

使用版本控制系统

Qt创建者支持的版本控制系统有：

| 版本控制系统 | 地址 | 笔记 |
|--------|---|--|
| 集市 | http://bazaar.canonical.com/ | |
| 清除大小写 | http://www-01.ibm.com/software/awdtools/clearcase/ | 默认情况下禁用。要启用该插件，请选择“帮助>关于插件>版本控制> ClearCase”。然后选择立即重启以重启Qt创建者并加载插件。 |
| 断续器 | http://www.nongnu.org/cvs/ | |
| 化石 | https://fossil-scm.org/index.html/doc/trunk/www/index.wiki | 要使用化石，您需要从源代码构建化石插件，并按照Qt Creator化石插件手册中的描述安装化石。 |
| 吉特 | http://git-scm.com/ | Git 版本 1.9.0 或更高版本 格里特版本 2.6 或更高版本 |
| 吉特实验室 | http://gitlab.com/ | 实验的 |
| 善变 | http://mercurial.selenic.com/ | |
| 佩力 | http://www.perforce.com | 服务器版本 2006.1 及更高版本 默认情况下禁用。要启用插件，请选择帮助>关于插件>版本控制> Perforce。然后选择立即重启以重启Qt创建者并加载插件。 |
| 颠覆 | http://subversion.apache.org/ | 子版本 1.7.0 及更高版本 |

设置版本控制系统

Qt Creator 使用版本控制系统的命令行客户端来访问您的仓库。若要允许访问，请确保可以使用环境变量找到命令行客户端。或者，在“编辑>首选项”>“版本控制”中“版本控制系统特定”选项卡的“命令”字段中指定命令行客户端可执行文件的路径。PATH

如果需要身份验证才能访问存储库，请在“用户名”和“密码”字段中输入用户凭据。

在“超时”字段中输入版本控制操作的超时。

应该准备好从Qt Creator使用该系统。

有关将 Git 用于 Windows 的更多信息，请参阅[将 Git 用于 Windows](#)。

设置常规选项

选择“**编辑**>**首选项**”>“**版本控制**”>“**常规**”以指定提交邮件的设置：

- 将**提交邮件的换行**时间限制在以下位置将提交邮件的行长度限制为指定的字符数。
- 提交消息检查脚本**是可用于在提交之前对提交消息执行检查的脚本或程序。提交消息作为脚本的第一个参数传入。如果出现错误，脚本应输出有关标准错误的消息，并返回非零退出代码。
- 用户/别名配置文件**是以邮件映射格式列出作者姓名的文本文件。对于每个作者，您必须指定真实姓名和电子邮件地址，还可以选择指定别名和第二个电子邮件地址。例如：

```
Jon Doe <Jon.Doe@company.com> jdoe <jdoe@somemail.com>
Hans Mustermann <Hans.Mustermann@company.com> hm <info@company.com>
```

在此字段中指定文件后，可以在“**昵称**”对话框中选择作者作为提交消息字段的值。

- 用户字段配置文件**是一个简单的文本文件，由指定将作者作为值的提交消息字段的行组成，例如：

```
Acked-by:
Initial-patch-by:
Reported-by:
Rubber-stamped-by:
Signed-off-by:
Tested-by:
```

在此字段中指定文件后，您可以在提交更改时将作者添加为提交邮件字段的值。如果还指定了**用户/别名配置文件**，则可以在“**昵称**”对话框中选择作者。

- SSH 提示命令**指定一个 ssh-askpass 命令，在使用 SSH 时，您可以使用该命令（在 Linux 上）提示用户输入密码。例如，或，具体取决于您使用的 ssh-askpass 实现。ssh-askpassx11-ssh-askpass
- Reset VCS Cache** resets the version control system configuration to a state known to Qt Creator after it has been changed from the command line, for example.

Creating VCS Repositories for New Projects

Qt Creator allows you to create repositories for version control systems that support local repository creation, such as Git, Mercurial, or Bazaar. When creating a new project by selecting **File > New File or Project**, you can choose a version control system on the final wizard page.

You can also select **Tools** and then select **Create Repository** in the submenu for the version control system.

To import a project that is under version control, choose **File > New Project > Project from Version Control** and select the version control system that you use. Follow the instructions of the wizard to import the project.

The **Tools** menu contains a submenu for each supported version control system. This section describes using the functions that are available for all the supported version control systems. For more information about the additional functions and options available for a particular version control system, see:

- › Using Bazaar
- › Using ClearCase
- › Using CVS
- › Qt Creator Fossil Plugin Manual
- › Using Git
- › Using GitLab
- › Using Mercurial
- › Using Perforce
- › Using Subversion

Version Control displays the commands that are executed, a timestamp, and the relevant output. Select **View > Output > Version Control** to open the view.

```

Version Control
warning: LF will be replaced by CRLF in examples/quick/demos/docks/docks.qml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in examples/quick/demos/docks/docks.qml.
The file will have its original line endings in your working directory.
Committed 1 files.
warning: LF will be replaced by CRLF in examples/quick/demos/docks/docks.qml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in examples/quick/demos/docks/docks.qml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in examples/quick/demos/docks/docks.qml.
The file will have its original line endings in your working directory.

```

Adding Files

When you create a new file or a new project, the wizard displays a page asking whether the files should be added to a version control system. This happens when the parent directory or the project is already under version control and the system supports the concept of adding files, for example, Perforce and Subversion. Alternatively, you can add files later by using the version control tool menus.

Viewing Diff Output

All version control systems provide menu options to *diff* the current file or project: to compare it with the latest version stored in the repository and to display the differences. In Qt Creator, a diff is displayed in a read-only editor. If the file is accessible, you can double-click on a selected diff chunk and Qt Creator opens an editor displaying the file, scrolled to the line in question.

```

Git Diff "examples/quick/demos/..." docks.qml
[5224e29] examples/quick/demos/clocks/clocks.qml
[Skipped 66 lines...] Rectangle {
    .....delegate: Content.Clock { city: cityName; shift: timeShift
    .....model: ListModel {
    .....ListElement { cityName: "New York"; timeShift: -4 }
    .....ListElement { cityName: "London"; timeShift: 0 }
    .....ListElement { cityName: "Oslo"; timeShift: 1 }
    .....ListElement { cityName: "Mumbai"; timeShift: 5.5 }
    .....ListElement { cityName: "Tokyo"; timeShift: 9 }
    .....ListElement { cityName: "Brisbane"; timeShift: 10 }
    .....ListElement { cityName: "Los Angeles"; timeShift: -8 }
[Skipped unknown number of lines...]

[98f9373] examples/quick/demos/clocks/clocks.qml
[Skipped 66 lines...] Rectangle {
    .....delegate: Content.Clock { city: cityName; shift: timeShift
    .....model: ListModel {
    .....ListElement { cityName: "New York"; timeShift: -4 }
    .....ListElement { cityName: "London"; timeShift: 0 }
    .....ListElement { cityName: "Berlin"; timeShift: 1 }
    .....ListElement { cityName: "Mumbai"; timeShift: 5.5 }
    .....ListElement { cityName: "Tokyo"; timeShift: 9 }
    .....ListElement { cityName: "Brisbane"; timeShift: 10 }
    .....ListElement { cityName: "Los Angeles"; timeShift: -8 }
[Skipped unknown number of lines...]

```

code pasting service.

Viewing Versioning History and Change Details

Display the versioning history of a file by selecting **Log** or **Filelog**. Typically, the log output contains the date, the commit message, and a change or revision identifier.

Annotating Files

Annotation views are obtained by selecting **Annotate** or **Blame**. Selecting **Annotate** or **Blame** displays the lines of the file prepended by the change identifier they originate from. Clicking on the change identifier shows a detailed description of the change.

To show the annotation of a previous version, right-click on the version identifier at the beginning of a line and choose one of the revisions shown at the bottom of the context menu. This allows you to navigate through the history of the file and obtain previous versions of it. It also works for Git and Mercurial using SHA-1.

The same context menu is available when right-clicking on a version identifier in the file log view of a single file.

Committing Changes

Once you have finished making changes, submit them to the version control system by choosing **Commit** or **Submit**. Qt Creator displays a commit page containing a text editor where you can enter your commit message and a checkable list of modified files to be included.

Reverting Changes

All supported version control systems support reverting your project to known states. This functionality is generally called *reverting*.

The changes discarded depend on the version control system.

A version control system can replace the **Revert** menu option with other options.

Viewing Status

You can select **Status** to view the status of the project or repository.

Updating the Working Tree

You can select **Update** to update your working tree with the latest changes from the branch. Some version control systems allow you to choose between updating the current project and updating all projects.

Deleting Files

You can select **Delete** to delete obsolete files from the repository.

[◀ Adding New Custom Wizards](#)

[Using Bazaar >](#)



Contact Us

Company

- About Us
- Investors
- Newsroom
- Careers
- Office Locations

Support

- Support Services
- Professional Services
- Partners
- Training

Community

- Contribute to Qt
- Forum
- Wiki
- Downloads
- Marketplace

Licensing

- Terms & Conditions
- Open Source
- FAQ

For Customers

- Support Center
- Downloads
- Qt Login
- Contact Us
- Customer Success