

# 配置主题

在皮肤部件 ([../../extension/EUI/skin/part/index.html](#))一节中介绍了如何将皮肤设置到逻辑组件上。但如果有一个全局通用的默认皮肤，我们需要每次实例化组件时都赋值一次皮肤，这样比较麻烦。因此我们提供了一个主题机制，能够让您为指定的组件配置默认皮肤，全局指定一次即可，之后直接实例化组件，不再需要显式设置组件的skinName属性。

## 配置主题

下面是一个主题配置文件的例子：

default.thm.json:

```
{
  "skins": {
    "eui.Button": "skins/ButtonSkin.xml",
    "eui.CheckBox": "skins/CheckBoxSkin.xml",
    "eui.HScrollBar": "skins/HScrollBarSkin.xml",
    "eui.HSlider": "skins/HSliderSkin.xml",
    "eui.Panel": "skins/PanelSkin.xml",
    "eui.ProgressBar": "skins/ProgressBarSkin.xml",
    "eui.RadioButton": "skins/RadioButtonSkin.xml",
    "eui.Scroller": "skins/ScrollerSkin.xml",
    "eui.ToggleSwitch": "skins/ToggleSwitchSkin.xml",
    "eui.VScrollBar": "skins/VScrollBarSkin.xml",
    "eui.VSlider": "skins/VSliderSkin.xml"
  },
  "exmls": [ ],
  "autoGenerateExmlsList": true
}
```

主题配置文件就是一个标准的JSON文件，

- `skins` 指定组件的默认皮肤，其中键是组件的类名，值是需要赋值给这个组件skinName属性的值。可以是exml文件路径，也可以是EXML文件上注册的类名（根节点上的class属性）。
- `exmls` 表示需要主题预加载的 EXML 文件列表。Theme 文件加载之后，它会优先加载这个列表中的 EXML文件，由于 EXML 可能会存在相互依赖，所以 Theme 会**按照列表中的顺序**编译 EXML。可以监听 `egret.Event.COMPLETE` 来确认该列表中的EXML已经加载完成。

- `autoGenerateExmlsList` 表示是否需要使用命令行工具自动生成 EXML 列表。`build` 命令会查找项目中的 EXML 文件，并根据 EXML 文件的依赖关系自动排序后，放到这个列表中。在项目发布时，命令行工具会将 EXML 文件的内容直接集成到 theme 文件中，减少网络请求的次数（参考下面的示例）。如果您想要自定义 EXML 列表，请将值设置为 `false`。

```
//发布后的主题文件
{
  "skins": {
    "eui.Button": "skins/ButtonSkin.exml"
  },
  "exmls": [{
    "path": "skins/ButtonSkin.exml",
    "content": "<?xml version='1.0' encoding='utf-8' ?>...</e:Skin>"
  }]
}
```

这里需要注意的是，引擎只会识别 `xxx.thm.json` 文件作为 theme 文件，其他格式的文件名不会自动生成。

## EXML 设置版本号

通常我们希望更新版本的时候避免被浏览器缓存，现在可以通过设置 EXML 版本号的方式来实现。

```
"exmls": [
  "resource/eui_skins/ButtonSkin.exml?v=20151211"
]
```

比如上面代码给 `ButtonSkin` 的设置了一个版本号,通过在 `default.thm.json` 配置加载 EXML 文件时加入后缀 `?=20151211` 来设置版本号。

## 启用主题

启用主题时，在项目初始化时调用一句代码即可：

```
class Main extends egret.Sprite {  
    public constructor(){  
        this.once(egret.Event.ADDED_TO_STAGE, this.onAddedToStage, this);  
    }  
    public onAddedToStage(event:egret.Event):void{  
        new eui.Theme("resource/default.thm.json", this.stage);  
    }  
}
```

创建了Theme之后，它会开始异步加载指定的主题文件并解析，在加载的过程中，如果已经有组件在创建，也不需要额外处理，这部分组件在主题加载完成后会自动重新查询一次默认皮肤。

特别注意，主题配置文件只是起到设置默认值的作用，并不能运行时切换所有默认皮肤。因为这么做需要遍历整个显示列表，开销较大。