**配置机制**

Git 使用简单的文本格式来存储每个存储库和每个用户的自定义项。

这样的配置文件可能如下所示：

#

# A '#' or ';' character indicates a comment.

#

; core variables

[core]

; Don't trust file modes

filemode = false

; user identity

[user]

name = "Junio C Hamano"

email = "gitster@pobox.com"

从配置文件中读取各种命令并相应地调整其操作。参见 [git-config[1]](https://git-scm.com/docs/git-config) 获取列表和有关配置机制的更多详细信息。

**标识符术语**

**<对象>**

指示任何类型的对象的对象名称。

**<博客>**

指示 Blob 对象名称。

**<树>**

指示树对象名称。

**<承诺>**

指示提交对象名称。

**<树形>**

指示树、提交或标记对象名称。

采用<树式>参数的命令最终希望对<树>对象进行操作，但会自动取消引用指向<树><提交>和<tag>对象。

**<承诺式>**

指示提交或标记对象名称。

采用<提交>参数的命令最终希望对<提交>对象进行操作，但会自动取消引用指向<提交><tag>对象。

**<类型>**

指示对象类型是必需的。

当前为：blobtree、commit、或 tag之一。

**<文件>**

指示文件名 - 几乎总是相对于树结构所描述的根。

GIT\_INDEX\_FILE

**符号标识符**

任何接受任何<对象>的 Git 命令也可以使用以下符号表示法：

**头**

表示当前分支的头部。

**<标签>**

有效的标签**名称**（即引用）。

refs/tags/<tag>

**<头>**

有效的头部**名称**（即引用）。

refs/heads/<head>

有关拼写对象名称方法的更完整列表，请参阅 [gitrevisions[7]](https://git-scm.com/docs/gitrevisions) 中的“指定修订”部分。

**文件/目录结构**

请参阅 [gitrepository-layout[5]](https://git-scm.com/docs/gitrepository-layout) 文档。

阅读 [githooks[5]](https://git-scm.com/docs/githooks) 了解有关每个钩子的更多详细信息。

更高级别的 SCM 可能会在 中提供和管理其他信息。$GIT\_DIR

**术语**

請參閱 [gitglossary[7]](https://git-scm.com/docs/gitglossary)。