

6SL3060-4AK00-0AA0连接电缆

产品名称	6SL3060-4AK00-0AA0连接电缆
公司名称	浔之漫智控技术-西门子PLC代理商
价格	666.00/件
规格参数	
公司地址	上海市松江区石湖荡镇塔汇路755弄29号1幢一层A区213室
联系电话	15221406036

产品详情

6SL3060-4AK00-0AA0连接电缆

pcs 7系统为os站提供标准的用户权限校验的功能，即当前登录的用户需要经过权限校验后，方能对其授权的区域和对象进行相应的授权操作，否则相应功能将对该用户屏蔽。

在创建一个标准的pcs 7项目时，系统将会为os项目创建默认的权限级别。如下图所示为pcs 7 v7.0默认建立的总共7个级别的权限等级。

图 1 pcs 7 v7.0默认权限等级

每一个权限等级所能完成的功能各不相同，并不叠加，详细内容请参考相应的pcs 7 os操作员手册中相关章节的内容，或者查询wincc inbbbbation system中working with wincc ? structure of the user administration ? overview of authorizations ? user hierarchies pcs 7中的介绍，此处不做详细介绍。

除此之外，pcs 7权限校验的另外一个特殊之处在于其“区域”的概念。如上图所示，在authorization列右侧，系统会根据当前os项目的图形树picture tree的结构（picture tree的结构是根据plant hierarchy的设置创建的），自动的创建区域列。在权限分配时，权限可以基于区域的方式来设置。也就是说，可以单独的来设置当前用户在不同区域下的权限。如上图所示，当前用户只拥有area1的5号权限，而没有area2的5号权限。

客户在使用user administrator进行项目的用户权限分配和管理过程中，常常会发现，按照系统关于权限的描述所定义的用户，在实际使用过程中并不具有其相应的操作功能，并不能操作其已经授权的某一些回路，常出现的问题有：

- 1) 从其它非授权区域拷贝到授权区域的block icon，用户不能通过面板进行正常的操作控制；
- 2) 在手动创建picture tree区域结构的情况下，用户不能操作其授权区域下所有的控制回路；

共同的故障特征就是：当前登录的用户明显拥有该区域的5级（process controlling）或6级（higher process controlling）权限，但在该区域的某些回路的操作面板中却无法执行相应的操作（手自动切换、参数的设定等）。本文将基于该类问题，对pcs 7的面板权限校验原理进行简单的描述，并就经常出现的这两类问题给出解释，在*后提出相应的意见和建议。

1. pcs 7面板权限校验

pcs 7标准的面板权限校验通常分为两个级别：第一级为普通的操作员级别，其所能进行的操作为回路的普通操作，例如，启动/停止命令、手/自动切换命令、设定值设定等；第二级为**的工程师级别，其所能进行的操作为回路的重要操作，例如，参数的设定、报警的抑制等（详情请参考各功能块的在线帮助信息中的描述）。面板操作权限的分配通过其对应的图标block icon来定义并传递。其默认分配的一级权限为图1所示的5级（process controlling）权限；默认分配的二级权限为图1所示的6级（higher process controlling）权限。

实际操作过程中当前登录的用户操作该面板时，只有在当前用户分配有图1所示的相应区域的5级权限后，方能进行该区域下回路面板的上述的普通操作员级别的操作，只有在当前用户分配有图1所示的相应区域6级权限后，方能进行该区域下回路面板的上述的**工程师级别的操作。否则当前登录的操作员即便能打开该区域的回路面板，面板中所显示的所有控件对象也均也只读的方式，不能通过其进行任何的操作。这就是pcs 7中各回路面板的权限校验方式。

当然，各回路操作面板的这两级的权限分配级别并不是固定不可修改的，可以通过其对应的图标block icon的两个属性（第一级操作对应的属性为processcontrolling_backup，第二级操作对应的属性为higherprocesscontrolling_backup）来指定，如下图2所示。

图2 回路对应的权限分配

此外，需要注意的是，如果在pcs 7用户管理器中不创建任何用户的前提下，则上述权限校验的过程将会自动关闭，即系统将不进行任何的面板操作权限的校验。用户可以通过面板对系统进行任何的操作。这种不基于用户登录和权限校验的方式一般只适合于对操作权限要求不高的小型过程控制系统场合。而且任何人员可以对系统进行任何操作，是一种非常不安全的方式，一般不建议用户使用。

2. 常见问题

2.1 不能通过面板操作从其它区域拷贝的回路面板

例如，有如下一个标准的pcs 7项目，根据工厂结构及控制需求总共划分有两个区域area1和area2，在各自区域下分别插入其对应的cfc程序和过程画面。

```
>; area1: cfc1 / motor1 , picture1
```

```
>; area2: cfc2 / motor2 , picture2
```

图3 工厂层级与区域对应关系

编译程序和os项目后，系统将会自动在用户权限管理器中创建area1和area2区域列，并在相应的区域画面中创建对应的block icon。根据用户控制要求，创建如下用户及分配权限：

1) 操作员“aaa1”具有操作“area1”下回路面板motor1的权限，分配有该区域的5级（process controlling）和6级（higher process controlling）权限，如下图4所示；

图4 区域area1和对应的用户权限设置

2) 操作员“aaa2”具有操作“area2”下回路面板motor2的权限，分配有该区域的5级（process controlling）和6级（higher process controlling）权限，如下图5所示。

图5 区域area2和对应的用户权限设置

运行os后，登录相应的用户，即可操作其授权区域的回路面板，未经授权的区域回路面板，即便能打开也不能进行任何操作。

在某些情况下，客户可能根据控制需求，要把“motor2”对应的block icon拷贝到监控区域“area1”的相应画面上。但是当操作员“aaa1”登录后，在picture1下对“motor2”回路进行操作控制的时候，打开面板发现其无法完成相应的操作。如下图6所示：

图6 操作员“aaa1”不能操作“motor2”

2.2 手动创建picture tree画面树而用户不能操作其授权区域下的所有控制回路

在进行os编译的编译过程中，不激活如下所示的picture tree选项，编译后，在picture tree manager中创建需要的画面树（区域结构）。

或者是通过os编译生成画面树结构后，打开picture tree manager进行一些编辑和修改，比如给区域重新命名。

图7 os编译对话框下的picture tree选项

图8 手动编辑picture tree

并按照标准的用户创建及权限分配的方式，创建用户并分配相应的区域权限，如下图所示：

图9 用户aaa1权限设置

运行os后，登录相应的用户，发现其无法对授权的区域下所有的回路面板进行相应的操作。

图10 用户aaa1不能操作motor2测量点

3. 问题分析及建议

pcs 7在进行回路面板的操作权限的校验时，是基于区域校验的方式，系统将会读取该回路所属的区域特性，该属性在cfc编程和工厂层级设置时即已决定。系统读取后会和用户权限管理器中当前用户所分配的区域进行比较，如果匹配，则可以进行相应操作，如果不匹配，则不能进行操作。

上述示例中motor2回路所对应的cfc2安插在区域area2中，则该回路motor2的区域特性为area2。

在上述两种故障情况下，当前用户aaa1分别分配了区域“area1”和“工艺区域1”的权限，其区域属性和cfc**能块motor2所属的区域属性（area2）并不一致，所以登录用户aaa1即便能打开该回路面板，也不能进行相应的面板操作。这就是pcs 7下面板基于区域权限校验的具体校验过程。

基于该面板权限校验的机制，为避免上述类似问题的发生，有如下几点建议：

- 1) 在编程之初，划分工艺操作区域时，充分考虑回路操作的要求，使之上下对应，减小在os上手动调整的回路图标的几率；
- 2) os编译时，激活picture tree选项，基于工厂层级架构的设置自动生成picture tree结构，不要对picture tree结构的手动修改，特别是*顶层画面的区域名的修改；
- 3) 尽量避免在不同的画面之间拷贝回路的图标，特别是在各个区域之间跨区域拷贝回路图标

使用opc scout监视s7-300/400的bbbbbb类型变量，但是在opc scout中总是没有字符显示。

如图1所示，在的db1中定义了两个字符串s1和s2。

图1

在ob1程序中为字符串s2赋值如图2。

图2

如图3所示，opc scout中定义了两个item，分别监视db1中的两个字符串s1和s2，结果第一个字符串可正常显示字符串内容，第二个字符串无字符显示。

图3

bbbbbb数据类型作为复杂数据类型无法在db块中直接监视，可通过变量表来监视但需要拆分为多个字节来监视和修改，如图4变量表中监视两个字符串的部分字节。

图4

对照图5所示string数据类型的结构来看，db1.dbb0和db1.dbb12分别为字符串s1和s2的*大长度，db1.dbb1和db1.dbb13为字符串s1和s2的实际长度，从变量表监视结果可看到，s1的实际长度为5个字符，而s2的实际长度为0，所以在opc scout中监视s2始终都没有字符显示。

图5

解决方法：

在程序中增加为s2实际长度字节赋值的指令。如将实际长度3通过move指令传递给db1.dbb13，在opc scout中能看到第二个字符串显示位“fal”。如果将s1的实际长度字节值修改为3，在opc scout中显示的字符串将是“abc”而不是现在的“abcde”