

# C-Lodop 技术手册

Ver 2.0.9.0

**Email: [gungho999@sina.com](mailto:gungho999@sina.com)**

**QQ: 932131686**

**梦泰尔软件(中国)**

**MTSoftware(CN)**

## 目 录

一、概述.....	3 -
二、安装.....	5 -
C-Lodop 的 32 位安装文件: .....	5 -
C-Lodop 的 64 位安装文件: .....	5 -
C-Lodop 的 32 位安装文件（适应 https 协议）: .....	5 -
安装后的运行文件: .....	5 -
安装后的设置窗口 .....	6 -
安装后的欢迎页面 .....	7 -
三、调用: .....	8 -
加载云打印 JS 文件的简单方法 .....	8 -
动态加载云打印 JS 文件 .....	8 -
云打印 JS 文件的优先级 .....	8 -
加载多个云打印 JS 文件 .....	9 -
获取 LODOP 对象.....	9 -
四、功能函数.....	10 -
Create_Printer_List .....	10 -
Create_PageSize_List.....	10 -
On_Return .....	10 -
On_Return_Remain.....	11 -
On_CLodop_Opened .....	11 -
demoCreateCLodopJSscript .....	12 -
PREVIEW .....	12 -
五、 升级.....	13 -
六、 https 调用 C-Lodop.....	15 -
安装文件 .....	15 -
服务端口 .....	15 -
进入 https 服务欢迎页 .....	15 -
安装后的运行文件 .....	16 -
判断 https 服务 .....	17 -
七、 特注.....	17 -

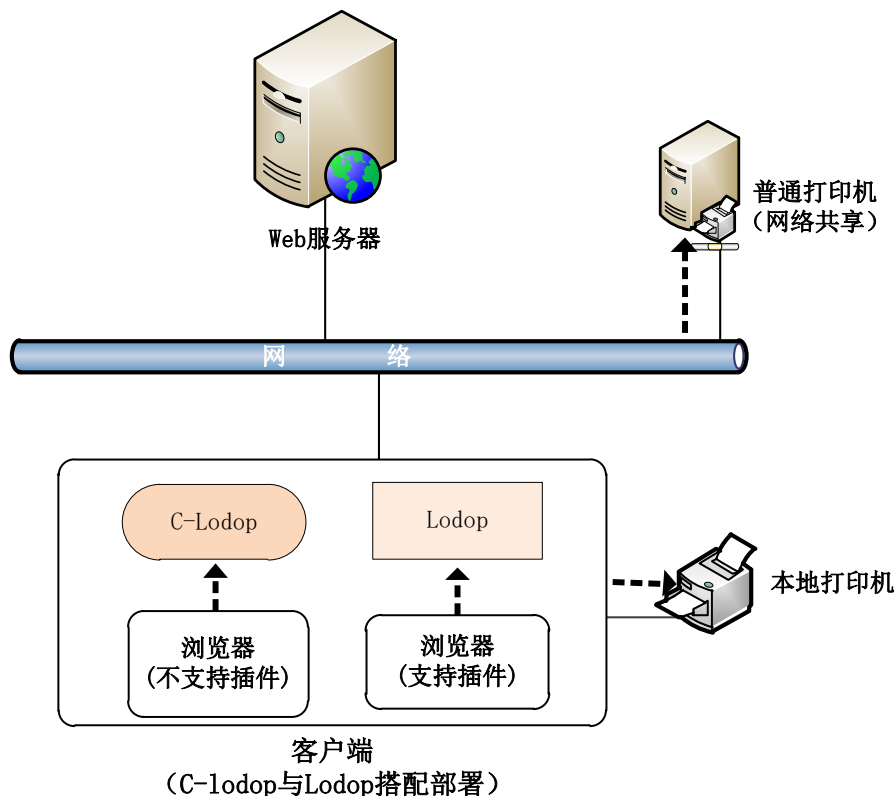
# 一、概述

C-Lodop 云打印是梦泰尔软件（MTSoftware）推出的一款精巧快捷的云打印服务产品，以 Lodop 功能语句为基础，开创了 JS 语句实现远程打印的先河，为“移动设备+Wifi+普通打印机+集中打印”奠定了编程基础。

C-Lodop 对客户端浏览器要求很低，几乎支持任何现代浏览器，除了 Windows 平台上的 Chrome(谷歌)、FireFox(火狐)、Opera、Safari、IE、edge 几大系列及其外壳浏览器（如 QQ 浏览器、搜狗、360、UC 等等）之外，客户端还支持“平板电脑”和“智能手机(无需安装 APP)”以及 Linux、OS(Mac)上的浏览器。

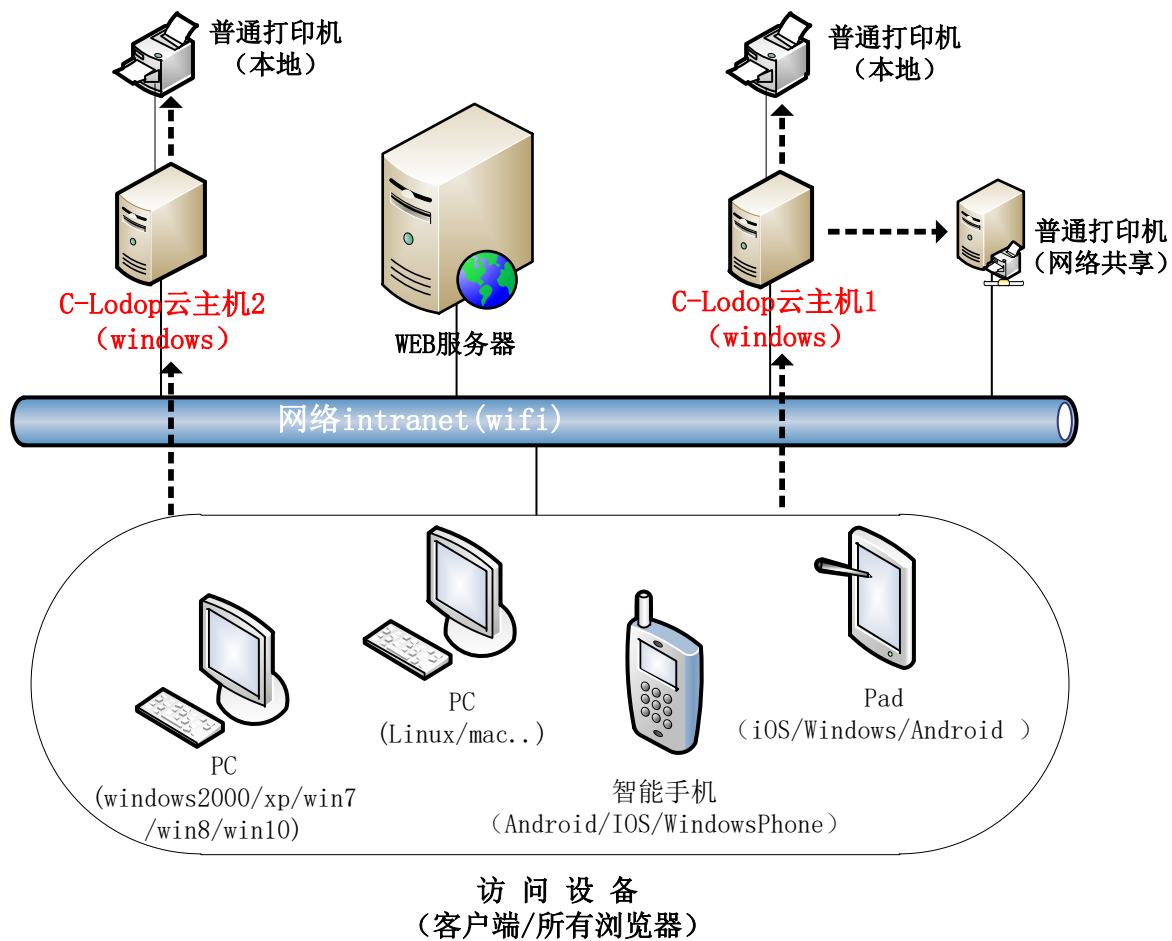
C-Lodop 可扮演两种角色，第一种是在 Windows 客户端 PC 本机上安装，从而替代 Lodop 控件功能，这也是常见基本用法；另一种角色是单独部署云服务器，适合后台集中打印。在第一种角色中，建议“云打印+控件”组合搭配使用。

## 1) C-Lodop 与 Lodop 搭配示意图：



(图 1-1)

## 2) C-Lodop 独立部署示意图：



(图 1-2)

## 二、安装

### C-Lodop 的 32 位安装文件：

文件名是 [CLodopPrint\\_Setup\\_for\\_Win32NT.exe](#) 可同时在 32 位和 64 位 Windows 操作系统上安装，适应面较广。

### C-Lodop 的 64 位安装文件：

文件名是 [CLodopPrint\\_Setup\\_for\\_Win64NT.exe](#) 只能在 64 位 Windows 操作系统上安装，性能较高，适合单独部署。

### C-Lodop 的 32 位安装文件（适应 https 协议）：

文件名是 [CLodop\\_Setup\\_for\\_Win32NT\\_https.exe](#) 可同时在 32 位和 64 位 Windows 操作系统上安装，适应面较广。

### 安装后的运行文件：

以 32 位文件为例：

#### 1、文件路径：

Program Files (x86)\MountTaiSoftware\CLodop32

#### 2、[云打印服务](#)主文件：

CLodopPrint32.exe

#### 3、[免登录启动](#)服务文件：

CLodopService32.exe

#### 4、样例目录

Program Files (x86)\MountTaiSoftware\CLodop32\Rootdir\CLodopDemos

样例中有一个 LodopFuncs.js 文件很关键，建议尽量分析理解透彻。

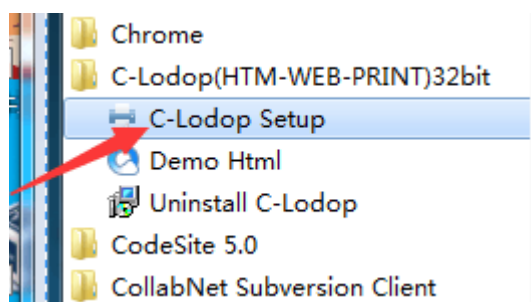
## 安装后的设置窗口

正常安装成功后，首先会弹出如下提示窗口：



(图 2-1)

这也是管理设置窗口，以后可以从 Windows 如下菜单重新进入以上界面。



(图 2-2)

在管理设置界面中，通过“设置->端口设置”菜单进入下面的对话框进行端口设置，默认端口是 8000 和 18000：



(图 2-3)

## 安装后的欢迎页面

点击以上（图 2-1）圈定的按钮链接，进入如下欢迎页面：

欢迎使用C-Lodop打印服务系统  
The Cloud Web Service System for Lodop HTML Print

**使用说明：**

一、C-Lodop是云打印服务，可接受JS语句实现远端输出，它是因满足平板电脑和手机浏览器打印而诞生的，由于其轻巧便捷，所以也可以安装在PC本地，是打印控件Lodop的升级产品。其在保持打印性能的同时，能适应所有浏览器的各种版本。

引用C-Lodop很简单，在页面head中[添加](#) 如下路径的js文件，就可以向本服务发打印请求了：

```
<script src="http://192.168.1.4:8000/CLodopfuncs.js"></script>
```

如果C-Lodop服务安装在PC本地，页面则用localhost或127.0.0.1向其发送打印请求：

```
<script src="http://localhost:8000/CLodopfuncs.js"></script>
```

页面引用以上js之后，就可以调用Lodop传统功能，点[打印预览](#)执行如下经典语句(注意区分大小写)看看效果：

```
var strHTML=document.getElementsByTagName("html")[0].innerHTML;
LODOP.PRINT_INITA(1,1,770,660,"测试预览功能");
LODOP.ADD_PRINT_TEXT(10,60,300,200,"这是测试的纯文本，下面是超文本：");
LODOP.ADD_PRINT_HTM(30,5,"100%", "80%", strHTML);
LODOP.PREVIEW();
```

下面是远程打印的常见四步骤：

第一步选择云主机：

第二步选择打印机：

第三步选纸张类型：

最后是[打印预览](#)，或者不预览而[直接打印](#)

二、默认情况下，只要能访问本页，都可以通过本服务控制打印机输出，为了阻止陌生请求，管理者可以从这里为guest用户[设置一个密码](#)（设置前需用admin身份登录，初始密码皆空）。

三、这是一个免费打印服务软件，您可以长期使用，但不能用来[商用](#)谋利。

四、如果出现故障，可以点这里[查看系统信息](#)

五、[最后欣赏用其实现 Lodop全部传统例子>>>](#)

（图 2-4）

如（图 2-4）红框中连接进入，可验证全部传统例子在 c-lodop 中的应用情况。

## 三、调用：

### 加载云打印 JS 文件的简单方法

调用 C-Lodop 很简单，在页面 head 中 添加 如下路径的 js 文件，就可以向其发打印请求了，如下几个例子都可以在欢迎页面或 Lodopfuncs.js 这个样例文件里找到：

```
<script src="http://Host 地址:8000/CLodopfuncs.js"></script>
```

这里的红色部分是固定内容，Host 地址是指安装 C-Lodop 的主机地址，可以是其 IP 地址或域名，如果是本地(第二种角色)打印，该值用 localhost、127.0.0.1 或主机名代替，端口默认是 8000 冗余 18000 端口(单独部署时，可以自行指定自己的端口)，如：

```
var head = document.head || document.getElementsByTagName("head")[0] || document.documentElement;
var oscript = document.createElement("script");
oscript.src = http://localhost:8000/CLodopfuncs.js?priority=1;
head.insertBefore( oscript,head.firstChild );

//引用双端口(8000和18000) 避免其中某个被占用:
oscript = document.createElement("script");
oscript.src = http://localhost:18000/CLodopfuncs.js?priority=0;
```

默认本地域名8000端口

8000被占时  
冗余支持18000端口

```
<script src="http://192.168.1.1:8000/CLodopfuncs.js"></script>
```

```
<script src="http://Localhost:8000/CLodopfuncs.js"></script>
```

### 动态加载云打印 JS 文件

为了方便控制或兼容老程序，以上静态方法可以用如下动态语句替代，效果一样：

```
var oscript = document.createElement("script");
oscript.src = "http://localhost:8000/CLodopfuncs.js?priority=1";
var head = document.head || document.getElementsByTagName("head")[0] || document.documentElement;
head.insertBefore( oscript,head.firstChild );
```

### 云打印 JS 文件的优先级

```
<script src="http://Localhost:8000/CLodopfuncs.js? priority=1"></script>
```

以上引用方法的参数 priority 设置的是优先级，优先级值是从 0 开始的阿拉伯数字，数字越大，优先级越高，默认是 0。当一个页面按后续章节同时引用多个云主机时，优先级设置很关键，否则会按下载顺序来处理。



## 加载多个云打印 JS 文件

如果一个页面通过以上语句同时引用了俩云主机，通常后一个引用会覆盖前一个引用，其先后顺序是以下载 js 速度快慢为准，此时如果页面希望其中一个引用有更高的优先级，可以用 url 参数 `priority` 来控制，该参数值越大优先级越高，默认值为 0，例如：

```
<script src="http://192.168.1.6:8000/CLodopfuncs.js"></script>
<script src="http://localhost:8000/CLodopfuncs.js? priority=1"></script>
```

这也是 C-Lodop 默认安装例子演示时，既可以让手机浏览器访问，也可以让电脑访问的神奇原因。如果希望页面的两个引用同时存在(不相互覆盖)，则需要用 url 的参数 `name` 来区分,例如：

```
<script src="http://192.168.1.1:8000/CLodopfuncs.js?name=LODOPA"></script>
<script src="http://192.168.1.2:8000/CLodopfuncs.js?name=LODOPB"></script>
```

此时调用打印时，不再用 LODOP 这个默认的 JS 变量名，而是用 LODOPA 或 LODOPB，二者分别向不同的云主机发打印指令，LODOPA 指向 192.168.1.1，而 LODOPB 指向 192.168.1.2，以此类推。

## 获取 LODOP 对象

引用以上 js 文件后，就可以在页面程序里通过 `getCLodop` (注意：如果 C-Lodop 与 Lodop 按图 1 搭配使用，要使用 `Lodopfuncs.js` 示中的 `getLodop`，其中就包含了 `getCLodop` 语句的调用) 函数获取主对象，赋值给一个普通 JS 变量，为了兼容已经开发好的打印程序，仍然统一用 LODOP 这个变量名：

```
var LODOP=getCLodop();
```

有了该对象之后，就可以调用云打印功能语句了，如

```
LODOP.PRINT();
LODOP.PREVIEW();
LODOP.PRINT_DESIGN();
```

更多语句参考后面的功能函数介绍。

## 四、功能函数

C-Lodop 的功能函数几乎涵盖了 Lodop 的所有语句，它们的详细功能说明参考《WEB 打印控件 Lodop 技术手册》。除此之外，新增了部分与页面整合相关的函数和属性，还有个别语句的参数值有所扩展，详解如下：

### Create\_Printer\_List

**作用：**C-Lodop 函数之一，获得云主机链接的所有打印机名单，并添加到当前页面某个 Select 标签元素中

**格式：**`Create_Printer_List(oElement);`

**参数：**`oElement` 目标对象(已经存在的某个 Select 标签元素)

**举例：**`CLODOP.Create_Printer_List(document.getElementById('Select01'));`

### Create\_PageSize\_List

**作用：**C-Lodop 函数之一，获得目标打印机所拥有的纸张类型名单，并添加到当前页面某个 Select 标签对象元素中

**格式：**`Create_PageSize_List(oElement, iPrintIndex);`

**参数：**`oElement` 目标对象(已经存在的某个 Select 标签元素)

`iPrintIndex` 目标打印机在 windows 系统内的序号

(操作系统对所添加打印机的顺序编号，从 0 开始)

**举例：**`CLODOP.Create_PageSize_List(document.getElementById('Select03'), 2);`

### On\_Return

**作用：**C-Lodop 属性之一，设置那些需要返回结果的动作语句在获得结果时触发的回调函数

**格式：**`On_Return=function(TaskID,Value){...};`

**说明：**和 Lodop 相比，C-Lodop 的动作语句不再立即返回执行结果，而是返回一个任务编号，页面程序在回调函数中通过该编号来区分是哪个语句返回的结果，并根据结果值在该回调函数中进行下一步的处理。

**参数：**`TaskID` 后续动作语句的任务编号，进一步参考后面 On\_Return\_Remain 的使用。

`Value` 该动作语句返回的结果值，类型为逻辑真假值或字符串值。

**举例：** ....

```
LODOP.On_Return=function(TaskID,Value){
    if (Value) alert("已发出实际打印命令！"); else alert("放弃打印！");
};
LODOP.PRINTA(); //这是一个典型的动作语句
```

## On\_Return\_Remain

**作用：**C-Lodop 属性之一，让所设置的回调函数持续保留，对后续多个动作都有效

**格式：**On\_Return\_Remain=真假值;

**说明：**正常情况下，回调函数执行完毕后会自动销毁（也就是本属性默认值是 false），而本属性设置为真值就可以阻止其销毁，从而让后面的所有动作语句都触发同一个回调函数，在回调函数中用 TaskID 的值来区分是哪个动作触发的回调。

**举例：**

```
var TaskID1, TaskID2
LODOP.On_Return_Remain=true;
LODOP.On_Return=function(TaskID,Value){
    if (TaskID== TaskID1) {
        alert("判断是否打印成功的结果是: "+ Value)
    } else if (TaskID== TaskID2) {
        alert("判断打印任务是否还存在的结果是: "+ Value)
    };
};
TaskID1=LODOP.GET_VALUE("PRINT_STATUS_OK",P_ID);
TaskID2=LODOP.GET_VALUE("PRINT_STATUS_EXIST",P_ID);
```

## On\_CLodop\_Opened

**作用：**C-Lodop 属性之一，该事件在页面 websocket 建立完毕并准备好时触发。

**格式：**On\_CLodop\_Opened =function();

**说明：**页面程序可以把调用打印的过程放在该事件中，避免因为过早打印造成“C-Lodop 未准备好”的频繁提示，同时也能尽快打印。事件触发正常工作之后，建议重置该属性为 null。

**举例：**

```
if (needCLodop()) {
    window.On_CLodop_Opened=function(){
        OpenPreview();    //执行打印预览
        window.On_CLodop_Opened=null;
    };
} else
window.onload = function(){OpenPreview();};
```

## demoCreateCLodopJScript

**名称:** C-Lodop 典型例子函数之一，在当前页面动态引入 C-Lodop 的 js 文件 [CLodopfuncs.js](#)，引入时清理掉此前的其它引用，从而唯一选择云主机。

**格式:** [demoCreateCLodopJScript\(strSrc\);](#)

**参数:** [strSrc](#) 参数值为目标云主机的 script 标签，其格式为：

[<script src='http:// Host 地址:端口/CLodopfuncs.js'></script>](#)

也可以仅是 src 值部分：

[http://Host 地址:端口//CLodopfuncs.js](#)

**举例:**

[demoCreateCLodopJScript \("<script src='http://192.168.0.8:8000/CLodopfuncs.js'></script>"\);](#)

或

[demoCreateCLodopJScript \("http://192.168.0.8:8000/CLodopfuncs.js"\);](#)

与前面说的在页面能同时引用多个云主机相比，本函数的动态选择比较常见，毕竟客户输出的目标打印机往往是很明确的。

## PREVIEW

**作用:** 打印预览语句扩展了三个参数 [oView,iW,iH](#)

**格式:** [PREVIEW\(oView,iW,iH\)](#)

**功能:** 以某种样式进行打印预览。

**参数:** 参数值及其含义

**[oView](#):** 让预览窗口**内含**在当前页面内（而不是弹窗），这里的“内含”不同于插件“内嵌”模式，进一步的内含样式有如下几种可选：

[\\_dialog](#):弹出对话框

[\\_blank](#):新窗口

[\\_self](#):在本窗口

[\\_top](#):新窗口

[\\_parent](#):本窗口

**具体 ID 值:**在某个现有 [iframe](#) 元素内（需要提前在页面内增加该 [iframe](#)）

**[iW](#):** 预览窗口显示时的宽度值(px);

**[iH](#):** 预览窗口显示时的高度值(px);

**举例:** [LODOP.PREVIEW\("\\_blank "\);](#) //在新的浏览器窗口显示打印预览

(注意浏览器阻止设置对其仍然有效)

[LODOP.PREVIEW\("001"\);](#) //在 ID 等于“001”的 [iframe](#) 元素内显示打印预览

## 五、升级

Lodop 6205 发行包里包含“[install\\_lodop32.exe](#)”、“[install\\_lodop64.exe](#)”、“[CLodopPrint\\_Setup\\_for\\_Win32NT.exe](#)”这 3 个安装文件和 50 个小例子及其 JS 文件。

文件“[CLodopPrint\\_Setup\\_for\\_Win32NT.exe](#)”是云打印“C-Lodop”的主安装文件，如果是配合部署需三个文件（见图 1），如果单独部署只需这一个文件。

由于云打印兼容控件传统语句，所以开发者要把自己的 web 系统由 Lodop6.1xx 升级到新版本(6.2xx+云打印)还是比较容易的，升级主要步骤有三个：

**第一、更新 [Lodopfuncs.js](#) 这个文件。**这虽说是一个 js 例子，但对大多数人来说最好直接使用它，此前已经自行编写 [getLodop](#) 过程的开发者需要理解本次变化后充实改进一下。其变化主要是增加了判断哪些浏览器调用传统控件，哪些浏览器调用 C-Lodop 云打印，并动态添加支持云打印的新 js 文件。

如果升级为**全面采用云打印（不再使用控件）**，即**单独部署 C-Lodop 云打印**，可修改如下函数直接返回 true

源代码如下（注意红色代码变化）：

```
//===判断是否需要安装CLodop云打印服务器:===  
function needCLodop() {  
    try{  
        var ua=navigator.userAgent;  
        if (ua.match(/Windows\sPhone/i) !=null) return true;  
        if (ua.match(/iPhone|iPod/i) != null) return true;  
        ...  
        ...  
    };  
    return false;  
} catch(err) {return true;};  
};
```

修改后如下：

```
//===判断是否需要安装CLodop云打印服务器:===  
function needCLodop() {  
    try{  
        return true;  
    } catch(err) {return true;};  
};
```

**第二、C-Lodop 语句对字母大小写敏感，要求大写。**此前控件例子中虽然都是大写，但小写或混合也是勉强可以的，但现在不行了，必须全是大写，例如 `LODOP.PRINT()` 写成 `LODOP.print()` 是非法的。

**第三、云打印语句返回结果的方式不一样，要指定回调函数。**此前控件语句本身会“同步返回结果”，而新的云打印前后指令不在同一个程序进程里，所以采用“异步回调函数”，函数名叫 `On_Return`。这个改动涉及面稍广，但做起来也不难，参考如下举例（样例 4），只需在原来(蓝色)代码前面增加这段新(红色)代码，如此以来就可以让页面程序同时兼顾老版本控件和新的云打印：

```
if (LODOP.CVERSION) { //用CVERSION属性判断是否云打印
    LODOP.On_Return=function(TaskID,Value){
        if (Value) alert("已发出实际打印命令!"); else alert("放弃打印!");
    };
    LODOP.PRINTA();
    return;
};
if (LODOP.PRINTA())
    alert("已发出实际打印命令");
else
    alert("放弃打印");
```

## 六、https 调用 C-Lodop

C-Lodop 云打印支持 https 协议，需要安装特殊版本，该版本的安装和端口详情如下：

### 安装文件

文件名称：CLodop\_Setup\_for\_Win32NT\_https\_2.060.exe；

注意文件名称中包含 https 字样。

### 服务端口

默认情况下 http 服务和 https 服务的端口差值 443，即：

http 服务端口默认 8000；

https 服务端口默认 8443；

端口 8443 与其他服务冲突时，自动变大。

页面引用方式举例如下：

```
<script src="https://localhost:8443/CLodopfuncs.js"></script>
```

注意其中域名必须是 localhost（不能是 127.0.0.1 或具体 ip 地址）；

引用方式也可以动态添加，动态加载方法可参考 第三部分“调用”。

### 进入 https 服务欢迎页

进入欢迎首页参考图 2-1、2-2，如果出现如下图 6-1 所提示的红框，则说明 C-Lodop 云打印 https 服务已启动。



(图 6-1)

## 安装后的运行文件

在 http 标准运行文件 (参考第 5 页) 基础上增加了 5 个文件, 如下:

1. 证书管理和安装: certmgr32.exe;
2. SSL 证书 key: localhost32.key;
3. SSL 证书文件: localhost32.crt;
4. https 服务辅助文件: libeay32.dll;
5. https 服务辅助文件: ssleay32.dll 。



## 判断 https 服务

页面程序通过 CLODOP.HTTPS\_STATUS 属性判断是否安装 https 服务：0-没安装 1-已安装 2-安装且已启动。

例如：

```
if (CLODOP.HTTPS_STATUS==undefined || CLODOP.HTTPS_STATUS<1) {  
    alert("请安装C-Lodop的https服务版本(特别版本)！");  
}
```

## 七、特注

由于运行在云端，所以 C-Lodop 不支持预览或设计窗口的前端**内嵌**功能，需要这项功能的业务系统，升级时最好兼顾“[控件\(插件\)+云打印](#)”，以便操作者根据需要选用支持插件的浏览器。尽管有此缺憾，但 C-Lodop 构思巧妙，未来前景远大，以后会发布带更多精彩功能的版本，甚至让最终用户能直接使用，满足“[移动设备+Wifi+普通打印](#)”的迫切需要。

<完>