

[TOC](#) [Previous](#) [Next](#)

---

# The C API

[C Compilers](#) [Data Structures](#) [Functions](#) [Example](#)

First let's get clear on names. The MySQL Native C API is a compiled C library for connecting to a MySQL database server. It has two other names: *libmysql*<sup>1</sup>, and since April 2009 *Connector/C*, under which name the library source is now available as a [package for separate compilation](#). The C API is also the kernel of other MySQL APIs, for example for Perl, Python and ODBC.

We can write MySQL-enabled applications in PHP, Java, Perl, Python, Ruby, Visual Basic, Microsoft Access, Visual C#, or a variety of other languages. The code in these apps will run more slowly than compiled C, but that difference is usually small compared with transmission time and query execution times. And development time for a C application may be ten times longer.

Then do we still need to know how to write to the MySQL C API? Yes. Many MySQL-enabled C applications are still running, and need to be maintained. Some computation-intensive modules and programs need to run in native code for specific classes of machine. The C API is the foundation for other MySQL APIs, so understanding the C API deepens our understanding of the others. And C++ is a viable development platform.

That said, the C API for MySQL 5&6 is now large enough to merit a book of its own. Our aim is to provide an entry point to help you get started.

To read the rest of this and other chapters, [buy a copy of the book](#)

---

[TOC](#) [Previous](#) [Next](#)

---

---

<sup>1</sup> not to be confused with *libmysqld*, which is the server library