

# 参数控件开发指南

## 概述

产品中的参数控件有时不能满足项目特殊需要，项目需要添加新的参数控件类型，此文档说明如何通过插件包在产品中新增参数控件。

## 实现说明

### 新增、编辑界面

通过插件开发方式，对产品smartbi.war\vision\js\freequery\sysresource\Parameter.js进行扩展，增加ext\vision\js\freequery\sysresource\Parameter.js.patch，在此文件中根据项目需要定制参数编辑界面。以下新增列表日期参数为例：

- 在Parameter.js.patch中增加以下代码：

```
Parameter.prototype._gddx_initControlType = Parameter.prototype.initControlType;  
Parameter.prototype.initControlType = function() {  
    this._gddx_initControlType();  
    this.ControlType.insertItems(new Array(["LIST_DATE", ""]));  
    this.ControlType.setReadOnly(true);  
    this.ControlType.setSelectedItem("COMBOBOX", "");  
};
```

以上代码在界面中则会相应增加可选型参数控件的类型，如图：

The screenshot shows a configuration window for a parameter named 'a' with the alias '结束时间 (northwind)'. The '数据类型' (Data Type) is set to '字符串' (String). The '控件类型' (Control Type) dropdown menu is open, showing a list of options: '可选型日期参数', '频度日期控件', '指标报表日期控件', '文本输入框', '列表对话框', '树对话框', '平铺勾选面板', and '可选型日期参数'. The '可选型日期参数' option at the bottom is highlighted with a red rectangular box.

- 保存时，需要把控件类型保存为OTHER，把真实控件类型写入扩展字段中，可拷贝以下代码使用：

```
Parameter.prototype._gddx_save = Parameter.prototype.save;  
Parameter.prototype.save = function() {  
    if (this.ControlType.getSelectedId() != "LIST_DATE") {
```

```

        this._gddx_save();
        return;
    }
    var name = this.ParamName.value;
    if (this.type == this.OptionType_ADD) {
        var ret = util.remoteInvoke("ParamService", "createParam", [
            this.dsId,
            this.parentFolderId,
            name,
            this.ParamAlias.value,
            this.ParamDesc.value,
            this.ParamType.getSelectedId(),
            this.getStandbyType(),
            this.getStandbyEditorValue(),
            this.getDefaultType(),
            this.getDefaultEditorValue(),
            this.NoShow.checked,
            this.NoWrite.checked,
            "OTHER",
            this.getComponentDefine(this.ControlType.getSelectedId(), this.getRootType(),
            this.getRootEditorValue(), "PARENT_CHILD",
            this.CheckValue.checked ? "true" : "false",
            this.MultSelect.checked,
            this.cacheableBox.getSelectedId(), "{\"componentType\": \"LIST_DATE\"}"]);
        if (ret.succeeded && ret.result) {
            alert("${Savesuccessful}${Exclamation}");
            this.needRefresh.fire(this, this.parentFolderId);
            this.contentChanged = false;
            return true;
        } else {
            if (ret.retCode == "CATALOG_DUPLICATE_FOLDER") {
                modalWindow.showServerError
                ("${Parameterswiththesamenameexists}${Comma}${Pleasereplacetheparametername}");
            } else {
                modalWindow.showServerError(ret);
            }
        }
    }
    } else if (this.type == this.OptionType_MODIFY) {
        var dv = this.getDefaultEditorValue()
        var sv = this.getStandbyEditorValue();
        var ret = util.remoteInvoke("ParamService", "updateParam", [
            this.paramId,
            this.ParamAlias.value,
            this.ParamDesc.value,
            this.ParamType.getSelectedId(),
            this.getStandbyType(),
            sv,
            this.getDefaultType(),
            dv,
            this.NoShow.checked,
            this.NoWrite.checked,
            "OTHER",
            this.getComponentDefine(this.ControlType
            .getSelectedId(), this.getRootType(),
            this.getRootEditorValue(), "PARENT_CHILD",
            this.CheckValue.checked ? "true" : "false",
            this.MultSelect.checked,
            this.cacheableBox.getSelectedId(), "{\"componentType\": \"LIST_DATE\"}"]);
        if (ret.succeeded) {
            alert("${Updatesuccess}${Exclamation}");
            this.contentChanged = false;
            this.standByValue = sv;
            this.defaultValue = dv;
            this.needRefresh.fire(this, this.paramId);
            return true;
        } else {
            modalWindow.showServerError(ret);
        }
    }
}
return false;
}

```

## 展现界面

修改插件包的vision\js\ext\ConfigurationPatch.js文件，增加以下Parameter.controls的配置项：

```
var ConfigurationPatch = {
  extensionPoints: {
    Parameter: {
      controls:[
        {type: "LIST_DATE", className:"gddx.parameter.ListDate"}
      ]
    }
  }
};
```

配置说明：

- type: 参数类型，此类型为以上保存在数据库中的“\“componentType\“:\“LIST\_DATE\“”中的LIST\_DATE
- className: 参数展现类路径

ListDate.js直接继承参数展现的基础类：smartbi.war\src\web\vision\js\freequery\control\BaseParameterControl.js，[API请见以下第3章节](#)。在捕获到参数值变化的方法里，需要抛出通知事件，代码为：

```
/**
 * @param {} this
 * @param {} oldValue
 * @param {} newValue
 * @param {} oldDisplayValue
 * @param {} newDisplayValue
 */
this.onselectChange.fire(combo, oldValue, newValue, oldDisplayValue, newDisplayValue);
/**
 * @param {} newValue
 * @param {} newDisplayValue
 */
this.oneditChange.fire(control, newValue, newDisplayValue);

ListDate.prototype.doEditBlur = function() {
  if (this.edit.value == "" && this.edit.value!=this.tempValue){ //${Comma}
    var transValue = this.edit.value;
    this.tempValue = transValue;
    this.onselectChange.fire(this, this.oldValue, transValue, this.oldValue, transValue);
    this.oldValue = transValue;
  }
  if (this.edit.value != this.tempValue){
    var transValue = this.edit.value;
    if(!util.checkDateTimeString(transValue, 1)){
      alert("${Dateformatnotcorrect}${Comma}${Pleasere_enter}${Exclamation}");
      return;
    }
  }
  var d = new Date(Date.parse(transValue.replace(/-/g, "/")));
  transValue = d.getFullYear() + "-" + this.appendZero(d.getMonth()+1) + "-" + this.appendZero(d.getDate());

  this.tempValue = transValue;
  this.onselectChange.fire(this, this.oldValue, transValue, this.oldValue, transValue);
  this.oldValue = transValue;
}
}
ListDate.prototype.editChange = function() {
  this.oneditChange.fire(this,this.edit.value, this.edit.value);
}
}
```

## 接口说明

### 1. a. BaseParameterControl.js

接口说明如下:

```
/**
 *
 * @param {} parentObject
 * @param {} parameterDefine
 */
var BaseParameterControl = function(parentObject, parameterDefine) {
}
/**
 *
 * @param {} parameter
 */
BaseParameterControl.prototype.render = function(parameter) {
}
/**
 *
 */
BaseParameterControl.prototype.custom = function() {
}
/**
 *
 * @param {} parameter
 */
BaseParameterControl.prototype.onRelateParamChanged = function() {
}
/**
 *
 * @param {} readonly
 */
BaseParameterControl.prototype.getValue = function() {
}
/**
 *
 * @param {} value
 */
BaseParameterControl.prototype.setValue = function(value, alias) {
}
/**
 *
 * @param {} readonly
 */
BaseParameterControl.prototype.setReadOnly = function(readonly) {
}
/**
 *
 * @param {} value
 */
BaseParameterControl.prototype.setParameter = function(parameter) {
}
/**
 *
 * @param {} value
 */
BaseParameterControl.prototype.getParameter = function() {
}
/**
 *
 * @param {} value
 */
BaseParameterControl.prototype.setParameterDefine = function(parameterDefine) {
}
/**
 *
 */
*/
```

```
BaseParameterControl.prototype.getParameterDefine = function() {
}
/**
 *
 */
BaseParameterControl.prototype.setDefaultValue = function(defaultValue) {
}
/**
 *
 * @param {} needReget
 */
BaseParameterControl.prototype.getDefaultValue = function(needReget) {
}
/**
 *
 */
BaseParameterControl.prototype.setStandbyValues = function(standbyValues) {
}
/**
 *
 * @param {} needReget
 */
BaseParameterControl.prototype.getStandbyValues = function(needReget) {
}
/**
 *
 */
BaseParameterControl.prototype.destroy = function() {
}
```

BaseParameterControl.js源码: [BaseParameterControl.rar](#)

## 示例工程源码

以上所涉及到的示例工程源码为: [GDDX\\_DataSystem.rar](#)。  
插件工程部署请见: [插件开发框架](#)第4点, 引入附件工程源码。