

SuperMap iDesktopX

扩展开发指南


(v10.1.0)

北京超图软件股份有限公司

2020 年 09 月 北京

法律声明

本资料的版权为北京超图软件股份有限公司所有，受《中华人民共和国著作权法》和著作权国际公约的保护。未经北京超图软件股份有限公司书面许可，不得以任何方式或理由对该资料的任何部分进行使用、复制、修改、抄录、传播或其它产品捆绑使用、销售，侵权必究。

“超图”、“SuperMap” 以及  为北京超图软件股份有限公司的注册商标，受法律保护。未经北京超图软件股份有限公司书面许可，不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播或其它产品捆绑使用、销售，侵权必究。

本资料并不代表供应商及其代理的承诺，北京超图软件股份有限公司可在不作任何声明的情况下对本资料进行修改。

本资料中提到其他公司和产品的商标所有权为该公司所有。未经该权利人的书面同意，不得以任何方式或理由进行使用、复制、修改、抄录、传播。

本资料中所涉及的软件产品及其后续升级产品均由北京超图软件股份有限公司研发、销售。

特此声明。

北京超图软件股份有限公司

地址：北京市朝阳区酒仙桥北路 甲 10 号院电子城 IT 产业园 107 楼 6 层

邮编：100015

电话：+86-10-59896655

传真：+86-10-59896666

技术支持与客户监督热线：400-8900-866

技术支持电子邮箱：support@supermap.com

客户监督电子邮箱：cs@supermap.com

网址：<http://www.supermap.com>

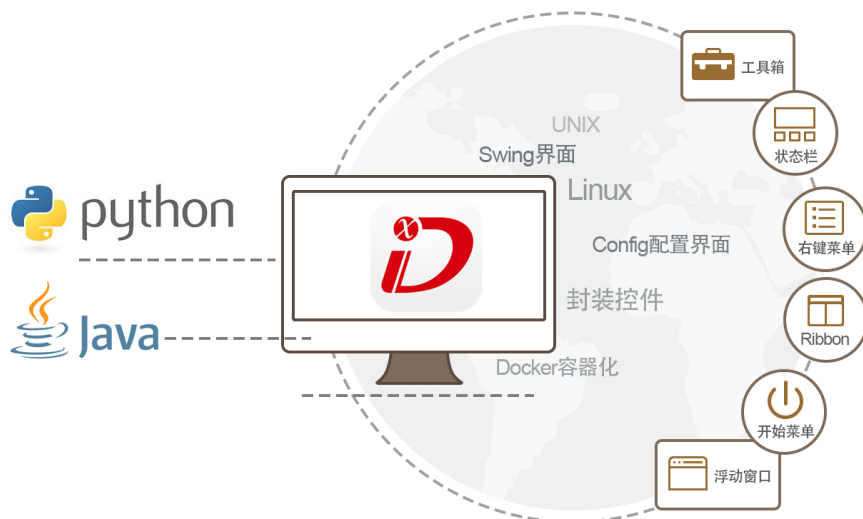
SuperMap 欢迎您的宝贵建议和意见。

前言

SuperMap iDesktopX 是超图研究院自 2015 年就推出的一款跨平台的桌面 GIS 软件。SuperMap iDesktopX 具有兼容 Windows 和 Linux，兼具最新 IT 技术和地理学技术的特点。经过四年多的发展和实践，SuperMap iDesktopX 越来越成熟，应用也越来越多，功能已经完全覆盖并超越了传统 GIS 桌面。同时，SuperMap iDesktopX 也采用的是插件式扩展开发框架，支持定制开发。

使用 SuperMap iDesktopX 定制开发有以下优势：

- 采用 Swing 图形界面框架，封装 60+GUI 控件，便于用户调用；
- 通过配置文件进行界面配置与管理，无须编程；
- 支持 Java、Python 两种语言扩展定制；
- 可定制开发界面功能及可视化的地理处理建模工具；
- 使用 IDEA 和 Maven 自动管理并部署依赖，实现一键部署。



目录

1 环境部署	6
1.1 软件安装	6
1.2 开发工程配置	6
2 工程模板	9
2.1 SampleCode Module	9
2.2 Develop module	9
2.2.1 MyStartUp	9
2.2.2 DevelopPlugin	9
2.2.3 CtrlActionSampleCode	9
2.2.4 配置 CtrlAction	10
2.2.5 工作流示例	10
3 配置文件	12
3.1 插件配置	12
3.2 界面配置	13
3.2.1 Ribbon 控件介绍	13
3.2.2 配置开始菜单<startMenu>	16
3.2.3 配置状态栏<statusbar>	17
3.2.4 配置右键菜单<contextMenu>	18
3.2.5 配置浮动窗口<dockbar>	19
3.2.6 配置工作流<Process >	20
3.3 属性信息	21
3.3.1 Plugin	21
3.3.2 Runtime	21
3.3.3 TaskBar	21
3.3.4 Ribbon	22
3.3.5 Group	22
3.3.6 Gallery	22
3.3.7 GalleryGroup	22
3.3.8 StartMenuPanel	23
3.3.9 StartMenu	23
3.3.10 SecondMenu	23

3.3.11	ProcessGroup.....	23
3.3.12	Process	24
3.3.13	Button.....	24
3.3.14	ButtonDropDown	25
3.3.15	ColorButton	25
3.3.16	GalleryButton	26
3.3.17	DropDownPanelButton	26
3.3.18	RibbonPopupMenuItem	27
3.3.19	Separator	27
3.3.20	RibbonPopupMenuSeparator	27
3.3.21	ComboBox/ ComboBoxFont	27
3.3.22	RibbonPopupMenuComboBox	28
3.3.23	Item	28
3.3.24	Label.....	28
3.3.25	Spinner	29
3.3.26	DockBar.....	29
3.3.27	ContentMenu	30
3.3.28	StatusBar.....	30
3.3.29	Textbox.....	30

1 环境部署

SuperMap iDesktopX 是通过 SuperMap iObjects Java 桌面核心库和 Java SE Development Kit (JDK) 构建的一款企业级、插件式、跨平台桌面 GIS 应用与开发平台,可基于 Java 语言在 Windows 或 Linux 系统中进行扩展开发。为了便于用户二次开发, SuperMap iDesktopX 提供了扩展开发实例模板工程 SampleCode ,只需要在配置文件中进行简单的 UI 配置,就可实现扩展开发。

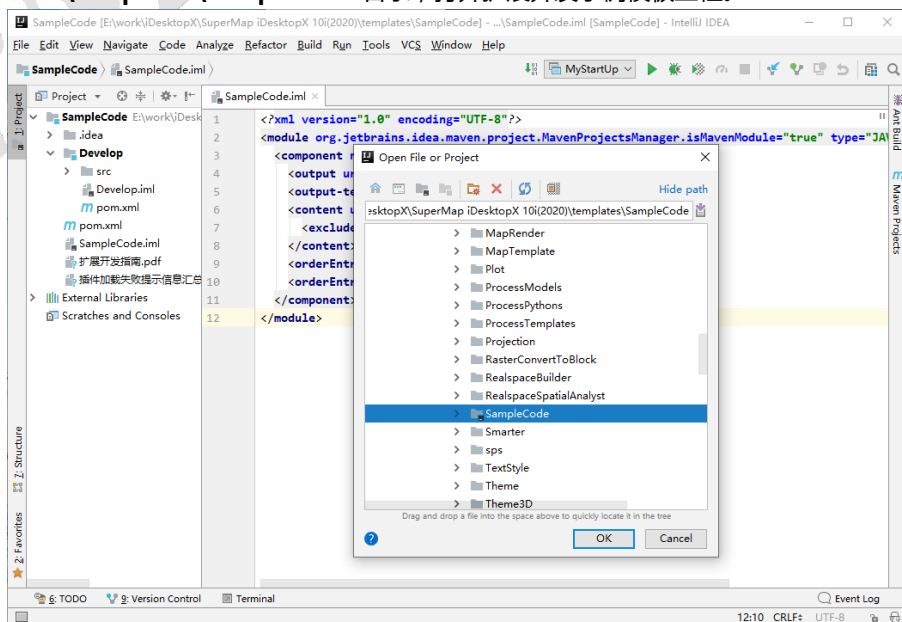
1.1 软件安装

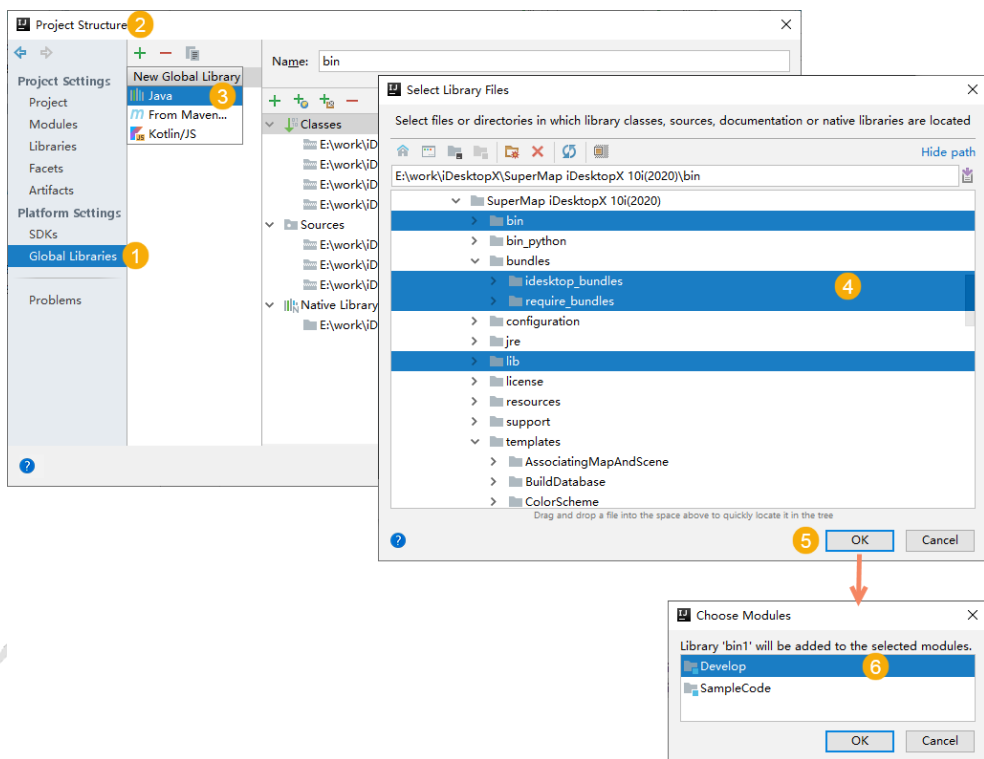
基于 SuperMap iDesktopX 扩展开发,需在计算机中安装 [JDK1.8](#) 和 Java 开发工具,开发工具推荐使用 [IDEA](#), 建议安装的 IDEA 为 2020.2.2 或更新版本。

1.2 开发工程配置

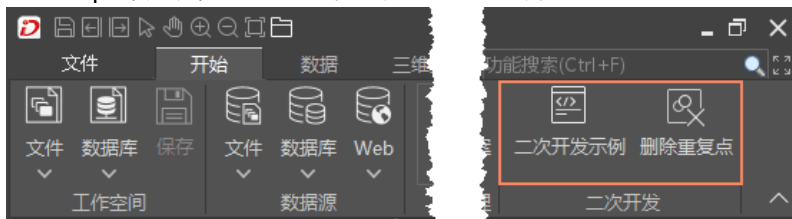
安装 IDEA 之后,需打开扩展开发的工程模板 SampleCode 并进行配置,详细说明如下:

1. 打开模板工程: IDEA->File->Open-> 选择 SuperMap iDesktopX 产品包 \templates\SampleCode 目录,打开扩展开发实例模板工程。





5. **编译**：单击工具栏中的 **Run** 按钮或者 **Debug** 按钮，编译后自动启动 SuperMap iDesktopX，启动后 Ribbon 中“**开始**”选项卡中会增加**二次开发**的分组，其中包含了**二次开发示例**、**删除重复点**两个功能按钮。此时，则表示环境配置成功，用户即可基于 SuperMap iDesktopX 进行扩展开发。**备注**：若未自动启动 SuperMap iDesktopX，建议将 IDEA 版本更新为 2020.2.2 或更新版本。



2 工程模板

示例模板工程即 SampleCode 工程，其中 Develop module 就是我们的开发 module。

2.1 SampleCode Module

SampleCode Module 主要通过此 module 的 pom 文件管理远程仓库地址及所有子 module（自定义插件），不实现任何具体功能。详情参见 SampleCode 的 pom.xml 文件。注意：jar 生成路径不可修改。

2.2 Develop module

Develop module 是扩展开发示例 module，内部主要的类有三个，分别为 MyStartUp、DevelopPlugin、CtrlActionSampleCode。

2.2.1 MyStartUp

MyStartUp 是二次开发启动的入口。

2.2.2 DevelopPlugin

DevelopPlugin 是**插件类**必须实现，其命名规则为 module name+ “Plugin”，必须继承实现 AbstractPlugin 类，否则 Develop module 所有功能都无法使用，具体实现方式参见 DevelopPlugin 类。

2.2.3 CtrlActionSampleCode

CtrlActionSampleCode 是示例开发类。CtrlActionSampleCode 类中包含了用于响应控件事件触发时所执行的内容，即与 UI 控件绑定的类，该类必须继承自 CtrlAction 类。具体实现方式参见 CtrlActionSampleCode 类。

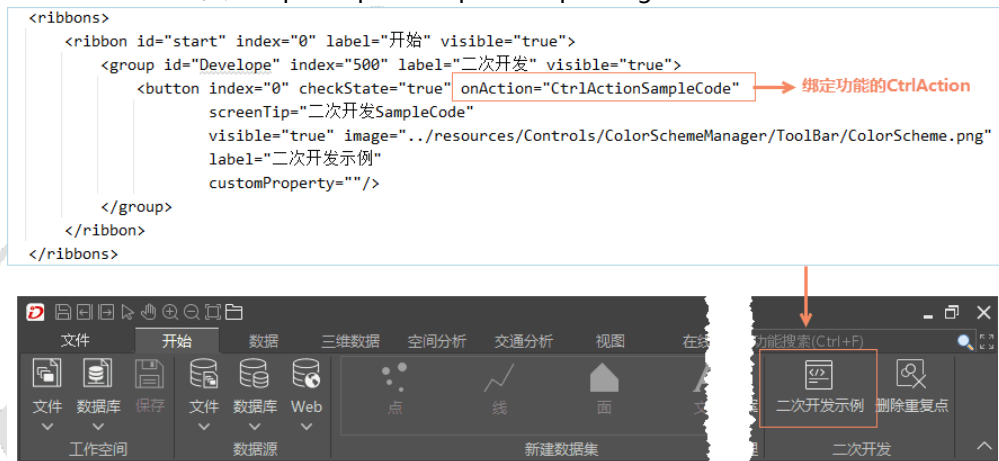
2.2.4 配置 CtrlAction

实现功能之后，就需要在相应配置文件中配置，跟 UI 绑定。即将功能对应的 CtrlAction 类名称写到 module 对应配置文件中相应控件的 onAction 属性中。

注意：一个 module 对应一个配置文件，配置文件必须存在且正确配置，否则无法加载对应 module。配置文件位于%模块目录%\src\main\resources\文件夹下。（自定义启动 module 不需要配置文件，只有界面功能需要配置文件）

当配置好配置文件后就可以通过 bat 文件或者 exe 启动桌面，查看效果。示例工程效果如下：右上角为配置控件，左下角输出信息为控件所绑定 CtrlAction 的事件运行结果。

示例配置文件参见 SuperMap.Desktop.Develop.config。



2.2.5 工作流示例

DesktopProcessSampleCode 类提供了一个 workflow 功能的完整实现代码。需要继承模型基类（**AbstractDesktopProcess** 类）来实现对应的方法，并注意以下五点：

1. 算子 name

算子构造函数第一句必须调用 **super** 方法，方法参数为算子 name，同一个包算子 name 不能重复，如下图所示：

```
super( name: "DeleteDuplicatePoints");
```

2. 构造输入参数及获取数据

以示例代码中为例讲述下如何构造输入参数，示例代码构造输入参数的语句为：

```
// 输入
```

```
this.datasetVectorISingleInput = init.defaultInput( name: "sourceDataset",DatasetVector.class);
```

- **sourceDataset** : 输入数据的 key , 一个功能可能有多个输入数据, 用于标识输入数据;
- **DatasetVector.class** : 限制输入的值类型, 本文示例是点数据集删除重复点, 所提输入的值类型是 DatasetVector.class。

获取输入数据需要调用相应的代码语句, 获取输入参数的数据的示例代码如下, 通过前面构造的输入参数对象变量调用 getValue 方法即可。

```
DatasetVector srcDataset = this.datasetVectorISingleInput.getValue();
```

3. 构造输出参数及输出数据

以示例代码中为例讲述下如何构造输出节点。示例代码构造输出节点的语句为:

```
// 输出
```

```
this.datasetVectorISingleOutput = init.defaultOutput( name: "resultDataset", DatasetVector.class);
```

- **resultDataset** : 是输出参数的 key , 标识输出参数, 因为一个功能可能有多个输出参数;
- **DatasetVector.class** : 限制输出的值类型, 本文示例是点数据集删除重复点, 所提输出的值类型是 DatasetVector.class。

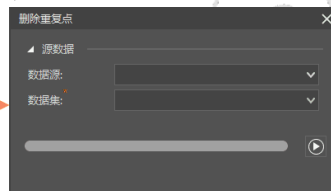
输出数据的需要调用相应的代码语句, 输出数据的示例代码如下, 通过前面构造的输出参数对象变量调用 setValue 方法即可。

```
this.datasetVectorISingleOutput.setValue(srcDataset);
```

4. 功能界面定制

SuperMap iDesktopX 封装了 70 余个控件, 如数据集控件、字段选择控件、单选框、复选框等, 用户在配置功能对话框界面时, 直接调用即可。对话框界面配置的示例代码如下:

```
<ProcessManager>
  <ProcessGroup id="WorkflowSampleCode" index="10000" title="工作流示例">
    <Process>
      <ClassName>com.supermap.desktop.develop.gp.DesktopProcessSampleCode</ClassName>
      <Title>删除重复点</Title>
      <Key>iDesktop_Factory-com.supermap.desktop.develop.gp.DeleteDuplicatePoints</Key>
    </Process>
  </ProcessGroup>
</ProcessManager>
```



5. childExecute 方法

执行功能的代码放在 childExecute 方法中, 且以 boolean 类型返回执行结果。当执行失败是返回 false, 执行成功时返回 true。

3 配置文件

3.1 插件配置

SuperMap iDesktopX 的配置文件都采用标准的 XML 语法编写，插件配置内容包含在 `<plugin>...</plugin>` 标签之间，如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<plugin xmlns="http://www.supermap.com.cn/desktop/config" name="Develop" author="SuperMap"
    url="www.supermap.com.cn" description="Develop Plugin">
    <runtime packageName="Develop.jar" pluginClassName="com.supermap.desktop.develop.DevelopPlugin" enabled="True" level="5"/>
    <startMenus>
    </startMenus>
    <ribbons>
        <ribbon id="start" index="0" label="开始" visible="true">
            <group id="Develope" index="500" label="二次开发" visible="true">
                <button index="0" checkState="true" onAction="CtrlActionSampleCode"
                    screenTip="二次开发SampleCode"
                    visible="true" image="../../resources/Controls/ColorSchemeManager/ToolBar/ColorScheme.png"
                    label="二次开发示例"
                    customProperty="" />
            </group>
        </ribbon>
    </ribbons>
</plugin>
```

其中：`<?xml version="1.0" encoding="utf-8"?>` 是 XML 文件的文件头，用以说明版本信息和采用的字符编码方式，下面详细介绍 `<plugin>...</plugin>` 标签的各个属性的含义与作用。

`xmlns` 为插件配置文件的命名空间，值为：`http://www.supermap.com/desktop`。

- **Name**：插件的名称。
- **Author**：插件开发者的相关信息。
- **url**：插件开发者可以提供一个 URL，供使用者访问或了解相关信息。
- **description**：插件的描述信息。
- **<runtime>**：标签用于提供插件运行库的信息，其中：
 - **packageName**：实现插件的 jar 文件名称（module 名称），该属性的值必须正确设置。
 - **pluginClassName**：实现“插件类”名称，该类从 `AbstractPlugin` 类继承。这里给定的实现插件的类名称需是全名，即需要指定其 package。该属性的值必须正确设置。
 - **enabled**：用于控制是否加载该插件。True 表示加载，可以使用该插件；false 表示不加载，禁用该插件。

3.2 界面配置

为使用户避免书写大量的代码来实现应用系统的界面构建，而是专注于业务功能的实现，SuperMap iDesktopX 提供了界面配置管理的方式，通过配置文件实现界面的布局和构建，并且界面风格采用 Ribbon 模式，即 Microsoft Office 2007 风格的界面；界面元素所绑定的功能也是在配置界面元素时指定的。

SuperMap iDesktopX 为每一个界面元素提供一个标签，向界面中添加界面元素时，只需在配置文件中完成该界面元素对应的标签的添加和设置即可，当添加多个界面元素时，就添加多个界面元素的标签，并对标签的属性进行设置即可，因此，这样可以很方便地将相应的界面元素，如按钮、文本框、组合框以及更为复杂的元素添加到 SuperMap iDesktopX 提供的基础框架中，并且各个界面元素所实现的功能也是在配置文件中指定。

SuperMap iDesktopX 采用的界面风格 Ribbon 风格，即所熟悉的 Microsoft Office 2007 的界面风格，下面详细介绍如何通过配置文件来构建界面，添加界面元素，并指定界面元素的行为（onAction）。

插件的界面配置主要分为功能区（Ribbon）配置、状态栏配置（statusbar）、开始菜单（startMenu）配置、右键菜单配置（contextMenu）和浮动窗口（dockbar）配置，这几类界面配置内容必须放置在插件配置文件中的 <plugin>...</plugin> 标签内，并且这几类界面配置处于相同级别，它们之间不存在包含与被包含关系，而是并列关系。

3.2.1 Ribbon 控件介绍

功能区上所承载的各类控件为 Ribbon 控件，Ribbon 风格界面的功能区上只能放置 Ribbon 控件，并且 Ribbon 控件只能放置在功能区上，下面详细介绍应用程序中所使用的所有 Ribbon 控件。

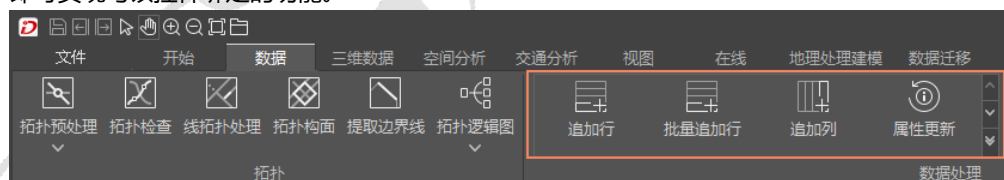
3.2.1.1 按钮控件（Button）

如下图所示为应用程序中的按钮控件，按钮上的显示内容分为两个部分，按钮上的图片为按钮的显示图标，而按钮上的文字内容为按钮的显示名称。通过点击按钮即可实现与该按钮绑定的功能。



3.2.1.2 GalleryButton 控件 (GalleryButton)

GalleryButton 控件只能放置在功能区中名为 Gallery 的容器控件中，如下图所示，GalleryButton 控件上的显示内容分为两个部分，控件上的图片为显示图标，而控件上的文字内容为控件的显示名称。GalleryButton 控件类似于按钮控件，通过点击 GalleryButton 控件即可实现与该控件绑定的功能。



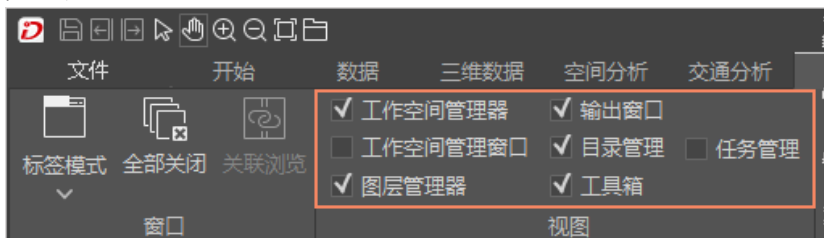
3.2.1.3 下拉控件 (RibbonPopupMenu)

如下图所示为一个下拉按钮控件，下拉按钮分为两个部分：一是按钮部分，点击该部分可以直接执行相应的功能，二是下拉按钮部分，点击该部分将弹出下拉菜单，通过选择下拉菜单中的项 来进一步实现相应的功能。下拉按钮的按钮部分显示下拉按钮的图标，下拉按钮的下拉按钮部分显示了下拉按钮的显示名称。



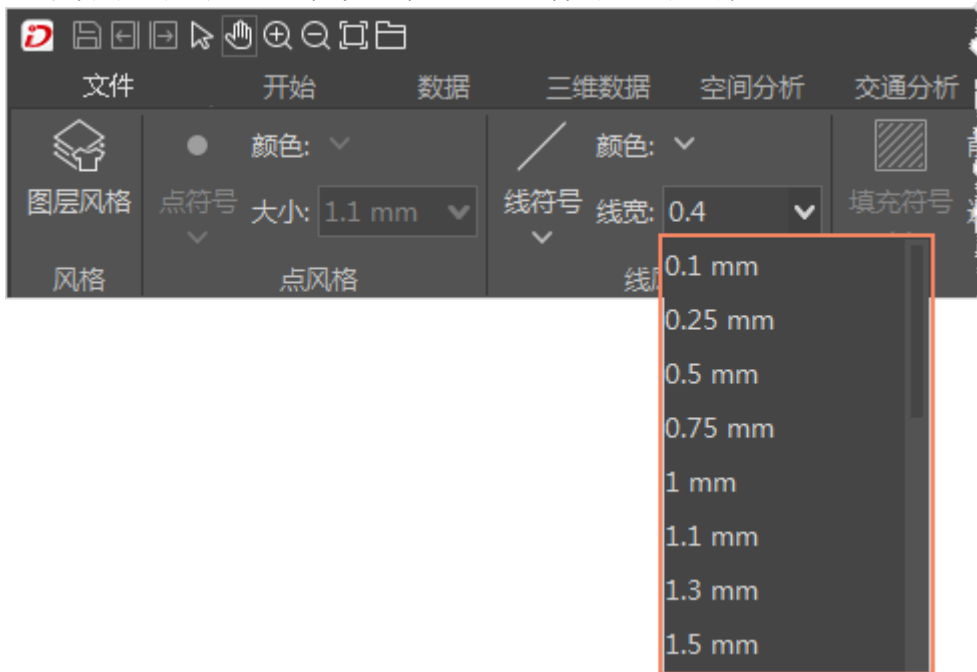
3.2.1.4 复选框控件 (CheckBox)

如下图所示为复选框控件，复选框控件上的文字内容为复选框控件的显示名称，用户通过选中和不选中复选框控件来与应用程序进行交互，应用程序通过判断复选框是否被选中来进行相应的操作的处理。



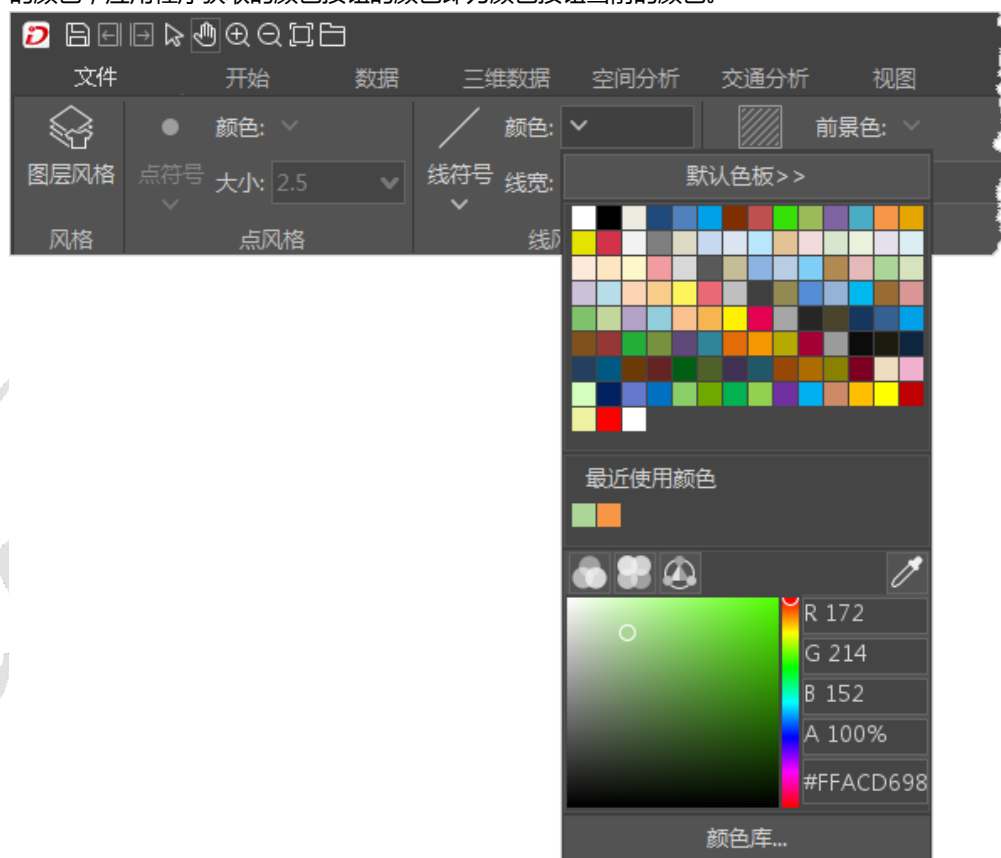
3.2.1.5 组合框控件 (ComboBox)

如下图所示为一个组合框控件，组合框控件由一个文本框和一个下拉列表组成，下拉列表中包含一系列子项，通常情况下，用户可以在文本框部分输入内容，也可以在下拉列表中选择某个项，应用程序会根据组合框中文本框里显示的内容来处理相应的操作。



3.2.1.6 颜色按钮控件 (ColorButton)

下图为颜色按钮控件，颜色按钮上显示的颜色为颜色按钮当前的颜色，即用户选择的颜色，点击颜色按钮右侧下拉按钮可以弹出颜色面板，用户可以选择需要的颜色，如果颜色列表中的颜色块不足以满足用户的需要，可以点击颜色面板最下方的“颜色库”按钮，通过弹出的颜色对话框可获得更多的颜色，用户选择颜色后，颜色按钮当前的颜色会变化为用户选择的颜色，应用程序获取的颜色按钮的颜色即为颜色按钮当前的颜色。



3.2.2 配置开始菜单 <startMenu>

应用程序用户界面的最左上角的按钮为“开始菜单”按钮，点击“开始菜单”按钮后会弹出如下图所示的开始菜单。在开始菜单上可以放置 Ribbon 控件，只需在开始菜单所对应的

<startMenu>...</startMenu> 标签中配置相关控件对应的标签即可。左侧是开始菜单，右侧是当前开始菜单的具体内容。



在配置文件中，文件菜单的配置内容必须放置在 <startMenu>...</startMenu> 或 <startMenuPanel>...</startMenuPanel> 标签中。详情可参考桌面配置文件。

3.2.3配置状态栏<statusbar>

状态栏 (statusbar) 配置，应用程序中的主窗口、子窗口都可以有自己的状态栏，并且每个窗口只能有一个状态栏，状态栏的配置通过 <statusbar>...</statusbar> 标签来配置，状态栏上可以放置各种 Ribbon 控件，向状态栏中添加 Ribbon 控件只需在 <statusbar>...</statusbar> 标签之间配置相应控件的标签即可。

```

<statusbars>
<statusbar formClass="com.supermap.desktop.mapview.FormMap" id="Map" visible="true">
<label index="1" label="选择个数:" visible="true" width="120"/>
<label index="2" label="比例尺:" visible="true" width="0"/>
<label index="3" label="比例尺:" visible="true" width="0"/>
<comboBox customProperty="" helpURL="" id="Scale" images="" index="4" label="比例尺" screenTip="" onAction="CtrlActionMapScale" screenTipImage="" visible="true" width="180"/>
<textBox index="5" label="经度纬度" onAction="" readOnly="true" screenTip="" screenTipImage="" text="经度, ***, 纬度, ***" visible="true" width="0"/>
<textBox index="6" label="坐标系" onAction="" readOnly="true" screenTip="" screenTipImage="" text="坐标系, GCS_WGS_1984" visible="true" width="0"/>
<label index="7" label="中心点:" visible="true" width="0"/>
<textBox index="8" label="X" onAction="" readOnly="true" screenTip="" screenTipImage="" text="0" visible="true" width="0"/>
<textBox index="9" label="Y" onAction="" readOnly="true" screenTip="" screenTipImage="" text="0" visible="true" width="0"/>
</statusbar>
</statusbars>

```

`<statusbar>...</statusbar>` 标签的配置内容要放置在配置文件中的 `<statusbars>...</statusbars>` 标签之间,即 `<statusbars></statusbars>` 标签表示该插件配置文件中所有状态栏的集合,下面详细介绍 `<statusbar>...</statusbar>` 标签的各个属性的含义与作用。

- **Visible** : 指定状态栏是否可见,该属性的值为 `true` 时,表示可见, `false` 为不可见。
- **formClass** : 指定状态栏所绑定的窗体,即该状态栏属于那种类型的窗口,设置该属性后,相应类型的窗口将出现状态栏。

状态栏中可以放置 Ribbon 控件,配置 Ribbon 控件的标签都组织在 `<statusbar></statusbar>` 之间。下面详细介绍 Ribbon 控件标签的各个属性的含义与作用。

- **index** : 用于排序控件,即当状态栏中存在多个控件时,每个控件将通过该属性的值来确定其排列次序。
- **label** : 控件的标题。
- **onAction** : 控件绑定的事件。
- **visible** : 指定用户自定义控件的可见性,该属性的值为 `true` 时,表示可见, `false` 为不可见。
- **screenTip** : 指定鼠标停留在用户自定义控件上时所显示的提示信息。

3.2.4 配置右键菜单 <contextMenu>

SuperMap iDesktop 应用程序界面中右键菜单的配置可以通过 `<contextMenu>...</contextMenu>` 标签来完成,一个标签对应一个右键菜单,一个配置文件中的右键菜单的配置要放置在 `<contextMenus>...</contextMenus>` 标签之间,下面详细介绍 `<contextMenu>...</contextMenu>` 标签的各个属性的含义与作用。

```

<contextMenu id="SuperMap.Desktop.II.WorkspaceControlManager.ContextMenuDatasetVector" index="1" label="工作空间管理-矢量数据集">
<group id="AddDatasetToMap" index="1" label="添加到地图" visible="true">
<button checkState="false" image="...resources/MapView/Menu/Dataset/AddToMap.png" index="0" label="添加到地图" onAction="CtrlActionDatasetAddToMap" screenTip="" screenTipImage="" shortcutKey="" visible="true"/>
<button checkState="false" image="...resources/MapView/Menu/Dataset/AddToCurrentMap.png" index="1" label="添加到当前地图" onAction="CtrlActionDatasetAddToCurrentMap" screenTip="" screenTipImage="" shortcutKey="" visible="true"/>
</group>
</contextMenu>

```

右键菜单中的配置项与功能区 (Ribbon) 中的配置项共用一套相同的配置标签, 但是, 这里暂时支持 group、button、buttonDropdown 配置。 <group>...</group> 标签中所包含的项目为一个分组, 如果右键菜单中配置有多个分组, 那么会在各个分组之间自动添加一个分隔条 (seperator)。 <group>...</group> 标签的各个属性的含义与作用如下:

- **Index**: 用于排序分组 (group), 如果右键菜单中存在多个分组 (group) 时, 每个分组 (group) 将通过该属性的值来确定其排列次序。
- **Id**: 组 (group) 支持多个不同配置文件里的项进行合并显示, 合并的依据就是通过 id 来实现的, 也就是说我们可以在 A 插件的配置文件中配置一个右键菜单的分组 (group), 指定一个 id, 同时在 B 插件里面配置同一个右键菜单的另一个分组 (group), 但指定相同的 id, 在系统显示时, 这个右键菜单中的两个分组 (group) 中的项将会合并到一起, 放到一个右键菜单的分组 (group) 上显示。
- **Visible**: 指定该分组 (group) 中的内容是否可见, 该属性的值为 true 时, 表示可见, false 为不可见。
- **Label**: 控件标题。

3.2.5 配置浮动窗口 <dockbar>

浮动窗口的相关配置内容必须放置在 <dockbar>...</dockbar> 标签中。

```
<dockbars>
<dockbar title="输出窗口" id="outputFrame" visible="true" dockDirection="bottom" dockState="minimized"
    component="com.supermap.desktop.controls.ui.output.OutputFrame"/>
<dockbar title="工作空间管理器" id="workspaceComponentManager" visible="true" dockDirection="leftTop"
    dockState="normal" component="com.supermap.desktop.controls.ui.WorkspaceComponentManager"/>
<dockbar title="图层管理器" id="layersComponentManager" visible="true" dockDirection="leftBottom" dockState="normal"
    component="com.supermap.desktop.controls.ui.LayersComponentManager"/>
<dockbar title="属性" id="dataPropertyContainer" visible="false" dockDirection="right" dockState="normal"
    component="com.supermap.desktop.controls.property.DataPropertyContainer"/>
</dockbars>
```

每个浮动窗口在配置文件中对应一个 < dockbar >...</ dockbar > 标签, 向界面中添浮动窗口, 只需添加 < dockbar >...</ dockbar > 标签, 并对标签的属性进行相应的设置即可完成界面中浮动窗口添加, 要添加多个浮动窗口, 就相应的添加多个 < dockbar >...</ dockbar > 标签, 下面详细介绍 < dockbar >...</ dockbar > 标签的各个属性的含义与作用:

- **Title**: 浮动窗口的标题。
- **dockState**: 指定浮动窗口的停靠模式, 该属性的值有四种: normal、minimized、maximized、float, 默认 normal。
- **dockDirection**: 指定浮动窗口停靠时的停靠位置, 即相对于主窗口的位置, 该属性的值有四种: left、right、bottom、top, 分别表示相对于主窗口的左、右、底

部、顶部位置。

- **visible**：指定浮动窗口是否可见，该属性的值为 true 时，表示可见，false 为不可见。
- **component**：嵌入在浮动窗口中的控件类，即 dockBar 所绑定的类所在的程序文件的名称，该属性的值需正确设置。

3.2.6 配置 workflow <Process>

3.2.6.1 配置到工具箱

workflow 配置到工具箱的相关配置需放置在 < ProcessGroup >...< /ProcessGroup> 标签中，而 ProcessGroup 需放置在 < ProcessManager >...< /ProcessManager> 标签中。如下图所示：

```
<ProcessManager>
  <ProcessGroup id="WorkflowSampleCode" index="10000" title="工作流示例">
    <Process>
      <ClassName>com.supermap.desktop.develop.gp.DesktopProcessSampleCode</ClassName>
      <Title>删除重复点</Title>
      <Key>iDesktop_Factory-com.supermap.desktop.develop.gp-DeleteDuplicatePoints</Key>
    </Process>
  </ProcessGroup>
</ProcessManager>
```

具体示例参见：

SampleCode\Develop\target\classes\SuperMap.Desktop.Develope.config 文件。每个节点属性参见属性信息 3.3.12 跟 3.3.13。

3.2.6.2 配置到 Ribbon

workflow 中的功能是支持同步配置到 Ribbon 上，有一点需要注意，默认是不需要配置 onAction 属性，会通过默认的 CtrlActionProcessBase 进行加载，如果 CtrlActionProcessBase 不能满足需求，那么可继承 CtrlActionProcessBase 类并将类名配置到 onAction 属性。示例中是配置成 button 控件，因此本文就以 workflow 配置成 Ribbon button 讲解下如何配置到 Ribbon。

Button 控件时必须配置在 <group>...</group> 标签中，而 group 必须配置在 <Ribbon>...</Ribbon> 标签中等等。如下图所示：

```
<?xml version="1.0" encoding="utf-8"?>
<plugin xmlns="http://www.supermap.com.cn/desktop/config" name="Develop" author="SuperMap"
    url="www.supermap.com.cn" description="Develop Plugin">
    <runtime packageName="Develop.jar" pluginClassName="com.supermap.desktop.develop.DevelopPlugin" enabled="True" level="5"/>
    <startMenus>
    </startMenus>
    <ribbons>
        <ribbon id="start" index="0" label="开始" visible="true">
            <group id="Develope" index="500" label="二次开发" visible="true">
                <button index="0" checkState="true" onAction="CtrlActionSampleCode"
                    screenTip="二次开发SampleCode"
                    visible="true" image="../resources/Controls/ColorSchemeManager/ToolBar/ColorScheme.png"
                    label="二次开发示例"
                    customProperty=""/>
                <button className="com.supermap.desktop.develop.gp.DesktopProcessSampleCode"
                    Key="iDesktop_Factory-com.supermap.desktop.develop.gp.DeleteDuplicatePoints"
                    checkState="false"
                    image="..\resources\SpatialAnalyst\Icon\Dataprocess\RegionToTrunkLine.png"
                    label="删除重复点" onAction=""
                    screenTip=""
                    screenTipImage="" shortcutKey="" visible="true"/>
            </group>
        </ribbon>
    </ribbons>
</plugin>
```

具体示例参见：
SampleCode\Develop\target\classes\SuperMap.Desktop.Develope.config 文件。每个节点属性参见属性信息 3.3.5 跟 3.3.14。

3.3 属性信息

3.3.1 Plugin

name	插件名称
author	作者名称
description	插件描述

3.3.2 Runtime

packageName	插件生成的 jar 名称
pluginClassName	插件类完整路径，大小写必须跟类名保持一致
enabled	插件是否可用，True 代表插件可用，False 代表不可用

3.3.3 TaskBar

id	唯一标识，不同配置文件中的 taskBar 根据 id 合并为一个 taskBar
----	---

index	索引，控制加载顺序
-------	-----------

3.3.4 Ribbon

id	唯一标识，不同配置文件中的 ribbon 根据 id 合并为一个 ribbon
index	索引，控制加载顺序
label	显示的文本信息
visible	控件是否可见，true 可见，false 不可见
formClass	Ribbon 绑定的窗口，当打开对应的窗口，窗口所绑定的 Ribbon 才会显示，默认是不显示

3.3.5 Group

id	唯一标识，不同配置文件中的 group 根据 id 合并为一个 group
index	索引，控制加载顺序
label	显示的文本信息
visible	控件是否可见，true 可见，false 不可见
image	显示的图片信息
rowCount	控制 flowGroup 的布局是几行布局，范围[1,3]

3.3.6 Gallery

id	唯一标识，不同配置文件中的 gallery 根据 id 合并为一个 gallery
index	索引，控制加载顺序
label	显示的文本信息
popupColumn	控制 Gallery 中一行最多显示控件的数量，大于 popupColumn 则换行显示

3.3.7 GalleryGroup

id	唯一标识，不同配置文件中的 galleryGroup 根据 id 合并为一个 galleryGroup
index	索引，控制加载顺序

label	显示的文本信息
maxLength	控制最大显示文本长度，当大于该长度时，会做字符串截取

3.3.8 StartMenuPanel

id	唯一标识
index	索引，控制加载顺序
Label	显示的文本信息
component	实现的 StartMenuPanel 类的完整路径
visible	控件是否可见，true 可见，false 不可见

3.3.9 StartMenu

id	唯一标识
index	索引，控制加载顺序
Label	显示的文本信息
onAction	CtrlAction 名称
visible	控件是否可见，true 可见，false 不可见

3.3.10 SecondMenu

index	索引，控制加载顺序
Label	显示的文本信息

3.3.11 ProcessGroup

id	唯一标识，不同配置文件中的 processGroup 根据 id 合并为一个 processGroup
index	索引，控制加载顺序
title	显示的文本信息
Platform	支持的操作系统平台，分别是 windows、Linux、All，windows 代表只支持 windows 平台，linux 代表 只支持 linux 平台，All 代表所有平台都支持，默认值是 All

ScreenTip	详细描述信息
-----------	--------

3.3.12 Process

ClassName	实现类的完整路径
loaderClass	加载实现类的加载器实现类完整路径,默认不需要写,此时会调用默认加载器加载实现类
index	索引,控制加载顺序
title	显示的文本信息
key	Process 的唯一标识,不能重复,格式为 iDesktop_Factory-算子所在包-算子 name
ScreenTip	Process 文本描述信息
Platform	支持的操作系统平台,分别是 windows、Linux、All,windows 代表只支持 windows 平台,linux 代表 只支持 linux 平台,All 代表所有平台都支持,默认值是 All

3.3.13 Button

checkState	控件是否选中, true 表示选中控件, false 表示未选中控件
customProperty	用户自定义属性信息
image	显示的图片信息
index	索引,控制加载顺序
label	显示的文本信息
onAction	CtrlAction 名称
screenTip	文本描述信息
screenTipImage	图片描述信息
shortcutKey	功能响应的快捷键
visible	控件是否可见, true 可见, false 不可见
id	唯一标识,不同配置文件中的 button 根据 id 合并为一个 button
style	控件尺寸, BIG, MEDIUM, SMALL 分别代表图片显示在上文本显示在下、图片跟文本同行显示、只有图片没有文本显示

Platform	支持的操作系统平台, 分别是 windows、Linux、All, windows 代表只支持 windows 平台, linux 代表 只支持 linux 平台, All 代表所有平台都支持, 默认值是 All
ClassName	针对 workflow 功能。功能实现类的完整路径
key	针对 workflow 功能。Process 的唯一标识, 不能重复, 格式为 iDesktop_Factory-算子所在包-算子 name
directLoadCtrlAction	是否直接加载 CtrlAction , 如果为 false 则会在桌面启动之后再加载 CtrlAction , 默认为 false

3.3.14 ButtonDropDown

id	唯一标识, 不同配置文件中的 ButtonDropDown 根据 id 合并为一个 ButtonDropDown
image	显示的图片信息
index	索引, 控制加载顺序
label	显示的文本信息
visible	控件是否可见, true 可见, false 不可见

3.3.15 ColorButton

checkState	控件是否选中, true 选中控件, false 不选中控件
customProperty	用户自定义属性信息
image	显示的图片信息
index	索引, 控制加载顺序
label	显示的文本信息
onAction	CtrlAction 名称
screenTip	文本描述信息
screenTipImage	图片描述信息
shortcutKey	功能响应的快捷键
visible	控件是否可见, true 可见, false 不可见
style	控件尺寸, BIG, MEDIUM, SMALL 分别代表图片显示在上文本显示在下、图片跟文本同行显示、只有图片没有文本显示

3.3.16 GalleryButton

checkState	控件是否选中，true 选中控件，false 不选中控件
customProperty	用户自定义属性信息
image	控件显示的图片信息
index	索引，控制加载顺序
label	显示的文本信息
onAction	CtrlAction 名称
screenTip	文本描述信息
screenTipImage	图片描述信息
shortcutKey	功能响应的快捷键
visible	控件是否可见，true 可见，false 不可见
ClassName	针对 workflow 功能。功能实现类的完整路径
key	针对 workflow 功能。Process 的唯一标识，不能重复，格式为 iDesktop_Factory-算子所在包-算子 name

3.3.17 DropDownPanelButton

checkState	控件是否选中，true 选中控件，false 不选中控件
customProperty	用户自定义属性信息
image	控件显示的图片信息
index	索引，控制加载顺序
label	显示的文本信息
onAction	CtrlAction 名称
screenTip	文本描述信息
screenTipImage	图片描述信息
shortcutKey	功能响应的快捷键
visible	控件是否可见，true 可见，false 不可见
style	控件尺寸，BIG，MEIDUM，SMALL 分别代表图片显示在上文本显示在下、图片跟文本同行显示、只有图片没有文本显示

3.3.18 RibbonPopupMenuItem

checkState	控件是否选中，true 选中控件，false 不选中控件
customProperty	用户自定义属性信息
image	控件显示的图片信息
index	索引，控制加载顺序
label	显示的文本信息
onAction	CtrlAction 名称
screenTip	文本描述信息
screenTipImage	图片描述信息
shortcutKey	功能响应的快捷键
visible	控件是否可见，true 可见，false 不可见
directLoadCtrlAction	是否直接加载 CtrlAction ,如果为 false 则会在桌面启动之后再加载 CtrlAction，默认为 false

3.3.19 Separator

index	索引，控制加载顺序
-------	-----------

3.3.20 RibbonPopupMenuSeparator

index	索引，控制加载顺序
label	显示的文本信息
customProperty	用户自定义属性信息

3.3.21 ComboBox/ ComboBoxFont

customProperty	用户自定义属性信息
index	索引，控制加载顺序
label	显示的文本信息
width	控件宽度
onAction	CtrlAction 名称
editable	控制控件是否可编辑，true 可编辑，false 不可编辑

screenTipImage	图片描述信息
shortcutKey	功能响应的快捷键
visible	控件是否可见，true 可见，false 不可见

3.3.22 RibbonPopupMenuComboBox

checkState	控件是否选中，true 选中控件，false 不选中控件
customProperty	用户自定义属性信息
image	显示的图片信息
index	索引，控制加载顺序
label	显示的文本信息
onAction	CtrlAction 名称
screenTip	文本描述信息
screenTipImage	图片描述信息
shortcutKey	功能响应的快捷键
visible	控件是否可见，true 可见，false 不可见
style	控件尺寸，BIG，MEDIUM，SMALL 分别代表图片显示在上文本显示在下、图片跟文本同行显示、只有图片没有文本显示

3.3.23 Item

index	索引，控制加载顺序
label	显示的文本信息
data	存储的值
image	显示的图片信息
customProperty	用户自定义属性信息

3.3.24 Label

customProperty	用户自定义属性信息
index	索引，控制加载顺序
label	显示的文本信息
width	控件宽度

screenTip	文本描述信息
screenTipImage	图片描述信息
visible	控件是否可见，true 可见，false 不可见

3.3.25 Spinner

customProperty	用户自定义属性信息
index	索引，控制加载顺序
label	显示的文本信息
width	控件宽度
onAction	CtrlAction 名称
minValue	值域范围最小值
maxValue	值域范围最大值
increment	每一步增加或减少的值
editable	控制控件是否可编辑，true 可编辑，false 不可编辑
screenTipImage	图片描述信息
shortcutKey	功能响应的快捷键
visible	控件是否可见，true 可见，false 不可见

3.3.26 DockBar

component	Dockbar 类的完整路径
dockDirection	Dockbar 显示位置，共有 left、leftTop、leftBottom、bottom、right 三种情况，分别代表 Dockbar 显示在桌面左边、左上、左下、下边、右边位置
id	唯一标识
title	Dockbar 标题
visible	控件是否可见，true 可见，false 不可见
dockState	DockBar 初始显示状态。共有 normal、minimized、maximized、float，分别代表普通、最小、最大、浮动。

3.3.27 ContenMenu

id	唯一标识，不可重复
label	右键菜单名称

3.3.28 StatusBar

id	唯一标识 不同配置文件中的 statusBar 根据 id 合并为一个 statusBar
visible	控件是否可见，true 可见，false 不可见
formClass	statusBar 绑定的窗口，当打开对应的窗口，窗口所绑定的 statusBar 才会显示在窗口底部，默认是不显示

3.3.29 Textbox

customProperty	用户自定义属性信息
index	索引，控制加载顺序
label	显示的文本信息
text	Textbox 显示的文本内容
width	控件宽度
visible	控件是否可见，true 可见，false 不可见
onAction	CtrlAction 名称