

第四课：宏代码库

宏代码库（您可以认为等同于工具类的作用）主要用来定义一些公共的宏方法。将一些使用频率较高的方法定义为公共的方法，便于代码重用，消除重复。

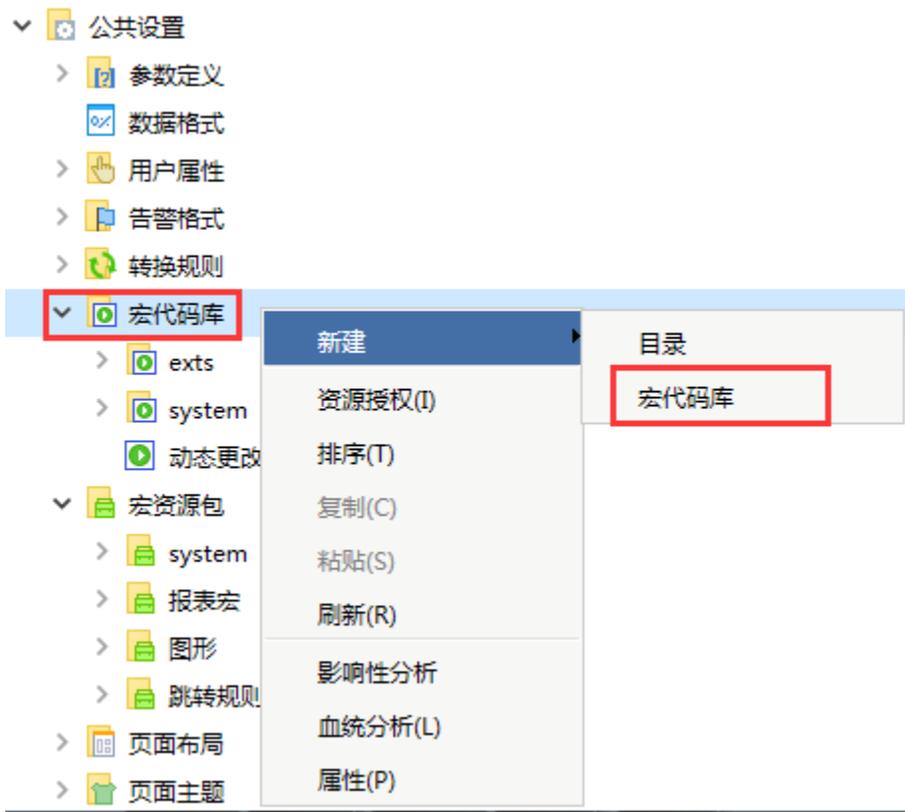
- 宏代码库中声明的公共方法不通过事件触发，因此需要通过报表宏或跳转规则引用宏代码库来调用这些公共方法。
- 系统默认内置的工具宏为：system > utils，分别针对服务端模块和客户端模块定义了一些常用的公共方法。
- 使用时用use(完整的路径)，譬如`var jmUtils = use("system.utils.JMClientUtils")`，这样JMClientUtils中的所有方法都可以jmUtils.方法名引用。

- 第一步：创建宏代码库
- 第二步：给宏代码库定义公共方法
- 第三步：调用工具宏定义的公共方法

系统中的宏代码库主要包括两项内容：

- **目录**：用来存储不同的宏代码文件。如内置的system就是一个目录。
- **宏代码库**：等同于一个代码文件，文件中定义了许多公共的方法。如内置的utils就是一个宏代码库。

创建宏代码库入口：在 **公共设置 > 宏代码库** 右键菜单中选择 **新建宏代码库**。



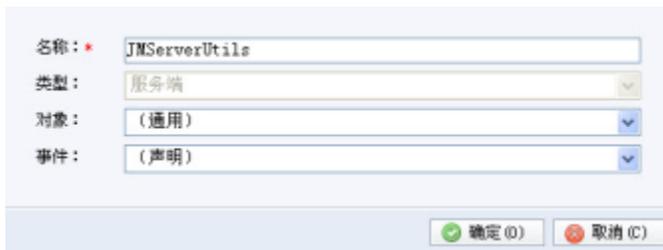
第一步：创建宏代码库

在“宏管理”界面的主界面区，直接点击“新建工具宏代码库”快捷方式，就会弹出“新建宏代码库”设置界面，输入“名称”为“utils”。



第二步：给宏代码库定义公共方法

1. 在“宏管理”界面的资源目录区中，对刚刚新建的宏代码库，在其右键菜单中选择“编辑宏”，进入宏编辑界面。
2. 对模块区的“服务端模块”节点，在其右键菜单中选择“新建模块”，弹出模块信息界面，并在“对象”中选择“（通用）”，“事件”中选择“（声明）”。

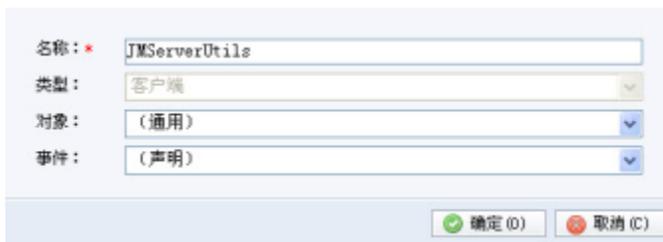


3. 在编辑区中，编写对应的宏代码。

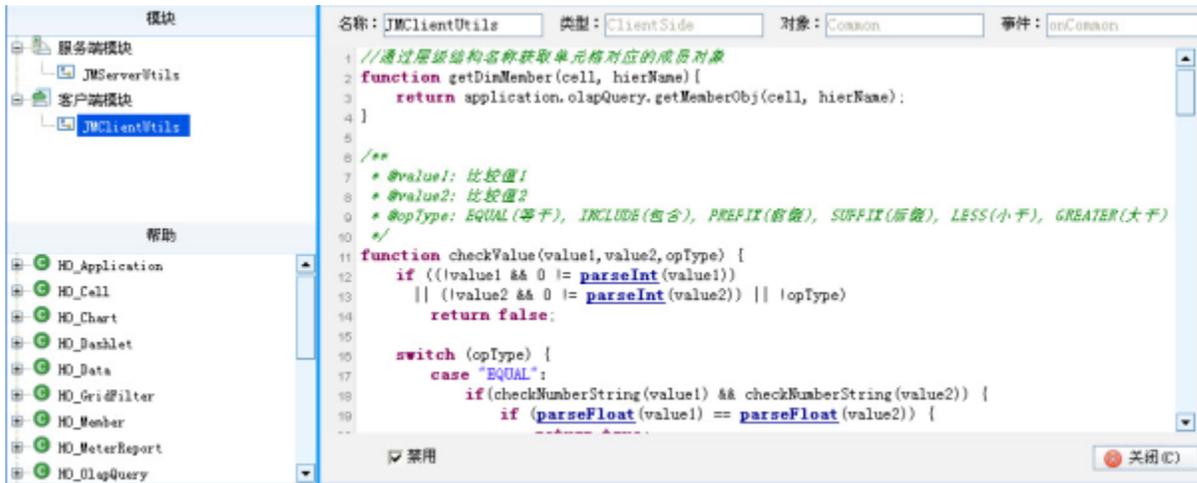


```
1 /**
2  * @cell: 单元格对象
3  * @levelName: 层次结构名称
4  * @levelType: NAME (成员名称), ALIAS (成员别名), VALUE (单元格的值)
5  * @value: 判断内容
6  * @opType: EQUAL (等于), INCLUDE (包含), PREFIX (前缀), SUFFIX (后缀), LESS (小于), GREATER (大于)
7  */
8 function checkValue(cell, levelName, levelType, value, opType) {
9     if (!cell || !levelType || !opType)
10        return false;
11
12     if (levelType.toUpperCase() == "VALUE") {
13         if (cell.data == null || cell.data.value == null)
14             return false;
15         var realValue = cell.data.value.toString();
16         var displayValue = cell.data.text;
17         if (cell.data.valueType == "DOUBLE" || cell.data.valueType == "LONG") {
18             value -= 0;
19             if (isNaN(value))
```

4. 对模块区的“客户端模块”节点，在其右键菜单中选择“新建模块”，弹出模块信息界面，并在“对象”中选择“（通用）”，“事件”中选择“（声明）”。



5. 在编辑区中，编写对应的宏代码。



第三步：调用工具宏定义的公共方法

工具宏定义的公共方法按照如下方式被调用：

例如：系统中有utils工具宏，该路径为“工具宏>system>utils”。在该工具宏中定义了JMClientUtils宏模块，模块中实现了headerMemberAliasEquals (cell, memberName, value)方法。

- 在宏代码中通过“var jmUtils = use(“system.utils.JMClientUtils”)”声明调用文件的路径。
- 在具体的宏代码中通过“变量名.方法”的方式，如“jmUtils.dataCellMemberAliasEquals(cell, “[Measures]”, “Profit”)”，来调用相应的方法。

应用方法如下所示：

```
var jmUtils = use("system.utils.JMClientUtils");

function main(olapTable) {
    if(!olapTable)
        return;

    //遍历数据区,满足条件的单元格添加链接
    setDataCellsClickable(olapTable.data);
}

function setDataCellsClickable(datas) {
    if(!datas)
        return;

    for(var i=0; i<datas.length; i++){
        var data = datas[i];
        var cell = {data: data, cellType: "Data"};
        setClickable(cell);
    }
}

function setClickable(cell) {
    if(!cell)
        return;

    //数据区链接显示处理
    if(jmUtils.dataCellMemberAliasEquals(cell, "[Measures]", "销售量")) {
        if(cell.data && cell.data.text != "") {
            cell.data.clickable = true;
            cell.data.groupName = cell.data.groupName + ", [销售明细表]";
            return;
        }
    }
}
```