关于 Online Change 的若干问题汇总

By Lizzy Chen 陈利君 2025.04.25

背景：对于大型项目调试时每次修改代码后Login速度太慢，有时明明是同一个项目程序下载时也会提示Online Change，以及有时不能Online Change只能重新Download，而Download会引起PLC Stop。如果机器正在生产，强行Download就可能后果严重。

本文通过Online Change的基本原理分析，结合Information System、相关同事的过往经验和文档，加上近期实际的测试结果，试图就以上典型现象进行深度研究，并给出可行的优化建议。

**目 录**

[1 Online Change的基本原理 3](#_Toc196496714)

[1.1 Online Change提示窗 3](#_Toc196496715)

[1.2 Online Change的机制 3](#_Toc196496716)

[1.2.1 在线修改时如何处理变量 3](#_Toc196496717)

[1.2.2 在线修改时如何处理程序 4](#_Toc196496718)

[1.3 在线修改的应用风险 4](#_Toc196496719)

[1.3.1 代码切换耗时，其间通讯暂停，或PLC周期溢出 4](#_Toc196496720)

[1.3.2 现有变量的值可能无效 5](#_Toc196496721)

[1.3.3 Pointers/References（指针/引用）可能无效 5](#_Toc196496722)

[1.4 Build 4024新功能：备用内存 5](#_Toc196496723)

[1.4.1 什么是备用内存？ 5](#_Toc196496724)

[1.4.2 如何设置备用内存 6](#_Toc196496725)

[1.4.3 如何查看TwinCAT占用内存大小 8](#_Toc196496726)

[1.5 观察在线修改次数 9](#_Toc196496727)

[2 不能直接Login的可能性及解决办法 10](#_Toc196496728)

[2.1 库版本自动升级导致的重新编译，产生了新的编译文件 10](#_Toc196496729)

[2.2 有关Visu 的内部库版本报错 10](#_Toc196496730)

[2.3 非倍福的Libray在发布时使用了相同的库版本号 11](#_Toc196496731)

[3 必须 Download的可能性及解决办法 11](#_Toc196496732)

[3.1 上传或者复制PLC源代码后缺少 .compileinfo编译文件 11](#_Toc196496733)

[3.2 修改了变量或者FB实例的类型而名称不变， 12](#_Toc196496734)

[3.3 添加/删除了校验功能，例如 CheckBounds 12](#_Toc196496735)

[3.4 使用了不同的编译器版本 13](#_Toc196496736)

[3.5 XAE无法区分不同版本的全局系统数据类型 13](#_Toc196496737)

[3.6 其它很少在Online Change时实际发生的情况 14](#_Toc196496738)

[4 显示Online Change的变更内容 14](#_Toc196496739)

[4.1 Build 4024，使用Details按钮 14](#_Toc196496740)

[4.2 Build 4022及以下，编译器定义 "uptodate\_in\_message\_out" 15](#_Toc196496741)

[4.3 如果控制器上有源代码，可直接使用XAE的Tools｜TwinCAT Project Compare 16](#_Toc196496742)

[5 如何加快Online Change的速度 16](#_Toc196496743)

[5.1 直接操作HMI界面，不必等待PC上的代码Online刷新 16](#_Toc196496744)

[5.2 升级编程电脑的CPU性能，可以提升Online Change的速度 17](#_Toc196496745)

[5.3 改善编程PC和PLC之间网络传输也可以提升Online Change的速度 17](#_Toc196496746)

[6 附：TC2项目的在线修改功能与TC3的区别 18](#_Toc196496747)

[6.1 观察在线修改的次数 18](#_Toc196496748)

[6.2 程序转移后如何才能直接Login 18](#_Toc196496749)

[6.3 TC2没有的功能 19](#_Toc196496750)

[6.3.1 TC2不能设置备用内存 19](#_Toc196496751)

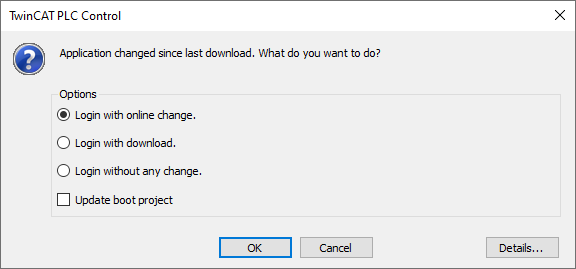
[6.3.2 TC2不能显示 Online Change的细节 19](#_Toc196496752)

[6.3.3 TC2不区分库文件的不同版本 19](#_Toc196496753)

# Online Change的基本原理

## Online Change提示窗

用一个 PLC 项目 Login，如果不同于目标机上的实际项目，系统就会弹出消息窗，要求选择登陆方式：



Online Change与Download的区别

|  |  |  |  |
| --- | --- | --- | --- |
| 选项 | Login with Download | Login with Online Change | |
| 机制 | 完整地切换程序，  并重新初始化 | 只装载变更部分。根据变更内容不同，可能需要进行简单的或更复杂的在线修改 | |
| 简单的 | 复杂的 |
| 仅更改程序 | 含新增变量  或，数据类型 |
| 可以简单/快速地完成在线修改 | 复杂的在线修改，  需要移动实例的内存 |
| 优点 | 可定义一致的初始状态 | 当前程序不会Stop，变量会保持原来的值。  可以在机器生产运行中执行 | |
| 缺点 | 当前程序将Stop。不能在机器运行过程中执行 Download！ | 部分代码切换存在应用风险！ | |

## Online Change的机制

### 在线修改时如何处理变量

#### 未做修改的变量

保留在其内存位置并保持原值

#### 删除的变量

会留下内存空隙，这些空隙最终会在下一次 Online Change 时填充（如果可能的话）。

#### 新建的变量

获得一个新的空闲内存位置：要么在空隙中，要么在已用内存块的末尾，要么在一个新开辟的内存块中。FB 实例的变量总是存储在内存中的一个块中（最终会留有间隙，但中间不会插入其他变量）。如果更改前 FB 实例的内存位置没有为新变量预留足够的空间，就会在 Online Change 的时候移动整个 FB 实例

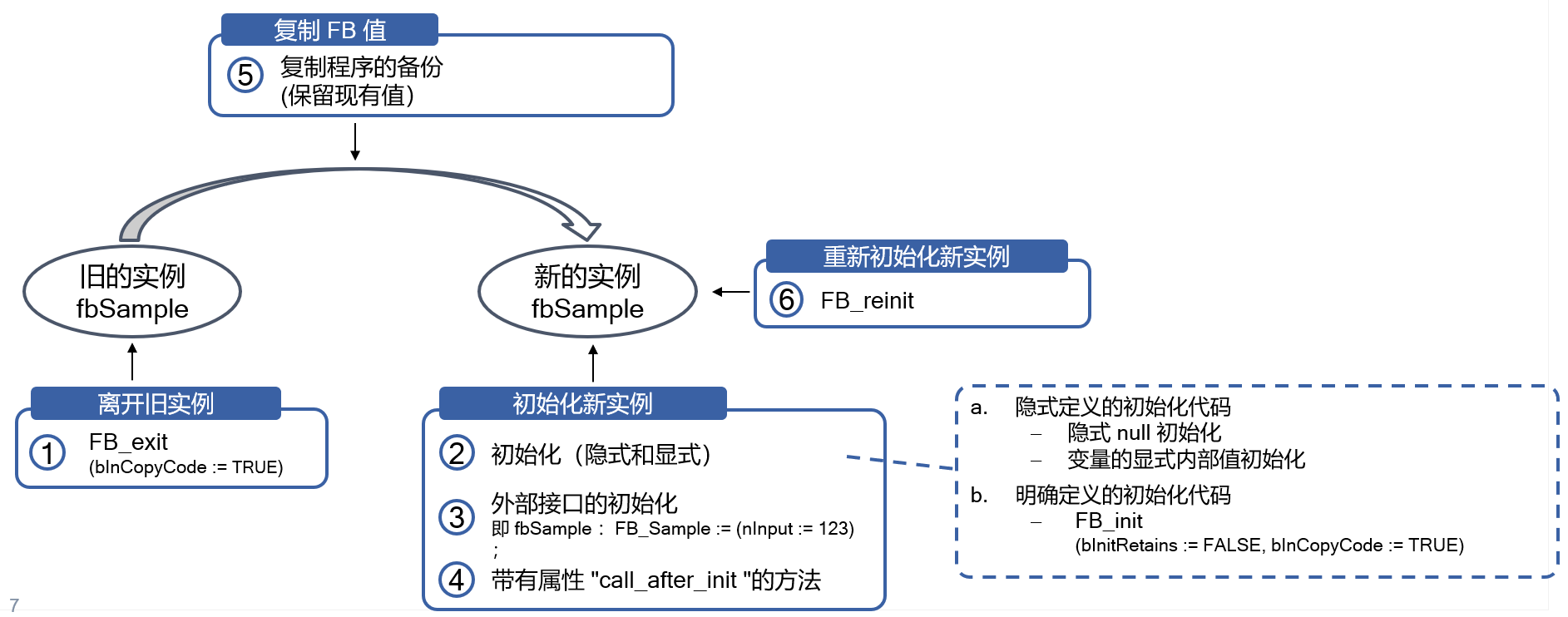
#### 数组和结构体的元素如果有变化

如果改变了数组变量的元素数量，或者在结构体类型中插入或者删除了元素、改变了元素的类型，Online Change时就会这种数组和结构体**实例**分配新的内存位置。现有元素的值被复制到新的位置，新增的元素测赋以初始值。

### 在线修改时如何处理程序

如果更改前 FB 实例的内存位置没有为新变量 **预留** 足够的空间，就会在 Online Change 的时候移动整个 FB 实例。

注意，仅在TC3.1.4024以后才有为FB预留内存的功能，而TC2和TC3.1.4022及以下的PLC在线修改时，如果FB新增了变量就总是会移动整个FB实例。



如果 FB 实例包含动态创建的内存（通过 \_\_NEW 创建），其指针变量值将被保留，并在 Online Change 时复制到“新的”FB 实例中。对于用户数据类型的\_\_NEW，在 Online Change 时更改，否则就会强制Download。

## 在线修改的应用风险

Online Change 后，更改后的代码直接生效。如果没有经过充分测试，就可能会出问题。并且如果发现不对，也不能撤消（undo）。只有新建的变量会初始化，而程序不会从头执行，使用时应充分考虑到可能的后续反应。

可能的风险还包括：

### 代码切换耗时，其间通讯暂停，或PLC周期溢出

再就是PLC只在只在当前语句执行完，下一行语句未开始的时候做代码切换。而代码切换期间不会刷新IO。如果改动涉及的程序和FB实例多且复杂，代码切换费时，可能会因通讯暂停而导致周期溢出或其它问题。

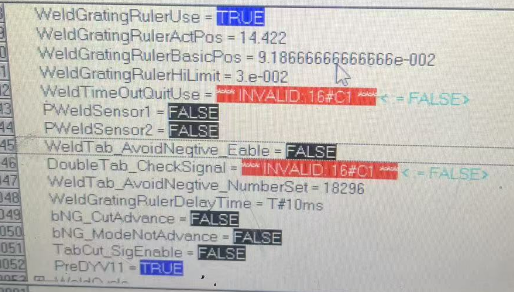
代码切换耗时主要是因为增加了变量或结构元素，而没有备用内存或者备用内存耗尽。对于Build 4024及以上，通过设置备用内存可大大缩短这个时间。

切换代码当期的Task excute time增加量，即使有所增加也是微秒级。所以对于ms级的任务周期，基本不用考虑。具体增加的时间，感兴趣可以通过跟踪全局变量 Last excute time来观察。

### 现有变量的值可能无效

同样，对于Build 4024及以上可以通过设置备用内存避免这种风险。而对于TC2和Build 4022及以下，应尽量避免增加FB的内部变量或者改动结构体的元素，因为在FB实例在内存中移动以及相关接口迁移的过程中，出错的风险也增加了。

现有变量值无效可能引起致命的 Runtime异常或程序误动作。最典型的数据无效是在BOOL变量中存放了非0非1的值，如果Login并定位到这个变量声明区，就可以看到有些变量值显示为红色，并标记为\*\*\*INVALID\*\*\*：



### Pointers/References（指针/引用）可能无效

在线修改时指针会保留 Online Change 之前的值，如果指针指向的变量在 Online Change 期间改变了位置，那么指针就会无效，可能引起Runtime 异常。

为了避免这种问题，指针应当每个周期更新，至少在 Online Change 后要更新。

相比之下 TC3 中使用的Interface，就会在Online Change 后自动更新，不存在指向内存无效的问题。

## Build 4024新功能：备用内存

### 什么是备用内存？

备用内存的全称是在线修改德用内存，即Online Change Memory Reserve，是TC3.1.4024及以后版本为FB增加的一个属性，用于为FB实例开辟一块额外的内存，供其Online Change(在线修改) 时存放新增的变量。

如前所述，PLC程序运行时，为所有FB实例分配了内存，其中包含了所有的FB局部变量。在Online Change时，如果新增了变量，为了不影响相邻内存区的值，只好把原来FB实例所占的内存复制到另一块内存并增加字节数。这样做有两个后果：一是原先指向FB局部变量的指针可能会失效，二是复制内存后，FB的input和Output接口和其它变量的映射关系会相应迁移。所以这种复制不仅花时间，还增加了出错的风险。

为此TC3.1.4024以后就增加了为FB提供了Online Change Memory Reserve的属性，用于设置在线修改时备用内存的大小。

根据Infosys中的描述Online Change Memory Reserve：

<https://infosys.beckhoff.com/content/1033/tc3_plc_intro/5356021131.html?id=7653146620267255574>

建议为功能块的Online Change配置备用内存。只要备用内存足够大，以后Online Change时在FB的变量声明中增加一个或多个变量后，功能块的实例就无需复制到新的存储空间，而是在备用内存中存放新增变量，Online Change的速度就会更快，出错的风险更小。

##### 什么时候修改设置备用内存？

最好是Login之前就设置好备用内存，否则在线修改的时候需要复杂的操作。（多复杂？）

##### 备用内存多大合适？

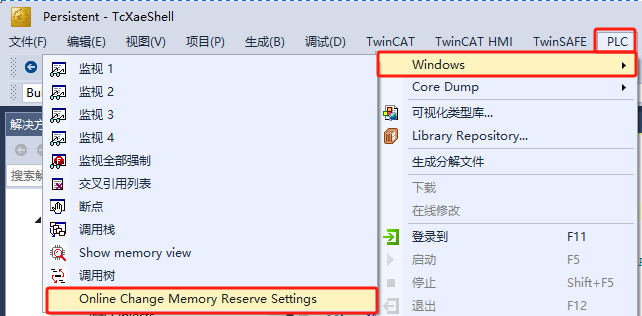
根据Information System中的描述[**Memory usage**](https://infosys.beckhoff.com/content/1033/cx7293/18089153803.html?id=4615977951752461161)

请注意，在 TwinCAT 中执行Online Change时，短时间内需要更多内存。如果要使用Online Change功能，最好始终保留一定的备用内存。在最极端的情况下，执行Online Change可能需要 **两倍** 于当前消耗的 PLC 程序。如果没有足够的内存用于Online Change，出错风险将会增加。

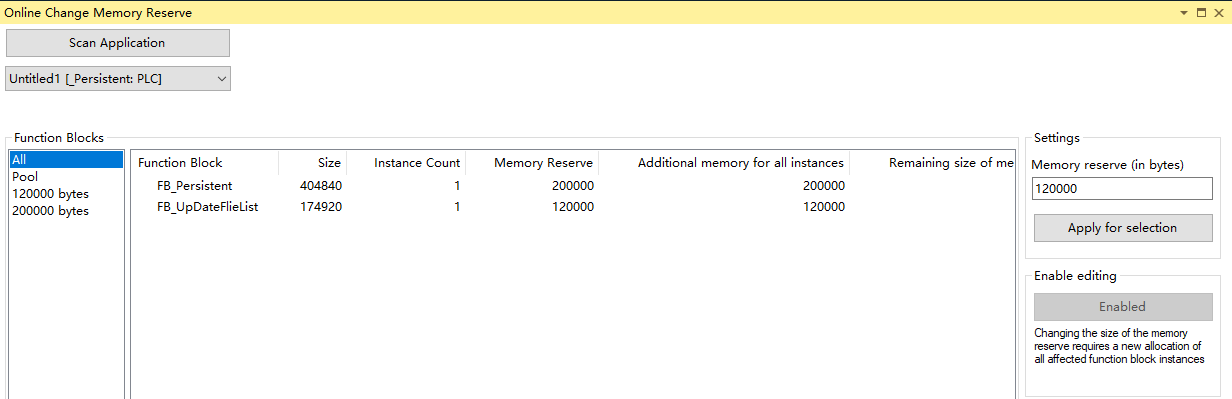
实际上，只要CX5100及以上的控制器，内存从2G、4G、8G到64G不等，而PLC的FB内存最多以kByte或者MByte计，所以通常有足够容量分配给备用内存。

### 如何设置备用内存

##### 菜单 PLC | Windows | Online Change Memory Reserve Settings



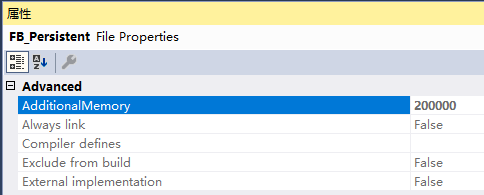
##### 界面 Online Change Memory Reserve



##### 操作步骤

|  |  |
| --- | --- |
| 如果项目中有多个PLC，需要选择目标项目 |  |
| 显示PLC中所有FB | 修改程序以后，都要点击上图中的“Scan Application”，以更新显示所有FB的名字及其内用内存（Online Change Memory Reserve）的占用情况 |
| 配置前先设置允许编辑 | 点击 Enable editing: |
| 修改目标值，最大可以设置到当前Size的一半或者同等大小。视当前控制器的内存容量而定 |  |
| 可以同时选择多个FB，在右图中设置预留内存，并点击“Apply……”，则所有选中的FB的备用内存都设置为该大小。 | 用此方法可以全部设置为0，以清除备用内存。 |
| 应用以后会提示编译 |  |

##### 备选：在属性中设置FB的备用内存



由于属性中看不到这个FB实际用了多少内存，所以还是在　PLC | Windows | Online Change Memory Reserve　中设置比较方便。

### 如何查看TwinCAT占用内存大小

如果担心为FB设置过大的Online Change Memory Reserve是否会导致内存不足，可以观察控制器的内存占用情况。

##### 控制器上的内存分配

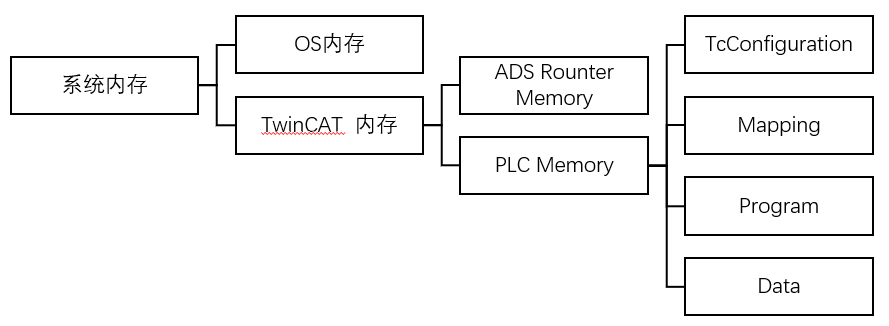
根据Information System中CX7000的内存描述，

<https://infosys.beckhoff.com/content/1033/cx7293/18089153803.html?id=4615977951752461161>

CX7293 有 512 MB 的主内存，用于固件（TwinCAT/RTOS）和 TwinCAT（TwinCAT 内存）。TwinCAT 内存又分为ADS Rounter Memory和 PLC Memory。ADS Rounter Memory用于 ADS 通信，PLC Memory用于实际 PLC 程序，包括 TcConfiguration、映射和数据。

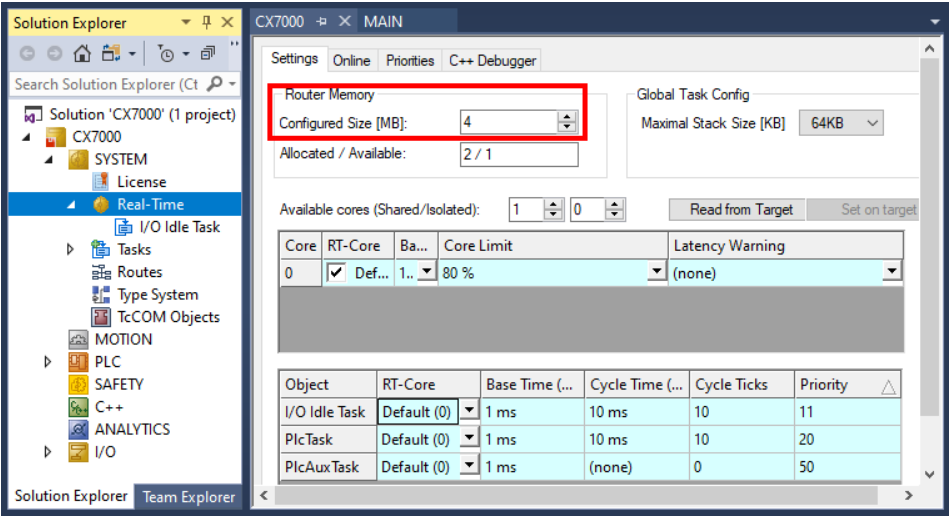
CX7293 可使用 112 MB 的 TwinCAT 内存。由于内存大小有限，因此必须检查内存使用情况，如果超出，则调整 PLC 项目。

所以，TwinCAT控制器内存应包括以下内容：



##### ADS Router Memory

一方面，您可以在 TwinCAT 中调整ADS Router Memory的大小，并根据实际使用的 ADS 通信设置较小的ADS Router Memory。

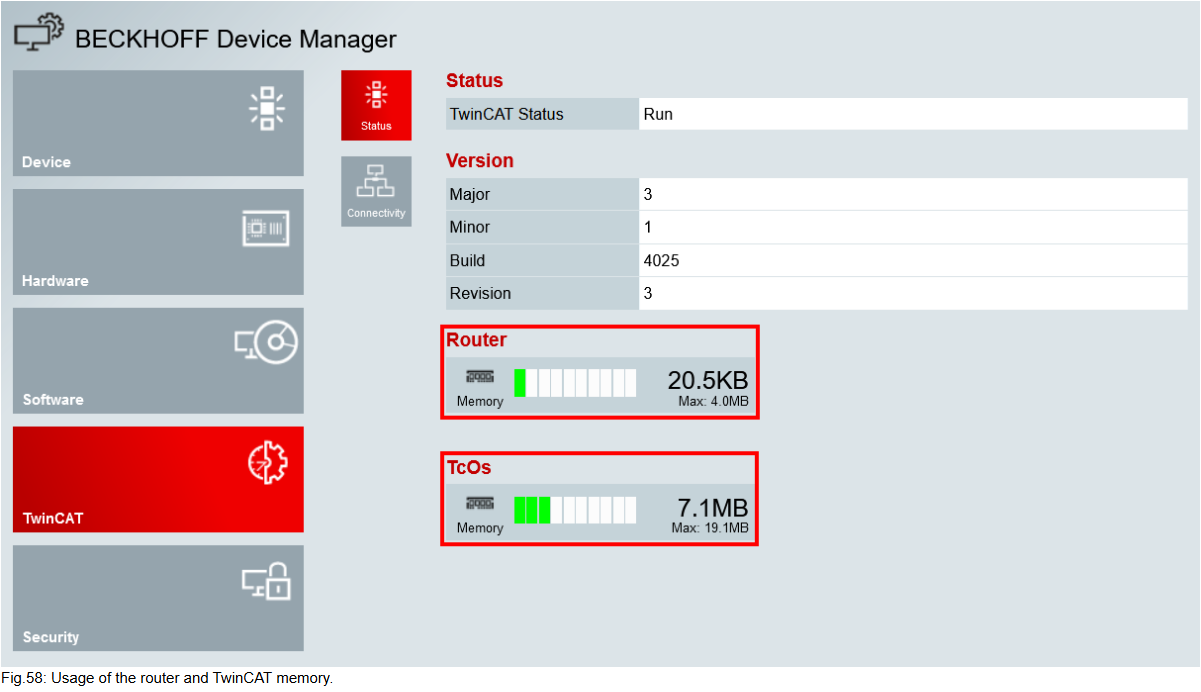


默认情况下，在 TwinCAT 中输入的值为 32 MB。建议 CX7293 的ADS Router Memory为 32 MB，如果几乎不使用 ADS 通信，内存可以更小。不过，ADS Router Memory至少应为 1 MB，且不应小于 1 MB。ADS Router Memory的使用量可通过功能块 FB\_GetRouterStatusInfo 或 Beckhoff Device Manger获取。

请注意，ADS Router Memory只有在**关机/开机**时才会重新创建。重新启动 TwinCAT 是不够的。根据经验，为 ADS 通信设置的ADS Router Memory越小，应用程序（即 PLC 程序、TcConfiguration、映射和数据）就越大。

##### 用Beckhoff Device Manager观察内存占用

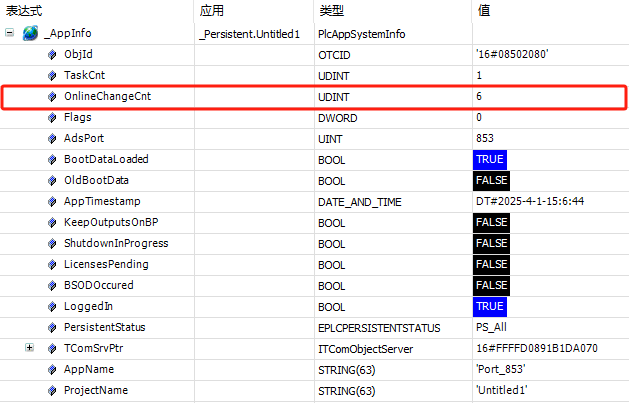
通过功能块 FB\_GetRouterStatusInfo，或者通过 Beckhoff Device Manager，可以确定 ADS Router Memory需要内存大小。



## 观察在线修改次数

#### Online Change次数

在全局变量 \_AppInfo中可以看到在线修改次数“OnlineChangeCnt”。每次Online Change都会额外消耗内存，所以次数越大，消耗的临时内存越多。



在备用内存设置界面，也可以查看其消耗情况。详见 1.4.2如何设置备用内存

#### Online Change文件在哪里

Online Change文件Port\_xxx.oce包含了自上次全新下载以来的所有在线修改信息。

**存储位置：**

< TC3.1.4026.0:  *C:\TwinCAT\3.1\Boot\Plc*

>=TC3.1.4026.0:  *C:\ProgramData\Beckhoff\TwinCAT\3.1\Boot\Plc*

**创建时间：Login或者Online Change的时候**

# 不能直接Login的可能性及解决办法

在一台编译PC上打开PLC项目，选择好目标控制器，选择Login时不会弹出Online Change对话框，就叫做直接Login。

通常调试过程中，上次Login之后没有任何程序修改，这一次就可以直接Login。实际在每次Login时编程PC都会读取控制器的C:\TwinCAT\3.1\Boot\Plc\port\_xxx\_act.tizip和Port\_xxx.oce的信息。解压tizip文件可以发现里面只有一个xxx.compileinfo文件，编程PC将它的信息与当前项目路径下的 …\TwinCAT\PLC\\_CompileInfo\ xxx. Compileinfo进行对比。**如果编译信息完全一致，**就可以直接Login，否则就会弹出Online Change对话框。

一旦编程PC上重新生成了编译文件，就会与目标控制器上不同，就不能直接Login了。

那么什么情况下编译PC会重新生成编译文件呢？最常见的当然是修改了代码或增减了变量，这是用户容易觉察的，还有一些情况是不易觉察的。

## 库版本自动升级导致的重新编译，产生了新的编译文件

常见一种不易觉察的重新编译，就是同一个项目放到另一台PC上运行，而那台PC上的XAE版本不同，如果PLC项目中引用的Library使用了默认设置“as always newest”，那么前后使用的Library中的同名FB可能变量和代码都变了，Login之前必然会再次编译。

##### 解决办法

为了避免这种问题，最简单的方法是在“Reference (引用) ”节点的右键菜单中选择 “Set to Effective Version （ 设置为有效版本 ）”，这样可以一次性设置所有Library/Placeholder。同时，TwinCAT 开发环境和 Runtime 要使用相同的版本（XAE/RM），如果不能一致，至少要安装相同大版本的 Remote Manager (远程管理器) ，用于将项目 Download到目标控制器。

## 有关Visu 的内部库版本报错

含PLC HMI界面的PLC程序，有时候一打开就弹出一长串Visu相关的报错。通常，选择Visu Profile配置文件，可以将Visu 相关的Library/placeholder自动设置为对应的版本。但在Build 4024中还无法手动修改Visu库的最终版本。原因是Visu 库内部引用了其它库，而这些库并不由 Visu Profile 设置。例如：

**4022.10 版 Visu Profile文件对应的内部引用库版本**



**4024.7 版 Visu Profile文件对应的内部引用库版本**



##### 解决办法

对于这些Library/Placeholder，也应当将其设置为有效版本。这也可以通过Reference 节点右键菜单中的“Set to Effective Version”实现。但如果此后使用不同版本的XAE打开项目或者更改了Visu Profile配置文件，就还得手动修改Visu内部库的版本。

更高版本的Build 4024版Visu Profile所引用库会不会再改动，得具体再看。

## 非倍福的Libray在发布时使用了相同的库版本号

用户自己封装的Library，如果发布时比较随意，填写Placeholder时和此前的版本重复，那么换了一台电脑，碰巧使用了之前版本的库，那么编程PC就很难发现了。Login之前也会认为程序有变化，会进行编译。

##### 解决办法

复制项目的时候，连引用的库一起复制，并在新的环境中安装覆盖原来的库。

# 必须 Download的可能性及解决办法

如前所述，Download时程序会完整下载并重新初始化，程序的连贯性一致性更好。但是这样必须会导致PLC Stop，如果设备正在生产时Download就会引起停机。那么调试过程中修改了程序试图Login时，哪些情况只允许Download呢？

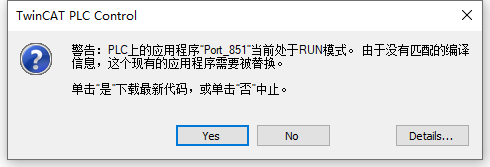
## 上传或者复制PLC源代码后缺少 .compileinfo编译文件

直接Login一个程序而不创建引导程序，或者激活配置下载程序Login之后如果删除编译文件，并删除控制器Boot下的所有文件，反复Logout/Login，都可以：

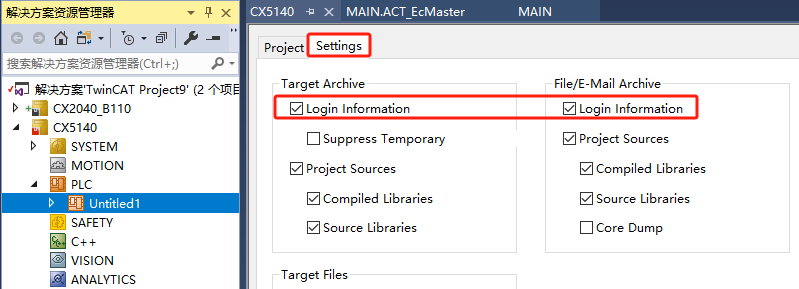
\PLC项目名称\**\_CompileInfo**\ 8527A260-770C-9835-6C83-EE04E01D8112.compileinfo

这说明编程电脑上的.compileinfo并不是跟控制器Boot下的tizip或其它文件进行对比，而是与编译电脑内存保存的上次登陆信息与控制器内存上的登陆信息进行对比。但是控制器重启，或者编程电脑关闭项目再重新打开时，这些信息就要从文件装载。编程电脑从.compileinfo装载，而控制器就从.tizip装载。

但是重命名或者删除这个文件，关闭当前项目再打开，就不能再直接Online，弹出提示：



所以**程序转移**时，或者**下载源代码再上载**时，都需要**选中Login信息**，目的就是复制这个文件夹的内容。



上图中，Target Archive设置选中Login Info，是为了创建引导程序时能同时把编译信息下载到控制器，以便将来从控制器上载源代码后可以直接Login。

File/E-Mail Archive设置选中Login Info，是为了项目打包归档转移到其它电脑后可以直接Login。当然，如果是整个项目文件夹直接压缩复制到别的电脑，这个选项就不影响了。

激活配置创建引导程序时，不同的Target Archive选项时控制器Boot生成的文件：

|  |  |  |
| --- | --- | --- |
| None | Login Infomation | Project Sources |
| .app  .cid  .crc  .occ  .oce | …boot.tizip  …act.tizip |  |

## 修改了变量或者FB实例的类型而名称不变，

例如：

从 "*fbTimer：TON*"

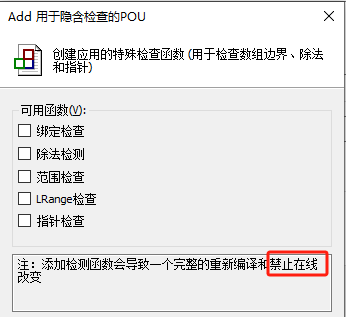
改为 "*fbTimer: TOF*"

##### 解决办法：避免直接修改类型

避免直接修改类型，但可以新建一个变量，并在程序中做变量替换。

## 添加/删除了校验功能，例如 CheckBounds

为了排查Page Fault问题，有时客户会使用“Add POU for implicit”。添加这种函数之后PLC程序会重新编译，不能再Online Change，而必须Download 重新启动PLC程序。



##### 解决办法：正常调试时不要增加Implicit函数

如果已经出现了Page Fault，其实就不在乎PLC是否重启了。并且在TC3中也已经有Core Dump功能，启用这个功能后再出现Page Fault就会在控制器中生成一个Dump文件。在XAE中装载这个文件，就能找出Page Fault之前停在哪一行。

当然这些隐含检查也有它的优势，只是正常调试的时候不要去增加这些函数。

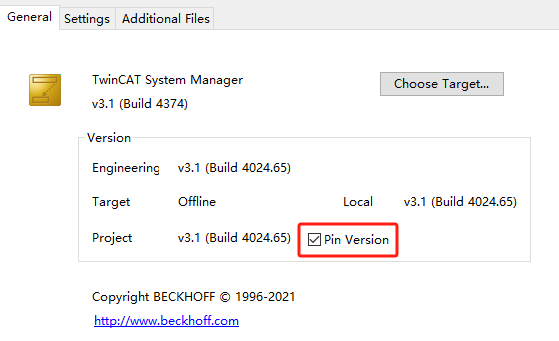
## 使用了不同的编译器版本

包括但不限于1.3 1) *库版本自动升级导致的重新编译*

在 PLC 项目属性中修改了它的编译器版本或者目标机上的项目在上次 Download 时使用的编译器版本，与当前开发环境中设置/使用的版本不同。

##### 提示1：设置 Pin version

所以为了避免引起停机，不要在Online Change前修改编译器版本，并在项目中选中“pin version”



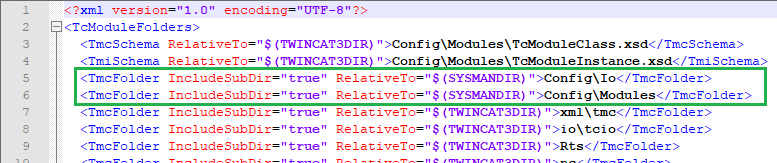
最好始终以相同的 XAE/RM 版本打开PLC项目，并且与控制器上XAR运行版本一致，即上图中的Engineering、Target和Project的版本保持一致。

## XAE无法区分不同版本的全局系统数据类型

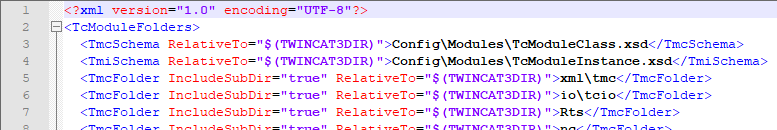
实际上很难做到所有工程师的编程PC都保持Engineering、Target和Project一致，除非使用多个虚拟机，每个虚拟机固定一个版本。通常的情况是工程师的编程PC上安装上相对较新的Build 4024，而为了维护若干年前交付的机器又安装了一些其它版本的RM。

全局系统数据类型包含一些基本的数据类型，比如全局变量\_PlcAppInfo的类型PlcAppSystemInfo，不是包含在库中，而通过指定路径下的.tmc文件定义。早年的TC3没有区分不同版本的.tmc文件，所以安装了新版本就覆盖上旧版本的.tmc文件，导致在旧版下创建的PLC项目在升级后的XAE中打开时这些.tmc文件中定义的全局系统数据类型可能会改变，再Login时就要求Download。

为了解决这个问题，倍福对每一个TC3大版本都做了升级，只要XAE/RM >= 4016.38、>= 4018.57、>= 4020.56、>= 4022.25 和 >= 4024.0，XAE所在的PC就会在TcModuleFolders.xml文件中指定.tmc文件（C:\TwinCAT\3.1\Config\Io\）的相对路径：



检查这个文件，如果发现这两行缺失，就说明系统无法区分不同版本的.tmc文件：



此时需要升级小版本。方法如下：

* 从 “C:\TwinCAT\3.1\Config ”中剪切 “IO ”文件夹，并粘贴到一个备份路径；
* 运行 PC 上安装的 XAE 开发环境的最高版本安装程序，选择 “ Repair ”模式；
* 将 “Isg ”文件夹从备份路径的 “IO ”文件夹粘贴到 “C:\TwinCAT\3.1\Config\Io ”

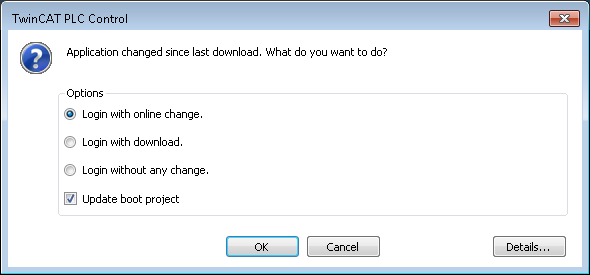
## 其它很少在Online Change时实际发生的情况

* 修改了 PLC 项目的task reference

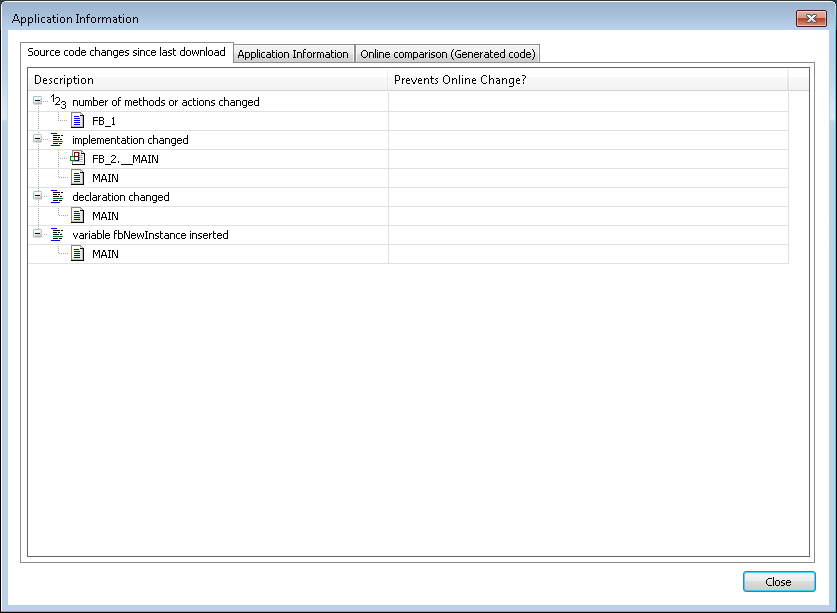
# 显示Online Change的变更内容

## Build 4024，使用Details按钮

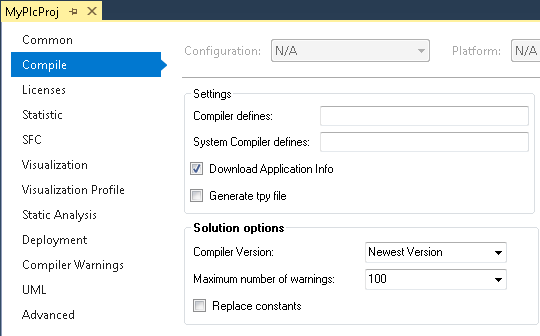
为了显示每次在线修改时具体有哪些内容变量，在TC3.1.4024中增加了Details 按钮：



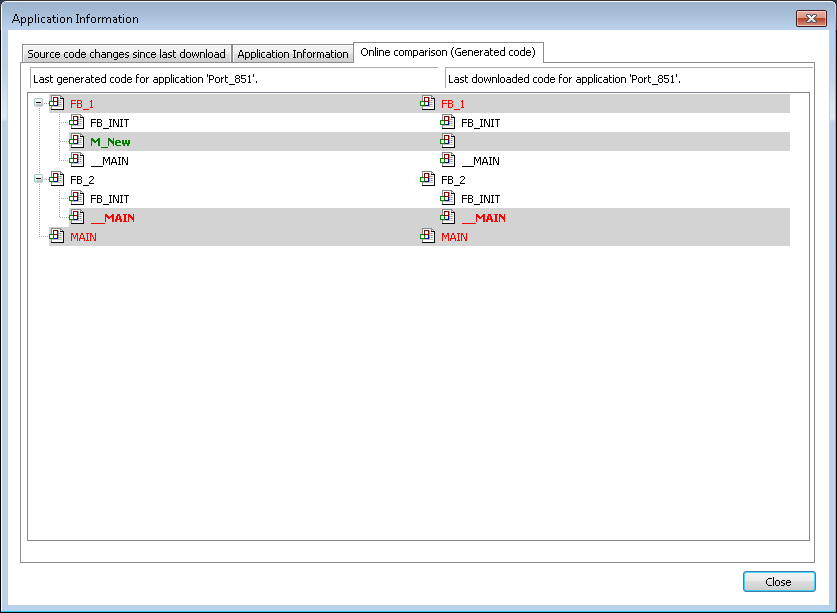
点击上图中的Details按钮，就会显示最近的代码变化：



如果在项目属性的编译选项中勾选了“Download Application Info”，还可以显示更多修改详情Online comparison（Generated code）：

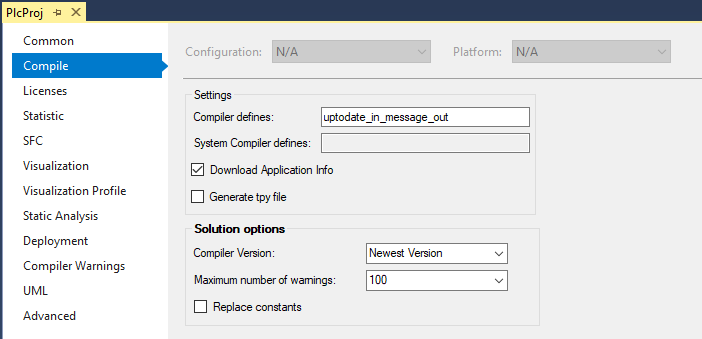


结果：



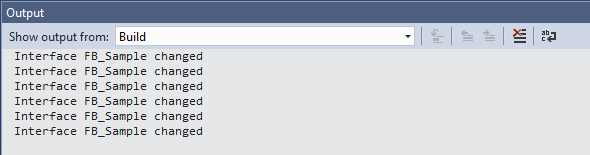
## Build 4022及以下，编译器定义 "uptodate\_in\_message\_out"

在 PLC 项目属性的 Compiler define 中输入 "uptodate\_in\_message\_out "，

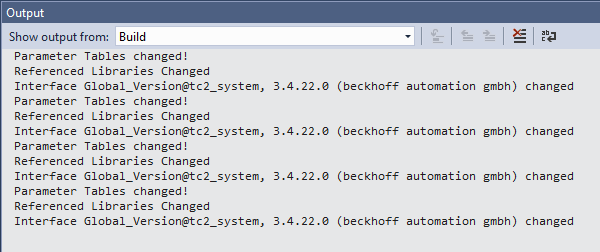


编译解决方案 Build Solution，Login，PLC 项目的更改内容将在输出窗口中以粗略的方式描述。例如：

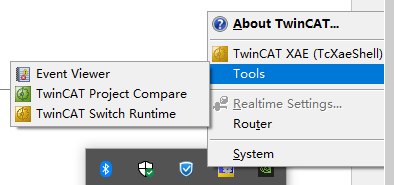
功能块的更改无法显示更改的具体内容



库版本的修改也可以在编译消息窗中提示：



## 如果控制器上有源代码，可直接使用XAE的Tools｜TwinCAT Project Compare



这种方法可以进行详细的比较，但前提是控制器上有源代码，并且与当前运行程序匹配。

# 如何加快Online Change的速度

在大型TwinCAT项目上，调试工程师经常反馈Online Change太慢，每次登陆都要等几十秒。通过实际测试一个编译文件为25M的项目，分析Wireshark抓取的数据帧时间、XAE界面录频的各状态切换时间以及PLC记录的输出实际切换的时间，分析得到以下可以加快Online Change速度的途径：

## 直接操作HMI界面，不必等待PC上的代码Online刷新

Online Change时，PLC上的逻辑更新远比POU显示刷新要快，因为每次Online Change时是先下载更新代码，PLC就可以按新逻辑运行，然后XAE再处理编译文件的更新和下传，不影响实时核的运行。所以XAE在处理编译文件的时候，PLC逻辑已经更新，此时操作人员可以正常操作HMI界面调试机器，不必等待POU的Online刷新，更不必等待Import Symbol过程结束显示“就绪”。

## 升级编程电脑的CPU性能，可以提升Online Change的速度

实际测试一个编译文件为25M的项目，发现同一台编程PC修改哪怕一行代码，无论Login至Local还是CX5140还是CX2030，需要的时间几乎相同，都是15秒左右。

因为编译过程、压缩文件这些耗时的步骤都是在编程电脑上完成的，所以对于大型项目，高性能的PC有助于加快这些过程。

## 改善编程PC和PLC之间网络传输也可以提升Online Change的速度

对于编译文件25M（.complileinfo）的项目，网线直联时也需要前后约5秒的时间用于数据传输。用Wireshark抓包发现从编程PC到PLC的有效数据量约为27 kByte和5.8 Mbyte（.tizip）。而大型项目的编译文件即使压缩后也可能高达几十M，花费的时间会更多。尤其是如果还经过了一个或者多个交换机，或者走Wifi而空间有遮挡，或者局域网中还有其它设备并繁忙通讯，那么传输的时间必然会成倍延长。

为了改善网络传输，大型程序调试时最好使用有线网络，在编程PC和PLC之间最好能够一条网线直连，如果无法做到真连，也可以单独用一个交换机（比如多人调试，可能随时有不同的人想连接这台PLC），不要跟其它无关的设备连到同一个网络。

# 附：TC2项目的在线修改功能与TC3的区别

在线修改的机制对于TC2和TC3是相同的，但是有的功能TC2中是没有的，有的功能在TC2中表现为不同的文件和变量。 所以调试TC2项目的时候，除了前面几个章节的内容外，有一些不同的地方需要注意：

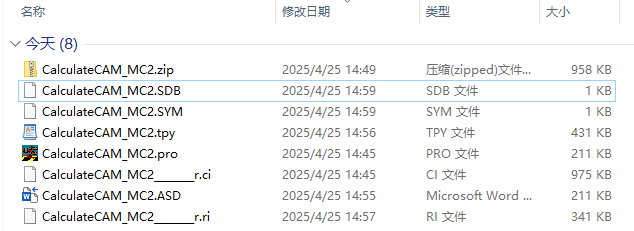
## 观察在线修改的次数

引用System.lib以后，在全局变量 SystemInfo.OnlineChangeCount中查看修改次数。

## 程序转移后如何才能直接Login

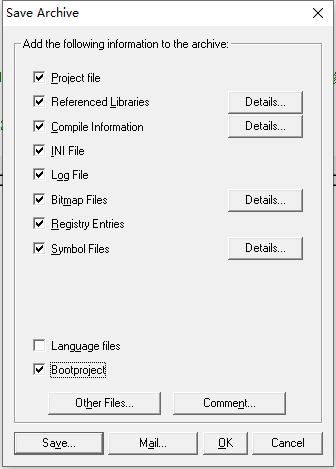
编程电脑要判断当前项目与目标控制器上运行的程序一致才能直接Login，否则就要编译，或者全编译，才能Login。

测试发现，在创建引导程序之前，目标设备上没有任何当前程序的信息，因而所有比较判断都在编程电脑上完成。TC2的PLC项目路径下内容包含：



|  |  |  |
| --- | --- | --- |
|  | 更新时间 |  |
| .pro  源代码 | 存盘时 | 只有.pro文件，无.ci，无.ri  需要编译，Download  成功Login时生成.ri文件  退出存盘时生成.ci文件 |
| .sdb  .sym  .tpy | 编译成功自动生成，不影响Login |  |
| \_r.ci | Logout后，再保存时 | 只有.pro +\_r.ci  不用编译，可以直接Login，  同时生成.ri文件。 |
| \_r.ri | 程序更新编译后，Download时 | 只有.pro +\_r.ri  需要编译，可以直接Login |

结论：如果要程序转移到另一台电脑进行调试，要能直接Login而不引起停机，就需要同时复制至少3个文件“.pro+\_r.ri+\_r.ci”。当然最好是用archive的方式，如6.3.3TC2不区分库文件的不同版本 所述，勾选上 Referenced Libraries和Compile Infomation、



Tip：不创建引导程序，TC重启后Pesistent变量也恢复了。

Tip：\_r.ci文件比.pro和\_r.ri文件之和还大，可能保存了二者信息供下次Login比较。

## TC2没有的功能

### TC2不能设置备用内存

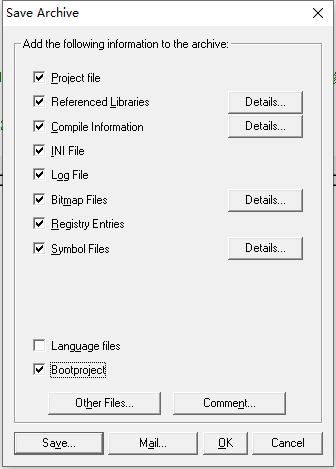
### TC2不能显示 Online Change的细节

### TC2不区分库文件的不同版本

TC2不区分库的版本，默认的.lib文件都存放在C:\TwinCAT\Plc\Lib下。

升级编程软件时，总是用新的Beckhoff库覆盖以前的库。TC2的PLC程序也没有Pin Version固定版本的功能。所以编程电脑升级后，打开以前的TwinCAT PLC程序，编译时就会用新版的引用库来替换旧的，表面上看用户自己写的代码没有改变，实际上它调用的FB可能已经变了。

所以备份一个PLC程序时，不仅要备份.pro和.tsm，还要把它当时引用的Lib也一并备份上。方法是通过菜单 File | Save/Mail archive…：



项目归档时，最好完整备份，以备将来复制到另一台电脑进行连机调试。推荐如上图所示选中所有可选的项。