**环境变量**

各种 Git 命令使用以下环境变量：

**Git 存储库**

这些环境**变量适用于所有核心** Git 命令。注意：值得注意的是，它们可能会被位于Git上方的SCMS使用/覆盖，因此如果使用外部前端，请小心。

**GIT\_INDEX\_FILE**

此环境允许指定备用索引文件。

$GIT\_DIR/index如果未指定，则使用默认值 。

**GIT\_INDEX\_VERSION**

此环境变量允许为新存储库指定索引版本。

它不会影响现有索引文件。默认情况下，使用索引文件版本 2 或 3。

参见 [git-update-index[1]](https://git-scm.com/docs/git-update-index) 了解更多信息。

**GIT\_OBJECT\_DIRECTORY**

如果通过此环境变量指定对象存储目录，则会在下面创建 sha1 目录 - 否则将使用默认目录。$GIT\_DIR/objects

**GIT\_ALTERNATE\_OBJECT\_DIRECTORIES**

由于 Git 对象的不可变性，旧对象可以存档到共享的只读目录中。

此变量指定一个 “：” 分隔（在 Windows 上为 “;” 分隔）的 Git 对象目录列表，可用于搜索 Git 对象。新对象将不会写入这些目录。

以（双引号）开头的条目将被解释为 C 样式的带引号路径，删除前导和尾随双引号并遵循反斜杠转义。例如，该值有两个路径：和 。""path-with-\"-and-:-in-it":vanilla-pathpath-with-"-and-:-in-itvanilla-path

**GIT\_DIR**

如果设置了环境变量，则它指定要使用的路径，而不是存储库基的默认值。命令行选项也会设置此值。GIT\_DIR.git--git-dir

**GIT\_WORK\_TREE**

设置工作树根目录的路径。这也可以由命令行选项和 core.worktree 配置变量控制。--work-tree

**GIT\_NAMESPACE**

设置 Git 命名空间;有关详细信息[，请参阅gitnamespaces[7]](https://git-scm.com/docs/gitnamespaces)。命令行选项也会设置此值。--namespace

**GIT\_CEILING\_DIRECTORIES**

这应该是以冒号分隔的绝对路径列表。如果设置，它是一个目录列表，Git 在查找存储库目录时不应将其 chdir 放入这些目录（对于排除加载缓慢的网络目录很有用）。

它不会排除当前工作目录或在命令行或环境中设置GIT\_DIR。通常，Git 必须读取此列表中的条目并解析可能存在的任何符号链接，以便将它们与当前目录进行比较。

但是，如果即使此访问速度很慢，您也可以向列表中添加一个空条目，以告诉 Git 后续条目不是符号链接，不需要解析;例， .GIT\_CEILING\_DIRECTORIES=/maybe/symlink::/very/slow/non/symlink

**GIT\_DISCOVERY\_ACROSS\_FILESYSTEM**

在没有“.git”存储库目录的目录中运行时，Git 会尝试在父目录中查找此类目录以查找工作树的顶部，但默认情况下它不会跨越文件系统边界。

此环境变量可以设置为 true，以告诉 Git 不要在文件系统边界处停止。

就像 ，这不会影响通过命令行或在命令行上设置的显式存储库目录。

GIT\_CEILING\_DIRECTORIESGIT\_DIR

**GIT\_COMMON\_DIR**

如果将此变量设置为路径，则通常位于 $GIT\_DIR 中的非工作树文件将从此路径中获取。

特定于工作树的文件（如 HEAD 或索引）取自 $GIT\_DIR。

有关详细信息[，请参阅 gitrepository-layout[5]](https://git-scm.com/docs/gitrepository-layout) 和 [git-worktree[1]](https://git-scm.com/docs/git-worktree)。此变量的优先级低于其他路径变量，如GIT\_INDEX\_FILE、GIT\_OBJECT\_DIRECTORY...

**GIT\_DEFAULT\_HASH**

如果设置了此变量，则新存储库的默认哈希算法将设置为此值。

克隆时当前忽略此值。

将改用远程存储库的设置。默认值为“sha1”。

这个变量是实验性的！参见 [git-init[1]](https://git-scm.com/docs/git-init)。--object-format