uncommitted) with SQL Server Compact 4.0.

|  |
| --- |
| **Note** |
| The NOLOCK hint does not affect Sch-S or Sch-X locks. |