



➤ 客户端下载

http://www.masocloud.com/download/PrintWorker_Install.exe

➤ 在线演示

http://demo.masocloud.com/DoYsUI/test/dlabel/integrate_dlabel.html

1. 关于 MASO

1.1. 什么是 MASO (码尚) 打印控件

MASO 打印控件是 [敖维科技](#) (上海敖维计算机科技发展有限公司) 自主研发的新款标签打印控件 (打印组件), 用于作为第三方控件独立发布。

1.2. 谁需要 MASO 打印控件

MASO 目标使用客户是 ISV (Independent Software Vendors)、SI (System Integrator) 或打印机厂商。

1.3. MASO 的设计目标是什么

极简、高效、轻量、易于集成。

作为 ISV/SI 客户, 只需半天左右的时间即可完成 MASO 打印控件的集成工作, 迅速解决标签打印相关问题。

1.4. MASO 技术特点

➤ Web 集成

可以直接集成在 Web 页面中。

标签设计基于 H5, 采用纯原生 JavaScript 技术 (ES8), 不依赖于任何第三方组件或技术。标签设计文件可以 JSON 字符串格式存储在服务端的数据库中, 也可以以文本文件形式存储在服务器或客户端的磁盘中。

➤ WinForm 集成

另外也可以集成在 WinForm 客户端（例如：C/S 架构的 MES 系统）。

规划未来支持 Android、Linux 客户端集成。

➤ **.NET 打印服务客户端**

支持网络打印。

标签打印服务客户端基于 .net Framework，安装简单，占用内存小，速度快。

额外提供打印任务管理、打印数据监控，历史数据查询及补打印。

打印机工作状态监控，耗材消耗统计等。

➤ **脚本支持**

内置支持原生 JavaScript 编程，脚本直接存储在每张标签文件中。可以实现最终用户的各种苛刻要求。开发人员只需将原始数据赋值给标签即可，脚本负责将数据二次加工成标签最终期望的展示形式。例如：根据产品不同，动态打印产品图片等。

➤ **标签支持**

支持主流的一维码（CODE-128, CODE-39, EAN-13）、二维码（QR, DataMatrix, PDF417）、GS1 等。

1.5. 解决客户痛点

➤ **缺乏 Web 版控件**

目前市面上的第三方打印控件普遍缺乏 Web 环境的集成支持，不适合 B/S 系统，大多仅支持 C/S 架构的系统集成。MASO 打印控件则两种都支持。

➤ **授权成本高**

对比 BarTender 等集成方案，MASO 打印控件提供更灵活，性价比更高的低成本集成解决方案。

2. 准备工作

2.1. 运行环境

➤ 操作系统

Windows 7 或 Windows 10 操作系统，用户具有管理员权限。

➤ 浏览器

Google Chrome、Microsoft Edge、Firefox 等主流操作系统，不支持 IE。

➤ .net Framework

客户端打印环境需要安装.net Framework 4.8，如果仅设计标签可以免安装。

下载地址：<https://dotnet.microsoft.com/download/dotnet-framework/net48>

2.2. 安装打印工作站客户端

使用前，需要提前安装 MASO 打印工作站客户端程序。提供标准安装包和绿色免安装包两种模式。

下载地址：邮件索取。

2.3. 下载样例程序

集成开发前，可以先下载集成样例程序，可以更快掌握控件集成方法。

下载地址：邮件索取。

3. 开发集成

3.1. 样例页面

➤ **在线演示地址**

http://demo.masocloud.com/DoYsUI/test/dlabel/integrate_dlabel.html

➤ **集成样例**

参考样例程序包中的 `integrate_dlabel.html` 页面。

3.2. 初始化控件

在集成页面中提供一个 `iframe`，`iframe` 样式可以自定义。

```
<iframe id="iframeLabel"></iframe>
```

指定 `iframe` 的宽高及边框样式。初始化标签，示例代码：

```
var lbl;  
  
var designMode = true; // true: 设计模式, false: 预览模式  
  
if (designMode) {  
    iframeLabel..src = ".../designer.html";  
}  
  
else {  
    iframeLabel..src = ".../preview.html";  
}  
  
iframeLabel.addEventListener("load", function (evt) {  
  
    // -- 获取 Label 控件 实例 --  
  
    lbl = ifr.contentWindow.getLabel();  
  
    // -- 注册并监听标签保存事件 (预览模式不需要) --  
  
    lbl.addEventListener("on-save", onSave, { a: "123", b: "456" });  
  
    // -- 获取打印机列表 --
```

```
lbl.getPrinterList().then((data) => {  
  
    console.log(data);    // -- data: 打印机列表 --  
  
}, false);
```

3.3. 调用 API

参考下一章节, 主要工作如下:

- 加载标签
- 标签赋值
- 标签打印

4. API

4.1. getLabel

获取 Label 控件对象实例。后续操作均通过该实例进行；

```
var lbl;  
lbl = ifr.contentWindow.getLabel();
```

参考样例程序，ifr 是集成页面的 iframe 对象。

4.2. addEventListener

添加标签事件监听。

➤ on-save

```
lbl.addEventListener("on-save", onSave, { a: "123", b: "456" });
```

标签保存时，会触发 on-save 事件，参数 onSave 是回调函数。示例：

```
function onSave(jsp) {  
    console.log(jsp);  
    let labelString = jsp.labelString;  
    // -- 此处添加代码，可以将标签文件字符串(labelString)发送到服务端存储。 --  
}
```

4.3. getPrinterList

获取打印机列表。示例：

```
lbl.getPrinterList().then((data) => {  
    for (let i = 0; i < data.length; i++) {  
        console.log(data[i]);  
    }  
});
```

4.4. loadLabel

加载标签。示例：

```
lbl.loadLabel(labelContent, { designMode: true });
```

➤ labelContent

标签内容字符串。

➤ designMode

true: 设计模式; false: 预览模式。

4.5. setValue

标签变量赋值。

```
lbl.setValue("variable_name_x", "变量值 X");
```

4.6. setElementValue

标签元素赋值。

```
lbl.setElementValue("element_name_x", "元素值 X");
```

4.7. compute

计算标签。

加载标签或对标签赋值后，调用 compute，对标签内容重新计算，包括执行自定义的脚本。同时标签界面会执行重绘。

4.8. getData

返回当前标签数据。

多张标签须多次获取。用于实际打印物理标签。示例：

```
let data = [];  
  
for (let i = 0; i < count; i++) {  
  
    // 标签赋值  
  
    lbl.compute(false); // false: 不重绘界面; true: 重绘  
  
    data.push(lbl.getData());  
  
}  
  
lbl.compute(true);
```

4.9. print

打印标签。

```
lbl.print({  
  
    data: data,  
  
    printerName: "printer name",  
  
    copies: 1  
  
})
```

➤ data

数据数组。

➤ printerName

打印机名称。

➤ copies

打印份数。

示例：当 copies = 2 时，打印结果为：1, 1; 2, 2; 3, 3。

如需打印 1, 2, 3; 1, 2, 3 这种排列，可以通过调用 print 方法 2 次实现。

5. 技术支持

➤ 商务合作

陈雷 rex

email: chenlei@always-china.com.cn

wechat: chenlei-rex

mobile: 135-2448-5915

➤ 开发支持

David Li

email: liwg@always-china.com.cn

wechat: liweiguo_cn