

SQLite ADO.NET Provider

SQLiteConnection Class

SQLite implementation of DbConnection.

For a list of all members of this type, see [SQLiteConnection Members](#).

[System.Object](#)

MarshalByRefObject

Component

DbConnection

SQLiteConnection

```
public sealed class SQLiteConnection : DbConnection, ICloneable
```

Remarks

The [ConnectionString](#) property can contain the following parameter(s), delimited with a semi-colon:

| Parameter | Values | Required | Default |
|------------------|---|----------|---------|
| Data Source | This may be a file name, the string ":memory:", or any supported URI (starting with SQLite 3.7.7). Starting with release 1.0.86.0, in order to use more than one consecutive backslash (e.g. for a UNC path), each of the adjoining backslash characters must be doubled (e.g. "\\Network\Share\test.db" would become "\\\\Network\Share\test.db"). | Y | |
| Uri | If specified, this must be a file name that starts with "file://", "file:", or "/". Any leading "file://" or "file:" prefix will be stripped off and the resulting file name will be used to open the database. | N | null |
| FullUri | If specified, this must be a URI in a format recognized by the SQLite core library (starting with SQLite 3.7.7). It will be passed verbatim to the SQLite core library. | N | null |
| Version | 3 | N | 3 |
| UseUTF16Encoding | True - The UTF-16 encoding should be used. False - The UTF-8 encoding should be used. | N | False |
| DefaultDbType | This is the default DbType to use when one cannot be determined based on the column metadata and the configured type mappings. | N | null |
| DefaultTypeName | This is the default type name to use when one cannot be determined based on the column metadata and the configured type mappings. | N | null |
| NoDefaultFlags | True - Do not combine the specified (or existing) connection flags with the value of the DefaultFlags property. | N | False |

| | | | |
|----------------------|--|---|-------------|
| | False - Combine the specified (or existing) connection flags with the value of the DefaultFlags property. | | |
| NoSharedFlags | True - Do not combine the specified (or existing) connection flags with the value of the SharedFlags property. False - Combine the specified (or existing) connection flags with the value of the SharedFlags property. | N | False |
| VfsName | The name of the VFS to use when opening the database connection. If this is not specified, the default VFS will be used. | N | null |
| ZipVfsVersion | If non-null, this is the "version" of ZipVFS to use. This requires the System.Data.SQLite interop assembly -AND- primary managed assembly to be compiled with the INTEROP_INCLUDE_ZIPVFS option; otherwise, this property does nothing. The valid values are "v2" and "v3". Using anyother value will cause an exception to be thrown. Please see the ZipVFS documentation for more information on how to use this parameter. | N | null |
| DateTimeFormat | Ticks - Use the value of DateTime.Ticks. ISO8601 - Use the ISO-8601 format. Uses the "yyyy-MM-dd HH:mm:ss.FFFFFFFF" format for UTC DateTime values and "yyyy-MM-dd HH:mm:ss.FFFFFFF" format for local DateTime values). JulianDay - The interval of time in days and fractions of a day since January 1, 4713 BC. UnixEpoch - The whole number of seconds since the Unix epoch (January 1, 1970). InvariantCulture - Any culture-independent string value that the .NET Framework can interpret as a valid DateTime. CurrentCulture - Any string value that the .NET Framework can interpret as a valid DateTime using the current culture. | N | ISO8601 |
| DateTimeKind | Unspecified - Not specified as either UTC or local time. Utc - The time represented is UTC. Local - The time represented is local time. | N | Unspecified |
| DateTimeFormatString | The exact DateTime format string to use for all formatting and parsing of all DateTime values for this connection. | N | null |

| | | | |
|----------------|--|---|-----------------------|
| BaseSchemaName | Some base data classes in the framework (e.g. those that build SQL queries dynamically) assume that an ADO.NET provider cannot support an alternate catalog (i.e. database) without supporting alternate schemas as well; however, SQLite does not fit into this model. Therefore, this value is used as a placeholder and removed prior to preparing any SQL statements that may contain it. | N | sqlite_default_schema |
| BinaryGUID | True - Store GUID columns in binary form False - Store GUID columns as text | N | True |
| Cache Size | If the argument N is positive then the suggested cache size is set to N. If the argument N is negative, then the number of cache pages is adjusted to use approximately abs(N*4096) bytes of memory. Backwards compatibility note: The behavior of cache_size with a negative N was different in SQLite versions prior to 3.7.10. In version 3.7.9 and earlier, the number of pages in the cache was set to the absolute value of N. | N | -2000 |
| Synchronous | Normal - Normal file flushing behavior Full - Full flushing after all writes Off - Underlying OS flushes I/O's | N | Full |
| Page Size | {size in bytes} | N | 4096 |
| Password | {password} - Using this parameter requires that the legacy CryptoAPI based codec (or the SQLite Encryption Extension) be enabled at compile-time for both the native interop assembly and the core managed assemblies; otherwise, using this parameter may result in an exception being thrown when attempting to open the connection. | N | |
| HexPassword | {hexPassword} - Must contain a sequence of zero or more hexadecimal encoded byte values without a leading "0x" prefix. Using this parameter requires that the legacy CryptoAPI based codec (or the SQLite Encryption Extension) be enabled at compile-time for both the native interop assembly and the core managed assemblies; otherwise, using this parameter may result in an exception being thrown when attempting to open the connection. | N | |
| | | | |

| | | | |
|-----------------|--|---|--------|
| Enlist | Y - Automatically enlist in distributed transactions N - No automatic enlistment | N | Y |
| Pooling | True - Use connection pooling. False - Do not use connection pooling. WARNING: When using the default connection pool implementation, setting this property to True should be avoided by applications that make use of COM (either directly or indirectly) due to possible deadlocks that can occur during the finalization of some COM objects. | N | False |
| FailIfMissing | True - Don't create the database if it does not exist, throw an error instead False - Automatically create the database if it does not exist | N | False |
| Max Page Count | {size in pages} - Limits the maximum number of pages (limits the size) of the database | N | 0 |
| Legacy Format | True - Use the more compatible legacy 3.x database format False - Use the newer 3.3x database format which compresses numbers more effectively | N | False |
| Default Timeout | {time in seconds} The default command timeout | N | 30 |
| BusyTimeout | {time in milliseconds} Sets the busy timeout for the core library. | N | 0 |
| WaitTimeout | {time in milliseconds} EXPERIMENTAL -- The wait timeout to use with WaitForEnlistmentReset method. This is only used when waiting for the enlistment to be reset prior to enlisting in a transaction, and then only when the appropriate connection flag is set. | N | 30000 |
| Journal Mode | Delete - Delete the journal file after a commit. Persist - Zero out and leave the journal file on disk after a commit. Off - Disable the rollback journal entirely. This saves disk I/O but at the expense of database safety and integrity. If the application using SQLite crashes in the middle of a transaction when this journaling mode is set, then the database file will very likely go corrupt. Truncate - Truncate the journal | N | Delete |

| | | | |
|------------------------|---|---|--------------|
| | <p>file to zero-length instead of deleting it.</p> <p>Memory - Store the journal in volatile RAM. This saves disk I/O but at the expense of database safety and integrity. If the application using SQLite crashes in the middle of a transaction when this journaling mode is set, then the database file will very likely go corrupt.</p> <p>Wal - Use a write-ahead log instead of a rollback journal.</p> | | |
| Read Only | <p>True - Open the database for read only access</p> <p>False - Open the database for normal read/write access</p> | N | False |
| Max Pool Size | The maximum number of connections for the given connection string that can be in the connection pool | N | 100 |
| Default IsolationLevel | The default transaction isolation level | N | Serializable |
| Foreign Keys | Enable foreign key constraints | N | False |
| Flags | Extra behavioral flags for the connection. See the SQLiteConnectionFlags enumeration for possible values. | N | Default |
| SetDefaults | <p>True - Apply the default connection settings to the opened database.</p> <p>False - Skip applying the default connection settings to the opened database.</p> | N | True |
| ToFullPath | <p>True - Attempt to expand the data source file name to a fully qualified path before opening.</p> <p>False - Skip attempting to expand the data source file name to a fully qualified path before opening.</p> | N | True |
| PrepareRetries | The maximum number of retries when preparing SQL to be executed. This normally only applies to preparation errors resulting from the database schema being changed. | N | 3 |
| ProgressOps | The approximate number of virtual machine instructions between progress events. In order for progress events to actually fire, the event handler must be added to the Progress event as well. | N | 0 |
| Recursive Triggers | True - Enable the recursive trigger capability. False - Disable the recursive trigger capability. | N | False |

Requirements

Namespace: [System.Data.SQLite](#)

Assembly: System.Data.SQLite (in System.Data.SQLite.dll)

See Also

[SQLiteConnection Members](#) | [System.Data.SQLite Namespace](#)

[Send comments on this topic.](#)

Generated from assembly System.Data.SQLite [1.0.113.0] by [NDoc3](#)