

---

# MySQL Connector/C Release Notes

## Abstract

This document contains release notes for the changes in each release of MySQL Connector/C.

For additional Connector/C documentation, see [MySQL Connector/C Developer Guide](#).

Updates to these notes occur as new product features are added, so that everybody can follow the development process. If a recent version is listed here that you cannot find on the download page (<https://dev.mysql.com/downloads/>), the version has not yet been released.

The documentation included in source and binary distributions may not be fully up to date with respect to release note entries because integration of the documentation occurs at release build time. For the most up-to-date release notes, please refer to the online documentation instead.

For legal information, see the [Legal Notices](#).

For help with using MySQL, please visit either the [MySQL Forums](#) or [MySQL Mailing Lists](#), where you can discuss your issues with other MySQL users.

Document generated on: 2019-05-17 (revision: 17900)

## Table of Contents

Preface and Legal Notices .....	1
Changes in MySQL Connector/C 6.1 .....	3
Changes in MySQL Connector/C 6.1.11 (2017-07-24, General Availability) .....	3
Changes in MySQL Connector/C 6.1.10 (2017-04-28, General Availability) .....	3
Changes in MySQL Connector/C 6.1.9 (2017-01-10, General Availability) .....	4
Changes in MySQL Connector/C 6.1.8 (2015-12-15, General Availability) .....	4
Changes in MySQL Connector/C 6.1.7 (2015-04-21, General Availability) .....	4
Changes in MySQL Connector/C 6.1.6 (2015-03-20, General Availability) .....	5
Changes in MySQL Connector/C 6.1.5 (2014-06-17, General Availability) .....	6
Changes in MySQL Connector/C 6.1.4 (2014-04-15, General Availability) .....	7
Changes in MySQL Connector/C 6.1.3 (2013-12-27, General Availability) .....	8
Changes in MySQL Connector/C 6.1.2 (2013-09-30, General Availability) .....	10
Changes in MySQL Connector/C 6.1.1 (2013-08-05, General Availability) .....	11
Changes in MySQL Connector/C 6.1.0 (2013-05-02, General Availability) .....	11
Changes in MySQL Connector/C 6.0 .....	11
Changes in MySQL Connector/C 6.0.3 (Not released) .....	11
Changes in MySQL Connector/C 6.0.2 (2009-08-10, General Availability) .....	12
Changes in MySQL Connector/C 6.0.1 (2009-04-21, Beta) .....	12
Changes in MySQL Connector/C 6.0.0 (2009-04-02, Beta) .....	12

## Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL Connector/C.

## Legal Notices

Copyright © 1997, 2019, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

## Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Changes in MySQL Connector/C 6.1

### Changes in MySQL Connector/C 6.1.11 (2017-07-24, General Availability)

- [Security Notes](#)
- [Bugs Fixed](#)

#### Security Notes

- The linked OpenSSL library for MySQL Connector/C 6.1 Commercial has been updated to version 1.0.2l. For a description of issues fixed in this version, see <http://www.openssl.org/news/vulnerabilities.html>.

This change does not affect the Oracle-produced MySQL Community build of Connector/C, which uses the yaSSL library instead. (Bug #26321042)

#### Bugs Fixed

- **Microsoft Windows:** Program Database (PDB) files (with file name extension `.pdb`) for debugging now are available as `-debug` Zip packages. They are intended for use with the main Connector/C Zip packages and should be unpacked on top of an unpacked main package. (Bug #23283189)
- If the `mysql_stmt_close()` C API function was called, it freed memory that later could be accessed if `mysql_stmt_error()`, `mysql_stmt_errno()`, or `mysql_stmt_sqlstate()` was called. To obtain error information after a call to `mysql_stmt_close()`, call `mysql_error()`, `mysql_errno()`, or `mysql_sqlstate()` instead. (Bug #25988703)

### Changes in MySQL Connector/C 6.1.10 (2017-04-28, General Availability)

- [Compilation Notes](#)
- [Security Notes](#)

#### Compilation Notes

- The Windows version of MySQL Connector/C Community is now built using the dynamic C runtime libraries (that is, with the `/MD` compiler option), with the following implications for users:
  - Target hosts running Windows applications that use MySQL Connector/C Community now need the [Visual C++ Redistributable for Visual Studio 2015](#) installed on them.
  - Client applications on Windows that use MySQL Connector/C Community should be compiled with the `/MD` compiler option.

## Security Notes

- The linked OpenSSL library for MySQL Connector/C 6.1 Commercial has been updated to version 1.0.2k. For a description of issues fixed in this version, see <http://www.openssl.org/news/vulnerabilities.html>.

This change does not affect the Oracle-produced MySQL Community build of Connector/C, which uses the yaSSL library instead. (Bug #25615447)

## Changes in MySQL Connector/C 6.1.9 (2017-01-10, General Availability)

- [Security Notes](#)
- [Functionality Added or Changed](#)

## Security Notes

- OpenSSL is ending support for version 1.0.1 in December 2016; see <https://www.openssl.org/policies/releasestrat.html>. Consequently, Connector/C Commercial builds now use version 1.0.2 rather than version 1.0.1, and the linked OpenSSL library for the Connector/C Commercial has been updated from version 1.0.1 to version 1.0.2j. For a description of issues fixed in this version, see <https://www.openssl.org/news/vulnerabilities.html>.

This change does not affect Oracle-produced MySQL Community builds of Connector/C, which use the yaSSL library instead. (Bug #24753384)

- A new `MYSQL_OPT_TLS_VERSION` option is available for the `mysql_options()` C API function to indicate the protocols the client permits for encrypted connections.

For more information, see [mysql\\_options\(\)](#). (Bug #22932026)

- A new `MYSQL_OPT_SSL_MODE` option is available for the `mysql_options()` C API function to indicate the security state to use for the connection to the server.

For more information, see [mysql\\_options\(\)](#). (Bug #22931973)

## Functionality Added or Changed

- Previously, Connector/C permitted user names up to a maximum of 16 characters. Connector/C now permits longer user names (up to 32 characters) if the server does. (Bug #22931954)
- Connector/C now includes support for the server GTID session tracker, introduced in MySQL 5.7.6. This tracker is exposed through the `session_track_gtids` system variable. (Bug #21040741)
- Connector/C now includes support for additional ranges of server errors, introduced in MySQL 5.7.6. (Bug #21040703)

## Changes in MySQL Connector/C 6.1.8 (2015-12-15, General Availability)

### Security Notes

- This release of Connector/C upgrades the linked OpenSSL library to version 1.0.1p. Issues fixed in the new OpenSSL version are described at <http://www.openssl.org/news/vulnerabilities.html>.

## Changes in MySQL Connector/C 6.1.7 (2015-04-21, General Availability)

## Security Notes

- Connector/C 6.1 Commercial has been updated to use OpenSSL version 1.0.1m, which has been publicly reported as not vulnerable to [CVE-2015-0286](#).

Since the only change in Connector/C 6.1.7 is the inclusion of OpenSSL libraries publicly reported as unaffected by CVE-2015-0286, and since Oracle-produced MySQL Community builds use YaSSL libraries which have been reported as not affected by CVE-2015-0286, Oracle will not produce builds for Connector/C Community for version 6.1.7. This means the Community edition of Connector/C will skip version 6.1.7. (Bug #20747718)

## Changes in MySQL Connector/C 6.1.6 (2015-03-20, General Availability)

- [Security Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

### Security Notes

- This release of Connector/C upgrades the linked OpenSSL library to version 1.0.1k. Issues fixed in the new OpenSSL version are described at <http://www.openssl.org/news/vulnerabilities.html>.

### Functionality Added or Changed

- yaSSL sources included in Connector/C source packages and used in GPL binary distributions were upgraded to version 2.3.7. (Bug #20201864)
- A new CMake option, `WITH_MSVCRT_DEBUG`, is available to control Visual Studio CRT memory leak tracing. The default is `OFF`. (Bug #73064, Bug #19031370)

### Bugs Fixed

- For source packages, it was not possible to build Connector/C with the client protocol test trace plugin enabled. (Bug #20316149)
- When there is no change in session state, the OK packet sent from server to the client contained an unneeded byte at the end of the packet. (Bug #19625718)
- The client protocol trace plugin did not account for the removal of the EOF packet from the client/server protocol in MySQL 5.7.5. (Bug #19512199)
- The `mysql_session_track_get_first()` and `mysql_session_track_get_next()` C API functions could cause a client crash if passed invalid arguments. (Bug #18769620)
- `mysql_config --libs_r` produces output containing link flags for `libmysqlclient_r`, even though that library was removed in MySQL 5.5 and replaced with a symbolic link to the underlying `libmysqlclient` library. The output now refers directly to `libmysqlclient`. (The implication is that it is no longer necessary to maintain the symbolic link for the sake of being able to use `mysql_config --libs_r`.) (Bug #73724, Bug #19506315)
- Invalid memory access could occur when using prepared statements if a `mysql` client connection was lost after statement preparation was complete and there was at least one statement that was in initialized state but not prepared yet. (Bug #70429, Bug #17512527)

- Client auto-reconnect did not work for clients linked against `libmysqlclient`, even with `MYSQL_OPT_RECONNECT` enabled.  
  
Also, if a `FEDERATED` table was accessed after `wait_timeout` expired, a `Lost connection to MySQL server` error occurred without an attempt to re-establish the connection. (Bug #70026, Bug #17309863, Bug #14874, Bug #11745408)
- `mysql_real_connect()` could close a file descriptor twice if the server was not running. (Bug #69423, Bug #19226740)

## Changes in MySQL Connector/C 6.1.5 (2014-06-17, General Availability)

- [Security Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

### Security Notes

- Connector/C 6.1.5 upgrades the linked OpenSSL library from version 1.0.1g to version 1.0.1h. Versions of OpenSSL prior to 1.0.1g are reported to be vulnerable to [CVE-2014-0224](#).

### Functionality Added or Changed

- Because there are new API functions (`mysql_session_track_get_first()`, `mysql_session_track_get_next()`), the client library ABI version is now 18.3. Shared library names now include 18.3 where appropriate.
- The server can now report to clients changes that occur to the client session state. Changes can be reported for these attributes of client session state:
  - The default schema (database)
  - Session-specific values for system variables
  - User-defined variables
  - Temporary tables
  - Prepared statements

The MySQL client/server protocol now includes tracker information so that session state changes can be detected. One use for the tracker mechanism is to provide a means for MySQL Connectors and client applications to determine whether any session context is available to permit session migration from one server to another. (To change connections in a load-balanced environment, it is necessary to detect whether there is session state to take into consideration when deciding whether a switch can be made.)

The following components comprise the user interface to control the tracker and retrieve state-change information from it, and thus enable implementation of state-change tracking on the client side:

- Clients can request notification when there is a change to any of the trackable session state-related values. Notification occurs in the form of a flag that is set in the OK packet received from the server after the change occurs. To control notification, enable or disable the `session_track_state_change` system variable. This variable is disabled by default.
- Clients can request notification of changed values for certain specific types of session state information:

- The default schema name. To control notification, enable or disable the `session_track_schema` system variable. This variable is enabled by default.
- The session values of system variables. Notification occurs for the system variables named by the `session_track_system_variables` system variable. Notification consists of the name and new value of each changed variable. By default, notification is enabled for `time_zone`, `autocommit`, `character_set_client`, `character_set_results`, and `character_set_connection`. (The latter three variables are those affected by `SET NAMES`.)
- To enable applications to extract the state-change information returned by the server, the MySQL C API provides a pair of functions:
  - `mysql_session_track_get_first()` fetches the first state-change information received from the server.
  - `mysql_session_track_get_next()` fetches any remaining state-change information received from the server. Following a successful call to `mysql_session_track_get_first()`, call this function repeatedly as long as it returns success.
  - Because there are new API functions, the client library ABI version is increased to 18.3. Shared library names now include 18.3 where appropriate.
- The `mysqltest` program has `enable_session_track_info` and `disable_session_track_info` commands to enable and disable tracking of session state-change information. See the MySQL Server Doxygen documentation, available at <https://dev.mysql.com/doc/index-other.html>.

For more information, see [Server System Variables](#) and `mysql_session_track_get_first()`. For information about the structure of the OK packet used to convey state-change information, see [OK\\_Packet](#).

## Bugs Fixed

- Calling `mysql_get_server_version()` with an invalid connection handler argument caused the client to exit. Now it returns 0 and reports a `CR_COMMANDS_OUT_OF_SYNC` error. (Bug #18053212)
- On Windows, calling `mysql_thread_init()` call without `mysql_init()` caused the client to exit. Now it returns a nonzero result because it is an error to call `mysql_thread_init()` before the client library is initialized with `mysql_library_init()`. (Bug #17514920)

## Changes in MySQL Connector/C 6.1.4 (2014-04-15, General Availability)

- [Security Notes](#)
- [Bugs Fixed](#)

## Security Notes

- Connector/C 6.1 Commercial has been updated to use OpenSSL version 1.0.1g, which has been publicly reported as not vulnerable to [CVE-2014-0160](#). Please see [Oracle Note #1645479.1](#) for further details.

Since the only change in Connector/C 6.1.4 is the inclusion of OpenSSL libraries publicly reported as unaffected by CVE-2014-0160, and since Oracle-produced MySQL Community builds use [YaSSL](#) libraries which have been reported as not affected by CVE-2014-0160, Oracle will not produce builds for Connector/C Community for version 6.1.4. This means the Community edition of Connector/C will skip version 6.1.4. (Bug #18533200)

## Bugs Fixed

- There was a difference in certificate handling by yaSSL and OpenSSL (used for Community and Enterprise, respectively). OpenSSL expected a blank certificate to be sent when not all of the `--ssl-ca`, `--ssl-cert`, and `--ssl-key` options were specified, and yaSSL did not do so. To resolve this, yaSSL has been modified to send a blank certificate when an option is missing. (Bug #68788, Bug #16715064)

## Changes in MySQL Connector/C 6.1.3 (2013-12-27, General Availability)

- [Security Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

## Security Notes

- A new `MYSQL_OPT_SSL_ENFORCE` option is available for the `mysql_options()` C API function to indicate whether to require the connection to use SSL. If enabled, an encrypted connection is attempted. If an encrypted connection cannot be established, the connection attempt fails.

For more information, see [mysql\\_options\(\)](#).

## Functionality Added or Changed

- Some dependencies between client-side plugin header files were removed:
  - The `MYSQL_PLUGIN_EXPORT` macro required by plugin declarations is now declared directly in `mysql/client_plugin.h` instead of getting the definition from `mysql/plugin.h`. That macro was the only thing required by client-side plugins and declared in server-side header `mysql/plugin.h`, so including `mysql/client_plugin.h` in an application no longer requires the application to also include `mysql/plugin.h`.
  - `mysql/plugin_trace.h` no longer uses `C_MODE_START` or `C_MODE_END`. Consequently, including `mysql/plugin_trace.h` in an application no longer requires the application to also include `my_global.h`.

Applications might require `mysql/plugin.h` or `my_global.h` for other reasons, of course. (Bug #17582168)

- The `mysql_version.h` file defines two new macros, `LIBMYSQL_VERSION` and `LIBMYSQL_VERSION_ID`, that indicate the string and numeric forms of the client library version.
  - In the client library included with MySQL Server distributions, these macros have the same values as `MYSQL_SERVER_VERSION` and `MYSQL_VERSION_ID`. For example, in MySQL 5.7.4, `MYSQL_SERVER_VERSION` and `LIBMYSQL_VERSION` are "5.7.4-m14", and `MYSQL_VERSION_ID` and `LIBMYSQL_VERSION_ID` are 50704.
  - In the client library included with Connector/C distributions, `MYSQL_SERVER_VERSION` and `MYSQL_VERSION_ID` have the values of the MySQL version on which the Connector/C distribution is based, whereas `LIBMYSQL_VERSION` and `LIBMYSQL_VERSION_ID` indicate the Connector/C version. For example, Connector/C 6.1.3 is based on MySQL 5.7.4, so `MYSQL_SERVER_VERSION` and `MYSQL_VERSION_ID` have values of "5.7.4-m14" and 50704, whereas `LIBMYSQL_VERSION` and `LIBMYSQL_VERSION_ID` have values of "6.1.3" and 60103.



In addition, the `mysql_get_client_info()` and `mysql_get_client_version()` C API functions in the client library now return values that reflect the type of distribution that provides the client library:

- In MySQL distributions, `mysql_get_client_info()` returns `MYSQL_SERVER_VERSION` and `mysql_get_client_version()` returns `MYSQL_VERSION_ID`. This is the same as before.
- In Connector/C distributions, `mysql_get_client_info()` returns `LIBMYSQL_VERSION` and `mysql_get_client_version()` returns `LIBMYSQL_VERSION_ID`. Previously, these functions returned the MySQL version, the same as in MySQL distributions.

(Bug #17171724)

- Because there are new API functions (`mysql_get_option()`, `mysql_reset_connection()`), the library ABI version is now 18.2. Shared library names now include 18.2 where appropriate.
- Connector/C is now included in MySQL Installer (Windows).
- When a connection is returned to the thread pool plugin, the connection thread context must be cleaned up. Previously, this was done using `COM_CHANGE_USER` (which is like the `mysql_change_user()` C API function). However, that operation reauthenticates, which is unnecessary network roundtrip overhead in this context.

Now it is possible for client connection state to be reset in a more lightweight manner without causing reauthentication. The API is exposed publicly through these changes:

- A new `COM_RESET_CONNECTION` protocol command (defined in `mysql_com.h`)
- A new `mysql_reset_connection()` C API function
- A new `resetconnection` command for the `mysql` client

Resetting a connection has effects similar to `mysql_change_user()` or an auto-reconnect except that the connection is not closed and reopened, and reauthentication is not done. See `mysql_change_user()` and see [C API Automatic Reconnection Control](#)).

For more information, see `mysql_reset_connection()` and [mysql — The MySQL Command-Line Client](#).

- A new `mysql_get_option()` C API function is available that returns the current value of applicable `mysql_options()` options. See `mysql_get_option()`.

## Bugs Fixed

- The C client library could leak memory when client plugins were used. (Bug #17933308)
- It was not possible to build client-side plugins using Connector/C because `client_plugin.h` referenced a macro defined in the `plugin.h` file, which is not included in Connector/C distributions. (Bug #17582228)

References: See also: Bug #17582168.

- Upgrading Connector/C using the 64-bit version of the Windows MSI package occurred in the default folder because registry search logic was hardcoded to use the 32-bit registry. (Bug #17515067)
- After the fix for Bug #16409270, it was not possible to `#include <mysql.h>` following `#include <windows.h>`. (Bug #17514554)

References: See also: Bug #16409270.

- A client crash occurred if `mysql_set_server_option()` or several other C API functions were called before `mysql_real_connect()`. (Bug #17338958)

## Changes in MySQL Connector/C 6.1.2 (2013-09-30, General Availability)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

### Functionality Added or Changed

- **Microsoft Windows:** The implementation of condition variables specific to Windows XP and Windows Server 2003 was removed from the source code because MySQL is not supported on those platforms as of MySQL 5.6.



#### Note

This change has the following implication: Previously, Connector/C 6.1, while not officially supported on Windows XP or Windows Server 2003, could be used on those platforms. This is no longer possible.

(Bug #17332056)

- Static libraries built with VS2008, VS2010, VS2012 are now distributed as part of Connector/C packages for Windows: ZIP and MSI. New directories named `vs9`, `vs10`, and `vs11` located under the `lib` directory contain static libraries and corresponding pdb files built with VS2008, VS2010, and VS2012, respectively.
- MySQL now supports the use of protocol trace plugins: client-side plugins that implement tracing of communication between a client and the server that takes place using the client/server protocol. Protocol trace plugins use the client plugin API.

In MySQL source distributions, a test protocol trace plugin is implemented in the `test_trace_plugin.cc` file in the `libmysql` directory. This can be examined as a guide to writing other protocol trace plugins.

For more information, see [Writing Plugins](#); in particular, [Writing Protocol Trace Plugins](#).

### Bugs Fixed

- The `CLIENT_CONNECT_WITH_DB` flag was improperly handled in the C client library. This could lead to a malformed packet sent to the server. (Bug #17351732)
- The `mysql_real_connect()` C API function could leak memory if it failed. (Bug #17337684)
- The `mysql_options()` C API function could leak memory if called more than once with the `MYSQL_SET_CLIENT_IP` option. (Bug #17297012)
- The Connector/C RPM package was missing the following files: `INFO_SRC`, `INFO_BIN`, `my_print_defaults`, `percona`. (Bug #17261610)
- The Connector/C MSI package was missing the following files: `ChangeLog`, `README`, `LICENSE`, `COPYING`, `INFO_SRC`, `INFO_BIN`. (Bug #17261526)
- On Windows, a MySQL client program that simply used `#include <mysql.h>` could not be compiled due to missing Windows declarations in that file. The same program would compile on other platforms. (Bug #16409270)

References: See also: Bug #17514554.

- The `libmysql.dll` library was missing several symbols: `my_init`, `mysql_client_find_plugin`, `mysql_client_register_plugin`, `mysql_load_plugin`, `mysql_load_plugin_v`, and `mysql_plugin_options`. (Bug #69204, Bug #16797982)

## Changes in MySQL Connector/C 6.1.1 (2013-08-05, General Availability)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

### Functionality Added or Changed

- The C API `libmysqlclient` shared-library `.so` files now have version 18.1.0 (up from version 18.0.0 used in 6.1.0). 18.1.0 can be used as a replacement for 18.0.0. (Bug #16809055)

### Bugs Fixed

- The `libmysql.dll` library was missing the `mysql_options4` symbol. (Bug #69204, Bug #16797982)

## Changes in MySQL Connector/C 6.1.0 (2013-05-02, General Availability)

### Functionality Added or Changed

- Connector/C 6.1 provides these features not present in 6.0:
  - Support for the pluggable authentication framework that enables implementation of authentication methods as plugins. This framework can be used for MySQL native authentication as well as external authentication methods. See [Pluggable Authentication](#).
  - Client-side support for the SHA-256, PAM, and Windows native authentication plugins. See [SHA-256 Pluggable Authentication](#), [PAM Pluggable Authentication](#), and [Windows Pluggable Authentication](#).

The older Connector/C 6.0 can connect only to accounts that use native MySQL passwords. If a client program attempts to connect to an account that requires a different authentication method, an “Access denied for user” error occurs.
- Support for connecting to accounts that have expired passwords. See [Server Handling of Expired Passwords](#).
- Support for prepared `CALL` statements. This enables client programs to handle stored procedures that produce multiple result sets and to obtain the final value of `OUT` and `INOUT` procedure parameters. See [C API Prepared CALL Statement Support](#).
- Support for connecting over IPv6. See [IPv6 Support](#).
- Support for binding client programs to a specific IP address at connect time. See [mysql\\_options\(\)](#).
- Support for specifying connection attributes to pass to the server at connect time. See [mysql\\_options\(\)](#), and [mysql\\_options4\(\)](#).

## Changes in MySQL Connector/C 6.0

### Changes in MySQL Connector/C 6.0.3 (Not released)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

## Functionality Added or Changed

- Connector/C now has the `--enable-local-infile` option enabled by default, which is consistent with other MySQL products. It can be disabled at configuration time using the `-DDISABLE_LOCAL_INFILE` CMake option. (Bug #54128)

## Bugs Fixed

- The definition of the `MY_INIT` macro in `my_sys.h` included an extraneous semicolon, which could cause compilation failure. (Bug #59080)

References: See also: Bug #53906.

- When run against MySQL Server 5.5, the Connector/C test suite failed in the `ps_bugs` test. When executing an `EXPLAIN` statement, MySQL Server 5.5 returned the string "information\_schema" in `metadatasdb_name`, instead of an empty string. (Bug #56012)
- `mysql_config` reported incorrect locations for library files. (Bug #54381)
- Connector/C did not support the `geostd8` character set. (Bug #54100)
- After a reconnect occurred followed by `mysql_stmt_execute()`, `mysql_stmt_errno()` always returned 0. (Bug #53311)
- When run against MySQL Server 5.0, `ctest` failed an `sql_mode` test after building Connector/C. (Bug #53289)
- `mf_iocache.c` failed to compile on some platforms. (Bug #46642, Bug #11754951)

## Changes in MySQL Connector/C 6.0.2 (2009-08-10, General Availability)

This is the first General Availability (GA) release of MySQL Connector/C.

### Bugs Fixed

- Connector/C did not support compression in the client/server protocol. (Bug #46206)
- Connector/C incorrectly linked against `libstdc++`. This dependency created various problems for Connector/OpenOffice.org. (Bug #45128)
- Executing the `mysql_config` program bundled with Connector/C resulted in a segmentation fault. (Bug #44698)

## Changes in MySQL Connector/C 6.0.1 (2009-04-21, Beta)

This is a new Beta development release.

### Functionality Added or Changed

- Connector/C now features a CMake-based build and packaging system.

## Changes in MySQL Connector/C 6.0.0 (2009-04-02, Beta)

This is a new Beta development release.