

# PDF 电子签章

## PdfSigner - 开发文档

### 1. PdfSigner 开发流程

PdfSigner 电子签章盖章的原理: 以 PDF 文件为原始数据,然后使用数字签名技术对原始数据进行数字签名并盖章。

盖章流程:

步骤	说明	接口	备注
1	装载控件		<object>... </object>
2	打开文件	Open	
3	在指定位置进行盖章	SingleSign 或 MultiSignByString 或 CrossSignByString	也可以模拟点击, 用户手动选择盖章的位置, 此时可以利用 Trigger 系列接口
4	保存或者上传文件	Save 和 Post	

### 2.WEB 页面调用控件:

```
<object id="pdfSigner" classid="clsid:66CDF03C-5C9A-44CA-89B1-2E82DD31AEC8"
width="700px" height="500px">
</object>
var activeX = document.getElementById("pdfSigner");//控件变量
```

### 3.接口说明

#### 3.1. 文件操作:

##### 3.1.1. Error Open(string url, string username, string password)

功能:

打开文件

说明:

其中以“http://”开头的 url 为 HTTP 文件, 以“ftp://”开头的为 FTP 文件, 其他默认为本地文件。打开本地文件时, username 和 password 不使用。

参数:

url 文件 URL, 目前支持文件、HTTP 文件和 FTP 文件

username 用户名

password 密码

返回:

错误码

示例:

```
var filePath = "D:\\1.pdf";  
var error = activeX.Open(filePath, null, null);
```

### 3.1.2. Error Save (string url, string username, string password)

功能:

保存文件，保存后所有签章都无法删除

参数:

url 文件 URL

username 用户名

password 密码

返回:

错误码

示例:

```
//存储文件的路径，需要注意的是文件夹必须存在，即 D:\\test 这个文件夹必须存在  
var filePath = "D:\\test\\1.pdf";  
var error = activeX.Save(filePath, null, null);
```

### 3.1.3. Error Close()

功能:

关闭文件

返回:

错误码

### 3.1.4. bool IsOpened ()

功能:

是否已打开文件

返回:

True:文件已打开

False: 文件未打开

### 3.1.5. string GetCurrentFileName()

功能:

获取当前打开文件的路径和名称

返回:

当前打开文件的路径和名称

### 3.1.6. int CurrentPageNumber

功能:

获取当前页码，从 1 开始

### 3.1.7. int PageCount

功能:

获取当前文档页数

### 3.1.8. void GoFirstPage ()

功能:

翻到首页

### 3.1.9. void GoNextPage ()

功能:

翻到下一页

### 3.1.10. void GoPrevPage ()

功能:

翻到上一页

### 3.1.11. void GoLastPage ()

功能:

翻到尾页

### 3.1.12. void GoPage(int number)

功能:

翻到指定页码

参数:

number 需要翻到的页码

## 3.2. HTTP 上传操作:

### 3.2.1. void HttpInit ()

功能:

HTTP POST 上传的初始化工作，主要是清空参数

### 3.2.2. void AddPostString(string name, string value)

功能:

添加上传参数

参数

name 参数名称

value 参数值

### 3.2.3. Error Post(string url, string filenameInPostForm)

功能:

Post 方法上传，之前添加的参数将编码在 URL 中

参数

url 上传页面 URL

filenameInPostForm 文件在 HTML 表单中的名称

返回:

错误码

说明:

进行 HTTP 上传操作时, 依次调用上述三个接口以完成上传工作

示例:

```
var uploadUrl = "http://192.168.0.136/Task/PostTest";
var fileName = "UploadFile";
//HTTP POST 上传初始化
activeX.HttpInit();
//可以增加随文件一同 POST 的其他参数, 可以多次调用多次添加
activeX.AddPostString("DictIndex", "1");
//通过 POST 方法上传, 第一个参数是上传的 URL, 第二个参数是文件在 HTML 表单
中的名称
var error = activeX.Post(uploadUrl, fileName);
```

### 3.3. 签章操作:

3.3.1. Error SingleSign(int pageNumber, double left, double bottom, double width, double height)

功能:

单页签章

参数:

pageNumber	页码, 从 1 开始
left	签章的左坐标
bottom	签章的底坐标
width	签章的宽度
height	签章的高度

返回:

错误码

示例:

```
var error = activeX.SingleSign (1, 100, 100,100,100);
```

3.3.2. Error MultiSignByString (string pageNumbers, double left, double bottom, double width, double height)

功能:

多页签章, 页码列表以字符串形式传输

参数:

pageNumbers	页码列表, 空格分割的数字字符串, 例如 "1 2 4"
left	签章的左坐标
bottom	签章的底坐标
width	签章的宽度
height	签章的高度

返回:

错误码

示例:

```
var error = activeX.MultiSignByString ("1 2 4", 100, 100,100,100);
```

### 3.3.3. Error CrossSignByString (string pageNumbers,

```
double width,
double height,
CrossSignMode mode,
double xPercent,
double yPercent,
double firstPercent
)
```

功能:

骑缝签章，页码列表以字符串形式传输

参数:

pageNumbers	页码列表，空格分割的数字字符串，例如“1 2 4”
width	签章大小
height	签章大小
mode	骑缝模式
xPercent	签章中心点距左侧的百分比
yPercent	签章中心点距上侧的百分比
firstPercent	第一部分的百分比

返回:

错误码

示例:

```
var error = activeX.MultiSignByString ("1 2 4", 100, 100, 1, 0.5, 0.5, 0.5);
```

### 3.3.4. Error TriggerSingleSign()

功能:

开始单页签章，效果与直接点击界面上的单页签章按钮相同

返回:

错误码

### 3.3.5. Error TriggerMultiSign ()

功能:

开始多页签章，效果与直接点击界面上的多页签章按钮相同

返回:

错误码

### 3.3.6. Error TriggerCrossSign ()

功能:

开始骑缝签章，效果与直接点击界面上的骑缝签章按钮相同

返回:

错误码

## 3.3.7. Error TriggerKeywordSign ()

功能:

开始关键字签章, 效果与直接点击界面上的关键字签章按钮相同

返回:

错误码

## 3.3.8. int SignatureCount

功能:

获取签章数量

## 3.3.9. string GetAllSignatureNumberString ()

功能:

获取空格分开的所有签章号码

## 3.3.10. int[] GetAllSignaturesNumber ()

功能:

获取所有签章号码的列表

## 3.3.11. ISignature GetSignature(int signatureNumber)

功能:

获取指定号码的签章信息

参数:

signatureNumber 签章号码

返回:

签章信息

说明:

ISignature 接口中包含签章的一些信息, 包括:

Number 签章号码

SignDateTime 签章时间

HostInformation 签发主机信息:

HostName 主机名

MainboardId 主板号

GetCertInfo(long infoType, string oid) 获取证书信息

infoType 信息类型——0: 证书主题

1: 颁发者主题

2: 序列号

3: 版本号

5: 有效期起始日期

6: 有效期终止日期

7: 证书扩展

oid 证书扩展的标识, 如果 infoType =7, 该输入项不能为空

示例:

```
var signature = activeX.GetSignature(1);
var result = signature.Number + "号签章\n" +
```

```
"签发于" + signature.SignDateTime + "\n" +  
"签发主机信息: \n" +  
"  主机名: " + signature.HostInformation.HostName + "\n" +  
"  主板号: " + signature.HostInformation.MainboardId + "\n" +  
"证书信息: \n" +  
"  证书序列号: " + signature.GetCertInfo(2, null) + "\n" +  
"  证书扩展: " + signature.GetCertInfo(7, "1.2.86.11.7.11") + "\n";
```

### 3.3.12. ISignature[] GetSignatures()

功能:

获取所有签章信息

返回:

所有签章信息列表

### 3.3.13. Error Delete(int signatureNumber)

功能:

根据对应的签章号码删除签章

参数:

signatureNumber 签章号码

返回:

错误码

## 3.4. 界面操作:

### 3.4.1. bool MenuBarVisible

功能:

设置以及获取菜单栏可见性, true 为可见, false 不可见

### 3.4.2. bool StatusBarVisible

功能:

设置以及获取状态栏可见性, true 为可见, false 不可见

### 3.4.3. bool ToolBarVisible

功能:

设置以及获取工具栏可见性, true 为可见, false 不可见

### 3.4.4. bool SideBarVisible

功能:

设置以及获取菜单栏可见性, true 为可见, false 不可见

## 3.5. 事件:

### 3.5.1. System.EventHandler AfterOpenFile

属性:

打开文件后事件

### 3.5.2. System.EventHandler AfterCloseFile

属性:

关闭文件后事件

### 3.5.3. System.EventHandler AfterSignPdf

属性:

签章完成后事件

### 3.5.4. System.EventHandler AfterDeleteSignature

属性:

删除签章后事件

## 3.6. 出错处理:

### 3.6.1. Error LastError

功能:

获取最后错误

### 3.6.2. string LastErrorDetail

功能:

获取最后错误的详细信息

### 3.6.3. string ErrorMessage(Error error)

功能:

将错误码转换为字符串

参数:

**error** 错误码

返回:

错误码对应的字符串