

更换产品显示方式_扩展包方式

看本文之前，建议了解如何查找定位对应图片、css或js，便于扩展本文中
没有涉及到的修改，见[Smartbi前端框架介绍及调试定位入门](#)。

在客户的生产环境中部署 Smartbi 程序时，通常需要对对其做一些简单的客户化定制，比如替换登录界面、以及主界面顶部的LOGO和图片、或者修改程序名称标识等。如果您遇到类似需求，请参考下面的步骤修改和部署 ChangeLogo 扩展包。



注意事项

所有第三方对smartbi的修改，必须放在扩展包中，不能直接修改war包里面的文件。

文档目录:

- 1、修改扩展包
 - 1.1 扩展包修改方法
 - 1.2 登录界面LOGO
 - 1.3 主界面和仪表盘界面logo
 - 1.4 屏蔽右上角图标以及下拉框菜单
 - 1.5 关于页面信息
 - 1.6 删除关于中的tab页
 - 1.7 自定义登录界面
- 2、部署扩展包

更多关于产品的OEM定制，譬如移动端换logo请见：[Smartbi产品OEM修改](#)

1、修改扩展包

1.1 扩展包修改方法

请首先下载 ChangeLogo 扩展包，下载地址：[ChangeLogo.ext](#)。如果需要修改其中的某个文件，请参照如下步骤进行。

- 首先用 WinRAR 之类的压缩工具，打开 ChangeLogo.ext 文件。
- 然后将需要修改的文件从 WinRAR 窗口中拖出来，对其进行修改。
- 接着将修改后的文件，拖放回 WinRAR 窗口中，覆盖以前的旧文件。
- 关闭 WinRAR 窗口保存修改即可。

1.2 登录界面LOGO

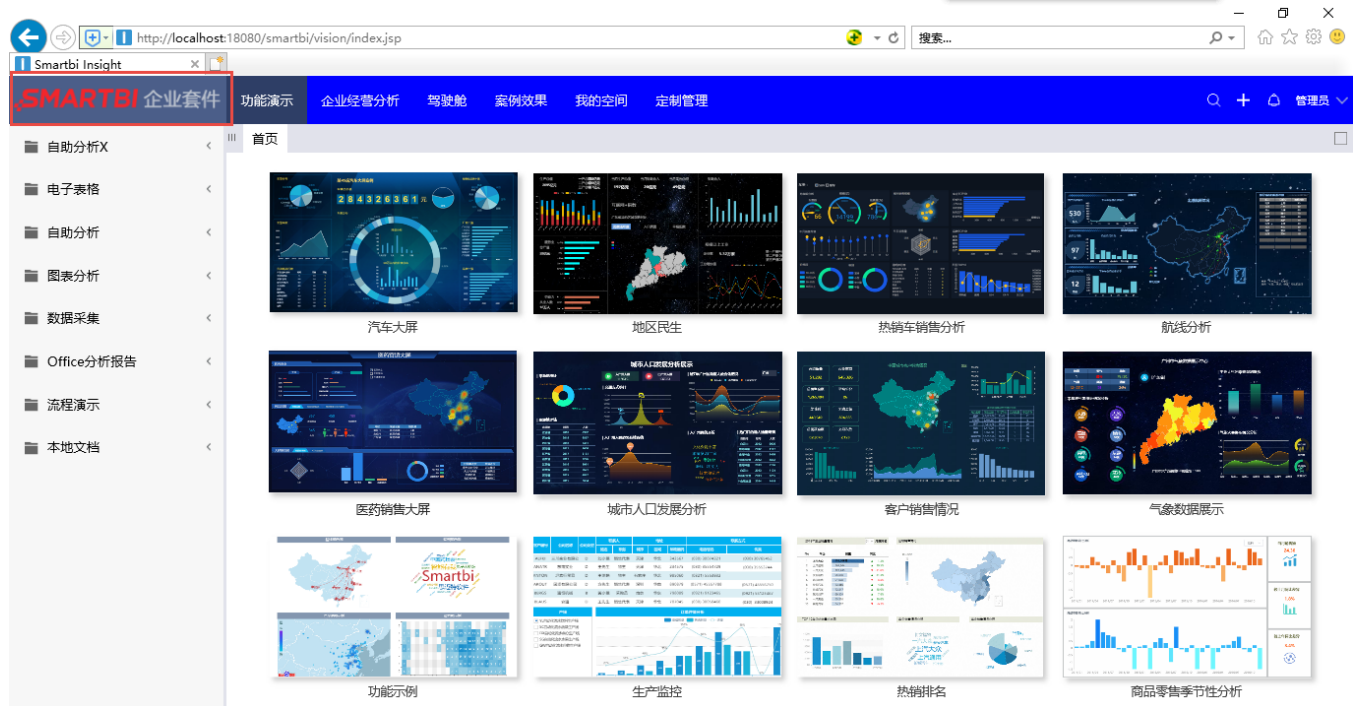
登录界面的图片文件路径是：ChangeLogo.ext\vision\img\login\logo.png 。直接将项目上所用的 LOGO 文件拖放到对应路径中，覆盖同名文件即可。



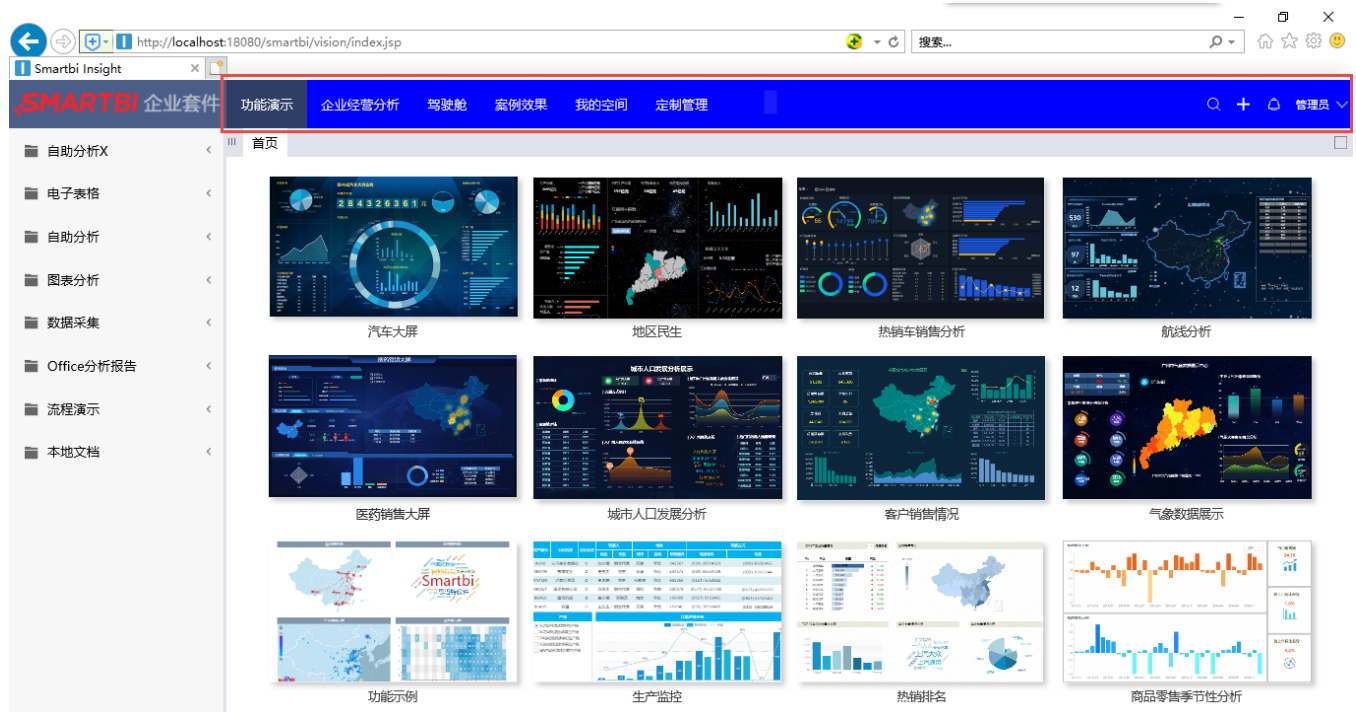
注意！如果是替换图片，则要求图片文件的名称、扩展名、图片的大小尺寸必须完全一致。

1.3 主界面和仪表盘界面logo

Smartbi 主界面的顶部logo文件和仪表盘界面logo是同一个，路径是：ChangeLogo.ext\ vision\img\mainframe\logo.png

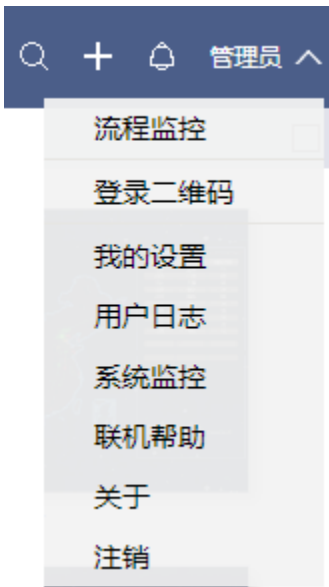


更改主页面最顶端的颜色，对应的文件路径为ChangeLogo.ext\vision\css\patch.css，修改background-color参数，如background-color:#0000ff;



1.4 屏蔽右上角图标以及下拉框菜单

系统右上角的功能图标入口及管理者的下拉菜单选项：



用户可以根据需求进行屏蔽和释放，通过修改文件“BannerView.js.patch”中的代码实现。

该文件对应的路径是：ChangeLogo.ext\vision\js\bof\baseajax\common。

文件链接：[BannerView.js.patch](#)

修改定制详情如下表：

修改内容	修改详情
搜索图标 (🔍)	<p>找到如下代码：</p> <pre>this.elemSearchText.style.display='none';</pre> <p>将这段代码的注释取消则屏蔽了“搜索”图标。</p>
新建图标 (+)	<p>找到如下代码：</p> <pre>this.elemCreateAnalysis.style.display='none';</pre> <p>将这段代码的注释取消则屏蔽了“新建”图标。</p>
我的任务图标 (🔔)	<p>添加如下代码：</p> <pre>this.bannertask.style.display='none';</pre> <p>添加位置如下图所示：</p> <pre>//this.elemSearchText.style.display='none'; //搜索 //this.elemCreateAnalysis.style.display='none'; //新建 //this.elemItemDropdown.style.display='none'; //管理员 this.bannertask.style.display='none'; //我的任务</pre>

管理员	<p>找到如下代码：</p> <pre>this.elemItemDropdown.style.display='none';</pre> <p>将这段代码的注释取消则屏蔽了“管理员”。</p>				
管理员下拉菜单	<p>流程监控、登录二维码、用户日志、系统监控这四个菜单的处理方法一致，在文档最后添加如下代码：</p> <pre>BannerView.prototype._changeLogo_addMenu = BannerView.prototype.addMenu; BannerView.prototype.addMenu = function(text, cmd, parentElem) { if (cmd == "MY_INSTANCE" cmd == "QRCode" cmd == "USER_LOGS" cmd == "SYS_MONITOR") { return ; } this._changeLogo_addMenu(text, cmd, parentElem); };</pre> <p>该代码中，对象关系如下：</p> <ul style="list-style-type: none">• MY_INSTANCE：流程监控• QRCode：登录二维码• USER_LOGS：用户日志• SYS_MONITOR：系统监控 <p>若想屏蔽对应的菜单，只需要在if()语句中将菜单对应的对象加上即可。如以上代码中，表示将“流程监控、登录二维码、用户日志、系统监控”都屏蔽，效果如下：</p> <table><tr><th>屏蔽前</th><th>屏蔽后</th></tr><tr><td></td><td></td></tr></table>	屏蔽前	屏蔽后		
屏蔽前	屏蔽后				
我的设置	<p>找到如下代码：</p> <pre>this.addMenu('\${My}\${Settings}', 'MY_SETTINGS', parentElem);</pre> <p>将这段代码注释即屏蔽了“我的设置”。</p>				
联机帮助	<p>找到如下代码：</p> <pre>this.addMenu('\${Onlinehelp}', 'ONLINE_HELP', parentElem);</pre> <p>将这段代码注释即屏蔽了“联机帮助”。</p>				
关于	<p>找到如下代码：</p> <pre>this.addMenu('\${About}', 'ABOUT_SMARTBI', parentElem);</pre> <p>将这段代码注释即屏蔽了“关于”。</p>				

注销	<p>找到如下代码：</p> <pre>this.addMenu('\${Logout}', 'LOGOUT', parentElem);</pre> <p>将这段代码注释即屏蔽了“注销”。</p>
----	---

1.5 关于页面信息

Smartbi “关于”对话框上的图片LOGO、公司名称、网站等信息的修改。

- 图片文件对应路径是：ChangeLogo.ext\vision\img\about\logo.png
其上的公司名称、网址和邮箱等信息，对应文件是：ChangeLogo.ext\vision\js\bof\baseajax\common\BannerView.js.patch



```
BannerView.prototype.showAbout = function() {
var data = [ registry.get('CompanyName'), registry.get('WebAddress'), registry.get('MailAddr') ] window ];
var dialogConfig = {
title : '关于Smartbi',
size : DialogFactory.getInstance().size.MIDDLE,
fullName : 'freequery.main.AboutDialog'
};
DialogFactory.getInstance().showDialog(dialogConfig, data);
};
```

- 上图data可参考下面代码修改：

```
var data = [ "公司名称", "公司网址", "mailto:公司邮箱", window ];
```

1.6 删除关于中的tab页

对应的文件路径是：ChangeLogo.ext\vision\js\freequery\main\AboutDialog.js.patch，将对应的红色框内代码，未屏蔽的部分代码进行屏蔽，原本屏蔽的//this.elemXXX.style.display='none'放出，则会删除对应的tab页

```

AboutDialog.prototype.initTab = function() {
    this.initLicenseInfo(this.elemLicenseInfoDiv);
    this.pagecontrol = new PagePanel(this.elemTabPanel);
    this.pagecontrol.contentBanner.style.backgroundColor = '#fff';

    this.aboutTab = this.pagecontrol.appendTab();
    this.aboutTab.setCaption("${About}");
    this.aboutTab.appendItem(this.elemAboutDiv);
    if (this.aboutTab.itemParent) {
        this.aboutTab.itemParent.style.verticalAlign = 'middle';
    }
    //this.elemAboutDiv.style.display='none';

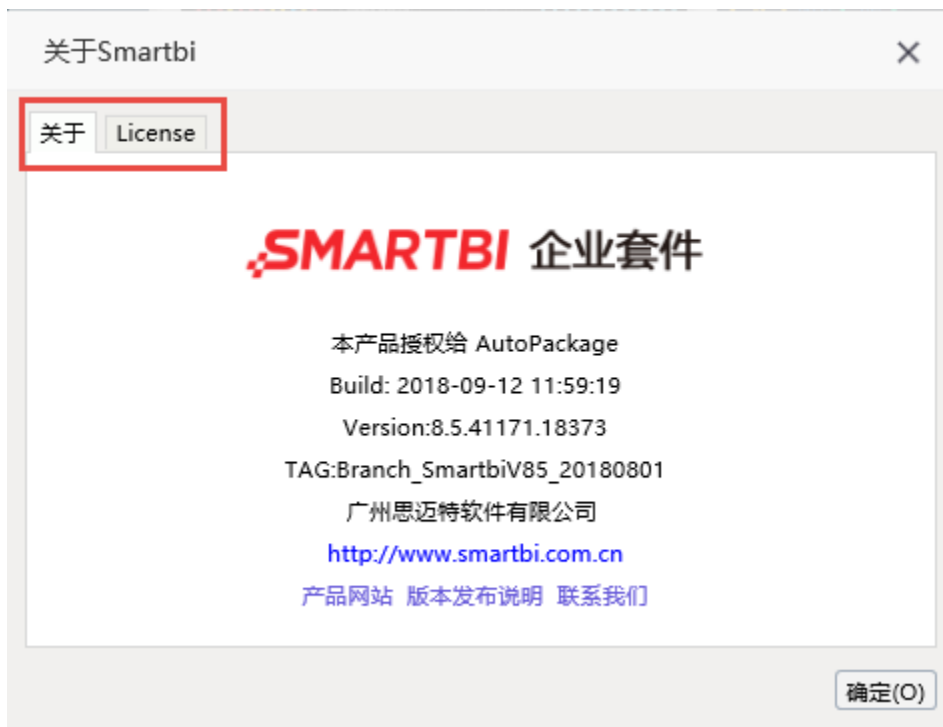
    this.licenseInfoTab = this.pagecontrol.appendTab();
    this.licenseInfoTab.setCaption("License");
    this.licenseInfoTab.appendItem(this.elemLicenseInfoDiv);
    //this.elemLicenseInfoDiv.style.display='none';

    this.pagecontrol.tabs[0].setActive();
    this.pagecontrol.autoFitResize();
}

```

关于

license



1.7 自定义登录界面

我们参照“1.3 登录界面图片”章节的方法，可以替换登录界面上的图片，但这在很多时候无法满足要求，客户希望把整个登录界面完全换成另外一种风格。针对此需求，可以参照如下步骤操作。

- 编写登录界面的 jsp 文件。可以参考 ChangeLogo.ext\vision\login_demo.jsp 示例文件，其中 javascript 代码的登录逻辑部分可以不用修改（直接拷贝使用），该文件的其余部分可以任意修改，把界面修改为客户所要求的设计风格。

```

<script type="text/javascript" src="js/freequery/lang/JSLoader.js"></script>
<script>
    <!-- Smartbi -->
    <!-- document.getElementById("user") -->
    var jsloader = new JSLoader();
    var util = jsloader.resolve("freequery.common.util");
    document.onkeydown = keydown;
    function init() {
        var userName = document.getElementById("user");
        if (userName)
            userName.focus();
    }

    function login() {
        var userName = document.getElementById("user"); //IDHTMLID
        var password = document.getElementById("password"); //IDHTMLID
        var ret = util.remoteInvokeEx("UserService", "login", [userName.value, password.
value]); //Smartbi

        if (!ret || !ret.result){
            alert("!");
            return;
        } else {
            window.location.reload();
        }
    }

    function keydown(e) {
        var theEvent = e || window.event;
        var code = theEvent.keyCode || theEvent.which || theEvent.charCode;
        if (code == 13) {
            login();
        }
    }
    <!------->
</script>

```

- 增加登录界面 jsp 文件配置项。对应的配置文件路径是：ChangeLogo.ext\vision\js\ext\ConfigurationPatch.js，将其中的 loginJspPath 属性值改为登录界面 jsp 文件路径。注意：此路径为相对于 smartbi.war/vision/ 的路径。

```

var ConfigurationPatch = {
    loginTitle : "Smartbi Vision",
    // loginJspPath: "login_demo.jsp", //自定义登录页面相对vision的地址
    applicationTitle : "Smartbi Vision",
    configTitle : "smartbi config",
    extensionPoints : {
        css : [ 'vision/css/patch.css' ]
    },
};

```

其它说明：

- 系统统一访问入口仍然为 <http://localhost:18080/smartbi/vision/index.jsp>；访问 Smartbi 时如果用户未登录，会自动跳转到 login_demo.jsp 页面中。
- 此登录界面的处理逻辑为：点击“登录”按钮时，调用 smartbi 的登录API进行登录，再重新刷新本页面。

自定义登录界面示例：



2、部署扩展包

根据项目上的客户化定制需求，修改完 ChangeLogo 扩展包后，需要将其部署系统中。完整的部署过程和操作步骤，请参考[扩展包部署](#)。