

即席查询（旧）如何实现根据不同参数值显示不同的报表？

 该宏示例在 V10.5上 验证通过

提示：本文档的示例代码仅适用于本文档中的示例报表/场景。若实际报表/场景与示例代码无法完全适配（如使用功能不一致，或多个宏代码冲突等），需根据实际需求开发代码。

- [示例说明](#)
- [设置方法](#)
- [宏类型](#)
- [宏代码](#)
- [资源下载](#)

示例说明

在即席查询中切换一个参数，不同的参数值打开不同的报表。如下图：

切换【报表】参数，值为1的时候显示报表1，值为2的时候显示报表2， 值为3的时候显示报表3：



即席查询- 根据参数值不同显示不同的报表1

报表* 1

共 77 行 每页 10 行 1 / 8 > >>

产品类别编号	产品名称	产品类别
1	苹果汁	饮料
1	牛奶	饮料
2	番茄酱	调味品
2	盐	调味品
2	麻油	调味品
2	酱油	调味品
7	海鲜粉	特制品
2	胡椒粉	调味品
6	鸡	肉/家禽
8	蟹	海鲜

设置方法

1. 在公共设置中，创建【报表】参数。

参数名称：***** 报表

参数别名：

描述：

数据类型：整型

控件类型：下拉框

标题宽度：

参数宽度：***** 175

参数下拉宽度：***** 175

参数下拉高度：***** 200

☒ 显示参数 ☐ 允许多选 ☐ 手工输入 ☐ 校验合法性 ☐ 不显示标题 ☒ 输出合并

☒ MDX使用引号 ☐ 允许为空（存储过程查询） 参数缓存：系统设置

备选值设置：

☐ SQL ☐ 对象 ☒ 静态列表 ☐ 函数

真实值	显示值	操作
1	1	↶ ↷ ↵ ↴ 面
2	2	↶ ↷ ↵ ↴ 面

默认值设置：

☐ SQL ☐ 对象 ☒ 静态列表 ☐ 函数

真实值	显示值
1	1

2. 创建一个“报表”过滤器，如下图：

名称：***** 报表

别名：

描述：

☐ 可选条件

报表 = 报表

表达式：*****

左侧树状图：

- northwind
 - DEFAULT
 - 计算字段
 - 过滤器
 - office分析报告
 - 跳转规则
 - 组合分析
 - 报表**
 - 产品
 - 产品类别过滤
 - 带参数

3. 在“分析展现”节点下分别创建即席查询报表1、报表2、报表3。

4. 切换到 公共设置 > 宏资源包 节点下，新建一个宏资源包，类型选择即席查询，适用范围选择报表1、报表2和报表3。

新建宏资源包

名称：

切换即席查询报表

别名：

类型：

即席查询

定义：

适用范围定义：(定义该规则适用于哪些报表?)

☐ 数据源<该数据源下的所有报表适用>

☒ 报表<只对具体的报表适用>

适用范围描述：

报表包含[报表1, 报表2, 报表3]

确定(O)

取消(C)

5. 双击此宏资源包进入报表宏界面。

6. 在报表宏界面新建客户端模块，在弹出的新建模块对话框中选择对象为simpleReport；事件为onParamValueChanged；并把下面宏代码复制到代码区域。

宏类型

类型	对象	事件
ClientSide	simpleReport	onParamValueChanged

宏代码

```
// 10.5.8

//
// : ClientSide      :simpleReport      : onRender
//
//
// 1. , , 0,
// 2. reportIdsid, 0

function main(simpleReport, simpleReportContext) {
    var reportIds = ['I8a8a9fd30183e906e90649ae0183ea57d1160a3f',
'I8a8a9fd30183e906e90649ae0183ea594eeb0ab5'];
    var switchReportParam = ''
    redefinedParamChangeFn(simpleReport, simpleReportContext, reportIds, switchReportParam);
}
//
function redefinedParamChangeFn (combinedQV, combinedQVNavigator, reportIds, switchReportParam) {
```

```

        if (combinedQV.paramPanelObj) {
            combinedQV.paramPanelObj.onSelectChange.unsubscribe(combinedQV.doParamChange, combinedQV);
            combinedQV.paramPanelObj.onSelectChange.subscribe(function(paramPanel, param, oldId, newId,
oldValue, newValue, notRefresh){
                if (isTargetReport(reportIds, this.combinedQuery.datas.queryId) && param.name ===
switchReportParam) {
                    if (reportIds[param.value]) {
                        var args = Array.prototype.concat.apply([reportIds],arguments);
                        switchReport.apply(this, args);
                    } else {
                        alert(" reportIds[" + param.value + "] id");
                    }
                    return;
                }
                this.doParamChange.apply(this, arguments);
            }, combinedQV);
        }
    }
    //
    function isTargetReport (reportIds, id) {
        return reportIds.findIndex(e => id === e) > -1;
    }
    //
    function switchReport(reportIds, paramPanel, param, oldId, newId, oldValue, newValue, notRefresh) {
        var targetReportId = reportIds[param.value];
        var currentReportId = this.combinedQuery.datas.queryId;
        if (targetReportId !== currentReportId) {
            //
            util.remoteInvokeEx('CombinedQueryService', 'closeCombinedQuery', [ this.combinedQuery.datas.
clientId ]);
            //
            var combinedQVCommand = this.combinedQuery.queryCmd;
            var node = combinedQVCommand.node;
            var commandFactory = node && node.tree && node.tree.commandFactory;
            if (commandFactory) {
                var command = commandFactory.getCommand('CombinedQueryCommand');
            } else {
                var CombinedQueryCommand = jsloader.resolve('smartbi.combinedquery.CombinedQueryCommand');
                command = new CombinedQueryCommand();
            }
            command.execute('OPENWITHNOREFRESH', {
                _id: targetReportId,
                _type: 'COMBINED_QUERY',
                tree: {
                    commandFactory: commandFactory
                },
                userParamsInfo: [{name: param.name, value: newId, displayValue: newValue}]
            });
            var timeout = null;
            var refreshFn = function() {
                if (command.combinedQuery.getQueryView()) {
                    command.combinedQuery.getQueryView().doRefresh();
                    clearTimeout(timeout);
                } else {
                    timeout = setTimeout(refreshFn, 10);
                }
            }
            timeout = setTimeout(refreshFn, 10);
        }
    }
}

```

资源下载

资源: [migrate.xml](#)