



手册

# Multiuser

TwinCAT 3

版本: 1.3  
日期: 2020-06-17

**BECKHOFF**



# 目录

1	前言 .....	5
1.1	文档说明 .....	5
1.2	安全说明 .....	5
2	概念 .....	7
3	工作流程 .....	9
3.1	创建新项目 .....	9
3.2	使用现有项目 .....	9
3.3	使用设置功能 .....	10
4	参照, 用户界面 .....	14
4.1	设置选项卡 .....	14
4.2	History (历史) 选项卡 .....	15
4.3	Diff (差异) 选项卡 .....	16
4.4	Status (状态) 选项卡 .....	17
4.5	Conflict (冲突) 选项卡 .....	18
4.6	Manual (手动) 选项卡 .....	20
5	参照, 项目设置 .....	21
6	参照, 服务器设置 .....	22
7	常见问题 .....	23
8	第三方授权 .....	24



# 1 前言

## 1.1 文档说明

本说明仅供熟悉适用国家标准的控制和自动化工程专家使用。  
在安装和调试元器件时，必须遵循本文档及以下注意事项和说明。  
技术人员应负责在每次安装和调试时使用已发布的文档。

负责人员必须确保所述产品的应用或使用符合所有安全要求，包括所有相关法律、法规、准则和标准。

### 免责声明

本文档经过精心准备。然而，所述产品正在不断开发中。  
我们保留随时修改和更改本文档的权利，恕不另行通知。  
不得依据本文档中的数据、图表和说明对已供货产品的修改提出赔偿。

### 商标

Beckhoff®、TwinCAT®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、Safety over EtherCAT®、TwinSAFE®、XFC®、XTS® 和 XPlanar® 均为倍福自动化有限公司的注册商标并由公司授权使用。  
本出版物中使用的其他名称可能是商标，第三方出于自身目的使用它们可能侵犯商标所有者的权利。

### 正在申请的专利

涵盖 EtherCAT 技术，包括但不限于以下专利申请和专利：：  
EP1590927、EP1789857、EP1456722、EP2137893、DE102015105702  
包括在其他各国家的相应专利申请或注册。

## EtherCAT®

EtherCAT® 是注册商标和专利技术，由德国倍福自动化有限公司授权使用

### 版权所有

© 德国倍福自动化有限公司  
未经明确授权，禁止复制、分发、使用本文档及擅自将内容与他人交流。  
违者将承担赔偿责任。在专利授权、工具型号或设计方面保留所有权利。

## 1.2 安全说明

### 安全规范

请注意以下安全说明和阐述！  
可在以下页面或安装、接线、调试等区域找到产品相关的安全说明。

### 责任免除

所有元器件在供货时都配有适合应用的特定硬件和软件配置。禁止未按文档所述修改硬件或软件配置，德国倍福自动化有限公司不对此承担责任。

### 人员资格

本说明仅供熟悉适用国家标准的控制、自动化和驱动工程专家使用。

### 符号说明

在本文档中，下列符号随安全指示或说明一起使用。必须仔细阅读并严格遵守安全说明！

**⚠ 危险****严重受伤的风险！**

未遵守带有此符号的安全说明将直接危及人员生命和健康。

**⚠ 警告****受伤的风险！**

未遵守带有此符号的安全说明将危及人员生命和健康。

**⚠ 谨慎****人身伤害！**

未遵守带有此符号的安全说明可能导致人员受伤。

**注意****危害环境或损坏设备**

未遵守带有此符号的安全说明可能危害环境或损坏设备。

**● 提示或指示**

**i** 此符号表示该信息有助于更好地理解。

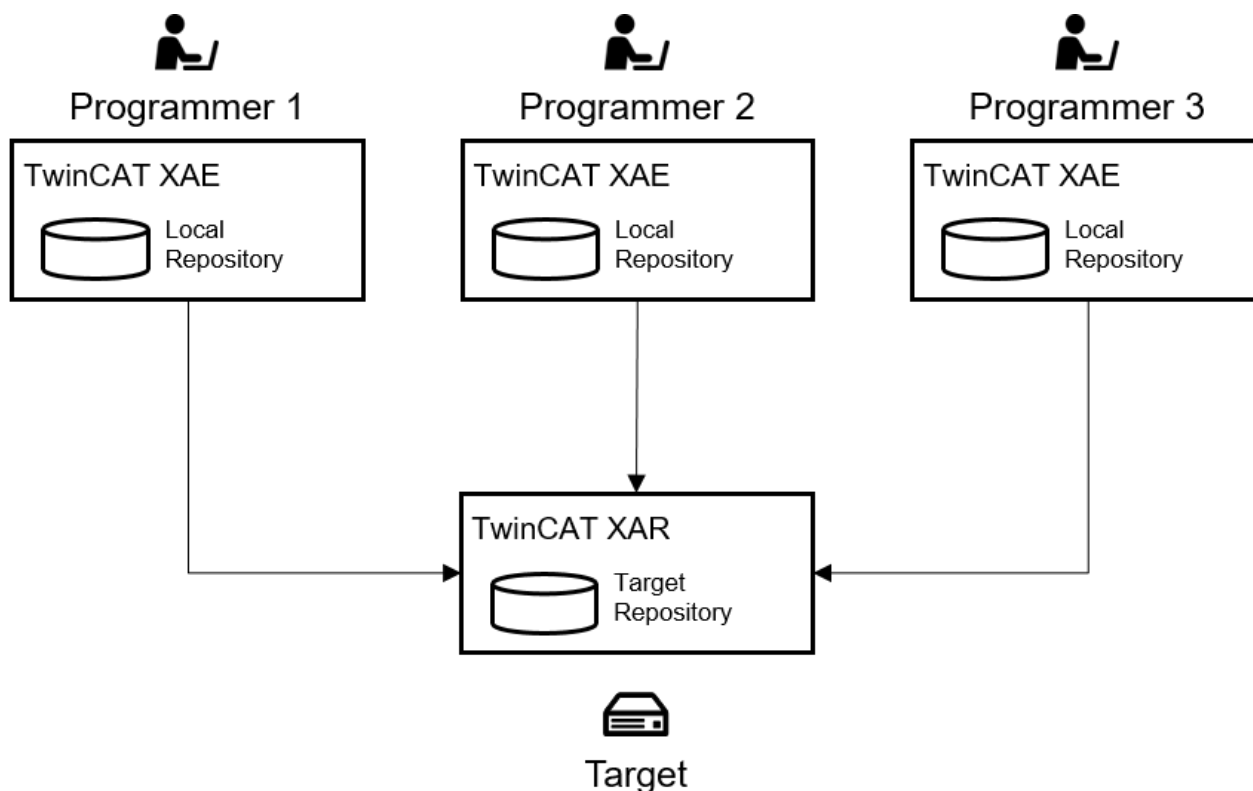
## 2 概念

越来越多的机器和系统功能被集成到软件中。为此，如今将会有包含多位程序员的各种团队参与到控制代码的创建以及这些机器和系统的后续调试中。这带来了许多挑战，特别是调试阶段：

- 在很多情况下，无法保证对源代码控制系统的访问
- 机器上的最新（激活）软件版本可能与本地版本不同
- 如果“下载”到目标系统的更改不正确，无法撤销“下载”。

TwinCAT Multiuser 应运而生。它是位于目标系统上的“本地”源代码控制系统，它的处理已被完全集成到现有工作流程中。这意味着，不需要对源代码控制系统的确切功能有深入了解，便能使用 TwinCAT Multiuser。

TwinCAT Multiuser 从 TC3.1.4024 版本开始提供。在该版本中，TwinCAT Multiuser 主要是方便了各自处理不同 PLC 项目的多名程序员之间的协作。如果多个 PLC 项目被集成在一个 TwinCAT 项目中，可分别为每个项目启用多用户功能。在这种情况下，将分别在本地和目标系统上为每个项目自动创建单独的存储库。下图对这一概念作了阐释。



附图 1:

在任何情况下，要将控制程序载入到目标系统中，需要一个到目标系统的连接，因此，在调试期间，目标系统被选为源代码控制服务器。这可免去附加的基础架构。

多用户集成的另一个目标是免去专门的源代码控制知识。源代码控制功能被集成在标准工作流程中。例如，在下载或在线更改项目时，当前项目状态将被自动传送给目标系统，无需分别触发发送。还会自动创建历史。所有更改（包括用户名、时间戳和更改）都将被记录在案。如有需要，可为每次更改传送设置注释查询。

为确保以这种方式创建的历史在调试后仍然可用，我们集成了一个源代码控制系统，从而允许多个存储库，还能将完整历史传送到所有存储库中。在成功调试之后，便可将机器/系统的活动项目状态传送到“开发存储库”中，包括在调试期间执行的所有步骤的追溯信息。后者使用源代码控制系统的板载资源完成。

多用户功能所基于的源代码控制系统为 Git。

### TwinCAT 源代码控制集成的不同之处:

TwinCAT 与源代码控制系统的常规接口连接不受多用户功能的影响，可独立运行。使用 TwinCAT 中集成的 Microsoft Visual Studio Shell 标准接口，大量源代码控制系统都可实现这一目的。除此之外，TwinCAT 项目的其他方面可由该集成功能处理。

TwinCAT Multiuser 功能基于 Git 标准。我们前面已经讲过，该功能的目的是让有多位程序员的团队在调试阶段实现控制程序协作，在该过程中，程序员无需特殊的基础架构或专门的源代码控制系统知识。

虽然可为两种功能使用不同的源代码控制系统，但只有两种功能都基于 Git，才能实现完整的功能。这指的是在调试期间将历史传送给“主”源代码控制系统。



## 3 工作流程

### 3.1 创建新项目

如果一个项目尚未通过多用户功能进行管理，要激活此项目的该功能，请执行下列步骤。

1. 对于需要通过多用户功能进行管理的 PLC 项目，在其项目设置中激活 **Use Multiuser (使用多用户)** 选项 (参见参照, 项目设置 [▶ 21])。
2. 通过 **View (视图)** -> **More Windows (更多窗口)** -> **Multiuser Explorer (多用户资源管理器)**, 打开 **Multiuser Explorer (多用户资源管理器)** 窗口。
3. 在 **Multiuser Explorer (多用户资源管理器)** 复选框中选择一个 PLC 项目。  
⇒ **Multiuser Explorer (多用户资源管理器)** 中将出现多个选项卡。
4. 选择 **Settings (设置)** 选项卡。
5. 在 **ADS Route (ADS 路由)** 字段中, 选择多用户存储库的目标系统。
6. 在 **Multiuser Repository (多用户存储库)** 字段中, 选择多用户存储库的名称。
7. 如有需要, 选中选项 **Ask for update message on each usage (在每次使用时询问更新消息)** (另请参见设置选项卡 [▶ 14])。
8. 单击 **Init local and remote (初始化本地和远程)** 按钮。  
⇒ 项目和目标系统中的多用户存储库将会进行初始化。

### 3.2 使用现有项目

如果要将现有的项目 (已在目标系统上设置了它的存储库) 连接到多用户服务器, 可按照设置功能相同的方式进行处理。唯一不同的是, 在目标系统名称复选框中, 不是创建新的存储库, 而是选择一个现有的存储库。

详细步骤如下:

1. 打开项目。
2. 在需要通过多用户功能进行管理的 PLC 项目的项目设置中, 检查是否已启用 **Use Multiuser (使用多用户)** 选项 (参见参照, 项目设置 [▶ 21])。如果未启用, 启用该项。
3. 通过 **View (视图)** -> **More Windows (更多窗口)** -> **Multiuser Explorer (多用户资源管理器)**, 打开 **Multiuser Explorer (多用户资源管理器)** 窗口。
4. 在 **Multiuser Explorer (多用户资源管理器)** 复选框中选择一个 PLC 项目。  
⇒ **Multiuser Explorer (多用户资源管理器)** 中将出现多个选项卡。
5. 选择 **Settings (设置)** 选项卡。
6. 在 **ADS Route (ADS 路由)** 字段中, 选择包含多用户存储库的目标系统。
7. 在 **Multiuser Repository (多用户存储库)** 复选框中, 选择项目要链接的多用户存储库的名称。
8. 如有需要, 选中选项 **Ask for update message on each usage (在每次使用时询问更新消息)** (另请参见设置选项卡 [▶ 14])。
9. 单击 **Init local and remote (初始化本地和远程)** 按钮。  
⇒ 项目中的多用户存储库现在开始初始化。



如果没有现有项目, 可创建一个 TwinCAT 项目。然后添加空的 PLC 项目, 再按照前文所述将其链接到目标系统上的多用户存储库。



Git 源代码控制系统使用存储库中的绝对路径。也就是说, 复制一个已安装多用户功能的项目 (包括 .TcGit 文件夹), 其多用户功能无法正常工作!

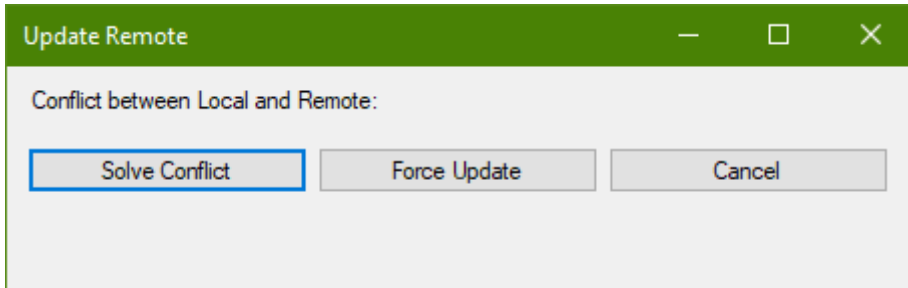
### 3.3 使用设置功能

如果通过创建新项目 [▶ 9]或使用现有项目 [▶ 9]中所述设置多用户功能，并将其链接到 PLC 项目，则该功能将被自动集成到现有工作流程中。

#### 将数据传送到目标存储库

在下载或在线更改期间，操作 PLC 项目时产生的更改将被自动传送到目标存储库（如果激活了多用户功能）。

将会对此进行检查，确认是否与目标存储库中的数据有冲突。如果操作该项目的其他程序员对相同项目部分（例如，POU、设置等）也作了更改并将它们传送到目标存储库，则会发生这种情况。如果发生冲突，将显示下列对话框。



在该选择对话框中，可选择：

- 使用 TwinCAT 项目比较工具解决冲突；
- 强制更新目标系统，包括目标存储库，或
- 取消登录。

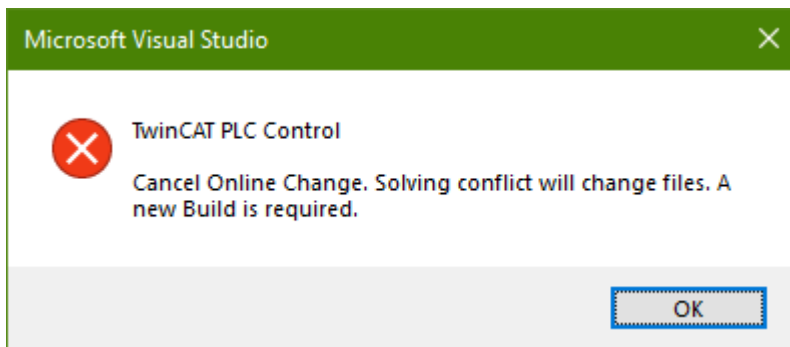
如果需要目标系统使用当前项目运行，不考虑该项目涉及的其他程序员所作的更改，可使用 **Force Update (强制更新)**。

项目协作的常规工作流程包括了冲突解决。在该步骤中，可合并您及其他程序员对项目所作的更改。

1. 为此，单击 **Solve Conflict (解决冲突)**。

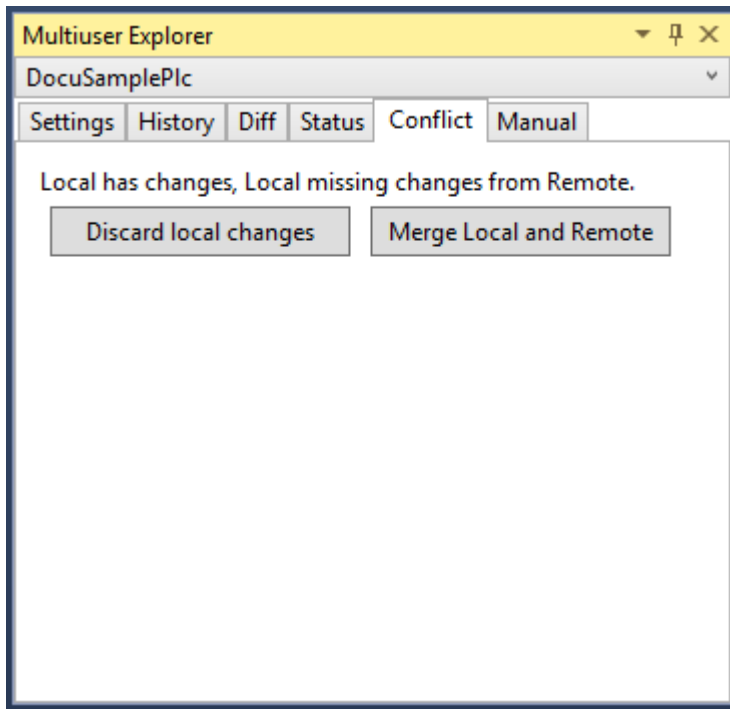
⇒ 在随后出现的对话框中，将通知解决冲突可能导致文件更改，如果发生更改，必须在合并操作后重新编译项目。

2. 使用 **OK (确定)** 确认该消息。

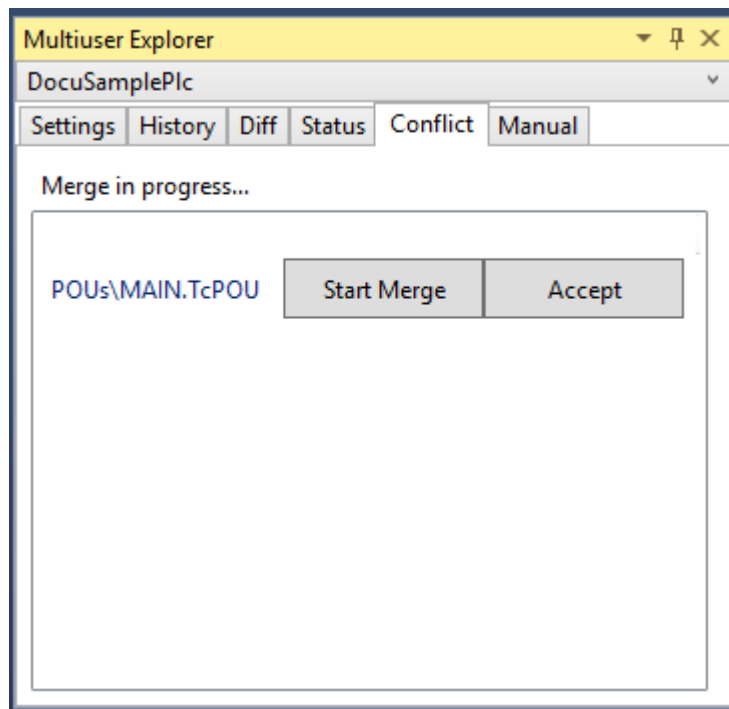


⇒ 将打开 **Conflict (冲突)** 选项卡。可在此选择放弃所有本地更改，或应用目标存储库中的版本，或合并两个版本。

3. 如要合并，单击 **Merge Local and Remote**（合并本地和远程）。

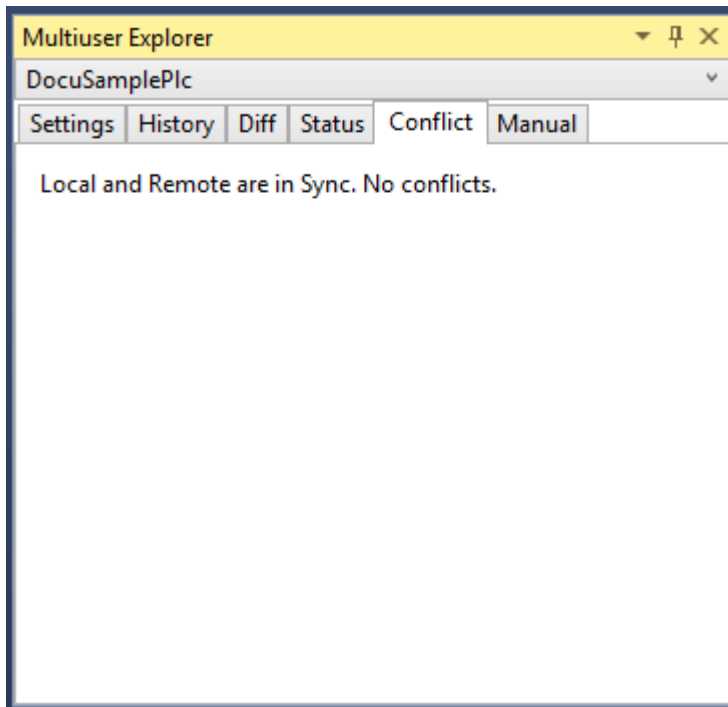


⇒ 将会显示一个总览画面，显示导致与目标存储库发生冲突的所有 POU（编程组织单元）。



4. 对于显示的每个冲突，单击 **Start Merge**（开始合并）。  
⇒ 将打开 TwinCAT 项目比较工具，可在此合并更改。
5. 使用 **Accept**（接受）接受结果。

⇒ 当合并好所有更改后，将给出通知，告知已没有冲突。

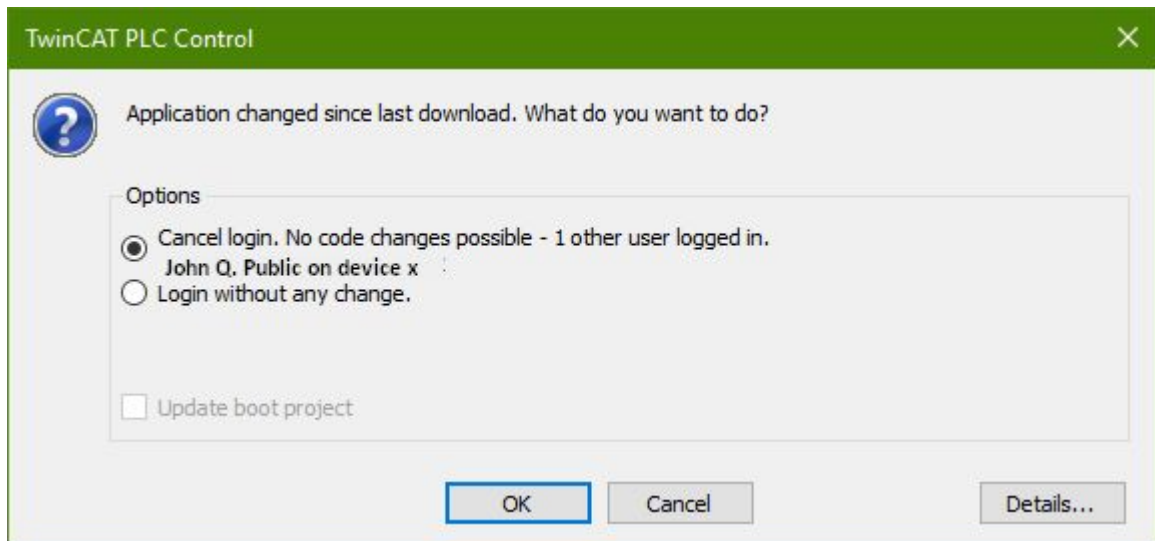


6. 现在重新登录。

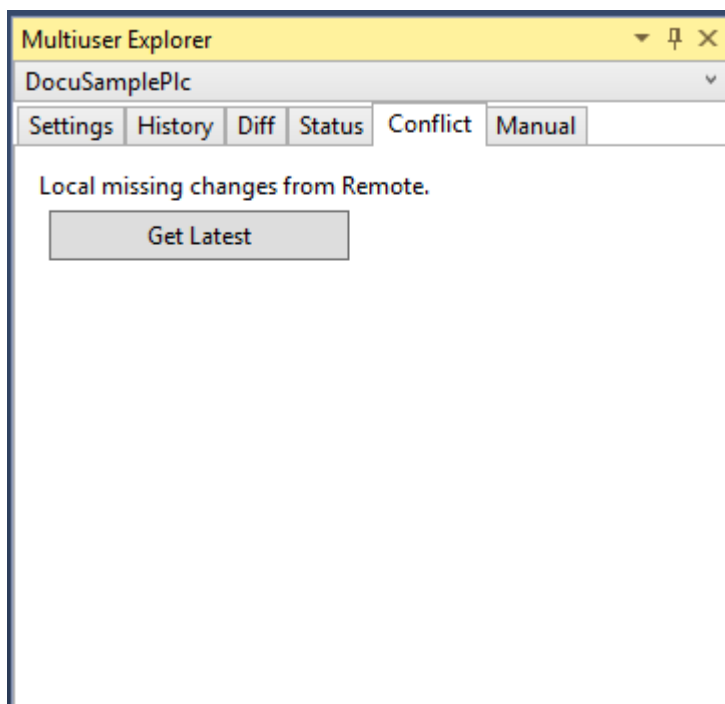
⇒ 项目当前和合并后的版本将被传送到目标系统并保存在目标存储库中。

## 每次仅一人

虽然多位程序员可同时登录目标，查看项目状态，但在任意时间只有一位程序员可进行更改。所有其他程序员都必须短暂退出。已登录的程序员显示在下列对话框中。

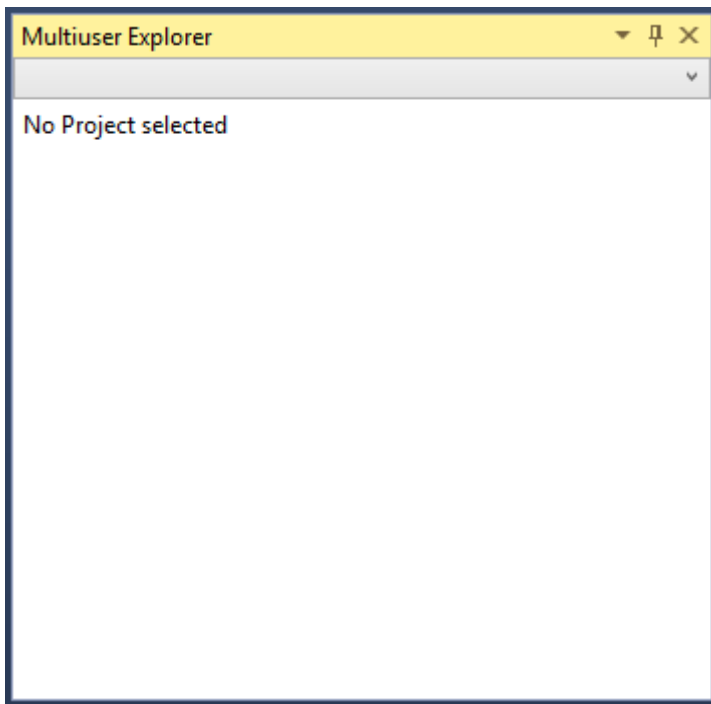


当作出更改后，其他程序员可重新登录，并触发上述工作流程，合并自己及其他程序员的更改。如果其他程序员尚未对项目进行进一步的更改，将会通知他们，告知远程系统上有新的版本。



## 4 参照，用户界面

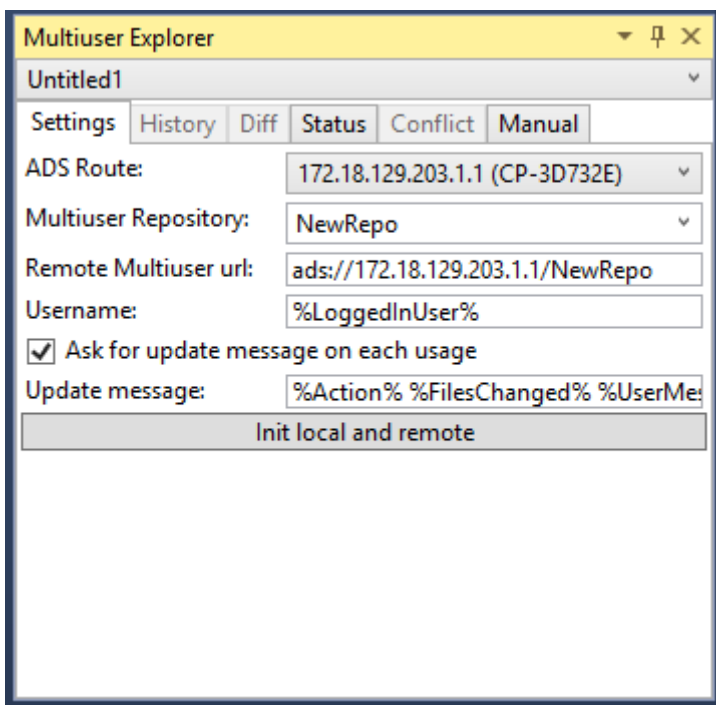
Multuser Explorer（多用户资源管理器）是多用户功能的集中管理工具。当没有项目与多用户功能相关联时，该窗口将保持空白。



当选择一个项目后，下图所示的选项卡将变得可见。

### 4.1 设置选项卡

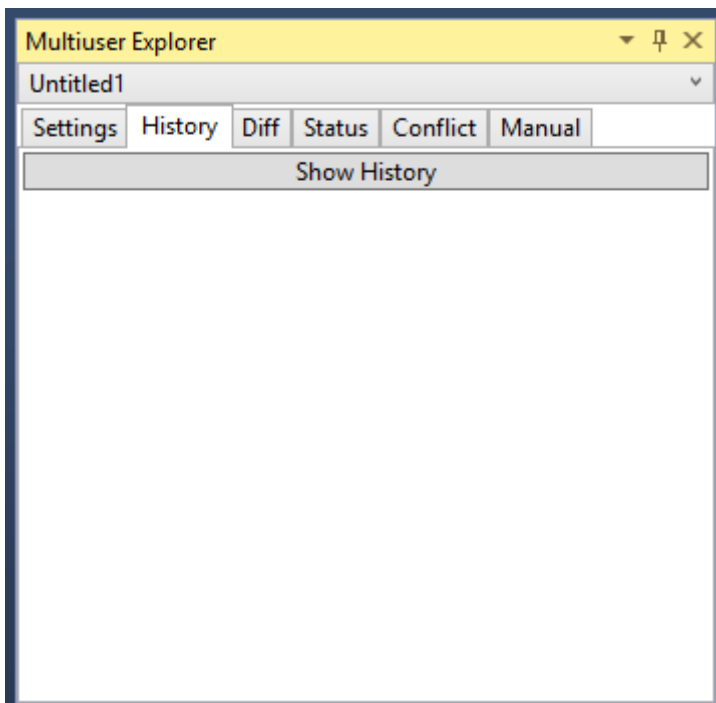
当选择一个 PLC 项目后，将显示如下图所示的 **Settings（设置）** 选项卡。



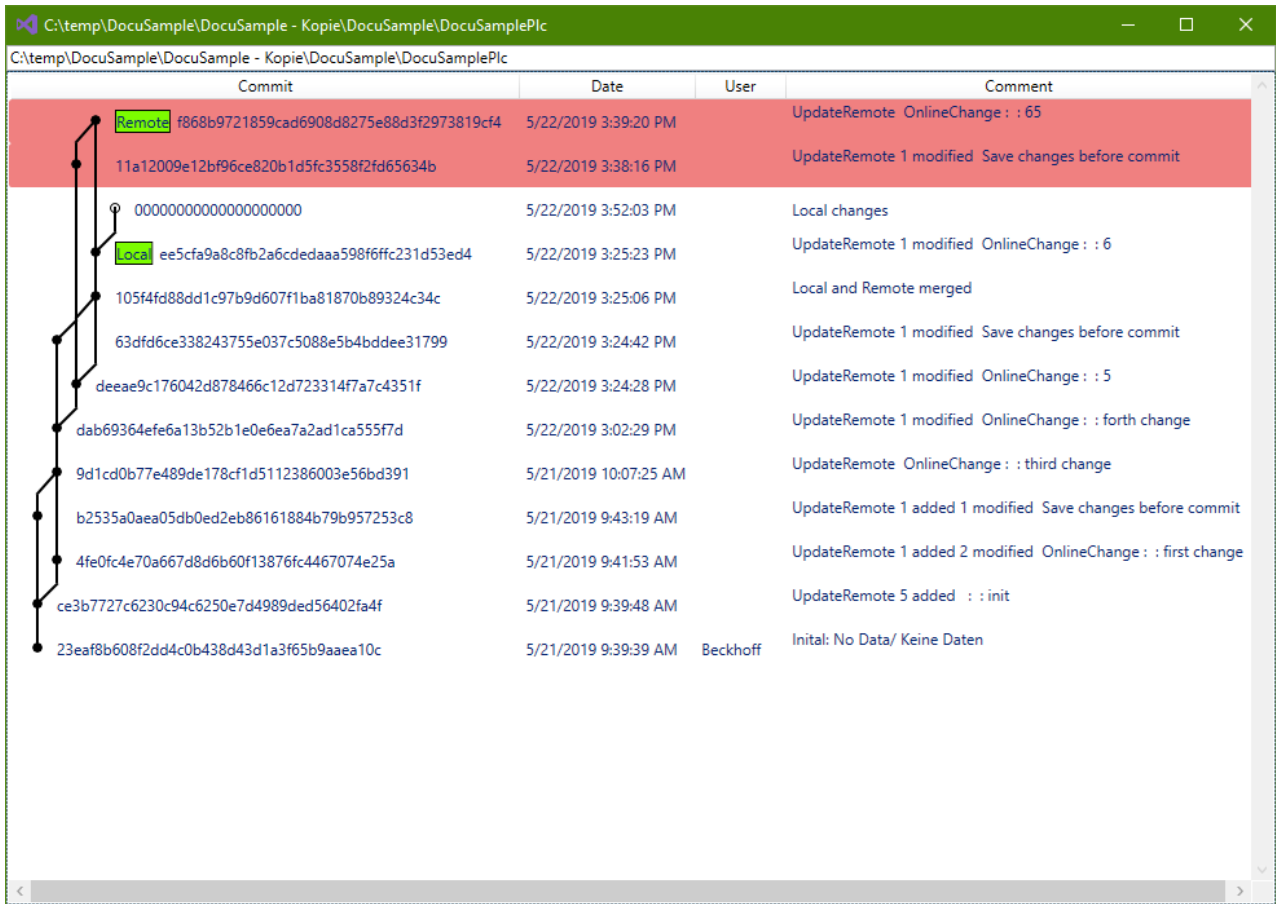
ADS route (ADS 路由)	用于选择现有 ADS 路由的组合框
Multiuser repository (多用户存储库)	用于选择现有 MU 存储库或创建新的 MU 存储库的复选框
Remote Multiuser URL (远程 Multiuser URL)	存储库的 URL
User name (用户名)	历史中的条目所使用的用户名。默认设置为占位符“%LoggedInUser%”。它将自动替换为本地登录的用户，作为注释中的用户名。
Ask for update message on each usage (在每次使用时询问更新消息)	用于询问更改消息的选项
Update message (更新消息)	更新消息的结构。在在线更改或下载期间，占位符“%Action%”、“%FilesChanged%”和“%UserMessage%”将被自动替换为相应的信息。如果删除了占位符，则相应的信息也将自动从生成的注释中删除。 占位符： “%Action%”：指示执行的是在线更改还是下载 “%FilesChanges%”：指示更改文件的次数和原因（例如，增加 2） “%UserMessage%”：如果启用了选项 <b>Ask for update message on each usage (在每次使用时询问更新消息)</b> ，则该占位符将替换为用户指定的注释。
Init local and remote (初始化本地和远程)	初始化本地和目标存储库

## 4.2 History (历史) 选项卡

History (历史) 选项卡中有一个 **Show History (显示历史)** 按钮。可用它打开一个工具窗口，显示当前项目的历史。



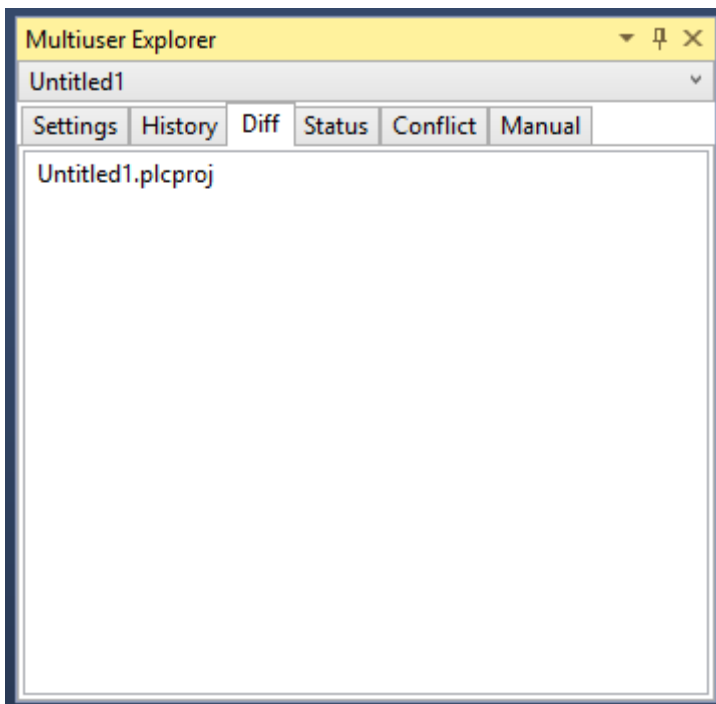
示例：文档历史示例。在该例中，目标系统和本地系统上均有更改。



### 4.3 Diff (差异) 选项卡

Diff (差异) 选项卡显示已作了更改但尚未将其传送到目标存储库中的所有文件。

下图是一个示例，显示项目 Untitled1 的 PLC 项目文件的更改。





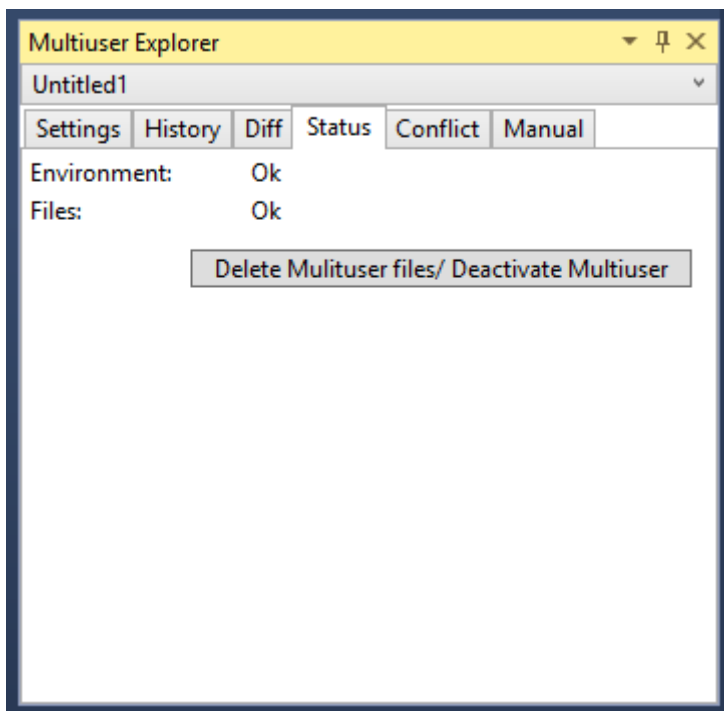
**i** 仅当代码更改时才会更新目标系统。

并不是显示的每个更改都是代码更改。仅当代码有更改，需要重新下载项目或项目的一部分时，才会将更改传送到目标系统。

示例：POU 中注释的更改是文件中的更改，不需要重新编译。因此，可登录而不进行在线更改，从而无需下载更改。

## 4.4 Status (状态) 选项卡

显示多用户功能的状态，提供禁用该设置的选项。这将删除本地存储库。



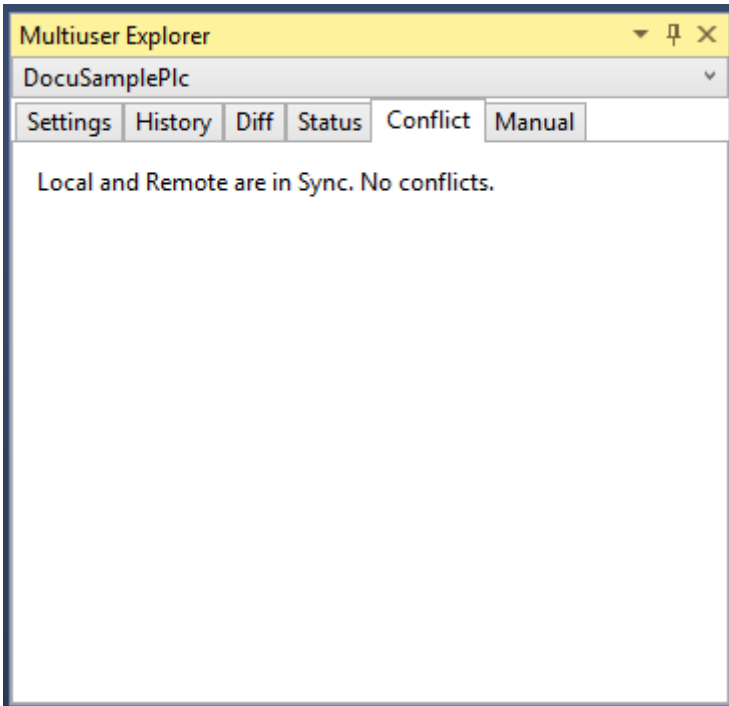
Environment (环境)	显示多用户服务器的状态： <b>OK (正常)</b> ：找到本地和目标存储库 <b>LocalNotExists (本地不存在)</b> ：未找到本地存储库 <b>RemoteNotExists (远程不存在)</b> ：未找到目标存储库
Files (文件)	显示文件的状态。 <b>OK (正常)</b> ：找到本地和目标存储库，但目标存储库上没有新的项目状态 <b>LocalNotExists (本地不存在)</b> ：未找到本地存储库 <b>RemoteNotExists (远程不存在)</b> ：未找到目标存储库 <b>LokalsbehindRemote</b> ：目标存储库上存在新版本
删除 Multiuser 文件/取消激活 Multiuser	删除本地存储库，禁用多用户功能

**i** 如果本地文件状态比存储库中的更新，Files (文件) 状态条目也将显示 OK (正常)。可在 Diff (差异) [▶ 16] 选项卡中查看哪些文件被更改了。

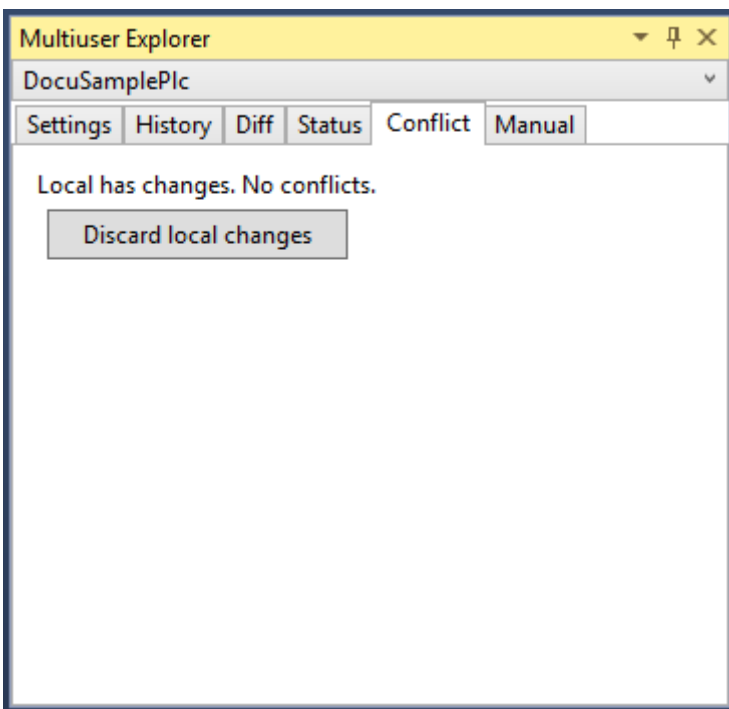
## 4.5 Conflict (冲突) 选项卡

**Conflict (冲突)** 选项卡显示本地状态和目标存储库中的状态是否有冲突，或者是否触发了合并。可能出现以下情况：

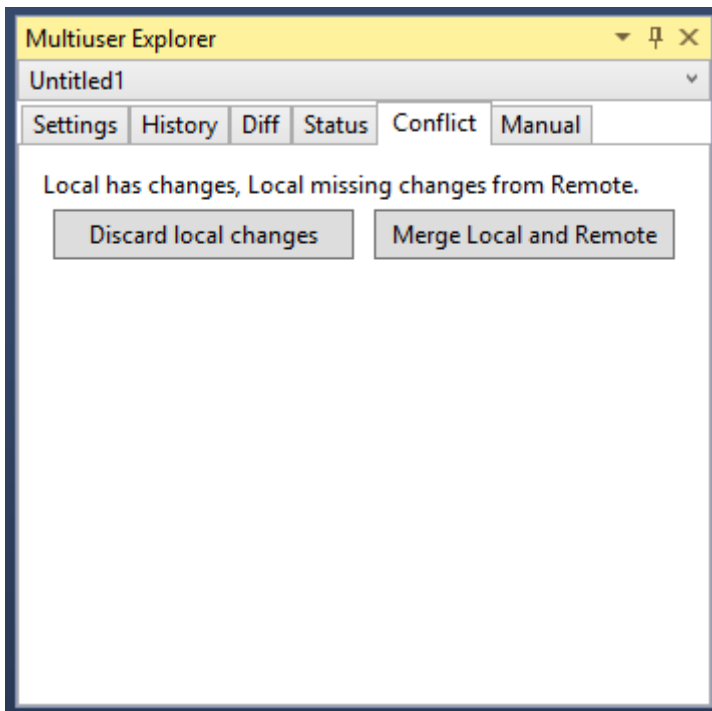
本地和目标存储库状态相同：



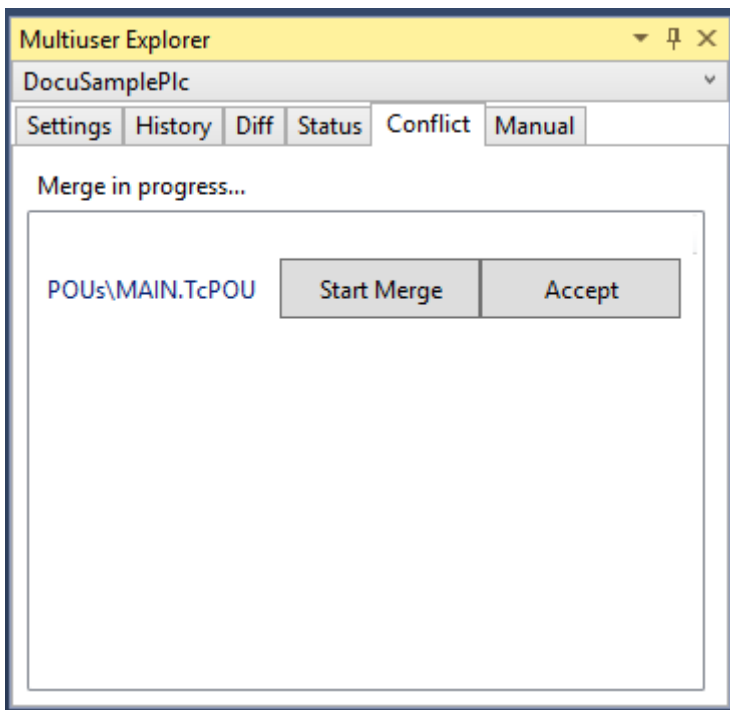
本地系统包含尚未被传送的更改：



本地和远程系统上均已作了更改：

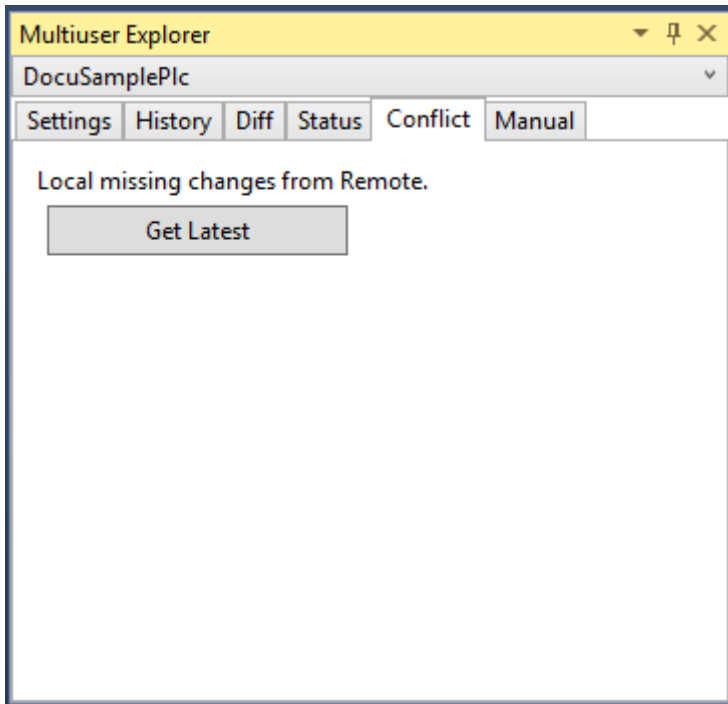


单击 **Merge Local and Remote (合并本地和远程)** 按钮显示更改列表。在下列视图中，可查看两端均已作了更改的所有对象（例如，下图中的 POU Main）。



要合并更改，单击 **Start Merge (开始合并)** 按钮。将打开 TwinCAT 项目比较工具，可在此合并对象。成功合并后，请在 TwinCAT 项目比较工具和 Multiuser Explorer (多用户资源管理器) 中使用 **Accept (接受)** 确认更改。

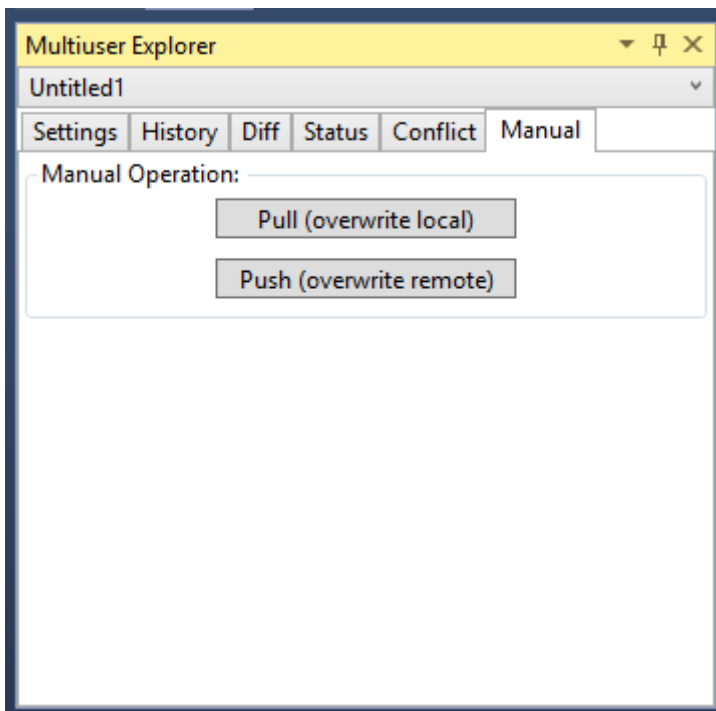
目标存储库包含本地尚未执行的更改：



单击 **Get Latest (获取最新)** 按钮，获取目标存储库的当前状态。TwinCAT 现在将通知您文件已在后台更改，询问是否要重新加载。使用 **OK (确定)** 确认该询问。

## 4.6 Manual (手动) 选项卡

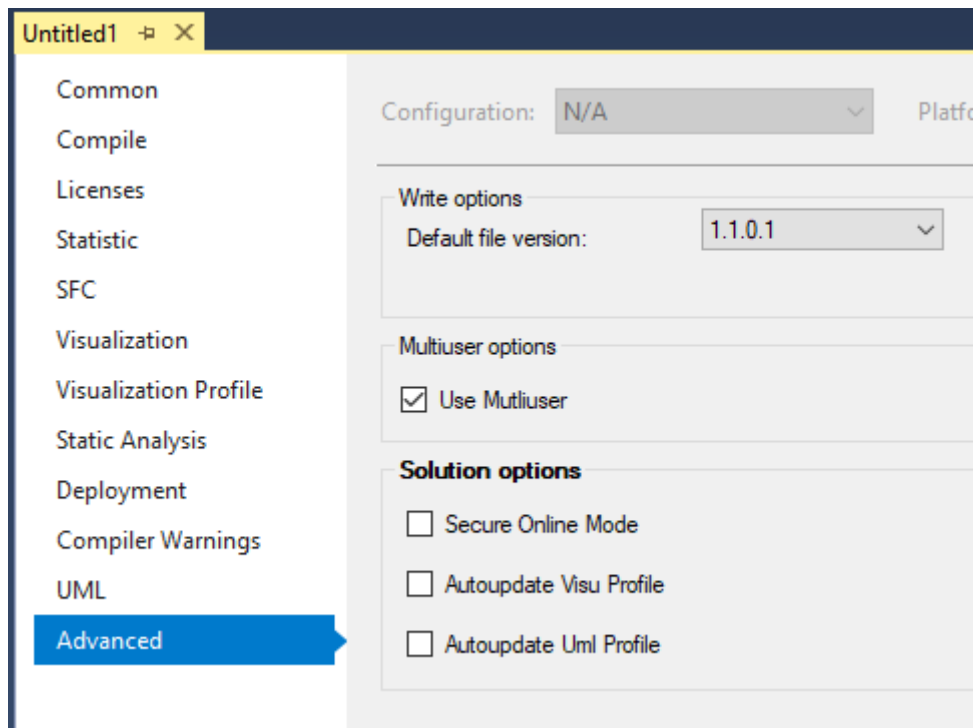
**Manual (手动)** 选项卡用于将当前项目状态写入到目标系统，或目标系统获取当前项目状态。



Pull (override local) (拉 (覆盖本地))	从目标系统获取当前项目状态，并覆盖本地项目。不合并项目状态。
Push (override remote) (推 (覆盖远程))	覆盖目标系统上的状态。不合并项目状态。

## 5 参照，项目设置

要在项目中使用 Multiuser，必须启用该功能。



Use Multiuser (使用 Multiuser)	启用 PLC 项目的多用户功能
------------------------------	-----------------

## 6 参照，服务器设置

TwinCAT Multiuser 功能基于 Git 源代码控制系统。它随“Minimal Git for Windows”版本自动安装。

### 更改 Git 存储库的存储位置

如有必要，可更改 Git 存储库在目标系统上的默认存储位置。默认位置为“C:\MultiuserRepository”。

要调整路径，更改文件“<TwinCAT Folder>/Functions/Multiuser/directorypath.config”中的条目，使其指向所需的路径。

## 7 常见问题

**Multiuser 功能的行为方式与描述的不同。我应该怎么处理？**

可在 **Status (状态)** 选项卡的 **MultiuserExplorer (多用户资源管理器)** 中查看 Multiuser 功能的**状态**。在使用 Multiuser 功能时，如果发现任何与预期不符的行为，首先应该查看这里（参见 [Status \(状态\) 选项卡 \[► 17\]](#)）。

还要检查目标系统上是否启用了 TwinCAT3 AdsGitServer Windows 服务，如有必要，重新启动该服务。

**如何重新启动 Multiuser 功能？**

在 MultiuserExplorer (多用户资源管理器) 的 **Status (状态)** 选项卡中，可使用 **DeleteMultiuser files/Deactive Multiuser (删除 Multiuser 文件/取消激活 Multiuser)** 按钮删除本地 Git 存储，以重新启动 Multiuser 功能（参见 [Status \(状态\) 选项卡 \[► 17\]](#)）。

**我收到消息：“RepositoryExistsException” (存储库存在异常)。如何处理这种状况？**

由于过去的无效操作，已为当前项目创建 Git 存储库。

在 MultiuserExplorer (多用户资源管理器) 的 **Status (状态)** 选项卡中，可使用 **DeleteMultiuser files/Deactive Multiuser (删除 Multiuser 文件/取消激活 Multiuser)** 按钮，删除该无效的存储库。

**我收到消息：“ADS error 0x745: Timeout elapsed” (ADS error 0x745: 已超时)。如何处理这种状况？**

在 MultiuserExplorer (多用户资源管理器) 中打开 **Status (状态)** 选项卡。消息“RemoteNotExists” (远程存储库不存在) 指示未创建远程存储库或它已被删除。

另外，也可在 Windows 资源管理器中，在目录 `C:\ProgramData\Beckhoff\MultiuserRepository` 下查看目标系统。

如果多次收到这条消息，在目标系统上重新启动 TwinCAT3 AdsGitServer Windows 服务。

**我收到消息：“ADS error 0x1: an internal error has occurred” (ADS error 0x1: 发生内部错误)。如何处理这种状况？**

请检查是否正确安装了 Multiuser 功能。除了别的文件之外，`C:\TwinCAT\Functions\Multiuser` 文件夹中必须存在文件 `LibGit2Sharp.dll`、`TcAdsGitPackage.dll` 和 `TcAdsGitServer.exe` 以及文件夹 `cmd` 和 `mingw32`。

## 8 第三方授权

TwinCAT Multiuser 功能使用 Git 作为源代码控制系统，它在 GNU General Public License 版本 2.0 中发布。

GNU 通用公共授权

1991 年 6 月，第 2 版

版权所有 (C) 1989, 1991 自由软件基金会

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

允许每个人复制和发布本授权文件的完整副本，

但不允许对它进行任何修改。

导言

大多数软件授权声明是设计用以剥夺您

共享与修改软件的自由。相反地，GNU 通用公共

授权力图保证您分享与修改自由软件的自由

——确保软件对所有的使用者都是自由的。通用

公共授权适用于大多数自由软件基金会的软件，

以及任何作者指定使用本授权的其他软件。

(有些自由软件基金会的软件，则适用

GNU 函式库通用公共授权规定。)您也可以让您的软件

适用本授权规定。

当我们在谈论自由软件时，我们所指的是自由，而不是

价格。我们的通用公共授权是设计用以确保

使您拥有发布自由软件备份的自由（以及

您可以决定此一服务是否收费），确保您能收到源码

或者在您需要时能得到它，确保您能变更软件或将它的一部分

用于新的自由软件；并且确保您知道您可以做上述的这些事情。

为了保障您的权利，我们需要作出限制：

禁止任何人否认您上述的权利，或者要求您放弃这些权利。

如果您发布软件的副本，或者对之加以修改，

这些限制就转化成为您的责任。

例如，假如您发布此类程序的副本，

无论是免费或收取费用，您必须将您所享有的一切权利给予

收受者。您也必须确保他们也能收到或得到

原始程序码。而且您必须向他们展示这些条款的内容，使他们知到他们所享有的

权利。

我们采取两项措施来保护您的权利：(1) 以著作权保护软件，以及

(2) 提供您本授权，赋与您复制、发布并且 / 或者



修改软件的法律许可。

同时，为了保护作者与我们，我们想要确定

每个人都明白，自由软件是没有担保

责任的。如果软件被他人修改并加以传播，我们

需要其收受者知道，他们所得到的并非原始版本，因此

由他人所引出的任何问题对原作者的声誉将不会有

任何的影响。

最后，所有自由软件不断地受到软件专利的威胁。

我们希望能避免自由软件的再发布者

以个人名义取得专利授权而使

程序专有化的风险。为了防止上述的情事发生，我们在此明确声明：

任何专利都必须为了每个人的自由使用而核准，否则就不应授与专利。

以下是有关复制、发布及修改的明确条款

及条件。

GNU 通用公共授权

复制、发布与修改的条款与条件

0. 凡著作权人在其程序或其他著作中声明，

该程序或著作会在通用公共授权条款下发布，

本授权对其均有适用。以下所称的“程序”，

是指任何一种适用通用公共授权的程序或著作；并且一个“基于本程序的著作”，

则指本程序或任何基于著作权法所产生的衍生著作，

换言之，是指包含本程序全部或一部的著作，

不论是完整的或经过修改的程序，以及（或）翻译成

其他语言的程序。（以下“修改”一词包括但不限于

翻译行为在内）。被授权人则称为“您”。

本授权不适用于复制、发布与修改以外的

行为；这些行为不在本授权范围内。执行

本程序的行为并不受限制，而本程序的输出

只有在其内容构成基于本程序所生的著作

（而非只是因为执行本程序所造成）时，始受本授权拘束。

至于程序的输出内容是否构成本程序的衍生著作，则取决于本程序的具体用途。

1. 您可以对所收受的本程序源代码，

无论以何种媒介，复制与发布其完整的复制物，

然而您必须符合以下要件：以显著及适当的方式

在每一份复制物上发布适当的著作权标示及无担保声明；维持所有

有关本授权以及无担保声明的原貌；

并将本授权的副本连同本程序一起交付予其他任何一位

本程序的收受者。

您可以对让与复制物的实际行为收取一定的费用，

您也可以自由决定是否提供担保以作为对价的交换。

2. 您可以修改本程序的一个或数个复制物或者本程序的任何部份，

以此形成基于本程序所生的著作，

并依前述第一条规定，

复制与发布此一修改过的程序或著作，但您必须符合以下要件：

a) 您必须在所修改的档案上附加显著的标示，

阐明您修改过这些档案，以及修改日期。

b) 您必须就您所发布或发行的著作，

无论是包含本程序全部或一部的著作，

或者是自本程序或其任何部份所衍生的著作，

整体授权所有第三人依本授权规定使用，且不得因此项授权行为而收取任何费用。

c) 若经过修改的程序在执行时通常以交互方式读取命令时，

您必须在最常被使用的方式下，

于开始进入这种交互式使用时，

列印或展示以下宣告：

适当的著作权标示及无担保声明（或者声明您提供担保）、

使用者可以依这些条件再发布此程序，

以及告知使用者如何浏览本授权的

副本。（例外：若本程序本身是以交互的方式执行，然而

通常却不会列印该宣告时，则您基于本程序所生的

著作便无需列印该宣告。）

这些要求对修改过的著作是整体适用的。倘著作中

可识别的一部份并非衍生自本程序，

并且可以合理地认为是一独立的、个别的著作，

则当您将其作为个别著作加以发布时，本授权

及其条款将不适用于该部分。然而当您

将上述部分，作为基于本程序所生著作的一部而发布时，

整个著作的发布必须符合本授权条款的规定，

而本授权对于其他被授权人所为的许可

及于著作整体。

因此，本条规定的意图不在于主张或剥夺

您对于完全由您所完成著作的权利；应该说，本条规定意在

行使对基于程序所生的之衍生著作或集合著作

发布行为的控制权。

此外，非基于本程序所生的其他著作与本程序

(或基于本程序所生的著作) 在同一储存或发布的媒介上的单纯聚集行为, 并不会使该著作因此受本授权条款约束。

3. 您可以依前述第一、二条规定, 复制与发布本程序

(或第二条所述基于本程序所产生的著作) 的目的码

或可执行形式, 但您必须符合以下要件:

a) 附上完整、相对应的机器可判读源码,

而这些源码必须依前述第一、二条规定

在经常用以作为软件交换的媒介物上发布; 或

b) 附上至少三年有效的书面报价文件,

提供任何第三人在支付不超过实际发布源码

所需成本的费用下, 取得相同源码的完整机器

可读复制物, 并依前述第一、二条规定

在经常用以作为软件交换的媒介物上

发布该复制物; 或

c) 附上您所收受有关发布相同源码的报价资讯。

(本项选择仅在

非赢利发布、且仅在您依前述 b 项方式

自该书面报价文件收受程序目的码或可执行形式时,

始有适用。)

著作的源码, 是指对著作进行修改时

适用的形式。对于一个可执行的著作而言, 完整的源码

是指著作中所包含所有模组的全部源码, 加上

相关介面的定义档, 还加上用以控制该著作

编译与安装的描述。然而,

特别的例外情况是, 所发布的源码并不需包含

任何通常会随著所执行作业系统的主要组成部分

(编译器、核心等等) 而发布的软件 (无论以源码或

二进位格式), 除非该部分本身

即附加在可执行程序。

若可执行码或目的码的发布方式, 是以指定的地点

提供存取位置供人复制, 则提供可自相同地点复制

源码的使用机会, 视同对于源码的发布,

然而第三人并不因此而负有将目的码

连同源码一起复制的义务。

4. 除本授权所明示的方式外, 您不得对本程序加以

复制、修改、再授权或发布。任何试图

以其他方式进行复制、修改、再授权或者发布本程序的行为均为无效，并且将自动终止您基于本授权所得享有的权利。

然而，依本授权规定自您手中收受复制物或权利之人，只要遵守本授权规定，他们所获得的授权并不会因此终止。

5. 因为您并未在本授权上签名，所以您无须接受本授权。然而，除此之外您别无其他修改或发布本程序或其衍生著作的授权许可。若您不接受本授权，则这些行为在法律上都是被禁止的。因此，藉由对本程序（或任何基于本程序所生的著作）的修改或发布行为，您表示了对于本授权的接受，以及接受所有关于复制、发布或修改本程序或基于本程序所生著作的条款与条件。

6. 每当您再发布本程序（或任何基于本程序所生的著作）时，收受者即自动获得原授权人所授予依本授权条款与条件复制、发布或修改本程序的权利。您不得就本授权所赋予收受者行使的权利附加任何进一步的限制。您对于第三人是否履行本授权一事，无须负责。

7. 若法院判决、专利侵权主张或者其他任何理由（不限于专利争议）的结果，使得加诸于您的条件（无论是由法院命令、协议书或其他方式造成）与本授权规定有所冲突，他们并不免除您对于本授权规定的遵守。若您无法同时符合依本授权所生义务及其他相关义务而进行发布，那么其结果便是您不得发布该程序。例如，若专利授权不允许其他人直接或间接取得复制物，通过您以免付权利金的方式再发布该程序，您唯一能同时满足该义务及本授权的方式就是彻底避免进行该程序的发布。

若本条任一部份在特殊情况下被认定无效或无法执行时，本条其余部分仍应适用，且本条全部于其他情况下仍应适用。

本条的目的并不在诱使您侵害专利或其他财产权的权利主张，

或就此类主张的有效性加以争执；

本条的唯一目的，是在保障藉由公共授权惯例所执行自由软件发布系统的

完整性。许多人信赖

该系统一贯使用的应用程序，而对经由此系统发布的大量软件有相当多的贡献；

作者/贡献者有权决定他或她是否希望

经由其他的系统发布软件，而被授权人则无该种选择权。

本条的用意在于将本授权其他不确定的部分彻底解释清楚。

8. 若因为专利或享有著作权保护的介面问题，而使得本程序的发布与/或使用局限于某些国家时，则将本程序置于本授权规范之下的原著作权人得增列明确的发布地区限制条款，将这些国家排除在外，而使发布的许可只限在未受排除的国家之内或之中。在该等情况下，该限制条款如同以书面方式订定于本授权内容中，而成为本授权的条款。

9. 自由软件基金会得随时发表通用公共授权的修正版与/或新版本。新版本在精神上将近似于目前的版本，然而在细节上或所不同以因应新的问题或状况。

每个版本都有不同的版本号码。若本程序指定有授权版本号码，表示其适用该版本或是“任何新版本”时，您可以选择遵循该版本或任何由自由软件基金会日后所发表新版本的条款与条件。若本程序并未指定授权版本号码时，您可以选择任一自由软件基金会所发表版本。

10. 若您想将部分本程序纳入其他自由程序，而其发布的条件有所不同时，请写信取得作者的许可。若为自由软件基金会享有著作权的软件，请写信至自由软件基金会；我们有时会以例外方式予以处理。我们的决定取决于两项目标：确保我们自由软件的所有衍生著作均维持在自由的状态，并广泛地促进软件的分享与再利用。

无担保声明

11. 由于本程序是无偿授权，因此在法律许可范围内，本授权对本程序并不负担担保责任。非经书面声明，著作权人与/或其他提供程序之人，无论明示或默许，均是依“现况”提供本程序而并无任何形式的担保责任，其包括但不限于，就适售性以及特定目的的适用性为默示性担保。有关本程序品质与效能的全部风险均由您承担。如本程序被证明有瑕疵，您应承担所有服务、修复或改正的费用。

12. 非经法律要求或书面同意，任何著作权人或任何可能依前述方式修改与/或发布本程序者，对于您因为使用或不能使用本程序所造成的一般性、特殊性、意外性或间接性损失，不负任何责任（包括但不限于，资料损失，资料执行不精确，或应由您或第三人承担的损失，或本程序无法与其他程序运作等），即便前述的著作权人或其他人已被告知该等损失的可能性时，也是一样。

-条文结束-

您的新程序该如何采用这些条款？

如果您开发了一个新程序，并且希望能够让它尽可能地被大众使用，达成此目的的最好方式就是让它成为自由软件，任何人依这些条款规定都能就该软件再为发布及修改。

为了做到这一点，请将以下声明附加到程序上。最安全的作法，是将声明放在每份源码档案的起始处，以有效传达无担保责任的讯息；

且每份档案至少应有“著作权”列以及本份声明全文位置的提示。

<用一行描述程序的名称与其用途简述>

版权所有 (C) <年份> <作者姓名>

本程序为自由软件；您可依据自由软件基金会所发表的 GNU 通用公共授权条款规定，就本程序再为发布与/或修改；无论您依据的是本授权的第二版或（您自行选择的）任一日后发行的版本。

本程序是基于使用目的而加以发布，

然而不负任何担保责任；亦无对适售性或特定目的的适用性所为的默示性担保。详情

请参照 GNU 通用公共授权。

您应已收到附随于本程序的 GNU 通用公共授权的副本；

如果没有，请写信至自由软件基金会：

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

同时附上如何以电子及书面信件与您联系的资料。

若程序是以交互方式运作时，请在交互式模式开始时，

输出简短提示如下：

Gnomovision 第 69 版，版权所有 (C) 年份 作者姓名

Gnomovision 不承担担保责任，欲知详情请键入 “show w”。

这是一个自由软件，欢迎您在特定条件下

再发布本程序；欲知详情请键入 “show c”。

所假设的指令 “show w” 与 “show c” 应显示

通用公共授权的相对应条款。当然，您可以使用

“show w” 与 “show c” 以外的指令名称；甚至以

鼠标键击或选菜单方式进行——只要是合于您程序需要的方式都可以。

如有需要，您亦应取得您的雇主（若您的工作为程序设计师）或

学校就本程序所签署的“著作权放弃承诺书”

。其范例如下，您只要修改姓名即可：

Yoyodyne 公司，兹此放弃 James Hacker 所写之 “Gnomovision” 程序

（该程序产出编译器所需资讯）所有的著作权利益。

<Ty Coon 公司签章>，1989 年 4 月 1 日

Ty Coon 公司，副总裁

本通用公共授权并不允许您将本程序并入

专有程序中。若您的程序是一子程序函数库时，您可能

认为允许专有应用程序与该函式库相联结会更有

帮助。若这是您所想做的，请使用 GNU 函式库通用公共授权

代替本授权。