# 有关功能权限的说明

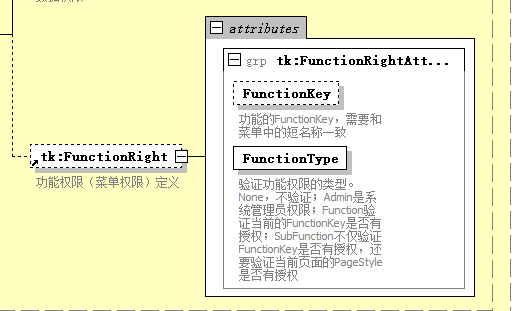
在Toolkit2,3,4版本中都定义了功能权限。即对应的菜单授权后，才能访问该功能。如果记住了功能的地址，没有获得授权，尝试通过敲地址的方式访问该功能，功能权限将拒绝访问。

由于Toolkit5对Source进行了重大修订，且大多数情况下，尝试通过敲地址的方式访问是个隐藏功能，不会出现，因此在移植中，忽略了功能权限的移植。

现在Toolkit5.5重新恢复了功能权限的功能。由于Toolkit5开始，对Module文件进行了革命性的调整，因此，有必要说明一下。

## 功能权限的Xml定义

在很多Source中，都可以找到功能权限的定义，如下：



一旦在Xml中定义了这个，功能权限的检查就会启动。

FunctionType有四个类型：

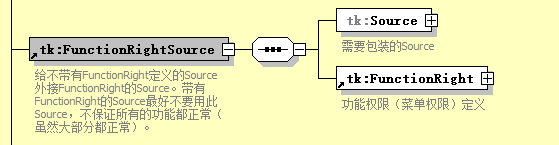
* None：不做功能权限检查，这个和没定义一样
* Admin：只要访问用户具有Admin权限就能访问功能，其他不管。
* Function：这个功能最常用，定义它时，需要同时配置FunctionKey，这个FunctionKey必须要和菜单的短名称一致。也就是说，只有这个菜单授权了，你才能访问。没授权，敲地址就会显示无授权页面（这个页面可以进行全局配置）
* SubFunction：这个功能除了具备Function的功能外，还能对Function中每个子操作检查是否授权。因此，使用这个配置，除了要配置FunctionKey，还需要配置每个子操作的Key。为了简化配置，从Toolkit4后，采用子操作的PageStyle作为Key。在没有操作权限时，这个功能防止进入未授权的功能，但是不能隐藏未授权的操作。现在有了操作权限，可以忽略这个选项。而且，它要求PageStyle必须子功能的Key一致。在当前的菜单子操作授权中，不能完全保证一致性（如果只有增删改，那没问题），所以使用此配置需要谨慎。

FunctionKey在FunctionType设置为Function或者SubFunction时配置，需要和菜单的短名称一致。如果不一致，即使获得授权，也一样不能访问。

## 关于功能权限的Source

对数据库操作的Source，如SingleDbSource，MasterDetailDbSource等等，都加了FunctionRight的配置标签。这些Source也是菜单上常用的Source，因此要开启功能权限应该非常方便。

但是，不是每种Source都添加了FunctionRight的配置，一些很简单的Source，通常无此配置，如果这些Source也需要添加功能权限的保护，那么，系统新开发了一个Source作为代理来完成此功能。



这个Source可以包装那些不带FunctionRight定义的Source。它只是一个代理，具体Source需要做的功能，依旧是原来的Source去做。只是在做之前，需要检查菜单功能是否授权。

## 关于SubFunction的功能增强

由于SubFunction的缺陷问题一直存在，原先采用PageStyle作为SubKey，只能解决常规问题。而强行在每个子功能指定Key，会使配置复杂，

好在tk5.5的功能定义已经全部可以通过Url获取，因此通过url可以获取一个子功能使用的Source和PageStyle，通过这两个参数可以基本上确定一个功能了。因此，对SubFunction进行相应的调整，SubKey以当前Url的PageStyle和Source（已经定义SubFunctionKey类，该类进行了相关处理，可以作为Dictionary的Key存储）作为Key传入API进行检查。

对应的，作为FunctionRight的实现，需要针对传入SubFunctionKey进行相关比对查找。此外，FunctionRight在存储用户信息时，最好也要通过分析Url来获取PageStyle和Source来进行相关比对。

Url的规则请查询相关文档。

Toolkit在SimpleRight也做了相应调整，如果有自定义的FunctionRight实现，可以参考SimpleRight中实现。

### 具体实例

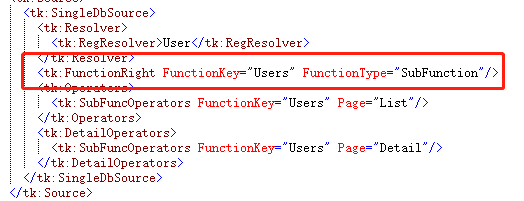
现在以用户管理功能为例，来说明SubFunction的配置。

在菜单管理中，可以看到用户管理的子功能，如下图：

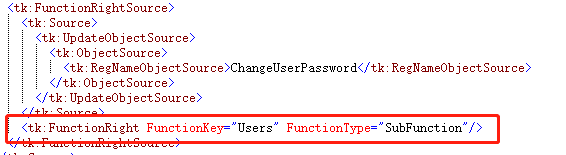


其中新建，修改，删除都是内置的功能。而重置密码和设置角色都有额外的Source来实现相应的功能。

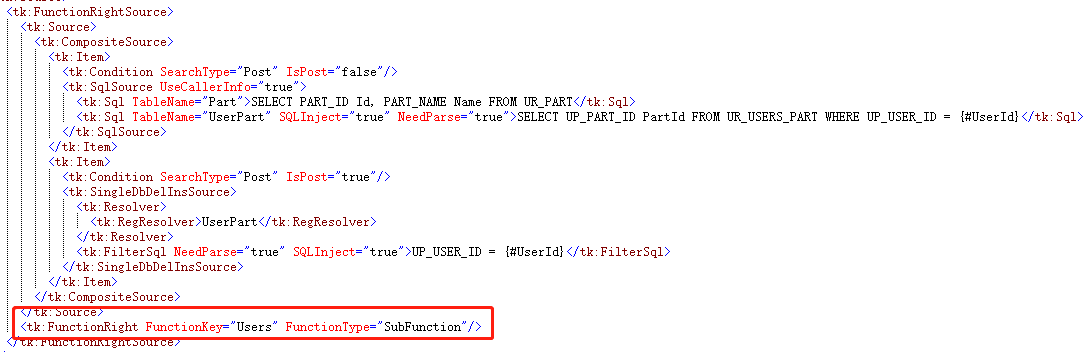
在User.xml中定义了新建，修改，删除的功能，这里使用的标准的SingleDbSource，因此可以直接配置FunctionRight的节点，如下图：



重置密码的配置在ChangeUserPasswd.xml中，这个Source没有FunctionRight节点，因此，我们使用前面定义的FunctionRightSource进行改写，如下图：



设置权限的配置在UserPart.xml中，这个Source同样没有FunctionRight节点，因此，我们同样使用前面定义的FunctionRightSource进行改写，如下图：



改造完成后，如果在角色配置中配置好相应的权限。使用相应的用户登录后，除了看不到相应功能外，如果尝试使用敲地址的方式进行访问，将看到如下界面：

