

# GalaxyPoint 表单开发手册

## 版本

|   |       |      |
|---|-------|------|
| 文件状态：<br><input checked="" type="checkbox"/> 草稿<br><input type="checkbox"/> 正式发布<br><input type="checkbox"/> 正在修改 | 文件标识： | Form |
|   | 当前版本： | V0.1 |
|   | 完成日期： |      |

## 历史版本

| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
|-------|----|-----|------|----|
|       |    |     |      |    |
|       |    |     |      |    |
|       |    |     |      |    |

# 目录

## Contents

|                          |    |
|--------------------------|----|
| GalaxyPoint 表单开发手册.....  | 1  |
| 目录.....                  | 2  |
| 前言.....                  | 6  |
| 1. 表单文件存放.....           | 6  |
| 2. 数据源和数据目标.....         | 6  |
| 2.1 数据源.....             | 6  |
| 2.1.1 说明.....            | 6  |
| 2.1.2 内置元素.....          | 6  |
| 2.1.3 默认值 (defval) ..... | 8  |
| 2.2 数据目标 .....           | 8  |
| 3. 文本框控件.....            | 8  |
| 3.1 样式 .....             | 8  |
| 3.2 普通文本框 .....          | 9  |
| 3.2.1 通用属性.....          | 9  |
| 3.3 数字控件 .....           | 9  |
| 3.3.1 特有属性.....          | 9  |
| 3.3.2 说明.....            | 9  |
| 3.3.3 举例.....            | 10 |
| 3.4 流水号 .....            | 10 |
| 3.4.1 特有属性.....          | 10 |
| 3.4.2 说明.....            | 10 |
| 3.5 日期控件 .....           | 10 |
| 3.5.1 特有属性.....          | 10 |
| 3.5.2 样式.....            | 10 |
| 3.5.3 说明.....            | 10 |

|       |                          |    |
|-------|--------------------------|----|
| 4.    | 多行文本域.....               | 11 |
| 4.1   | 属性.....                  | 11 |
| 4.2   | 样式.....                  | 11 |
| 4.3   | 自适应高度.....               | 11 |
| 5.    | 组合框.....                 | 11 |
| 5.1   | 属性.....                  | 11 |
| 5.2   | 样式.....                  | 12 |
| 5.3   | 说明.....                  | 12 |
| 5.3.1 | selectbindtable.....     | 12 |
| 5.3.2 | selectvaluecolumn.....   | 12 |
| 5.3.3 | selectdisplaycolumn..... | 12 |
| 5.3.4 | datamap.....             | 12 |
| 5.3.5 | selectfilter.....        | 13 |
| 5.3.6 | defval.....              | 13 |
| 6.    | 单选按钮.....                | 13 |
| 6.1   | 属性.....                  | 13 |
| 6.2   | 样式.....                  | 13 |
| 6.3   | 说明.....                  | 14 |
| 7.    | 多选按钮.....                | 14 |
| 7.1   | 属性.....                  | 14 |
| 7.2   | 样式.....                  | 14 |
| 7.3   | 说明.....                  | 14 |
| 8.    | 子表.....                  | 15 |
| 8.1   | 属性.....                  | 15 |
| 8.2   | 样式.....                  | 15 |

|        |                           |    |
|--------|---------------------------|----|
| 8.2.1  | 表格样式.....                 | 15 |
| 8.2.2  | 表头样式.....                 | 15 |
| 8.3    | 功能.....                   | 15 |
| 8.3.1  | 新增行.....                  | 15 |
| 8.3.2  | 删除行.....                  | 16 |
| 8.3.3  | 上移行.....                  | 16 |
| 8.3.4  | 下移行.....                  | 16 |
| 8.3.5  | 移到顶部.....                 | 16 |
| 8.3.6  | 移到底部.....                 | 16 |
| 9.     | 自定义逻辑.....                | 16 |
| 10.    | 提交时验证.....                | 19 |
| 11.    | 通用弹出层.....                | 20 |
| 11.1   | 属性.....                   | 20 |
| 11.2   | 取值.....                   | 20 |
| 11.3   | 调用.....                   | 21 |
| 11.4   | 子页面配置.....                | 22 |
| 11.4.1 | 说明.....                   | 22 |
| 11.4.2 | modalyType 等于 div.....    | 22 |
| 11.4.3 | modalyType 等于 iframe..... | 23 |
| 12.    | 通用附件上传.....               | 23 |
| 12.1   | 属性.....                   | 23 |
| 12.2   | 上传成功后返回对象.....            | 24 |
| 12.3   | 示例：.....                  | 24 |
| 13.    | 通用选择页面.....               | 24 |
| 13.1   | 选用户.....                  | 25 |
| 13.1.1 | 页面路径.....                 | 25 |
| 13.1.2 | 参数配置.....                 | 25 |
| 13.1.3 | 调用示例.....                 | 25 |
| 13.2   | 选部门.....                  | 25 |

|        |                          |    |
|--------|--------------------------|----|
| 14.    | 通用 Ajax.....             | 25 |
| 14.1   | 参数.....                  | 25 |
| 14.2   | 返回结果.....                | 26 |
| 14.3   | 示例.....                  | 26 |
| 15.    | 操作 Json.....             | 26 |
| 15.1   | C#.....                  | 26 |
| 15.2   | js.....                  | 27 |
| 15.2.1 | 将 js 对象转换为 Json 字符串..... | 27 |
| 15.2.2 | 将 Json 字符串转换为 js 对象..... | 27 |
| 16.    | 常用取值.....                | 27 |
| 17.    | 表单通用按钮样式.....            | 28 |

# 前言

表单功能模块存放于 BMPWebNew/Form 目录下。该文档提及到的文件或者子目录将省略根路径名称。

表单控件使用相关代码可以在” 表单控件使用实例” 流程中查看, 对应表单 demo.htm。

**建议:**在表单大数据量时, 特别是在审批页面, 大多时候只是用来查看数据时, 建议替换成文本控件, 提高表单加载速度;

例如: 不修改时, 下拉框替换成文本框;

不涉及计算时, 去掉文本框 total 属性;

不需要修改日期时, 去掉文本框 texttype 属性。

## 1. 表单文件存放及支持浏览器

1. 表单文件存放于对应的流程所属公司的目录中。Form/ formtemplate 目录包括所有的公司目录, 公司目录的名称由数据库表 SYS\_Company 的 CompanyCode 字段命名。

2. 表单支持 IE7、IE8、IE9、IE10、Chrome、Firefox、Safari 等主流浏览器。

特别注意在Firefox下请将meta的charset属性设置为utf-8或者meta标签整个删除。

(老系统的表单自动生成的为unicode, 在火狐下会乱码)

3. 表单文件可以没有<html><head><title><body>等标签, 只留常规html元系及js即可。

4. 表单引用JS、样式等文件, 相对于Form/formtemplate/liveflowdefault.htm页面(相对于根目录../.. /)

```
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
```

## 2. 数据源和数据目标

### 2.1 数据源

#### 2.1.1 说明

数据源(formDataSrc) 有两部分构成, 一是系统内置的元素, 如登陆用户名等, 二是对应表单的数据库表的字段名。

#### 2.1.2 内置元素

| 元素                  | 说明        |
|---------------------|-----------|
| sysLoginUser.UserId | 登录用户的用户编号 |

|                            |                                |
|----------------------------|--------------------------------|
| sysLoginUser.UserAccount   | 登录用户的账号                        |
| sysLoginUser.HRID          | 登录用户的员工号                       |
| sysLoginUser.UserDspName   | 登录用户的显示名                       |
| sysLoginUser.JobTitleCode  | 登录用户的职位                        |
| sysLoginUser.UserTitle     | 登录用户的职称                        |
| sysLoginUser.UserLevel     | 登录用户的级别                        |
| sysLoginUser.EMail         | 登录用户的邮箱                        |
| sysLoginUser.Department    | 登录用户的业务部门                      |
| sysLoginUser.JoinDate      | 登录用户的入职日期                      |
| sysLoginUser.LineManagerId | 登录用户的直接领导编号(Id)                |
| sysLoginUser.UserNameEn    | 登录用户的英文名                       |
| sysLoginUser.CompanyId     | 登录用户的公司编号                      |
| sysLoginUser.GroupName     | 登录用户所在组织的组织名(以切换的 Group 为主)    |
| sysLoginUser.GroupId       | 登录用户所在组织的组织编号(以切换的 Group 为主)   |
| sysLoginUser.UserName      | 姓名                             |
| sysLoginUser.HireType      | 雇佣类型(正式用户,临时用户)                |
| sysLoginUser.Birthday      | 登录用户的生日                        |
| sysLoginUser.Gender        | 登录用户的性别(0:男,1:女)               |
| SysTaskID                  | 任务号                            |
| SysToday                   | 系统当前日期(yyyy-MM-dd)             |
| SysCurrentDateTime         | 系统当前日期和时间(yyyy-MM-dd hh:mm:ss) |
| SysCurrentTime             | 系统当前时间(hh:mm)                  |
| sysLoginUser.Tel           | 登录用户的电话                        |
| sysLoginUser.Fax           | 登录用户的传真                        |
| sysLoginUser.Mobile        | 登录用户的手机号码                      |
| sysLoginUser.CostCenter    | 登录用户成本中心                       |
| sysLoginUser.Location      | 登录用户区域代码                       |

### 2.1.3 默认值 (defval)

默认值可以绑定内置元素和固定字符串，当使用固定字符串时，文本框会默认显示固定字符串，下拉框将会默认选中值为固定字符串的项。值为固定字符串的单选框和复选框将会被选中。处理表单中的控件如果设置默认值，数据源将使用默认值，查看表单时默认值不起作用。

### 2.1.4 示例

人员信息: defval=" sysLoginUser. xxxxxxxx"

固定字符串: defval=" 有研硅股"

## 2.2 数据目标

数据源(formDataTag)，表单提交后，控件的值存放于对应数据库表的某个字段。

## 2.3 span 和 label

span 和 label 控件，可以绑定 defval 和 formdatasrc 属性，formdatatag 设置无效。

### (3).Link

|             |   |
|-------------|---|
| formdatasrc | 数据源(显示值)                                      |
| taskid      | taskid="2725"<br>taskid="tableName.fieldName" |
| 下边框         | class="underbar"                              |

```
<a target="_blank" taskid="2725">2725</a>
```

在链接中配置taskid属性，表单加载后，自动解析taskid并加密。点击后打开任务表单。

绑定formdatasrc属性, formdatasrc属性表示显示值。

## 3. 文本框控件

### 3.1 样式

|     |                  |
|-----|------------------|
| 有边框 | class="box"      |
| 无边框 | class="noborder" |
| 下边框 | class="underbar" |



## 3.2 普通文本框

### 3.2.1 通用属性

formdataTag 数据目标

formdataSrc 数据源

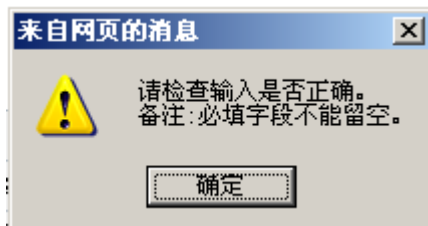
defval 默认值

check 需要验证的类型。具体请参见”提交时验证章节”。

checktitle 提交时验证不通过时的提示标题，具体请参见”提交时验证章节”。

示例：其中”备注：必填字段不能留空”中的”备注”就是 checktitle 绑定的值。

```
<input formdataSrc="T_OA_productionStatus.checktest"  
check="notnull" checkTitle="备注" formdataTag="T_OA_productionStatus.checktest"  
value="b" />
```



## 3.3 数字控件

### 3.3.1 特有属性

| 属性名      | 说明                                  |
|----------|-------------------------------------|
| textType | textType="number" 表示该控件为数字控件        |
| neg      | 是否可以输入负数,true/false, 不设置该属性,默认false |
| sep      | 是否千分位,true/false, 不设置该属性,默认false    |
| dec      | 小数点保留几位, 不设置该属性, 默认 2 位             |
| max      | 最大值, 不设置该属性不限制大小                    |
| min      | 最小值, 该属性设置, neg属性无效, 不设置该属性不限制大小    |

其他属性请参照”文本框控件—通用属性”。

### 3.3.2 说明

数字控件只能输入数字类型的字符，可以通过设置以上属性来适应业务需求；

其中千分位设置以后，输入的时候会自动千分位显示，如果审批页面有计算时，可以不设置该属性，只在查看表单中设置。

### 3.3.3 举例

```
<input style="width: 97%" name="G_CurrentMonth" class="noborder" checktitle="本月数"  
class="box" texttype="number" sep="true" dec="4" neg="false" max="5000" min="0"/>
```

## 3.4 流水号

### 3.4.1 特有属性

| 属性名      | 说明                        |
|----------|---------------------------|
| textType | textType="SN" 表示该控件为流水号控件 |

其他属性请参照“文本框控件—通用属性”。

### 3.4.2 说明

流水号控件将会自动显示当前任务的流水号，规则请在流程设计页面配置。

Post 页面请勿绑定 formdatasrc 和 formdatatag 属性

## 3.5 日期控件

### 3.5.1 特有属性

| 属性名       | 说明  |
|-----------|---|
| textType  | textType="date" 表示该控件为日期控件  |
| startDate | 开始日期(选择日期范围最小值)   |
| endDate   | 结束日期(选择日期范围最大值)   |
| datetype  | format属性未配置时有效<br>datetype="date" (默认)//只显示日期<br>datetype="datetime"//显示日期和时间   |
| format    | 日期格式<br>format ="yyyy-mm-dd"(默认)<br>format ="yyyy-mm-dd hh:ii:ss"<br>format ="mm-dd-yyyy "<br>format ="mm-dd-yyyy hh:ii:ss" |

### 3.5.2 样式

系统自动设置样式。

### 3.5.3 说明

查看页面中，请勿将 textType="date" 属性移除，只需加入 readonly=true 属性。

format 与 datatype 的区别:

1. datatype="date" 只显示日期 ;datatype="datetime" 显示日期及时间,
2. 当 format 属性未配置的情况下, datatype 属性有效, 日期格式将从站点 web.config 获取。

## 4. 多行文本域

### 4.1 属性

| 属性名            | 说明                              |
|----------------|---------------------------------|
| formDataSource | 数据源                             |
| formDataTag    | 数据目标                            |
| defval         | 默认值                             |
| checkTitle     | 需要验证的类型。具体请参见“提交时验证章节”。         |
| check          | 提交时验证不通过时的提示标题, 具体请参见“提交时验证章节”。 |

### 4.2 样式

|     |                  |
|-----|------------------|
| 有边框 | 系统自动设置           |
| 无边框 | class="noborder" |

### 4.3 自适应高度

表单打开时, 自动适应高度。

## 5. 组合框

### 5.1 属性

| 属性名            | 说明                              |
|----------------|---------------------------------|
| formDataSource | 数据源                             |
| formDataTag    | 数据目标                            |
| defval         | 默认值                             |
| checkTitle     | 需要验证的类型。具体请参见“提交时验证章节”。         |
| check          | 提交时验证不通过时的提示标题, 具体请参见“提交时验证章节”。 |
| selectprompt   | 提示, 例: selectprompt="请选择"       |

|                     |  |
|---------------------|--|
| selectbindtable     | 绑定到的表(数据来自绑定的表)  |
| selectvaluecolumn   | 绑定的值字段   |
| selectdisplaycolumn | 显示字段   |
| datamap             | 值映射, 下拉框 <b>change</b> 后, 将选中表的当前行的一个(多个)字段映射给其他一个(多个)控件 |
| selectfilter        | 数据过滤(sql 过滤)   |
| querySql            | 查询条件(selectbindtable 和 querySql 选择其一)                    |

## 5.2 样式

### 系统自动设置

## 5.3 说明

### 5.3.1 绑定表、视图或者 Sql 语句

selectbindtable:设置该属性, 下拉框拉绑定数据库的指定表, 下拉框的 option 项将来源来绑定的表。

querySql:设置该属性, 下拉框拉绑定自定义 sql 语句

selectbindtable 和 querySql 选择其中之一。

### 5.3.2 selectFilter 过滤 Sql 条件

例: `selectfilter="g_pid<>'33' and G_companyName is not null"`

`g_pid` 和 `G_companyName` 为表字段。

### 5.3.3 selectvaluecolumn

option 项的 value 指向绑定表中的指定字段。

### 5.3.4 selectdisplaycolumn

option 项的 text 指向绑定表中的指定字段。

### 5.3.5 datamap

下拉框 change 时触发, 将绑定表当前选中项, 所对应绑定表的某行的一个(多个)字段的值映射到其他控件。多个映射用";"(分号)分隔。

格式: `datamap= "映射字段—映射控件所绑定的 formdatasrc"`

```
dataMap="G_productCategory--T_OA_productionStatus.G_category;  
G_prickle--T_OA_productionStatus.G_prickle;  
G_productCategoryName--T_OA_productionStatus.G_categoryName"
```

### 5.3.6 cascade 下拉框级联

✧ 级联过滤:可通过上级下拉框选中项的值进行过滤。

下拉框级联。父级统一在 cascade 属性中设置联系条件, cascade="数据库字段名--匹配值", 多条件用分号(;)分隔。

如: cascade="G\_comName--T\_OA\_product.submitDept;G\_code--T\_OA\_product.G\_pid"  
表示找到下级下拉框 formdataTag="T\_OA\_product.submitDept" 的控件根据 G\_comName 字段过滤。

#### 注意事项:

1. 主表下拉框的上级控件必须是主表控件。
2. 子表下拉框的上级控件可以是主表也可以是子表, 当上级控件为主表控件时, 上级控件 change 后会将子表中多行中满足条件的下拉框同时过滤; 当上级控件为子表控件时, 只允许子表中同一行的控件(即: 同行控件只能过做为同行控件的筛选条件。

### 5.3.7 defval

表单加载后会自动选中 defval 设置的值的项。具体请参考”数据源和数据目标-数据源”章节。

## 6. 单选按钮

### 6.1 属性

| 属性名         | 说明                              |
|-------------|---------------------------------|
| formDataSrc | 数据源                             |
| formDataTag | 数据目标                            |
| checkTitle  | 需要验证的类型。具体请参见”提交时验证章节”。         |
| check       | 提交时验证不通过时的提示标题, 具体请参见”提交时验证章节”。 |

### 6.2 样式

1. 系统自动设置。

2. 通过自定义代码创建出的单选按钮统一样式方法示例：

```
$("#createRdo").checkbox({ cls: "hradio" });
```

请勿查找所有 radio，只须查找自动创建的 radio 进行调用。

## 6.3 说明

主表中，绑定相同 formdatasrc 的单选按钮将会视为一组。

子表中，同行，绑定相同 formdatasrc 的单选按钮将会视为一组。

表单提交后，将会把一组中选中的单选按钮的值存入数据库。

# 7. 多选按钮

## 7.1 属性

| 属性名            | 说明                             |
|----------------|--------------------------------|
| formDataSource | 数据源                            |
| formDataTag    | 数据目标                           |
| checkTitle     | 需要验证的类型。具体请参见”提交时验证章节”。        |
| check          | 提交时验证不通过时的提示标题，具体请参见”提交时验证章节”。 |

## 7.2 样式

系统自动设置。

通过自定义代码创建出的单选按钮统一样式方法示例：

```
$("#createChk").checkbox({ cls: "hcheckbox" });
```

请勿查找所有 radio，只须查找自动创建的 radio 进行调用。

## 7.3 说明

主表中，绑定相同 formdatasrc 的多选按钮将会视为一组。

子表中，同行，绑定相同 formdatasrc 的多选按钮将会视为一组。

表单提交后，将会把一组中选中的多选按钮的值存入数据库，逗号(,)分隔。

## 8. 子表

### 8.1 属性

| 属性名             | 说明                  |
|-----------------|---------------------|
| gridHeadRows    | 表头行数                |
| formGrid        | 表示该表格为子表 true/false |
| footerRowCount  | 表尾行数                |
| gridInitRows    | 发起任务时，页面加载后初始行数     |
| formBindTable   | 绑定的表或视图             |
| gridDynamicRows | Add 时，一次性增加几行       |

以上属性必须设置

### 8.2 样式

#### 8.2.1 表格样式

系统自动设置

#### 8.2.2 表头样式

1. 当表头为<th>时，系统自动设置，文字居中。
2. 当表头为<tr>时，class=" staticcell\_center"，该样式文字居中，有背景；  
class="staticcell"，该样式文字居左 16px，有背景。

### 8.3 功能

```
<td style="text-align: right" colspan="12">  
  <input class="button" value="添加" type="button" buttontype="addnew" copyrow="1" />  
  <input class="button" value="删除" type="button" buttontype="delete">  
  <input class="button" value="上移" type="button" buttontype="moveup">  
  <input class="button" value="下移" type="button" buttontype="movedn">  
  <input class="button" value="移到顶部" type="button" buttontype="movetotop">  
  <input class="button" value="移到底部" type="button" buttontype="movetoend">  
</td>
```

#### 8.3.1 新增行

在表格的表尾处加添按钮，并且设置 buttonType="addnew"，按钮样式系统自动设置。

复制上一行数据：将新增按钮添加属性 copyRow=" 1"

### 8.3.2 删除行

在表格的表尾处加添按钮，并且设置 `buttonType="delete"`，选中行将会被删除。  
按钮样式系统自动设置。

### 8.3.3 上移行

在表格的表尾处加添按钮，并且设置 `buttonType="moveup"`，选中行将会被上移，  
按钮样式系统自动设置。

### 8.3.4 下移行

在表格的表尾处加添按钮，并且设置 `buttonType="movedn"`，选中行将会被下移，  
按钮样式系统自动设置。

### 8.3.5 移到顶部

在表格的表尾处加添按钮，并且设置 `buttonType="movetotop"`，选中行将会被移动到顶部，按钮样式系统自动设置。

### 8.3.6 移到底部

在表格的表尾处加添按钮，并且设置 `buttonType="movetoend"`，选中行将会被移到底部，按钮样式系统自动设置。

## 8.4 计算

计算不分主表和子表，这里将计算当做子表功能小节介绍。

### 8.4.1 规则属性

| 属性名 | 说明  |
|-----|---|
| @m  | 全称Main，表示控件的id属性(主表控件)                      |
| @d  | 全称Detail，表示控件的name属性(大多情况代表子表中的name控件，主表也可) |
| @v  | 表示 js 变量                                    |
| @f  | 代表方法名                                       |

公式为计算表达式，只是特殊数字用属性名表示，将计算表达式绑定在结果控件的 `total` 属性中，支持百分比。

### 8.4.2 举例

例 1：下例表示 `id="txtResult"` 的控件的值等于



Id= txtA 的控件的值 加 Id= txtB 的控件的值 加 js 变量 a 的值 的总和，保留

3 位小数。

```
var a = "5";  
<input id="txtA" value="10"/>  
<input id="txtB" value="100"/>  
<input total="@m_txtA+@m_txtB+@v_a" id="txtResult" dec="3"/>
```

例 2：子表行内统计(自动执行 a 方法，目前不支持传参)

```
Function a(){  
    return 2+2;  
}  
<input name="txtA" value="10"/>  
<input name="txtB" value="100"/>  
<input total="(@d_txtA+@d_txtB+@f_a)/10%" name="txtResult" dec="2"/>
```

例 3：子表列求和

```
<input name="txtA" value="10"/>  
<input total="@d_txtA" id="txtResult" dec="2"/>
```

例 4：主表子表控件混合计算

```
<input id="txtA" value="20"/> //主表(该值改变，name=txtResult 的所有元素都会改变值)
```

```
<input name="txtB" value="10"/> //子表
```

```
<input total=" @m_txtA*@d_txtB" name="txtResult" dec="2"/> //子表
```

### 8.4.3 说明及注意事项

#### 注意：

- 1.切勿双向绑定，造成死循环。
2. 控件 a 等于 b+c,控件 d 等 a+1; 那么控件 d 的 total 可以直接写成 b+c+1, 效率会高些。

#### 说明：

1. `<input total="@a+1" id="txtResult">`, 下拉框 change 将控件 a 的值改变, 那么结果 控件 txtResult 的值也会改变。

这里的 change 是指下拉框的 datamap 映射, 如果自定义 js 代码改变了文本框 a 的值, 可以手动执行计算方法, `$("#a").focusout();`

例:

```

<input id="txtA" value="5"/>
<input id="txtResult" total="@m_txtA*10%" />
<select onchange="change()"></select>
function change(){
    $("#txtA").val("10");//值改变

    $("#txtA").focusout();//重新计算, txtResult 结果为 10
}

```

2. 子表 Add 和 Delete 动作时, 将会自动重新计算列合计; 行内合计不执行;
3. 复制上一行的数据, 重新计算列合计, 行内合计不执行

## 8.5 Excel 数据导入子表

1. 定义好模板供用户下载, 必须是 97-2003 版本。

|   | A  | B  | C  | D  |
|---|----|----|----|----|
| 1 | 列1 | 列2 | 列3 | 列4 |
| 2 | 10 | 2  | 大  | 工  |
| 3 | 20 | 3  | ih | 工  |
| 4 | 30 | 4  | 大  | 工  |

```
<input type="button" id="btnExcel"/>
```

```

$("#btnSmall").excel({
    title: "导入excel",
    tableId: 'table1',
    beforeShow: function (target) { //打开前
        alert(target.prop('id'))
    },
    callback: function (data, target) { //导入后回调, data:导入的数据, target:事件源
        alert(target.prop('id'));
    },
    appendType: 'first', //追加方式, first:删除已有行, 从第一行开始新增行, append(默认):从尾行新增
    fields: ['Text1', 'Text2', 'Text3', 'Text4'] //和Excel的列头个数相对应, 例: 将第一列赋值给formdatatag='表名.Text1'的控件
    //如为空, 默认取Excel列头
});

```

## 9. 自定义逻辑

```
function afterNewRow(row) { } //新增行后,执行方法(row,新增行对象tr)
function afterDeleteRow() { } //删除行后执行的方法

////模板加载成功后,解析之前的
function StartOnload() { } //5.7
function customInit() { } //6.0

function RejectClick() { } //5.7 拒绝前(return true or false)
function EndRejectClick() { } //拒绝后

////数据加载完成后
function EndOnload() { } //5.7
function customAfterLoadData() { }; //6.0

function StartSave() { } //保存前(return true or false)
function EndSave(draftId) { } //保存后,发起保存参数是草稿Id,审批保存参数为taskId

////动作提交前(return true or false)
function StartApprovelClick(actionname) { } //5.7
function customBeforeAction(actionname) { }; //6.0

////动作执行后
function EndApprovelClick(actionLinkName,tid) { }; //5.7
function customAfterAction(actionLinkName, tid) { }; //6.0
|
//数据撤回之后
function customAfterRecedeBack() {}
```

## 10. 提交时验证

系统中已有的验证规则有:

- "email":电子邮件地址
- "notnull":控件值不能为空
- "needrow":子表不能为空

示例:

- 1:<input type="radio" dataformtag="table1.field1" check="notnull"  
checktitle="单选" />  
<input type="radio" dataformtag="table1.field1"/>//多选和单选只需设置  
其中一个元素
- 2:<input check="notnull,email" checktitle="电子邮件" />
- 3:<table formgrid check="needrow" checktitle="子表 1"></table>

# 11. 通用弹出层

## 11.1 属性

|           |                      | 属性                            | 描述   |
|-----------|----------------------|-------------------------------|--|
| gModal    | title                | 标题                            |  |
|           | template             | 页面                            |  |
|           | width                | 层宽 (默认 800)                   |  |
|           | height               | 层高 (默认 600)                   |  |
|           | param                | 参数                            |  |
|           | beforeShow           | 打开前方法, 返回 <b>false</b> 弹出层不显示 |  |
|           | callback             | 层确定后, 执行方法                    |  |
|           | closeAfterSuccess    | 确定成功后是否关闭                     |  |
|           | displayButton        | 是否显示层自代的确定和取消按钮(默认 true)      | "确定"按钮自动调用 <b>callback</b> 方法<br>"取消"按钮自动关闭层 |
|           | div                  | 自定义 div 的 Id, 若为空, 将自动创建      |  |
|           | top                  | 居上高度                          | 不传 自动计算居上高度                                  |
| modalType | 层弹出的类型,iframe 或者 div |                               |  |

## 11.2 取值

| 取值                  |                        |                     |                   |
|---------------------|------------------------|---------------------|-------------------|
| 子页面取父页面             |                        | 子页面获取参数             |                   |
| modalType 值为 iframe | parent.opener.document | modalType 值为 iframe | parent.cuop.param |
| modalType 值为 div    | document               | modalType 值为 div    | cuop.param        |

| 子页面关闭弹出层            |                      | 获取当前弹出层的 id         |                 |
|---------------------|----------------------|---------------------|-----------------|
| modalType 值为 iframe | parent.cuop.close(); | modalType 值为 iframe | parent.cuop.div |
| modalType 值为 div    | cuop.close();        | modalType 值为 div    | cuop.div        |

D

|                  |
|------------------|
| 子页面获取事件源(事件按钮对象) |
|------------------|

|                     |                    |
|---------------------|--------------------|
| modalType 值为 iframe | parent.cuop.target |
| modalType 值为 div    | cuop.target        |

## 11.3 调用

### 11.3.1 主表

```

function functionA(para1,para2){
    alert(para1);
}
////数据加载完成后
function customAfterLoadData() {
    button.gModal({
        title: "加签人员",
        template: "../com/Wf_SelectUser1.htm",
        width: "890",
        height: "540",
        closeAfterSuccess: false,
        callback: functionA,
        beforeShow: function () {
            if ($("#aa").val() == "") {
                alert('请输入必填项');
                return false;//return false 将不打开弹出层
            }
        },
        modalType: "iframe",
        param: { mode: "single", containsSelf: false, showMemo: true }
    });
};

```

### 11.3.2 子表

```

function showFun(btn) { // btn为当前按钮对象
    //打开层前会执行 return ture or false
}
////新增行后执行方法 (row :新增行的对象tr)
function afterNewRow(row) {
    $("#input[name='btnAaa']").each(function () { //新增行中 name为btnAaa的按钮对象
        $(this).gModal({
            title: "来文单位",
            template: "../../GrinnFormHandler/plugin_laiwenmanager/SelectLaiwenDetail.htm",
            width: "800",
            height: "550",
            top: "100",
            modalType: "div",
            callback: function (param1,param2) { //这里的参数 在子页面里传过来
                //这里的方法 点击确定后,会执行
            },
            beforeShow: showFun //这里的方法在打开层前会执行
        });
    });
};
}

```

## 11.4 子页面配置

### 11.4.1 说明

子页面加载时执行的函数请放在 `init()` 方法中。回调执行函数请放在子页面的 `callback()` 方法中。子页面 `callback` 中需要添加 `curop.callback(para)` 来调用父页面的 `callback` 方法， 参数可选。

### 11.4.2 modalyType 等于 div

```
<div style="padding-left: 10px;">
|   <div>
|     <br />
|     <label>...</label>
|     <span id="authUserID"></span>
|     <br />
|     <br />
|     <label>...</label>
|     <input type="password" id="authPwd" class="login_inputtext" />
|     <br />
|     <br />
|   </div>
| </div>
| <script type="text/javascript" id="oPopHelper">
|   function init() {
|     var userDspName = curop.param.userDspName;
|     $("#authUserID").html(userDspName);
|   }
|   function callback() {
|     var pwd = $.trim($("#authPwd").val());
|     if (pwd != "") {
|       pwd = hex_md5(pwd);
|     }
|     if (curop.callback) {
|       curop.callback(pwd);
|     }
|   }
| }
| </script>
```

### 11.4.3 modalyType 等于 iframe

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml
|<html xmlns="http://www.w3.org/1999/xhtml">
|<head>...</head>
|<body>...</body>
|<script src="../../Commonjs/jquery-1.9.1.js" type="text/javascript"></script>
|<script type="text/javascript">
    //param: { handlerUrl: "../", mode: "single", containsSelf: false, showMemo: true }
    var showMemo = parent.cuop.param.showMemo; //获取参数
1   function callback() {
1       if (dataArr.length < 1) {alert("请选择用户");
           return false;
           }
           var selArray = [];
1       for (var i = 0; i < dataArr.length; i++) {
1           var user = {
               UserId: dataArr[i].userid(),
               UserAccount: dataArr[i].useraccount()
           }
           selArray.push(user);
           }
           parent.cuop.callback(returnData);|
    };
    var currentUser = "";
1   $(function () {
1       $.ajax({
           url: "../FormHandler/AjaxUserInfo.ashx?req=U003",
           type: "get",
1       success: function (data) {
               currentUser = JSON.parse(data);
           },
           error: function (data) {return null;}
       });
    });
    ..

```

## 12. 通用附件上传

附件上传通用模板页面路径：站点根/Form/com/uploadtemplate.htm

### 12.1 属性

| 参数 param 的属性: |   | 例   |
|---------------|---|---|
| 属性名           | 描述  |   |
| resourceType  | 源类型(必传) 没有特定类型可以根据业务要求自定义                                     | 表单上传附件<br>resourceType=表单上传<br>resourceId=任务 Id |
| resourceId    | 源 Id(必传) 没有特定类型的 id, 可以输入-1                                   |   |
| count         | 允许上传附件个数(不传将设置默认允许上传个数)                                       |   |
| ext           | 允许上传附件类型(不传将设置默认允许上传类型)                                       |   |
| savePath      | 上传到的路径(必传)<br>相对于 uploadtemplate.htm 的路径<br>存放文件的目录不存在, 将自动创建 |   |

## 12.2 上传成功后返回对象

| 上传成功后返回 js 对象包含属性 |                  | 上传成功 自动执行方法  |
|-------------------|------------------|--|
| ExtensionName     | 扩展名              | dspAttach(filesList, target)//请在 gModal 的 callback 的 js 方法, 上传成功后将自动调用<br>参数 filesList 返回的是已上传附件对象的集合<br>属性请参考左边属性列表 |
| PhysicalName      | 物理名(上传后的随机名+扩展名) |  |
| DisplayName       | 显示名              |  |
| ResourceType      | 源类型              |  |
| ResourceId        | 源 Id             |  |
| Url               | 所在目录路径(相对上传页面)   |  |

## 12.3 示例:

```

1 $("[name='fujian']", row).gModal({
    modalType: "iframe",
    title: "选择上传文件",
    template: "../com/uploadtemplate.htm",
    width: "460",
    height: "330",
    displayButton: false,
    param: { resourceId: 1, resourceType: "Portal_Advise", count: 10, ext: "png,jpg,gif,jpeg", savePath: "../../At
1   callback: function (data, target) { //data:上传附件的信息, target:事件源
        var tr = target.parents('tr').index();
        alert(target.parents('tr').index() + 1)
    }
});

```

子表请放在自定义逻辑 afterNewRow(row) 中调用。

## 12.4 附件上传配置版

```

<input type="button" value="上传" displaytype="thumbnail"
        buttontype="attachment"
        ext="doc,jpg,bpm" formdatatag="Sys_FormDemo.Text3"
        formdatasrc="Sys_FormDemo.Text3" />

```

只需在表单上添加一个按钮, 按钮属性配置如下:

```

buttontype="attachment"    //必须
displaytype    //显示方式thumbnail: 上传后立即以缩略图方式呈现在表单上
                  link: 上传后以链接形式打开(默认)
jpgWidth       //显示方式为thumbnail时有效, 设置图片显示宽度
jpgHeight      //显示方式为thumbnail时有效, 设置图片显示高度
ext            //上传格式, displaytype 为 link 时有效
formdatatag    //存入的字段
formdatasrc    //显示的字段(与 formdatatag 必须相同)

```

## 13. 通用选择页面



## 13.1 选用户

### 13.1.1 页面路径

| 路径                              | 说明   |
|---------------------------------|------|
| 站点根/Form/com/Wf_SelectUser.htm  | 有组织树 |
| 站点根/Form/com/Wf_SelectUser1.htm | 无组织树 |

### 13.1.2 参数配置

| 参数名                        | 值        | 说明  |
|----------------------------|----------|-----|
| mode                       | single   | 单选  |
|                            | multiple | 多选  |
| containsSelf<br>是否包含当前登录用户 | true     | 包含  |
|                            | false    | 不包含 |

### 13.1.3 调用示例

```
actionBtn.gModal(  
{  
  title: "加签人员",  
  template: "../com/Wf_SelectUser1.htm",  
  width: "890",  
  height: "545",  
  closeAfterSuccess: false,  
  callback: Cloud.action.endorseTask,  
  modalType: "iframe",  
  param: { mode: "single", containsSelf: false, showMemo: true }  
});
```

## 13.2 选部门

页面路径：站点根/Form/com/Wf\_SelectGroup.htm

示例请参照选用户章节介绍。

## 14. 通用 Ajax

### 14.1 参数

| 参数   | 值        | 作用 | 例 |
|------|----------|----|---|
| type | get/post |    |   |

|          |            |  |                      |
|----------|------------|--|----------------------|
| url      |            | 后台处理页面   | ../Form/Handler.ashx |
| param    |            | 提交到后台的参数   |                      |
| dataType | text/json  | 返回值类型<br>text 表示 返回纯字符串<br>json 表示 返回 json 对象<br>不传, 默认 text |                      |
| async    | true/false | 同步/异步 不传 默认 false  |                      |

## 14.2 返回结果

| 属性       | 值          | 说明                          |
|----------|------------|-----------------------------|
| result   | true/false | true 成功 false 失败            |
| errorMsg |            | result 值为 false 时, 所返回的错误信息 |
| data     |            | 返回数据                        |

## 14.3 示例

```

var para = { pid: '1111', lan: 'ch' } //参数
var oResult = oform.ajax("post", "Handler.ashx", para); //调用
if (oResult.result == false) { //获取操作结果, 成功或失败
    alert(oResult.errorMsg); //错误信息
    return false;
}
var data=oResult.data; //返回结果

```

# 15. 操作 Json

## 15.1 C#

### a) 引用

```
using System.Web.Script.Serialization;
```

### b) 实例化

```
JavaScriptSerializer js = new JavaScriptSerializer();
```

### c) 序列化: js.Serialize(formBasicData);

```

SYSUserInfo user = new SYSUserInfo
{
    UserAccount = "user001",
    UserDspName = "张三"
};
JavaScriptSerializer jss = new JavaScriptSerializer();
string userJson = jss.Serialize(user);

```

d) 将json字符串转换成后台对象

```

string jsonStr= "[{"UserId":1084,"UserName":"\u8521\u632f\u5e73"}, {"UserId":1046,"
    UserName":"\u8521\u4e00\u745c"}]";
List<HandlerUser> userList =
    (List<HandlerUser>)js.Deserialize(userJson, typeof(List<HandlerUser>));

```

## 15.2 js

### 15.2.1 将 js 对象转换为 Json 字符串

```

var defaults = {
    tid: '111',
    pid: '1205'
};
var defJson = JSON.stringify(defaults);

```

### 15.2.2 将 Json 字符串转换为 js 对象

```

var defJson = '{"tid":"111","pid":"1205"}';
var defaults = JSON.parse(defJson);

```

## 16. 常用取值、加密解密

| 获取当前任务信息 (oform 对象) |                  | 值   |
|---------------------|------------------|---|
| 草稿 Id               | oform.draftId    |   |
| 流程 Guid             | oform.flowGuid   |   |
| 流程名                 | oform.flowName   |   |
| NodeId              | oform.nodeId     |   |
| 任务 Id               | oform.tid        |   |
| 步骤 Id               | oform.pid        |   |
| 任务状态                | oform.taskStatus | 0: 运行中 1: 已暂停 2: 已停止 4: 已完成 5: 已拒绝 6: 已删除 |
| 任务名                 | oform.taskName   |   |
| 流水号                 | oform.serialNum  |   |

| 当前登录用户 (oform.loginUser) |                         |
|--------------------------|-------------------------|
| 所在组织 Id                  | oform.loginUser.GroupId |

|                       |                             |
|-----------------------|-----------------------------|
| 所在组织 Name             | oform.loginUser.GroupName   |
| 职位                    | oform.loginUser.JobTitle    |
| 公司 Code               | Oform.loginUser.CompanyCode |
| 其他属性请参看数据库 SYS_User 表 |                             |

| js 方法名                      | 参数         | 返回值           | 作用   | 例  |
|-----------------------------|------------|---------------|--|--|
| oform.checkStep(t)          | directSend | true<br>false | 判断当前步骤是否是直送过来的步骤                                   | oform.checkStep('directSend')              |
|                             | recall     | true<br>false | 判断当前步骤是否是取回的步骤                                     | oform.checkStep('recall')                  |
| oform.getAttach(tid)        |            |               | 1. 传参, 返回指定任务的附件集合;<br>2. 不传参, 返回当前任务/草稿已上传的附件对象集合 | var fileList = oform.getAttach()           |
| oform.DESEncrypt(str, type) |            |               | str: 要加密/解密的字符串<br>type: 0 加密, 1 解密                | var tid='1111'<br>oform.DESEncrypt(tid, 1) |
| \$.query.get("t")           |            |               | 获取页面 url 参数中 t 的值                                  | ../handler.html?t=a                        |



## 16.1 后台调用加密解密方法

```
using Galaxy.Framework;
```

```
string returnStr=ESEncrypt.Encrypt(str) //加密
```

```
string returnStr=ESEncrypt.Decrypt(str) //解密
```

## 17. 表单通用按钮样式

| 按钮样式  | 样式   | 描述   |
|-------|--|--|
| 普通按钮  | 小: class="btnSmall"<br>中: class="btnMedium"<br>大: class="btnBig"<br>特大: class="btnBiger" |            |
| 弹出层按钮 | class="browse"   | value=""  |

## 18. 代理任务

## 18.1 代理发起任务

在表单中添加一个选择用户的功能，当发起人在提交表单前，弹出页面选择被代理人，将选择后的被代理者的 UserId 数据绑定到 id=" fco\_ownerUserId"的文本框中即可。fco\_ownerUserId 文本框在表单模板页面 "liveflowdefault.htm" 中已存在。

## 18.2 代理审批任务

可通过代理设置页面进行设置，或者在表单的更多操作中的“委托”功能。

# 19. 历史表单中控件处理

历史表单表示从可查看，已发起、已处理等打开的只能查看的表单。

表单引擎做了以下处理。

1. 所有的 textbox、textarea 只读；
2. Radio、checkbox、select 禁用；
3. Button 隐藏

特殊情况下，按钮控件在历史表单中需要显示的时候，请在控件上定义

属性 show="1"

# 20. 表单通用查询页面插件

## 20.1 属性说明

| 属性    | 类型     | (默认值) 值               | 必填 | 描述    |
|-------|--------|-----------------------|----|-------|
| title | string | ('')                  |    | 标题    |
| mode  | string | ('single'),'multiple' |    | 单选/多选 |

|              |            |             |   |   |
|--------------|------------|-------------|---|---|
| width        | [number]   | (830)       |   | 宽度  |
| height       | [number]   | (610)       |   | 高度  |
| top          | [number]   | ('')        |   | 居上 (不设置自动计算)  |
| tableName    | [string]   | ('')        | 是 | 数据库表名或视图 (tableName/sql 选其一)                              |
| sql          | [string]   | ('')        | 是 | 查询Sql语句 (tableName/sql 选其一)                               |
| uniqueFields | [string]   | ([])        | 是 | 主键或者判断唯一的字段集合。单选模式必须配置uniqueFields: 'UserId, UserDspName' |
| filterSql    | [string]   | ('')        |   | 条件<br>例: useraccount <> 'sa' and username <> 'zhangsan'   |
| defaultOrder | [string]   | ('')        | 是 | 默认排序字段 (支持field1 asc, field desc)                         |
| showFields   | [array]    | ([])        | 是 | 表头配置字段集合  |
| search       | [array]    | ([])        | 是 | 查询区域配置集合  |
| callback     | [function] | (undefined) |   | 数据选择后执行的脚本  |
| beforeShow   | [function] | (undefined) |   | 页面打开前执行的脚本  |
| pageSize     | [number]   | (10)        |   | 每页显示多少条数据   |

## 20.2 列头配置说明

### 属性 showFields 配置列头

例 20.2-1 :

```
showFields: [{ fieldName: 'UserId', dspName: '用户Id', align: 'center', width: '10%' },
{ fieldName: 'UserAccount', dspName: '用户账号', align: 'center', width: '20%' }]
```

如上例, showFields为集合, 元素为每列的配置信息, 多列配置用逗号(,)分隔, 列的配置信息包括:

fieldName: '', //该列显示的字段名

dspName: '', //该列的显示名

align: '', //文字显示方式 left, center, right

width: '', //宽度

format: '', /\*显示格式:

1. 日期例配置'yyyy-MM-dd'、'yyyy-MM-dd hh:mm:ss' 或者  
'yyyy-mm-dd'、'yyyy-mm-dd hh:ii:ss'

2. 特定列需要转换值, 例如数据库存入0, 1, 2等, 但是显示时转成用户可以看懂的数据 [{key: 0, value: "男"}, {key: 1, "value": "女"}]

\*/

split: '' // format类型为2时(特定列)有效, 数据以某字符分隔, 如逗号、分号等; 默认逗号

## 20.3 查询区域配置说明

**说明** : search 为集合, 元素为每个查询控件的配置信息, 多查询用逗号(,)分隔, 配置信息包括

日期查询配置、下拉框配置、文本框配置、子查询页面配置.

### 20.3.1 文本框配置

#### 一、文本

```
{
  fieldName: 'UserDspName', //查询的字段, 不区分大小写(*)
  dspName: '用户名',      //字段显示名(*)
  tagName: 'text',        //控件类型, 文本框(*)
  searchType: 'like',     //查询类型(like, equals), 默认like
  defaultValue: 'sa'     //查询默认值, 默认空
}
```

#### 二、数字

```
{
  fieldName: 'UserId',
  dspName: '用户Id',
  searchType: 'like',
  tagName: 'number',      //控件类型:文本框, 只能输数字
  dec: 2,                 //小数点位数
  defaultValue: ''       //默认值
}
```

### 20.3.2 日期查询配置

```
{
  fieldName: 'Birthday', //查询的字段
  dspName: '生日',       //字段显示名
  searchType: 'between', //查询类型(between)
  tagName: 'date',       //控件类型
  defaultValueStart: '1950-1-1', //默认查询的初始日期, 默认空
  defaultValueEnd: '2013-12-31' //默认查询的结束日期, 默认空
}
```

### 20.3.3 下拉框配置(自定义项)

```
{
  fieldName: 'Gender', //查询的字段(*)
}
```

```

dspName: '性别', //字段显示名(*)
tagName: 'select', //控件类型, 文本框(*)
searchType: 'equals', //查询类型(like, equals), 默认like
defaultValue: '0', //查询默认值, 默认空
options: [{value: '', text: '--请选择--' },
          { value: '0', text: '男' },
          { value: '1', text: '女' }]
}

```


### 20.3.4 下拉框配置(绑定表)

```

var department = oform.loginUser.Department;
{
  fieldName: 'department', //查询的字段(*)
  dspName: '部门', //字段显示名(*)
  tagName: 'select', //控件类型, 文本框(*)
  searchType: 'equals', //查询类型(like, equals), 默认like
  defaultValue: department, //查询默认值, 默认空
  bindTable: 'vw_sys_Department', //绑定的表或视图
  filterSql: "department not in('有研分院1', '有研分院2')", //过滤条件
  valueField: 'Department', //值字段(不区分大小写)
  dspField: 'Department', //显示字段(不区分大小写)
  prompt: '--please select--' //提示
}

```

### 20.3.5 从另一个选择页面带出查询数据

如图 ，某些情况下查询区域的个别条件是需要从另外一个弹出页面带出的，例如，选择人员，显示文本框里保存用户名，隐藏字文本框里保存用户账号，查询的时候从取隐藏文本框的值做为条件。

```

{
  tagName: 'choose', //类型:通过另外一个弹出页面选择带出数据
  fieldName: 'CompanyId', //要查询的字段
  dspName: '公司名',
  returnDspField: 'CompanyName', //带出的显示字段的值 (只显示不做查询条件)
  returnHiddenField: 'CompanyId', //带出的隐藏字段 (做查询条件)默认空时取
  //returnDspField的值
  searchType: 'equals',
  defaultValueDspField: '有研总院',
  defaultValueHiddenField: '1',
  chooseConfig: //子查询页面的配置与本页面相同tableName, sql, height, width等都支持; 没有
  //callback, beforeShow属性, 只能单选
}

```



```

    sql: 'select * from Sys_Company where Active=1',
    filterSql: '',
    defaultOrder: "CompanyName",
    showFields: [{ fieldName: 'CompanyId', dspName: "公司编号", align: "center", width:
        '15%' },
        { fieldName: 'CompanyCode', dspName: "公司代码", align: "center", width: '40%' },
        { fieldName: 'CompanyName', dspName: "公司名", align: "center", width: '40%' }],
    search: [{ fieldName: 'CompanyName', dspName: '公司名', searchType: 'like',
        tagName: 'text', defaultValue: ''}],
    title: '选择公司',
    pageSize: '6' //条件查询页面默认条每页显示5条
}
}

```

PS:子页面的配置与当前页的配置参数相同(不包括: `callback`, `beforeShow`), 配置属性放在 `chooseConfig`对象中, 固定单选模式

## 20.4 举例

### 20.4.1 简单查询配置 ( 单选 )

```

var dept = oform.loginUser.Department;
<input id= 'btnSelect' type= 'button' />
$("#btnSelect").showQuery({
    title: "选择部门",
    mode: "single", //single(单选),multiple(多选),
    width: "900",
    height: "500",
    tableName: "vw_sys_Department",
    filterSql: "Department <> '" + dept + "'",
    defaultOrder: "CompanyId", //默认排序字段
    uniqueFields: 'UserId, UserDspName', //主键或者判断唯一的字段集合。单选模式
        必须配置, 逗号分隔

    showFields: //配置表头/
    [{ fieldName: 'CompanyId', dspName: "CompanyId", align: "center", width: '20%' },
        { fieldName: 'Department', dspName: "部门", align: "center", width: '70%' }],
    //配置查询区域
    search: [{ fieldName: 'Department', dspName: '部门', searchType: 'like', tagName:
        'text', defaultValue: ''}],
    callback: function (data, target) { //target:事件源, data:返回数据
        var department = data.department; //单选模式取值, 属性部小写
        alert('选中了' + department);
        //如果选中的数据不符合要求, return false;弹出框就不关闭
    },
    beforeShow: function (target) { //target:事件源(按钮btnSelect)

```

```

//打开页面前验证可以写在这里, return false 不打开选择页面
return true;
}
});

```

选择部门
✖

部门

共 115条记录 页次: 1/12

查询

首页 | 上一页 | 下一页 | 末页

| 选择                    | 公司Id | 部门                |
|-----------------------|------|-------------------|
| <input type="radio"/> | 0    | Corporate Affairs |
| <input type="radio"/> | 0    | Internal Audit    |
| <input type="radio"/> | 0    | Project Dept      |
| <input type="radio"/> | 0    | 测试部门              |
| <input type="radio"/> | 0    | 院长部               |
| <input type="radio"/> | 1    | 测试部门              |
| <input type="radio"/> | 1    | 产业发展部             |
| <input type="radio"/> | 1    | 超导材料中心            |
| <input type="radio"/> | 1    | 电子材料所             |
| <input type="radio"/> | 1    | 分析测试所             |

### 20.4.3 复杂查询配置 ( 多选 )

```

var uid = oform.loginUser.UserId;
var department = oform.loginUser.Department;

$( ":button[name=' btnSearchDetail' ]", row ).showQuery( {
    title: "选择人员",
    mode: "multiple", //single(单选),multiple(多选)
    width: "900",
    height: "500",
    sql: "select * from sys_user",
    filterSql: "UserId <>" + uid,
    defaultOrder: "UserDspName",
    showFields:
    [ { fieldName: 'UserId', dspName: "用户Id", align: "center", width: '10%' },
      { fieldName: 'UserAccount', dspName: "用户账号", align: "center", width: '20%' },
      { fieldName: 'UserDspName', dspName: "用户显示名", align: "center", width: '20%' },
      { fieldName: 'Birthday', dspName: "用户生日", align: "center", width: '20%', format:
      'yyyy-MM-dd' }, //yyyy-MM-dd hh:mm:ss
      { fieldName: 'HRID', dspName: "员工号", align: "center", width: '20%' },
      { fieldName: 'Gender', dspName: "性别", align: "center", width: '5%',
      format: [ { key: 0, value: '男' }, { key: 1, value: '女' } ]
    }
  ]
},
search: //配置查询区域

```

```

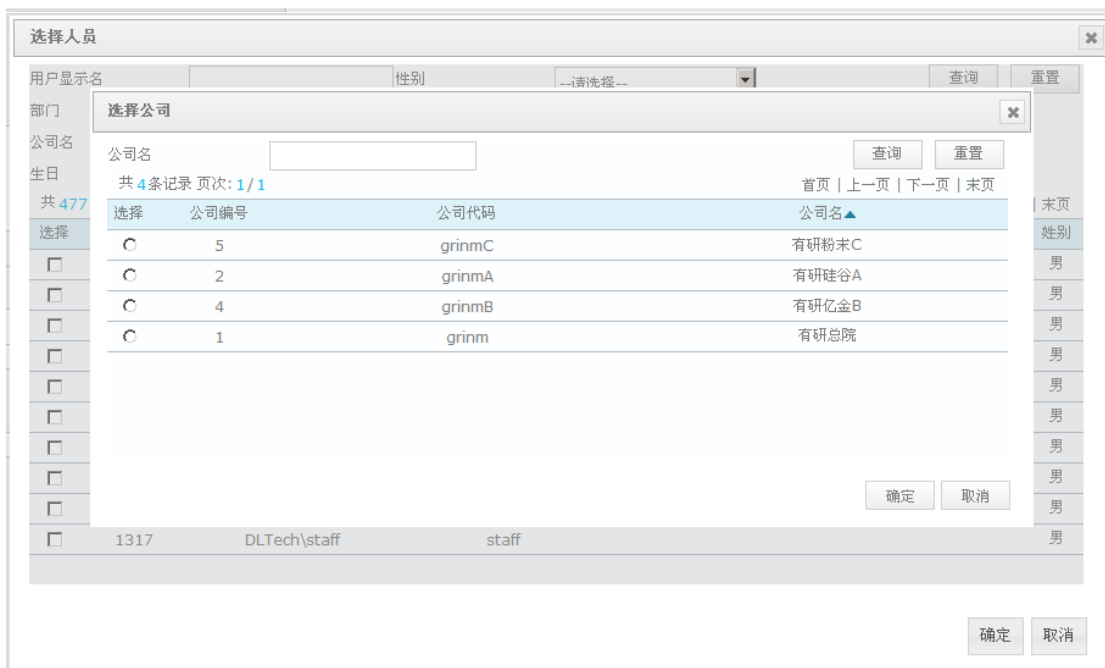
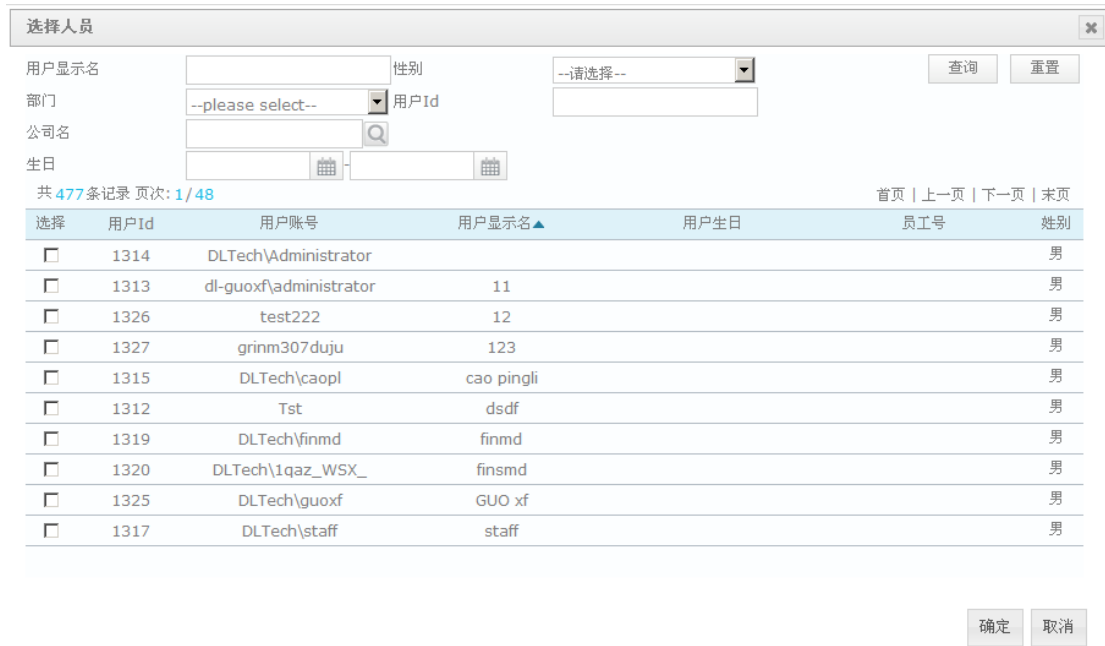
    [{ fieldName: 'UserDspName', dspName: '用户显示名', searchType: 'like', tagName: 'text' },
      { fieldName: 'Gender', dspName: '性别', searchType: 'equals', tagName: 'select',
        defaultValue: '1', options: [//自定义下拉框选项(非绑定表)
          { value: '', text: '--请选择--' },
          { value: '0', text: '男' },
          { value: '1', text: '女' }
        ]
      },
    ],
    { fieldName: 'department', dspName: '部门', searchType: 'equals', tagName: 'select',
      bindTable: 'vw_sys_Department',
      valueField: 'Department', //值字段
      dspField: 'Department', //显示字段
      defaultValue: department, //默认值
      prompt: '--please select--' //提示),
      { fieldName: 'Birthday',
        dspName: '生日',
        searchType: 'between',
        tagName: 'date',
        defaultValueStart: '1950-1-1', //默认查询的初始日期
        defaultValueEnd: '2013-12-31' //默认查询的结束日期
      },
    ],
    {
      tagName: 'choose', //类型:通过另外一个弹出页面选择带出数据
      fieldName: 'CompanyId', //要查询的字段
      dspName: '公司名',
      returnDspField: 'CompanyName', //带出的显示字段的值 (只显示不做查询条件)
      returnHiddenField: 'CompanyId', //带出的隐藏字段 (做查询条件)
      searchType: 'equals',
      defaultValueDspField: '有研总院',
      defaultValueHiddenField: '1',
      chooseConfig:
      {
        sql: 'select * from Sys_Company where Active=1',
        filterSql: '',
        defaultOrder: "CompanyName",
        showFields: [{ fieldName: 'CompanyId', dspName: "公司编号", align: "center", width:
          '15%' },
          { fieldName: 'CompanyCode', dspName: "公司代码", align: "center", width: '40%' },
          { fieldName: 'CompanyName', dspName: "公司名", align: "center", width: '40%' }],
        search: [{ fieldName: 'CompanyName', dspName: '公司名', searchType: 'like',
          tagName: 'text', defaultValue: '' }],
        title: '选择公司',
        pageSize: '6' //条件查询页面默认条每页显示5条
      }
    }
  ],

```

```

{ fieldName: 'UserId', dspName: '用户Id', searchType: 'like', tagName: 'number', dec: 2}],
callback: function (data, target) { //data:返回据, target:事件源
    alert('选中了' + data.length + '个数据');
    for (var i = 0; i < data.length; i++) {
        var userAccount1 = data[i].useraccount;
    }
    //如果选中的数据不符合要求, return false;弹出框就不关闭
},
beforeShow: function (target) {
    return true;
}});

```



## 21. 表单通用查询页面配置版

### 21.1 说明:

弹出框配置版是和PageConfig相结合的功能。

步骤1: 首先要从弹出框工具中配置好相应的数据源;

步骤2: 在表单页面中添加一个button, 并配置button的特定属性

### 21.2 button 属性说明

```
<input type="button" buttontype="browse" class="browse"  
      pageconfigname="DialogGroup" mode="multiple"  
      datamap="GroupName--sys_formdemoDetail.Text2;companyname--sys_formdemo.textnumertest"/>
```

|                     |                                 |
|---------------------|---------------------------------|
| buttontype="browse" | 必须                              |
| pageconfigname      | 弹出框工具中配置后的名称                    |
| mode                | 单选/多选single:单选(默认) multiple: 多选 |
| datamap             | 选择确定后映射的字段(多选情况下将以逗号分隔方式返回值)    |

补充: 如配置版本不满足需求, 请使用第20章的代码版或者自定义弹出页面