

附录E MySQL 程序参考

本附录介绍以下列出的 MySQL 程序。附录的后部分详细介绍了每个程序，其中包括短描述、调用语法、所支持的选项，以及所具有的任何内部变量。

isamchk 和 myisamchk 对表进行检查和修复、执行键分布分析以及释放和重新激活索引的实用程序。

myisampack 和 pack_isam 产生压缩的只读表的实用程序。

mysql 具有行编辑能力的交互式程序，可以将查询发送到 MySQL 服务器中，还可以批处理方式使用来执行存储在一个文件中的查询。

mysql.server 启动和关闭 MySQL 服务器的脚本。

mysqlaccess 测试访问权限的脚本。

mysqladmin 执行管理操作的实用程序。

mysqlbug 产生故障报告的脚本。

mysqld MySQL 服务器。该程序必须一直运行使得客户机具有对服务器所管理的数据库访问。

mysqldump 转储数据库表内容的实用程序。

mysqlimport 将数据批量装载到表中的实用程序。

mysqlshow 提供关于数据库或表的信息的实用程序。

safe_mysqld 启动和监控 MySQL 服务器的脚本。

在语法描述中，可选的信息由方括号 ([]) 指明。

E.1 指定程序选项

大多数 MySQL 程序能理解影响程序操作的一些选项。选项可在命令行中指定，或在选项文件中指定。另外，有些选项可能由设置环境的变量指定。命令行中指定的选项优先于其他方式指定的选项，选项文件中的选项优先于环境变量的值。

从 MySQL 程序中获得联机帮助

除 mysqlbug、mysql.server 和 safe_mysqld 外，每个 MySQL 程序都理解 --help 选项。该选项提供一种快速的方法从程序本身获得有关该程序的联机帮助。例如，如果您对如何使用 mysqlimport 没有把握，可按以下形式调用该程序：

```
% mysqlimport --help
```

? 选项与 help 相同，尽管外壳程序可以将其解释为文件名的通配符：

```
% mysqlimport -?
```

```
mysqlimport: No match.
```

如果发生这种情况，试着用下列命令进行替换：

```
% mysqlimport -\?
```

大多数选项同时有短（单字母）格式和长（完整的词）格式。 -? 和 --help 选项就是这样的例子。如果短格式选项后跟一个值，在大多数情况下该选项和值可能由空格分隔。例如，

当您指定一个文件名时, `-upaul` 等价于 `-u paul`。`-p` (口令) 选项是一个例外, 任何口令值都必须跟在 `-p` 的后面, 中间没有空格。

每个程序的描述中都列出了程序可以理解的所有选项。如果程序好像不认识描述中所列出的某个选项, 可能您的版本是该选项引入之前的旧版本 (仔细检查该语法以确信您没有指定错误的选项)。

E.1.1 标准的 MySQL 程序选项

有几个选项在多个 MySQL 程序中都有标准的含义。它们将在下面一次性列出, 而不是在每个程序的描述中给出, 在每个程序的“支持的标准选项”一节指明程序可以理解这些选项中的哪几个 (本节只列出长格式名, 但程序也理解相应的短格式选项, 除非另有所指)。

标准选项是:

`-C, --compress` 允许客户机和服务器之间通信所使用的协议压缩技术 (如果服务器支持该技术的话)。该选项只由客户机使用。它是在 MySQL 3.22.3 中引入的。

`# debug_options, --debug=debug_options` 开启调试输出。除非 MySQL 是用调试支持环境建立的, 否则该选项不起作用。`debug_options` 串由冒号分隔的选项组成。`d:t:o:file_name` 就是一个典型的值, 它允许调试、开启函数调用项并退出跟踪, 以及发送输出结果到文件 `file_name` 中。

如果您希望进行大量的调试, 应该检查 MySQL 源程序分发包中的文件 `dbug/dbug.c`, 它描述了可以使用的所有选项。

`--defaults-file=path_name` 指定选项文件的路径名。通常, 程序在几个位置中搜索选项文件, 但是, 如果 `--defaults-file` 被指定, 则只能读取这个指定的文件。如果您使用该选项, 它必须是命令行中的第一个选项。该选项是在 MySQL 3.22.23 中引入的。

`-, --` 告诉程序打印帮助消息并退出。

`-h host_name, --host=host_name` 指定连接的主机 (即, 服务器正在运行的主机)。该选项由客户机程序使用。

`--on-defaults` 取消任何选项文件的使用。如果使用该选项, 它必须是命令行的第一个选项。

`-p[password], --password[=password]` 指定在连接到服务器时要使用的口令。如果 `password` 没有在该选项名后指定, 程序将请求输入口令。如果给出 `password`, 则必须紧跟在该选项名的后面, 中间没有空格。该选项只由客户机程序使用。

`-w, --pipe` 使用一个指定的管道连接到服务器。该选项只能用于运行在 Windows 下的客户机程序。

`-p port_num, --port=port_num` 对于客户机程序, 这是在连接到服务器时要使用的端口号。它用于 TCP/IP 的连接 (主机不是 localhost 的连接)。对于 `mysqld`, 该选项指定监听 TCP/IP 连接的端口。

`--print-defaults` 如果您调用程序而在命令行中没有带该选项的话, 则打印将要使用的该选项的值。它列出了将从选项文件 (和环境变量中) 中拾取的值。`--print-defaults` 有助于验证选项文件设置的正确性。它也有助于识别 MySQL 正在使用的、而您从来就没有指定过的选项。可以使用 `--print-defaults` 确定选项是否是从某个选项文件中拾取

的。如果使用该选项，它必须在命令行的第一个位置。

`-O var-value, --set-variable var=value` 几个程序都有可设置的变量（操作参数）。`--set-variable` 选项可以设置这些变量。`var` 是分配给变量的值。变量的值以字节为单位，表示缓冲区的大小或长度。值可以由精确的数字表示，或以带有后缀 ‘K’ 或 ‘M’ 的数字表示以表明千字节或兆字节。后缀不是大小写敏感的，‘k’ 和 ‘m’ 等价于 ‘K’ 和 ‘M’。每个程序的变量都在该附录的程序描述中给出，如果调用带有 `--help` 选项的程序时，还可以显示这些变量。

`-s, --silent` 指定安静（silent）方式。这并不意味着程序是完全安静的，只不过它产生的数据结果比正常的要少。几个程序都允许指定该选项多次使得程序逐渐安静（该选项也可在选项文件中工作）。

`-S path_name, --socket=path_name` 对于客户机程序，它是在连接到服务器时要使用的套接字文件的全路径名。套接字文件用于对 `localhost` 的连接。

`-u user_name, --user=user_name` 对于客户机程序，它是在连接到服务器时要使用的用户名。如果没有指定该选项，则缺省值是您在 UNIX 中的注册名和 Windows 中的 ODBC。对于 `mysqld`，该选项指明（用于运行服务器的）UNIX 账号名（为使该选项更有效，服务器必须作为 `root` 来启动）。

`-v, --verbose` 指定详细方式。程序将产生比常规更多的输出结果。几个程序都允许多次指定该选项使程序逐渐详细（它也可在选项文件中工作）。

`-V, --version` 告诉程序打印版本信息串并退出。

E.1.2 选项文件

选项文件是在 MySQL 3.22 中引入的。它们允许存储 MySQL 程序选项，使得您不必每调用一个程序都在命令行输入选项。选项文件由 `isamchk`、`myisamchk`、`myisampack`、`mysql`、`mysqladmin`、`mysqld`、`mysqldump`、`mysqlimport`、`mysql.server` 和 `pack_isam` 读取。在 MySQL 安装目录的 `share/mysql` 目录中，或在源程序分发包的 `support-files` 目录中，可以找到一个实例选项文件 `my-example.cnf`。

选项文件中指定的任何选项都可以通过在命令行中直接指定该选项的另一个值而被忽略。

MySQL 程序通过咨询几个文件来查看选项，但是，丢失一个选项文件也不是什么错误。在 UNIX 中，可以对选项按给定的顺序查看下列文件：

| 文件名 | 内 容 |
|-----------------------------|---------|
| <code>/etc/my.cnf</code> | 全局选项 |
| <code>DATADIR/my.cnf</code> | 服务器专用选项 |
| <code>~/my.cnf</code> | 用户专用选项 |

在 Windows 中可读取下列文件：

| 文 件 名 | 内 容 |
|--------------------------------|---------|
| <code>SYSTEMDIR \my.ini</code> | 全局选项 |
| <code>c:\my.cnf</code> | 全局选项 |
| <code>DATADIR \my.cnf</code> | 服务器专用选项 |

`DATADIR` 表示机器中的数据目录的路径名（这是编译到服务器中的路径名，它不能用 `--`

datadir 选项改变)。在 Windows 中, DATADIR 是 c:\mysql\data。SYSTEMDIR 代表 Windows 系统目录的路径名(通常可能是 C:\Windows 或 C:\WinNT)。尽管 Windows 的选项文件名使用 ‘\’ 作为路径名部分的分隔符来指定,但在文件中指定的路径名应该使用 ‘/’ 作为路径名分隔符来书写。如果要想用 ‘\’ 作为分隔符,应使用 ‘\\’,因为 ‘\’ 被认为是转义符。

全局选项文件由能够理解选项文件的所有程序使用。服务器的数据目录中的文件只能由相应的服务器使用。用户专用的文件由该用户所运行的程序使用。

选项按组指定,例子如下:

```
[client]
user=paul
password=secret
```

```
[mysql]
no-auto-rehash
```

```
[mysqlshow]
status
```

组名在方括号中给出,通常它对应一个程序名。在上例中,[mysql] 指明 mysql 客户机的选项组,而 [mysqlshow] 指明 mysqlshow 的选项组。专用的组名 [client] 允许指定可应用于所有客户机程序的选项。标准的 MySQL 客户机程序可同时查看到 [client] 组和与客户机名相同名字的组。例如,mysql 可查看到 [client] 和 [mysql] 组,而 mysqlshow 可查看到 [client] 和 [mysqlshow] 组。

跟随在组名下面的任何选项都与该组相关。一个选项文件可能包含任何数量的组,后列出的组优先于先列出的组。如果一个给定的选项在一个程序可以查看到的多个组中多次出现,则最后列出的那个值被采用。

每个选项都应在单独的一行上指定。行中的第一个词是选项名,必须用不带前导虚线的长名字格式指定(例如,如果要在命令行中指定压缩选项,可使用 -c 或 --compress,但在选项文件中只能使用 compress)。程序支持的任何长格式的选项都可以列在一个选项文件中。如果选项需要值,则用 ‘=’ 作为分隔符列出名字和值。

考虑下列命令行:

```
mysql --compress --user=paul --set-variable max_allowed_packet=24M
```

为了在一个选项文件的 [mysql] 组中指定同样的信息,可按以下格式书写:

```
[mysql]
compress
user=paul
set-variable=max_allowed_packet=24M
```

可注意到,在选项文件中,除了在变量名和值之间的 ‘=’ 外, set-variable 后面还跟一个 ‘=’ 字符。

选项文件的行如果为空,或以 ‘#’ 或 ‘;’ 开始,则该行被认为是注释而忽略。

某些转义符序列可用于选项值以指明特殊的字符:

| 序列 | 含义 |
|----|----|
| \b | 退格 |
| \n | 换行 |
| \r | 回车 |

| | |
|----|-----|
| \s | 空格 |
| \t | Tab |
| \\ | 反斜杠 |

小心不要将选项放在 [client] 组中，因为这个组只能由一个单独的客户端来使用。例如，如果将 mysql 专有的 skip-line-numbers 选项放在 [client] 组中，将会发现其他的客户端程序（如 mysqlimport）不再工作（您会得到错误消息，后跟帮助消息）。反之，将 skip-line-numbers 移到 [mysql] 组中，一切正常。

由读取 [client] 组的所有 MySQL 程序可理解的选项是 compress、debug、help、host、password、pipe port、select、user 和 version（当然，将 help 或 version 放在选项文件中没有太大的意义）。

保持用户专用的选项文件为专有的

要确保每个用户专用的选项文件由适当的用户所拥有，并且将该方式设置为 600 或 400，使得其他的用户不能读取它。除了使用该文件的用户外，不要将 MySQL 用户名和口令信息暴露给任何人。

E.1.3 环境变量

MySQL 程序可查看到几个环境变量的值以获得选项设置。环境变量的优先权较低，用环境变量指定的选项可能会因在选项文件或命令行中所指定的选项而被忽略。。

MySQL 程序检查下列环境变量：

MYSQL_DEBUG 调试时使用的选项。除非 MySQL 是用调试支持环境建立的，否则该变量没有作用。设置 MYSQL_DEBUG 类似于使用 --debug 选项。

MYSQL_PWD 建立对 MySQL 服务器的连接时使用的口令。设置 MYSQL_PWD 类似于使用 --password 选项。

MYSQL_TCP_PORT 对于客户，它是建立服务器的 TCP/IP 连接时使用的端口号。对于 mysqld，它是监听 TCP/IP 连接的端口。设置 MYSQL_TCP_PORT 类似于使用 --port 选项。

MYSQL_UNIX_PORT 对于客户机，它是建立对运行在 localhost 上的服务器的 UNIX 域套接字连接时使用的套接字文件路径。对于 mysqld，它是监听本地连接的套接字。设置 MYSQL_UNIX_PORT 类似于使用 --socket 选项。

TMPDIR 创建临时文件的目录路径名。设置该变量类似于使用 --tmpdir 选项。

USER 它是在对服务器连接时使用的 MySQL 用户名。该变量只由运行在 Windows 中的客户端程序使用，设置它类似于使用 --user 选项。

mysql 客户端检查下列环境变量的值：

MYSQL_HISTFILE 用于存储命令行历史的文件的名字。如果该变量没有被设置，则缺省值为 \$HOME/.mysql_history，这里的 \$HOME 是主目录的位置。

MYSQL_HOST 在建立对服务器的连接时连接到的主机。设置该变量类似于使用 --host 选项。

E.2 isamchk 和 myisamchk

这些实用程序允许检查和修复损坏的表，显示表的信息，执行索引键值分布分析，以及禁止或允许索引。第 4 章提供了关于键分析和禁用索引的详细信息。第 13 章提供了有关表的检查和修复的详细信息。

myisamchk 用于使用较新的 MyISAM 存储格式的表。这些表有带有 .MYD 和 .MYI 后缀的数据文件名和索引文件名。可对使用旧 ISAM 存储格式的表使用 isamchk。这些表有带有 ISD 和 ISM 后缀的数据文件名和索引文件名。如果告诉任一个实用程序对一个错误类型的表进行操作，它将打印警告消息并忽略该表。

不要在 MySQL 服务器正访问一个表时执行该表的检查或修复！第 13 章讨论了如何防止服务器在 myisamchk 或 isamchk 对表进行处理时使用该表。

E.2.1 用法

```
isamchk [options] tbl_name[.ISM] ...
myisamchk [options] tbl_name[.MYI] ...
```

如果没有选项，这些实用程序将检查指定表的错误。否则，表将按照特定选项的含义进行处理。*tbl_name* 参数可以是表名或该文件的索引文件名（.ISM 或 .MYI）。使用索引文件名是方便的，因为您可以使用文件名的通配符，对在单个命令中指定存储类型的所有表进行操作。例如，可以按如下检查某个目录中的所有表：

```
% isamchk *.ISD
% myisamchk *.MYI
```

这些实用程序不对表文件的位置在哪里做判断。您必须指定这些文件的路径名（如果它们不在当前目录中）。由于表文件不被假定是存放在服务器的数据目录中，因此可以将表文件拷贝到另一个目录中并对这些拷贝而非原始文件进行操作。

有些选项涉及到索引的编号。索引从 1 开始编号。可以发布 SHOW INDEX 查询或使用 mysqlshow --key 命令来查找指定表的索引编号。key_name 列按 myisamchk 和 isamchk 查看索引的顺序列出索引。

MYSQL_PWD 是危险因素

使用 MYSQL_PWD 变量来存储口令产生了一个危险因素，这是因为系统中的其他用户可以轻易地发现口令的值。例如，PS 实用程序将显示其他用户的环境变量的设置。

E.2.2 isamchk 和 myisamchk 支持的标准选项

```
--debug          --set-variable  --verbose
--help           --silent        --version
```

--silent 选项表示只打印错误的消息。在任一个程序带有 --description 或 --extend-check 选项时，或在 myisamchk 带有 --check 选项时，--verbose 选项都将打印更多的信息。可以通过多次指定 --silent 和 --verbose 选项来增加效果。

E.2.3 isamchk 和 myisamchk 共有的选项

isamchk 和 myisamchk 有许多共有的选项：

-a, --analyze 指定键分布分析。它可以帮助服务器更快地执行基于索引的查找和连接。通过再次运行带有 --description 和 --verbose 选项的 isamchk 和 myisamchk 进行分析后，可获得有关键分布的信息。

-b *n*, --block-search=*n* 打印包含块 *n* 的块头部的表行的开始信息。它只用于调试操作。

-d, --description 打印关于表的描述信息。

-e, --extend-check 执行一个扩展的表检查。该选项很少需要使用，因为 isamchk 和 myisamchk 用不扩展的检查方式应该能发现错误。

-f, --force 迫使表被检查或修复，即使该表的临时文件已经存在也是如此。通常，如果 isamchk 和 myisamchk 发现一个 *tbl_name*.TMD 文件，在打印一个错误消息后将简单地退出，因为那表示该程序的另一个实例正在运行。但是，如果您在程序运行时取消它前面的调用，该文件还能够存在，在这种情况下该文件被安全地删除（另一种可选的方法是，手工删除该临时文件）。

如果在检查表时使用 --force，程序将利用 -recover 自动重新启动来检查任何有问题的表。

-i, --information 打印关于表的内容的统计信息。

-k *n*, --keys-used=*n* 与 --recover 一起使用。告诉 MySQL 只更新前 *n* 个索引。换句话说，它释放编号大于 *n* 的索引。该选项可用来提高 INSERT、DELETE 和 UPDATE 操作的性能。如果指定索引号为 0 将释放所有的索引。如果指定索引号为最大的索引编号将激活所有的索引。

-l, --no-symlinks 如果一个表参数是符号连接，则通常 symlink 所指向的表要修复。当指定该选项时，不跟随 symlink，而是将 symlink 替换成该文件的新版本。

-q, --quick 该选项与 --recover 一起使用将比单独使用 --recover 时修复得快。当给出这两个选项时不触及数据文件。为了迫使程序修改数据文件（即使发现重复的键值），指定 --quick 选项两次。如果使用该选项两次，肯定首先有一个数据文件备份。

-r, --recover 执行常规的恢复操作。它可以修正大多数问题，除了在索引文件中发现的重复值（它们应该是惟一的）外。

-o, --safe-recover 使用比 --recover 方法要慢的一种恢复方法，但它能够修复 --recover 不能修复的问题。

-S, --sort-index 将索引块排序以提高块顺序读取的速度。

-R *n*, --sort-recoerds=*n* 按照记录在索引 *n* 中的顺序对数据记录进行排序。基于该给定索引的后续检索应该更快。当首次对表执行该操作时，它可能很慢，因为记录将要重新排序。

-u, --unpack 解开一个打包文件，myisamchk 可以解开利用 myisamchk 打包的 MyISAM 文件。isamchk 可以解开利用 pack_isam 打包的 ISAM 文件。该选项可以用来将压缩的只读文件表转换成可修改的格式。

-w, --wait 如果一个表被锁定，要一直等待它可用为止。如果没有 --wait 选项，程序将为锁等待 10 秒中，如果没有得到任何的锁，将打印错误消息。

E.2.4 myisamchk 专有的选项

尽管 isamchk 没有专有的选项，但 myisamchk 有几个专有的选项。

-c, --check 检查表的错误。如果不指定任何选项，该选项是缺省的。

-D *n*, --data-file-length=*n* 当重新建立一个已经装满的数据文件时，该选项指定该数据文件应该允许增长的最大长度（当文件到达 MySQL 所施加的大小限制或由操作系统的大小约束所施加的大小限制时，这种情况会发生。它还发生在行数达到内部表的数据结构所施加的限制时）。该值以字节为单位指定。该选项只在与 -recover 或 --safe-recover 一起使用时有效。

-F, --fast 不检查自上次检查以来没有被修改的表。

-m, --medium-check 用一种比扩展检查更快的但稍有点不彻底的方法来检查表（myisamchk 帮助消息声称此方法“只能”发现 99.99% 的错误）。这种检查方法应该满足大多数的环境。medium 检查方法通过计算索引中键的 CRC 值进行工作，并将这些键与从数据文件的索引列中所计算出的 CRC 值进行比较。

-T, --read-only 不标记已经检查的表

-A [*n*], --set-auto-increment[=*n*] 迫使 AUTO_INCREMENT 值起始为 *n*（或为较高的值，如果该表已经包含带有与 *n* 一样大小的 AUTO_INCREMENT 值的记录）。如果没有指定任何值，则该选项将下一个 AUTO_INCREMENT 设置为一个比当前最大值更大的值。

-t *path_name*, --tmpdir=*path_name* 指定用于临时文件的目录路径名。如果该变量没有被设置，则缺省值为 TMPDIR 环境变量的值或 /tmp。

E.2.5 isamchk 和 myisamchk 的变量

以下的 isamchk 和 myisamchk 变量可以用 --set-variable 选项来设置：

key_buffer_size 索引块所使用的缓冲区的大小。

read_buffer_size 读缓冲区的大小。

write_buffer_size 写缓冲区的大小。

sort_buffer_size 键值的存储操作所使用的缓冲区大小（该变量用于 --recover，但不用于 --safe-recover）。

sort_key_blocks 该变量与索引使用的 B-树结构的深度有关。一般不需要修改它。

decode_bits 在对压缩的表进行解码时使用的比特位。值越大导致操作的速度越快，但将需要更多的内存。一般情况下使用缺省值就足够了。

E.3 myisampack 和 pack_isam

myisampack 和 pack_isam 实用程序产生压缩的只读表。它们可以使存储需求减少 40% 至 70% 以维持快速的记录访问。myisampack 将 MyISAM 表打包并处理所有列类型。pack_isam 将 ISAM 表打包并只处理不包含 BLOB 或 TEXT 列的表。

缺省设置时，myisampack 和 pack_isam 不包括在 MySQL 中，但可用于订购了 10 个或更多许可证的客户，或者是有扩展支持级或更高级别上的支持协议的客户。这些是可以改变的，以便使程序在与 MySQL 分发包相同的条件下可用（有关当前的支持环境和许可信息，请参阅 MySQL 参考手册）。

所有的 MySQL 版本都不需要读取已经用这些实用程序打包的表。这使得它们对这样的

应用特别有用：即希望分发一个表，该表包含归档的或百科全书式的信息，并且是只读的，无须更新。如果想将一个打包文件转换到解包的和可修改的格式，可以使用 `myisamchk --unpack`（对于 MyISAM 表）或 `isamchk --unpack`（对于 ISAM 表）来进行。

`myisampack` 和 `pack_isam` 对数据文件进行打包，但不对索引文件打包。为了更新索引，可在运行 `myisampack` 后运行 `myisampack -rq`，或在运行 `pack_isam` 后运行 `isamchk -rq`。

E.3.1 用法

```
myisampack [options] tbl_name ...
pack_isam [options] tbl_name ...
```

`tbl_name` 参数可以是表名也可以是该表的索引文件名（对于 MyISAM 表是 `.MYI` 文件，对于 ISAM 表是 `.ISM` 文件）。如果您不在那个目录中，应指定表所在的目录路径名。

E.3.2 `myisampack` 和 `pack_isam` 支持的标准选项

```
--debug          --silent          --version
--help           --verbose
```

E.3.3 `myisampack` 和 `pack_isam` 共有的选项

`-b, --backup` 告诉 `myisampack` 或 `pack_isam` 在对 `tbl_name` 打包前做每个 `tbl_name` 参数的备份，备份文件名为 `tbl_name.OLD`。

`-f, --force` 强迫表打包，即使作为结果的打包文件比原始文件大或者该表的临时文件已经存在也是如此。通常，`myisampack` 或 `pack_isam` 如果发现一个文件 `tbl_name.TMD`，则在打印一个错误消息后退出，这是因为那已经表明程序的另一个实例正在运行。但是，如果在其运行时取消程序的上一次调用，该文件可能还存在，这时该文件可以被安全地删除。如果您了解这种情况，可使用 `--force` 告诉任一个实用程序对该表打包，即使临时文件存在。

`-j join_tbl, --join=join_tbl` 将命令行指定的所有表连接到单个的打包表中，即 `join_tbl`。所有被合并的表必须有相同的结构（列名、类型和索引必须相同）。

`-t, --test` 以测试方式运行。运行打包测试，并打印有关在实际对表打包时应获得的信息。

`-T path_name, --tmpdir=path_name` 指定临时文件使用的目录路径名。

`-w, --wait` 等待并重新测试表是否在使用中。如果当在打包时表可能会更新，就不要对其进行打包。

E.3.4 `pack_isam` 专有的选项

尽管 `myisampack` 没有专有的选项，但 `pack_isam` 有一个，即 `-p n, --packlength=n`。该选项允许 `pack_isam` 使用一个 `n` 字节的记录长度存储尺寸，这里的 `n` 是一个从 1 到 3 的整数。`pack_isam` 试图自动地确定记录打包表中每个记录长度所需要的字节数。在某些情况中，它能够确定可使用的较小长度。如果是这样的话，`pack_isam` 为该结果发布一条消息。你可以

再次运行 pack_isam 并用 --packlength 选项明确指定较小的长度，以达到节省某些附加的空间。

E.4 mysql

mysql 客户机是一个交互式程序，它允许对服务器进行连接，发布查询并浏览结果。如果您将该命令的输入重定向为从一个文件中读取，mysql 还能以批操作方式进行使用以执行存储在该文件中的查询。例如：

```
% mysql -u paul -p -h pit-viper.snake.net samp_db < my_query_file
```

在交互式方式中，当 mysql 启动时，它显示一个提示符 'mysql>' 来表示正在等待输入。要想发布一个查询，在这里键入 mysql（如果需要可使用多行），然后键入 ';'（分号）或 '/g' 表示该查询的末尾。mysql 将该查询发送到服务器上，显示结果，然后打印另一个提示符以表示正准备接收另一个查询。

mysql 通过改变提示符来表示它正在等待您输入什么：

```
mysql>          等待一个新查询的首行
->              等待当前查询的下一行
'>              等待当前查询中的单引号串的结束
">              等待当前查询中双引号串的结束
```

'>' 和 '>' 提示符表示您已经在上一行中开始了一个单引号或双引号串，并且还未输入终结引号。通常，它在您忘记终止一个串时会发生。如果这种情况出现，要想离开串集合方式，可输入相配的引号（即由提示符指定的引号），后面跟 '\c' 以取消当前的查询。

当 mysql 用于交互式方式时，它在历史文件中保存查询。该文件在缺省时为 \$HOME/.mysql_history，或者可通过设置 MYSQL_HISTORY 环境变量来明确指定文件名。查询可以从该命令的历史中重新调用并重新发布，而不论是否进一步进行了编辑。以下列表出一些编辑命令（箭头键在 Windows 下不工作）：

| 键 序 列 | 含 义 |
|---------------|---------------|
| 上箭头, Ctrl-P | 调用上一行 |
| 下箭头, Ctrl-N | 调用下一行 |
| 左箭头, Ctrl-B | 向左移动光标（向后） |
| 右箭头, Ctrl-F | 向右移动光标（向前） |
| Escape Ctrl-B | 向后移动一个词 |
| Escape Ctrl-F | 向前移动一个词 |
| Ctrl-A | 移动光标到行头 |
| Ctrl-E | 移动光标到行尾 |
| Ctrl-D | 删除光标下的字符 |
| Delete | 删除光标左边的字符 |
| Escape D | 删除词 |
| Escape 空格 | 删除光标左边的词 |
| Ctrl-K | 删除从光标至行尾的所有内容 |
| Ctrl_ | 撤消上次的修改，可以重复 |

某些选项禁止使用历史文件。通常是一些表示mysql 非交互式使用的选项，如 --batch、--html 和 --quick。

E.4.1 用法

`mysql [options] [db_name]`

如果指定一个 `db_name` 参数，则该数据库成为您会话的当前（缺省的）数据库。如果没有指定 `db_name` 参数，则 `mysql` 的启动不带当前数据库，您需要或者限定所有的表带数据库名进行引用，或者发布 `USE db_name` 语句指定一个缺省的数据库。

E.4.2 mysql 支持的标准选项

| | | | |
|-------------------------|-----------------------------|------------------------|------------------------|
| <code>--compress</code> | <code>--password</code> | <code>--silent</code> | <code>--version</code> |
| <code>--debug</code> | <code>--pipe</code> | <code>--socket</code> | |
| <code>--help</code> | <code>--port</code> | <code>--user</code> | |
| <code>--host</code> | <code>--set-variable</code> | <code>--verbose</code> | |

`--silent` 和 `--verbose` 可以指定多次以增加效果。

E.4.3 mysql 专有的选项

`-B, --batch` 指定 `mysql` 应该以批方式运行。查询结果以制表符分隔形式显示（每行在一个单独的显示行上，列值之间用制表符分隔）。这对于生成要想导入到另一个程序（如电子表格）的结果特别方便。查询结果包括缺省时的一个列标题的初始行。为了禁用这些标题，可使用 `--skip-column-names` 选项。

`-T, --debug-info` 在程序终止时打印调试信息。

`-e, query, --execute=query` 执行查询并退出。应该用引号将该查询括起，以防外壳程序作为多命令行的参数来解释它。通过在查询串中使用分号来分隔查询，可以指定多个查询。

`-f, --force` 通常，当 `mysql` 从文件中读取查询时，如果出现错误则退出。该选项促使 `mysql` 保持正在处理的查询，而不管是否有错误。

`-H, --html` 产生 HTML 的输出结果。该选项是在 MySQL 3.22.26 中引入的。

`-i, --ignore-space` 指示服务器忽略在函数名和导入参数列表的 ‘(’ 字符之间的空格。通常，函数名后面必须紧跟括号，中间没有空格。

`-A, --no-auto-rehash` 当 `mysql` 启动时，它对数据库、表和列名进行散列以构造供名字快速完成使用的数据结构（您可以在输入一个查询时键入名字的初始部分，然后按 Tab 键。`mysql` 将完成该名字，除非该名字不明确）。该选项禁止散列计算，这样可使 `mysql` 启动得更快，尤其是有许多表的时候。如果想要在启动 `mysql` 后使用名字自动完成，可在 ‘`mysql>`’ 提示符下使用 `rehash` 命令。

`-o, --one-database` 该选项是在从更新日志文件的内容中更新数据库时使用。它指示 `mysql` 只更新缺省数据库（该数据库在命令行指定）。对其他数据库的更新将被忽略。如果命令行未指定任何数据库，则不执行任何更新。

`-q, --quick` 通常，`mysql` 在显示查询之前从服务器中检索整个查询结果。该选项使每行在检索时显示，它使用很少的内存并可以成功地完成一些大的查询。但是，该选项不应该对交互式使用。如果用户暂停输出或暂停 `mysql`，服务器将继续等待，可能干扰其他客户机。

-r, --raw 书写没有扩展任何特殊字符的列值。该选项与 --batch 选项组合使用。

-N, --skip-column-names 禁止列名作为查询结果的列标题显示。也可以通过指定 --silent 选项两次来达到相同的效果。

-L, --skip-line-numbers 在出现错误时不写行号。

-t, --table 以表格形式产生输出结果，每行值由横杠分隔并垂直排列。

-n, --unbuffered 在每个查询后刷新用于与服务器通信的缓冲区。

-E, --vertical 垂直打印查询结果，即，结果集的每个数据行（row）作为一个行（line）集显示，每列一行。每行由一个列名和值组成。在结果集中表示行号的行先于每个数据行的显示。垂直显示格式在查询产生很长的行时很有用。

如果没有指定该选项，可以通过用 ‘ \G ’ 而非 ‘ ; ’ ‘ \g ’ 终止单个的查询，来对这些查询开启垂直显示格式。

该选项是在 MySQL 3.22.5 中引入的。

-w, --wait 如果不能建立对服务器的连接，则等待并重试。

E.4.4 mysql 的变量

以下 mysql 的变量可用 --set-variable 选项来设置：

max_allowed_packet 用于服务器和客户机之间通信的缓冲区大小的最大值。

net_buffer_length 用于服务器和客户机之间通信的缓冲区的初始大小。该缓冲区最大可扩展到 max_allowed_packet 字节。

E.4.5 mysql 命令

mysql 除了允许向服务器发送 SQL 语句外，本身还认识几个命令。每个命令都必须在一个单独的行中给出；行的结尾不需要分号。大多数这些命令都有一个长格式（由词组成）和一个短格式（由虚线后跟单字符）。长格式的命令不区分大小写。短格式的命令必须使用下列清单中的大小写字母指定：

clear, \c 清除（取消）当前查询。当前查询是您正在输入的查询。该命令不取消已经发送到服务器的查询以及 mysql 正在为其显示输出结果的查询。

connect [db_name [host_name]], \r [db_name [host_name]] 连接到指定主机上的指定数据库。如果数据库名或主机名忽略了，则使用当前 mysql 会话中所使用过的最新值。

edit, \e 编辑当前查询。要使用的编辑器由 EDITOR 和 VISUAL 环境变量（或者是 vi，如果它们都没有被设置的话）的值决定。

ego, \G 将当前查询发送到服务器中并垂直排列地显示其结果。该命令是在 MySQL 3.22.11 中引入的。

exit 同 quit。

go, \g, ; 将当前查询发送到服务器中并显示其结果。

help, \h, ? 显示描述可用的 mysql 命令的帮助消息。

print, \p 打印当前查询（是查询文本本身，并非执行该查询所得到的结果）。

quit, \q 退出 mysql。

rehash, \# 重新计算数据库、表和列名自动完成所需要的信息。参见关于 --no-auto-

rehash 选项的描述。

stats, \s 检索和显示服务器的状态信息。如果要想检查服务器的版本和当前数据库，该选项有用。

use db_name, \u db_name 选择指定的数据库为当前数据库。

E.5 mysql.server

mysql.server 是一个脚本，它通过调用 safe_mysqld 启动和停止 mysqld 服务器。

用法

```
mysql.server start
mysqlserver stop
```

通常，mysql.server 用在 System V 风格的系统中并安装在 /etc 目录下的一个运行时目录中。系统通过调用带有 start 参数的该脚本在系统引导期间启动服务器。系统通过调用带 stop 参数的脚本在系统关闭期间关闭服务器。该脚本还可以用相应的参数手工调用以启动或关闭服务器。

E.6 mysqlaccess

该脚本允许连接到服务器、检索权限信息，以及测试指定用户权限的结果。它使用 mysql 数据库中的 user、db 和 host 表的拷贝来进行（但 mysqlaccess 不能用来试验表或列的权限）。也可以将您在这些临时表中做的改变提交回 mysql 数据库的实际表中。

要想使用 mysqlaccess，必须有足够的访问授权表的权限。

E.6.1 用法

```
mysqlaccess [host_name [user_name [db_name]]] options
```

E.6.2 mysqlaccess 支持的标准选项

```
--host --password --user --version
```

E.6.3 mysqlaccess 专有的选项

-b, --brief 以单行格式显示结果。

--commit 将临时授权表拷贝回到 mysql 数据库中。然后必须执行 mysqladmin flush-privileges 命令，使服务器注意到这些变化。

--copy 将授权表加载到临时表中。

-d db_name, --db=db_name 指定数据库名。

--debug=n 指定调试级别。n 应为 0 到 3 的整数。

--howto 显示某些例子，说明如何使用 mysqlaccess。

--old_server 说明该服务器的版本比 MySQL 3.21 的旧。这样将使 mysqlaccess 对发送到服务器的查询做某些调整。

--plan 显示 mysqlaccess 未来发行版中将计划增加的功能清单。

--preview 显示实际和临时授权表之间的权限差别。
--relnotes 打印 mysqlaccess 发行版的说明。
-H *host_name*, --rhost=*host_name* 指定要连接的远程服务器。
--rollback 撤消对临时授权表所做的改变。
-P *password*, --spassword=*password* 为 MySQL 超级用户（具有足够的修改授权表的权限的用户）指定口令。
-U *user_name*, --superuser=*user_name* 指定 MySQL 超级用户的用户名。
-t, --table 按表格形式显示结果。

E.7 mysqladmin

mysqladmin 实用程序与 MySQL 服务器进行通信来执行各种管理操作。可以使用 mysqladmin 从服务器的操作中获得信息或控制服务器的操作、设置口令，以及创建或删除数据库。

E.7.1 用法

```
mysqladmin [options] command ...
```

E.7.2 mysqladmin 支持的标准选项

| | | | |
|------------|------------|----------|-----------|
| --compress | --host | --port | --user |
| --debug | --password | --silent | --version |
| --help | --pipe | --socket | |

如果 mysqladmin 不能连接到服务器的话，--silent 将导致 mysqladmin 默默地退出。

E.7.3 mysqladmin 专有的选项

-f, --force 该选项有两个作用。首先，它可使 mysqladmin 不对 *db_name* 命令进行确认。其次，当在该命令行上指定多个命令时，mysqladmin 试图执行每个命令，即使出现了错误。通常，mysqladmin 将在第一个错误出现后退出。
--relative 显示在以 --sleep 方式使用时的当前的和以前的值之间的差别。通常，该选项只与 extended-status 命令一起工作。
-i *n*, --sleep=*n* 以 *n* 秒的间隔重复执行命令行中指定的命令。
-t *n*, --timeout=*n* 在当试图连接到服务器时在超时前等待 *n* 秒。该选项是在 MySQL 3.22.1 版本中引入的。
-w [*n*], --wait[=*n*] 指定不能建立对服务器的连接时的等待和重试次数。如果未指定 *n* 的值，则缺省值为 1。

E.7.4 mysqladmin 的命令

在命令行的任何参数之后，可以指定一个或多个以下的命令。命令的名字可缩短成一个词头，只要该词头是惟一的即可。例如，processlist 可以缩短成 process 或 proc，但不能是 p。

有几个命令等价于 SQL 语句，它们在描述中说明。有关这些 SQL 语句的含义的详细信息，请参阅附录 D。

`create db_name` 创建指定名称的新数据库。

该命令类似于 SQL 语句 `CREATE DATABASE db_name`。

`drop db_name` 删除指定名的数据库和在该数据库中的任何表。要小心使用该命令，因为不能使该数据库再恢复。除非给出 `--force` 选项，否则 `mysqlamin` 要对该命令进行确认。

该命令类似于 SQL 语句 `DROP DATABASE db_name`。

`extended-status` 显示服务器状态变量的名字和值。

该命令类似于 SQL 语句 `SHOW STATUS`。

`extended-status` 命令是在 MySQL 3.22.10 中引入的。

`flush-hosts` 刷新主机高速缓存。

该命令类似于 SQL 语句 `FLUSH HOSTS`。

`flush-logs` 刷新（关闭并重新打开）日志文件。

该命令类似于 SQL 语句 `FLUSH LOGS`。

`flush-privileges` 重新加载授权表。

该命令类似于 SQL 语句 `FLUSH PRIVILEGES`。

`flush-status` 清除状态变量（它将几个计数器复位）。

该命令类似于 SQL 语句 `FLUSH STATUS`

`flush-tables` 刷新表的高速缓存。

该命令类似于 SQL 语句 `FLUSH TABLES`。

`kill id,id, ...` 取消由给定的标识符号指定的服务器线程。如果指定多个号，则 ID 列表应该不包含空格以便它不会与跟在 `kill` 命令后面的另一个命令相混淆。要想查找当前有哪些线程在运行，可使用 `mysqladmin processlist`。

该命令类似于对每个线程 ID 发布的 SQL 语句 `KILL`。

`password new_password` 当运行 `mysqladmin` 命令时为连接到服务器的 MySQL 用户改变口令。新的口令是 `new_password`。您必须能作为 MySQL 用户（是要修改口令的用户）先连接到服务器，这起到了对您是否知道旧口令的验证。

该命令类似于 SQL 语句 `SET PASSWORD`。

`password` 命令是在 MySQL 3.22 中引入的。

`ping` 检查 MySQL 服务器是否在运行。

`processlist` 显示当前正在运行的服务器线程的列表。

该命令类似于 SQL 语句 `SHOW PROCESSLIST`。

`refresh` 刷新表的高速缓存并关闭和重新打开日志文件。

`reload` 重新加载授权表。

该命令类似于 SQL 语句 `FLUSH PRIVILEGES`。

`shutdown` 指示服务器关闭。

`status` 显示服务器简短的状态消息。

`variables` 显示服务器变量的名字和值。

该命令类似于 SQL 语句 `SHOW VARIABLES`。

`version` 检索并显示服务器版本信息串。

E.8 mysqlbug

MySQL 参考手册的第 2 章包含提交故障报告的详细过程。该过程后面的帮助信息确保您的报告是有用的并包含有足够的信息以解决所报告的问题。该过程的关键部分是使用 `mysqlbug` 脚本。它可用来在您发现有关 MySQL 的问题时创建和发送一个故障报告到 MySQL 邮件清单中。`mysqlbug` 收集有关系统和 MySQL 配置的信息，然后带您进入一个编辑器，其中包含要发送的该邮件消息的内容。编辑该消息以便尽可能地增加有关正在报告的该问题的信息，然后保存该消息的变化并退出编辑器。`mysqlbug` 将询问是否要发送该报告，确认后发出。

请使用 `mysqlbug` 来报告故障，但不要轻易地使用它。在许多情况中，“故障”其实根本就不是故障，或者是一个已经编入参考手册中的某个问题的报告。例如，有一个作为故障被报告了几次的问题是：语句 `DELETE FROM tbl_name` 报告了一个受影响的行值为零。这是事实，但它是预期的行为。它也在参考手册中做了说明，如果您想要得到真正的计数就不应该使用这个查询。MySQL 邮件清单档案是另一个有用的信息来源，可以在 <http://www.mysql.com/doc.html> 找到它们。

用法

```
mysqlbug [address]
```

故障报告缺省时被发送到 MySQL 的邮件清单中。如果您在命令行指定一个电子邮件地址，报告将送至该地址。您可以指定自己的地址来测试 `mysqlbug`，而不是发送报告到邮件清单中。对于首次使用该脚本并且不清楚它是如何工作的您来说，这个主意不坏。

E.9 mysqld

`mysqld` 是 MySQL 服务器。它向客户机程序提供数据库访问，因此，它必须一直运行，否则客户机不能使用由该服务器管理的数据库。当 `mysqld` 启动时，它打开网络端口进行监听，然后等待客户机的连接。`mysqld` 是多线程的并使用一个单独的线程处理每个客户机的连接以提供客户机间的并发性。写给服务器的查询被自动执行，当服务器开始执行像这样的查询时，它将不执行含有该表的其他查询，直到当前查询结束为止。由于这个原因，两个客户机不能同时在表中修改同一行。

E.9.1 用法

```
mysqld [options]
```

E.9.2 mysqld 支持的标准选项

```
--debug      --port      --user
--help       --socket    --version
```

请注意，尽管它支持 `--socket`，但不支持 `-S`，因为服务器使用 `-Sg` 选项，其含义与套接字文件无关。

如果给出 `--user` 选项，它指定运行服务器时要使用的账号的用户名。此时，当服务器启动

时，它从口令文件中查找该用户和该账号组 ID 的值，然后修改其用户和相配的组 ID。这样，服务器将使用与该用户相关的权限进行运行，而非 root 的权限（为了使 --user 选项有效，服务器必须作为 root 启动，否则它将不能改变其用户的 ID）。该选项是在 MySQL 3.22 中引入的。

E.9.3 mysqld 专有的选项

--b *path_name*, --basedir=*path_name* 指定 MySQL 安装目录的路径名。如果它们以相对的路径名给出，则许多其他的路径名被认为是在该目录下的。

--big-tables 通过将所有临时结果保存到磁盘而不是将其保存在内存中，允许处理大的结果集。这样避免了大多数作为内存不足的结果的“table full（表已满）”消息出现，以便存放大的结果集。该选项自 MySQL 3.23 以来不再需要。

--bind-address=*ip_addr* 连接指定的 IP 地址。通常，mysql 与服务器运行的主机的缺省 IP 地址连接。如果主机有多个地址的话，该选项可用来选择一个要连接的地址。

--bootstrap 该选项在您首次安装 MySQL 时由安装脚本使用。

--chroot=*path_name* 运行固定到特定的根目录上的 MySQL 服务器。该选项指定该目录的路径名。有关在 chroot()-ed 环境中运行的详细信息，请参阅 chroot() UNIX 人工页。该选项是在 MySQL 3.22.2 中引入的。

--h *path_name*, --datadir=*path_name* 指定到 MySQL 数据目录的路径名。

--default-table-type=*type* 指定缺省的表存储类型。type 值应该为 isam、myisam 或 heap（该值不区分大小写的）。如果该选项没有指定，则将缺省使用 MyISAM 表。该选项是在 MySQL 3.23 中引入的。

--enable-locking 对于像 Linux 这样的系统，允许外部锁定（文件系统锁定），缺省为 --skip-locking。该选项是在 MySQL 3.22.4 中引入的。

--T [*n*], --exit-info[=*n*] 在程序终止时打印调试信息。该选项是在 MySQL 3.22 中引入的。

--init-file=*file_name* 指定在启动时要执行的 SQL 语句的文件。

--L, --language=*lang_name* 用指定的语言向客户机显示错误消息。通常，lang_name 将是 english 或 german 这样的值。

--l [*path_name*], --log[=*path_name*] 开启常规日志文件。常规日志包含有关客户机连接和查询的信息。如果没有给出 *path_name*，则该日志文件名为 *host_name.log*，这里的 *host_name* 是服务器主机的名字。如果 *path_name* 作为相对路径给出，它将被认为是在数据目录下的。

--log-isam[=*file_name*] 开启索引文件。该选项只用于调试 ISAM/MyISAM 的操作。

--log-long-format 将附加信息写到更新日志中。该选项是在 MySQL 3.22.7 中引入的。

--log-update[=*path_name*] 开启更新日志文件。更新日志包含修改数据库表的任何查询的文本。如果没有给出 *path_name*，则更新文件名为 *host_name.nnn*，这里的 *host_name* 是服务器主机的名字，*nnn* 是比前一个更新日志大 1 的序号。

如果给出 *path_name* 且最后的部分中没有扩展名，则服务器增加一个格式为 *nnn* 的数值扩展名（如前面所描述的）。如果路径名的最后部分中包含扩展名，则该名字作为更新日志文件名使用，而无须修改。如果 *path_name* 以相对路径给出，则它被认为是在数据目录下的。

--low-priority-updates 给予写操作比检索低的优先权。该选项是在 MySQL 3.23 中引入的（从 MySQL 3.22.5 到 MySQL 3.23，它被称为 --low-priority-inserts）。

-n, --new 使用新的、可能不安全的例程。这些例程是还未表明稳定性的 MySQL 的功能。该选项只有在您喜欢冒险时使用。

-o, --old-protocol 使用 MySQL 3.21 以前的客户机/服务器通信使用的协议。该选项在服务器与版本很旧的客户机程序进行通信时可能需要。

--one-thread 使用单个线程运行，用于在 Linux 中的调试，因为通常最少使用三个线程。该选项是在 MySQL 3.22.2 中引入的。

--pid-file=path_name 当 mysql 启动时，它将其进程的 ID（PID）写入文件中。该选项指定 PID 文件的路径名。该文件可以由确定服务器进程号的其他进程使用，尤其是为发送一个信号的目的。例如，mysql.server 在它向服务器发送一个关闭信号时读取该文件。

--safe-mode 该选项类似于 --skip-new，但禁止更多的东西。如果 MySQL 出现不稳定或复杂的查询产生不正确的结果时可试一试该选项。如果使用该选项提高服务器的操作，则在使用 mysqlbug 报告您遇到的问题时应注释该情况。

-Sg, --skip-grant-tables 不使用授权表检验客户机的连接。该选项给予任何客户机做任何事情的所有访问权。通过发布 mysqladmin flush-privileges 命令可以指示服务器开始使用授权表。

--skip-host-cache 禁止使用主机名高速缓存。

--skip-locking 禁止使用系统锁定（文件系统锁定）。如果使用该选项，则必须在使用 isamchk 或 myisamchk 修复表之前关闭服务器，否则将有毁坏这些表的危险。

--skip-name-resolve 不解决主机名。如果指定该选项，授权表必须由 IP 号或以 localhost 指定主机

--skip-networking 不允许 TCP/IP 的连接。客户机必须使用 localhost 从本地主机中连接。

--skip-new 不使用新的、可能不安全的例程。参见 --new。

--skip-show-database 不允许未授权用户发布 SHOW DATABASES 查询或在用户没有任何访问权的数据库上使用 SHOW TABLES。该选项时在 MySQL 3.23 中引入的。

--skip-thread-priority 通常，修改表的查询比检索数据的查询更优先。如果不希望这样，该选项使得服务器将不给予不同类型的查询以不同的优先级。

-t path_name, --tmpdir=path_name 指定临时文件使用的目录路径名。该选项是在 MySQL 3.22.4 中引入的。

Windows 专有的选项如下：

--console 为错误消息显示一个控制台窗口。该选项是在 MySQL 3.22.4 中引入的。

--flush 在每次更新后刷新磁盘上的所有表。这减少了崩溃事件中表讹误的风险，但严重降低了性能。该选项是在 MySQL 3.22.9 中引入的。

--install 安装作为服务的 mysqld-nt（只对 Windows NT）。

--remove 删除作为服务的 mysqld-nt（只对 Windows NT）。

--standalone 启动作为独立程序的而非作为服务的 mysqld-nt（只对 Windows NT）。

E.9.4 mysqld 的变量

要查看 mysql 将使用哪些缺省值，可用下列命令：

```
% mysqld --help
```

要查看当前正运行的 mysqld 使用了哪些值，可用以下命令：

```
% mysqladmin variables
```

通过发布 SHOW VARIABLES 查询还可以检查当前变量的值。能使用 --set-variable 选项设置的 mysqld 变量如下：

back_log 未决的连接请求数量，这些请求可能在处理当前连接时排队等待。

connect_timeout 在初始连接信号交换时 mysql 等待报文分组的秒数。

delayed_insert_timeout 当 INSERT DELAYED 操作的处理完成插入排队的行时，它等待该变量值这么多秒钟，看看是否有任何新的 INSERT DELAYED 行到达。如果有，它将处理它们，否则终止。

delayed_insert_limit 来自 INSERT DELAYED 语句中的行数，这些行在被插入到表中之前检查是否有针对该表的任何新的 SELECT 语句到达。如果有，则插入操作暂停，让检索操作执行。

delayed_queue_size 为 INSERT DELAYED 语句排队等待的行数。如果队列满，将阻塞后面的 INSERT DELAYED 语句，直到队列中有空间为止。

flush_time 如果该变量有非零值，则关闭表，每 flush_time 秒向磁盘刷新一次未决变化。Windows 的缺省值为 30 秒；使用 --flush 选项可以改变该值。

join_buffer_size 全连接缓冲区的尺寸（即不使用索引来完成的连接的缓冲区）。

在 MySQL 3.23 之前该变量称为 join_buffer。

key_buffer_size 索引块使用的缓冲区的大小。该缓冲区在连接处理程序之间共享。

该选项在 MySQL 3.23 之前称为 key_buffer。

long_query_time 定义一个“慢”查询的秒数。任何花费的时间比该值长的查询将引起 Slow_queries 计数器增加。

max_allowed_packet 服务器和客户机之间进行通信所使用的缓冲区尺寸的最大值。分配给该缓冲区的初始值为 net_buffer_length 字节，但可根据需要增长到最多 max_allowed_packet 字节。

max_connections 所允许的同时发生的客户机连接的最大数量。

max_connect_errors 在阻塞主机后面的连接请求之前允许的该主机不成功连接的最大次数。使用这个变量是因为某人可能打算从该主机中闯入。FLUSH HOST 语句或 mysqladmin flush-host 命令可以用来清除该主机的高速缓存以便重新使阻塞的主机有效。

max_delayed_insert_threads 将要启动处理 INSERT DELAYED 语句的处理程序的最大个数。在已经到达处理程序最大值的情况下被接收的任何这样的语句将认为是非 DELAYED 语句。

max_join_size MySQL 优化器估计一个连接将返回多少行。如果该估计超过 max_join_size 行，将返回错误。如果用户喜欢编写其返回的行数无规律的 SELECT 语句，则该变量会有帮助。

max_sort_length 使用每个值的第一个 max_sort_length 字节存储 BLOB 或 TEXT 的值。

net_buffer_length 服务器和客户机之间进行通信所使用的缓冲区大小的初始值。该缓

缓冲区最大可扩充到 `max_allowed_packet` 字节。

`record_buffer` 执行连续表扫描的线程所使用的缓冲区大小。

`sort_buffer` 执行排序操作的线程（`GROUP BY` 或 `ORDER BY`）所使用的缓冲区的大小。每个线程一个值。通常，如果您有许多同时进行排序的线程，则使该值太大是不明智的（超过 1MB）。

`table_cache` 可打开的表的最大数量。该高速缓存在线程之间共享。

`tmp_table_size` 临时表所允许的最大字节数。如果表超过这个大尺寸，`mysqld` 将向客户机返回一个错误消息。

`thread_stack` 每个线程的堆栈大小。

`wait_timeout` 在服务器关闭一个连接之前该保持连接空闲的秒数。

E.10 mysqldump

`mysqldump` 程序将数据库表的内容写到文本文件中。这些文件可用于各种目的，如数据库备份、移动数据库到另一个服务器中，或基于已有数据库的内容建立测试数据库。

在缺省设置时，每个转储表的输出结果由重新创建该表的 `CREATE TABLE` 语句组成，后跟一组重新加载该表的内容的 `INSERT` 语句。如果指定 `--tab` 选项，表的内容将以行的形式写入数据文件，并且创建表的 `SQL` 语句写入一个单独的文件中。

E.10.1 用法

```
mysqldump [options] db_name [tbl_name] ...
```

如果没有指明跟在数据库名 `db_name` 后的任何表，则该数据库中的所有表都进行转储。否则只转储指定的表。

使用 `mysqldump` 最常用的方法如下：

```
% mysqldump --opt db_name > backup_file
```

E.10.2 mysqldump 支持的标准选项

| | | | |
|-------------------------|-------------------------|-----------------------------|------------------------|
| <code>--compress</code> | <code>--host</code> | <code>--port</code> | <code>--user</code> |
| <code>--debug</code> | <code>--password</code> | <code>--set-variable</code> | <code>--verbose</code> |
| <code>--help</code> | <code>--pipe</code> | <code>--socket</code> | <code>--version</code> |

E.10.3 mysqldump 专有的选项

下列选项控制怎样操作 `mysqldump`。下一节“数据格式选项”将介绍可与 `--tab` 选项组合使用来指定数据文件格式的选项。

`--add-drop-table` 在每个 `CREATE TABLE` 语句之前增加 `DROP TABLE IF EXISTS` 语句。

`--add-locks` 在加载每个表数据的 `INSERT` 语句组前后增加 `LOCK TABLE` 和 `UNLOCK TABLE`。

`--allow-keywords` 允许创建是关键字的列名。

`-c`, `--complete-insert` 使用 `INSERT` 语句指明要插入的每个列。

`--delayed-insert` 写 `INSERT DELAYED` 语句。

`-e`, `--extended-insert` 写插入多行的 `INSERT` 语句。

-F, --flush-logs 在开始转储前刷新服务器日志文件。

-f, --force 继续，即使有出现错误。

--full 增加对 mysqldump 产生的 CREATE TABLE 语句的附加信息，如表的类型、起始的 AUTO_INCREMENT，等等。这是您可以在 CREATE TABLE 语法的 table_options 部分指定的信息（参见附录 D）。

-l, --lock-tables 在转储表之前得到对所有表的锁。

-t, --no-create-info 不写 CREATE TABLE 语句。

-d, --no-data 不写表数据。

--opt 优化表的转储速度并写转储文件，该文件的重新加载速度是最佳的。该选项开启 --add-drop-table、--add-locks、--all --extended-insert、--quick 和 --lock-tables。

-q, --quick 在缺省时，mysqldump 将表的全部内容读到内存中，然后再进行写操作。该选项使每行（只要一从服务器中读出来）被写到输出文件中。

如果使用该选项，不应该挂起 mysqldump；这会导致服务器等待，干扰其他的客户机。

-T *dump_dir*, --tab=*dump_dir* 该选项使 mysqldump 对每个表写两个文件，文件的位置为 *dump_dir*。该目录必须是已经存在的。对于每个表 *tbl_name*，写文件 *dump_dir/table_name.txt* 和 *dump_dir/table_name.sql*，前者包含该表的数据，后者是该表的 CREATE TABLE 语句。使用该选项必须具有 FILE 的权限。

为了避免混淆，当使用该选项时最好在服务器主机上运行 mysqldump。否则，有些文件被写在服务器上而另一些则写在客户机中。对于 *.sql 文件，*dump_dir* 用于服务器主机中，对于 *.txt 文件，*dump_dir* 用于客户机。*.sql 文件将属于用来运行服务器的账号，*.txt 文件将属于您自己。

缺省设置时，数据文件包含换行终止（newline-terminated）格式的行，由制表符分隔的列值组成。该格式可以用“数据格式选项”中介绍的选项改变。

-w *where_clause*, --where=*where_clause* 只转储通过由 *where_clause* 指定的 WHERE 子句选择的记录。应该用引号将该子句括起来，以防止外壳程序将其解释为多个命令行的参数。该选项是在 MySQL 3.22.7 中引入的。

E.10.4 数据格式选项

如果指定 --tab 或 -T 选项为每个表生成单独的数据文件，则有几个附加的选项。您可能需要用适当的引用字符将该选项的值括起来。这些选项类似于 LOAD DATA 语句的数据格式选项。参见附录 D 的 LOAD DATA 条目。

--fields-enclosed-by=*char* 指明列值应括在指定的字符中，通常是引号字符。缺省时列值不括在任何字符中。

--field-escaped-by=*char* 为转义特殊的字符指定转义符。缺省时无转义符。

--fields-optionally-enclosed-by=*char* 表明列值应括在指定的字符中，通常使用引号字符。该字符用于非数值的列。缺省时列值不用任何字符括起。

--fields-terminated-by=*char* 指定文件所使用的列值分隔符。缺省设置时，列值由制表符分隔。

--lines-terminated-by=*str* 指定表示输出结果行末尾的串（可以是多字符的）。缺省时用换行符。

E.10.5 mysqldump 的变量

以下 mysqldump 变量可用 --set-variables 选项设置：

max_allowed_packet 用于服务器和客户机之间通信的缓冲区尺寸的最大值。

net_buffer_length 用于服务器和客户机之间通信的缓冲区的初始大小。该缓冲区最大可扩充到 max_allowed_packet 字节。

E.11 mysqlimport

mysqlimport 实用程序是一个批量加载器，它将文本文件的内容读到已有表中。它对 LOAD DATA 的 SQL 语句起到命令行接口的作用，是将行输入到表中的有效方法。

E.11.1 用法

```
mysqlimport [options] db_name file_name ...
```

db_name 参数指定数据库，该数据库中包含您要加载数据的表。要加载数据的表由文件名参数确定。对于每个文件名，去除其任何扩展名，留下的基名就作为要将文件加载进来的表名。例如，mysqlimport 将加载 president.txt 的内容到 president 表中。

E.11.2 mysqlimport 支持的标准选项

| | | | |
|------------|------------|----------|-----------|
| --compress | --host | --port | --user |
| --debug | --password | --silent | --verbose |
| --help | --pipe | --socket | --version |

E.11.3 mysqlimport 专有的选项

下列选项控制 mysqlimport 如何处理输入文件。下一节“数据格式选项”将介绍用来指定输入文件数据格式的选项。

-d, --delete 在加载任何数据到表中之前清空每个表。

-f, --force 继续加载行，即使出现错误。

-i, --ignore 当一个输入行包含某个唯一键值，且该值已存在于表中，则保留已有的行并丢弃该输入行。

-L, --local 缺省时，mysqlimport 允许服务器读数据文件，这意味着文件必须在服务器主机中加载，并且您必须有 FILE 权限。指定 --local 选项将指示 mysqlimport 读取数据文件本身并将它发送到服务器中。如果您没有 FILE 权限，在另一台机器上运行 mysqlimport 与在服务器主机上相比，以及包括在服务器主机上，该选项尽管工作但较慢。

--local 选项是在 MySQL3.22.15 中引入的

-l, --lock-tables 在加载数据到表中以前锁定每个表。

-r, --replace 当一个输入行包含一个唯一键值，且该值已经存在于表中，则可用输入行替换已有的行。

E.11.4 数据格式选项

缺省时，mysqlimport 假定数据文件包含换行终止行，并由制表符分隔的值组成。可以使

用下列选项更改成希望的格式。可能需要用适当的引用符号将该选项值括起。这些选项类似于 LOAD DATA 语句的数据格式选项。参见附录 D 的 LOAD DATA 条目。

--fields-enclosed-by=*char* 指明列值应括在指定的字符中，通常用引号。缺省时，假定列值不括在任何字符中。

--fields-escaped-by=*char* 指定用于转义特殊字符的转义符。缺省表示无转义符。

--fields-optionally-enclosed-by=*char* 表示列值应括在指定字符中，通常是引号。

--fields-terminated-by=*char* 指定分隔列值的字符。缺省时，假定列值由制表符分隔。

--lines-terminated-by=*str* 指定结束输入行的串（可以是多字符的）。缺省时假定由换行符终止行。

E.12 mysqlshow

mysqlshow 显示数据库、数据库中的表，或有关表中列或索引的信息。它担当 SHOW SQL 语句的命令行接口的作用。

E.12.1 用法

```
mysqlshow [options] [db_name [tbl_name [col_name]]]
```

如果没有指定数据库名，mysqlshow 将列出服务器主机中所有数据库。如果指定数据库名但未指定表名，则列出该数据库中所有的表。如果数据库和表名都被指定，但未指定列名，则列出该表的所有列。如果所有的名字都被指定，mysqlshow 将显示有关指定列的信息。

如果最后的参数中包含外壳程序通配符（‘*’或‘?’），输出结果将被限制为与该通配符相配的值。‘*’和‘?’被视为 LINK 操作符的‘%’和‘_’SQL 通配符。

E.12.2 mysqlshow 支持的标准选项

| | | | |
|------------|------------|----------|-----------|
| --compress | --host | --port | --version |
| --debug | --password | --socket | |
| --help | --pipe | --user | |

E.12.3 musqlshow 专有的选项

-i, --status 显示由 SHOW TABLE STATUS 语句显示的相同类型的表的信息。

--status 选项是在 MySQL 3.23 中引入的。

-k, --keys 显示除表列信息外的表索引信息。该选项只在指定表名时才有意义。

E.13 safe_mysql

safe_mysql 是启动 mysql 服务器和监控程序的一个脚本。如果服务器死机，safe_mysql 重新启动它。

用法

safe_mysql 理解与 mysql 相同的选项。