

优道 PDF 阅读控件

用户使用说明书

版本号:V2.4

www.uudoc.com

珠海市优道信息科技有限公司

目 录

第一章、使用前必读	- 2 -
1.1 如何安装优道 PDF 阅读控件	- 2 -
1.2 控件安装程序	- 3 -
1.3 控件的依赖文件	- 3 -
1.4 控件的卸载	- 4 -
1.5 版本升级历史	- 4 -
第二章、优道 PDF 阅读控件的功能	- 5 -
2.1 控件的功能列表	- 5 -
2.2 控件的运行环境	- 6 -
2.3 如何在网页中嵌入控件	- 6 -
2.4 服务器文档加密	- 9 -
2.5 服务器端版权控制	- 11 -
2.6 控件属性	- 13 -
2.7 控件方法	- 15 -
2.8 控件事件	- 17 -
2.9 错误码说明	- 18 -

第一章、使用前必读

1.1 如何安装优道 PDF 阅读控件

请仔细阅读安装说明，否则将不能正确安装和使用优道 PDF 阅读控件。以下为 Web 网络应用的安装方式：

在浏览器的地址栏上输入“<http://localhost/虚拟目录/程序文件>”回车后，页面自动提示安装，您将看到以下界面：



请按照提示信息点击安装链接执行控件的安装，安装完成之后需要刷新当前页面让浏览器重新加载控件。

如果控件进行了升级，也会出现该页面提示进行升级，您可以参考我们的演示程序对安装提示页进行修改。

如果是升级安装，可以不卸载原来的安装程序，新版本会直接覆盖旧版本。

从 v2.2 版本开始，为了兼容最新版本的 Windows 操作系统，我们使用了 msi 格式的安装包。如果您的电脑无法打开 msi 文件执行安装，很可能是您的操作系统没有安装 Windows Installer 服务，您可以从微软网站获取该软件，下载地址是：

<http://www.microsoft.com/zh-cn/download/details.aspx?id=25>

1.2 控件安装程序

axUdPdf32.msi 32 位 ActiveX 控件的安装程序，安装后可支持 32 位 IE 浏览器和桌面应用

axUdPdf64.msi 64 位 ActiveX 控件的安装程序，安装后可支持 64 位 IE 浏览器和 64 位的桌面应用程序

npUdPdf.msi 基于 NPAPI 标准的浏览器插件安装程序，安装后可以支持 FireFox、Chrome、Safari 等浏览器，目前只提供 32 位

我们也提供了基于 Visual Studio 2010 的 MSI 文件的安装程序制作示例，您可以根据自己的需要参考制作个性化的安装向导。您可以和我们的客服人员联系获取该示例的解决方案和源代码。

1.3 控件的依赖文件

优道 PDF 控件支持嵌入到 Windows 桌面应用程序和所有主流浏览器当中，如果在桌面应用程序或 IE 浏览器中使用，主要有以下依赖文件：

UdPdf32.ocx 优道 PDF ActiveX 控件 32 位主程序，只能给 32 位的应用程序和 32 位的 IE 浏览器调用。如果在 Windows 桌面程序中使用，使用前需使用安装工具（可参考我们提供的安装示例）或调用 regsvr32.exe 命令进行注册。

如果在 64 位系统中使用，需调用 c:\windows\SysWOW64\regsvr32.exe 进行注册

UdPdf64.ocx 优道 PDF ActiveX 控件 64 位主程序，只能在 64 位 Windows 操作系统中给 64 位的 IE 浏览器或应用程序调用，使用前需使用安装工具或调用 c:\windows\system32\regsvr32.exe 命令进行注册。

ATL100.dll 微软活动模板库程序，位于 Windows 系统目录 System32 下，使用前需复制到客户机器上。此文件为控件的必需依赖文件，不需要注册。

如果在 64 位 Windows 操作系统上，32 位的 atl100.dll 放在 Windows 系统的 SysWOW64 目录下，64 位的 atl100.dll 放在 Windows 系统目录 System32 下。

如果需要使用防截图功能，还要使用防截图组件，不需要防截图功能就不需要以下文件：

UdAntiCap32.exe 优道 32 位防截图保护程序

UdSafety32.dll 优道 32 位防截图安全组件

UdAntiCap64.exe 优道 64 位防截图保护程序

UdSafety64.dll 优道 64 位防截图安全组件

因为 64 位系统上可同时运行 32 位和 64 位的程序，所以就算是 32 位的安装包，64 位防截图组件也需一并安装

1.4 控件的卸载

打开控制面板 - 程序和功能

选择“UdPdf Control(32Bit)”后点击“卸载/更改”按钮，选择“是”，即可完成 32 位 ActiveX 控件的卸载。

选择“UdPdf Control(64Bit)”后点击“卸载/更改”按钮，选择“是”，即可完成 64 位 ActiveX 控件的卸载。

选择“UdPdf NPAPI Plugin”后点击“卸载/更改”按钮，选择“是”，即可完成 NPAPI 浏览器插件的卸载。

1.5 版本升级历史

V1.0 支持基本的 PDF 查看和打印功能，支持放大、缩小、旋转、搜索等基本功能

V1.1 添加了防截屏功能，支持嵌入到浏览器当中，可通过网页在线异步下载文件并查看

V1.2 添加了对 Windows 程序开发的支持，可嵌入到桌面窗口程序当中

V1.3 增加了绝对路径和相对路径的支持，能从不同地址下载并查看文件

V1.4 修正了一些 BUG，增加了页码和缩放比率的获取功能

V1.5 采用全新的渲染引擎，可对大文件进行快速查看，修改了不能进行查找和复制的问题

V1.6 增加了鼠标右键菜单、全屏显示功能，支持对属性设置和方法调用进行加密

V1.7 增加了动态水印功能，支持服务器端版权属性设置，支持对加密的 PDF 文件进行读取

V2.0 增加了对非 IE 浏览器的支持，可支持所有使用 NPAPI 标准的浏览器

V2.1 提供了 64 位版本，支持 64 位操作系统和 64 位 IE 浏览器

V2.2 采用了性能更好的防截图引擎，处理了 UAC 帐户控制和 IE 保护模式下发现的问题，提供了安装程序的制作示例

V2.3 增加了多语言界面的支持，目前支持英文、中文简体、中文繁体等用户界面

V2.4 增加了滚动、打印和文档关闭等事件处理接口，调整了示例程序

第二章、优道 PDF 阅读控件的功能

优道 PDF 阅读控件（UUDoc PDF Control）是一款支持反截屏功能的 PDF 文档数字版权保护控件，支持对 PDF 文档的在线加密保护，可支持多种不同类型的浏览器和桌面应用程序使用。控件通过简单的属性设置即可完成对文档的复制、打印和截图的授权控制，独特的服务器端配置模式，支持在服务器端对控件进行属性设置，防止了客户端通过脚本对版权控制功能进行篡改的可能。控件支持内存方式加载文档和加密传输方式远程加载文档，控件小巧可独立运行，不需要另外再安装其它庞大的 PDF 阅读器。

控件自带换页、查找、缩放、拖拽、旋转、适应页面、适应页宽等常用文档浏览器的功能，可以轻松的嵌入到各类网站、Web 应用系统和桌面应用程序当中，能有效的和您目前的应用进行无缝集成。控件支持 PDF 动态水印定义功能，能给您最强有力的文档阅读和数字版权保护功能支持。

2.1 控件的功能列表

序号	功能	说明
1	独立核心	为独立控件，客户端不需要再另外安装其它庞大的 PDF 阅读器软件
2	使用简单	通过属性的设置和方法的调用即可完成 PDF 文档阅读器的复杂功能
3	兼容性强	支持嵌入到各种常用的主流浏览器和桌面应用程序当中，支持各类编程语言进行调用
4	大文件支持	可以对大尺寸文档和图片进行很好的渲染
5	版权保护支持	可以通过属性设置完成对复制、截屏和打印等数字版权保护功能的控制，支持动态添加水印
6	服务器端配置	属性设置可以在服务器端进行，防止了在客户端对版权控制功能进行篡改的可能
7	高强度加密	采用国际权威的高强度加密算法在内存中完成加密解密，远程加密传输，不留临时文件
8	功能齐全	附带有换页、查找、放大、缩小、旋转、颜色定制等文档浏览器常用功能，支持通过 JavaScript 与网页进行交互
9	支持各种服务器	服务器端支持 Windows 及 Linux 的服务器，支持 HTTP、HTTPS、FTP 协议，支持各种编程语言
10	国际权威认证	软件经过权威机构安全数字签名认证，通过多种环境及杀毒软件测试，可安全运行

2.2 控件的运行环境

服务器端：支持 Windows 和 Linux 操作系统，支持任何的服务器编程语言，例如 asp, asp.net, php, java, python, ruby 等

支持使用 HTTP、HTTPS、FTP 协议从服务器远程加载文档，支持服务器身份验证

控件客户端支持的操作系统：WindowsXP/2003/2008/Vista/Win7/Win8

桌面开发支持的编程语言：VB、CSharp 、VB.NET、Delphi、C++等

支持32位及64位操作系统和浏览器

支持嵌入到基于 ActiveX 技术的 IE 浏览器和使用 NPAPI 技术的非 IE 浏览器中

支持英文、中文简体、中文繁体等多种语言用户界面

2.3 如何在网页中嵌入控件

您可以参考我们提供的[演示文件](#)在网页中嵌入优道 PDF 阅读控件：

控件的 PlogId 为:UdPdf.Viewer

控件的 ClassId 为:8C3D3705-1100-4D52-9D04-7E9A435CE220

控件的 ContentType 为: application/udpdf-plugin

```
<object id="pdfCtl" width="100%" height="600"
classid="CLSID:8C3D3705-1100-4D52-9D04-7E9A435CE220"
codebase="UdPdf/axUdPdf.msi#Version=2,4,0,0">
<param name="License" value="" />

<embed id="pdfPlugin" width="100%" height="600" type="application/udpdf-plugin"
pluginspage="UdPdf/npUdPdf32.msi#Version=2,4,0,0"
License=""
onReady="f_onReady" onDownload="f_onDownload" onPageChanged="f_onPageChanged"
onZoom="f_onZoom" onError="f_onError"></embed>
</object>
```

其中 axUdPdf32.msi 为控件 32 位 ActiveX 版的安装程序, 安装后控件支持 IE 访问; npUdPdf.msi 为 NPAPI 版的安装程序, 安装后支持 FireFox、Chrome、Safari 等等使用 NPAPI 标准的非 IE 浏览器使用。

判断控件是否安装及是否有新版本，并获取 PDF 控件对象：

```
function f_initPdf(){
    var udPdfCtl = null;
    if(window.ActiveXObject){//if IE
        var axobj = null;
        var setupFile = "UdPdf/axUdPdf32.msi";//PDF控件安装文件
        if(window.navigator.platform == "Win64"
        || window.navigator.cpuClass == "x64")//64位浏览器安装文件
            setupFile = "UdPdf/axUdPdf64.msi";

        try{
            axobj = new ActiveXObject('UdPdf.Viewer');
        }catch(e){}
        if(axobj) {
            if(f_checkNewVersion(axobj.GetVersion())){
                document.getElementById("udPdfDiv").innerHTML = "<a href=" + setupFile
                    + "' class='inst' target='_blank'>PDF控件有新版本v"
                    + UDPDF_VERSION + "， 点这里升级安装...</a>";
            }
            else{
                udPdfCtl = document.getElementById("udPdfCtl");
                return udPdfCtl;
            }
        }
        else{
            document.getElementById("udPdfDiv").innerHTML = "<a href=" + setupFile
                + "' class='inst' target='_blank'>您尚未安装PDF控件， 点这里进行安装...</a>";
        }
    }
    else
    {
        //NOT IE
        var setupFile = "UdPdf/npUdPdf.msi";
        if (navigator.plugins && navigator.plugins.length > 0){
            var plugin = navigator.plugins["UdPdf Plugin"];
            if(plugin){
                udPdfCtl = document.getElementById("udPdfPlugin");
                if(f_checkNewVersion(udPdfCtl.GetVersion())){
                    document.getElementById("udPdfDiv").innerHTML = "<a href=" + setupFile
                        + "' target='_blank' class='inst'>PDF控件有新版本v"
                        + UDPDF_VERSION + "， 点这里升级安装...</a>";
                }
            }
        }
    }
}
```



```
        else{
            return udPdfCtl;
        }
    }
    else{
        document.getElementById("udPdfDiv").innerHTML = "<a href='" + setupFile
            + "' target='_blank' class='inst'>您尚未安装PDF插件，点这里进行安装...</a>";
    }
}
else{
    document.getElementById("udPdfDiv").innerHTML = "<span class='inst'>当前浏览器不支持插
件<span>";
}
}
return null;
}

//版本比较，检查是否安装了新版本
function f_checkNewVersion(instVer){
    var newVer = UDPDF_VERSION.split(".");
    var curVer = instVer.split(".");
    if(parseInt(newVer[0]) > parseInt(curVer[0]))
        return true;
    if(parseInt(newVer[0]) == parseInt(curVer[0]) && parseInt(newVer[1]) > parseInt(curVer[1]))
        return true;
    if(parseInt(newVer[0]) == parseInt(curVer[0]) && parseInt(newVer[1]) == parseInt(curVer[1])
        && parseInt(newVer[2]) > parseInt(curVer[2]))
        return true;
    return false;
}

//初始化并获取控件对象
var ctl;
function f_onload() {
    ctl = f_initPdf();//失败返回null
    if(ctl){
        ctl.FileId="PdfSample"
        ctl.LoadFromWeb("GetFile.asp");
    }
}
```

属性的设置示例：

```
<script type="text/javascript">
    //根据复选框判断是否显示工具条
    function f_toolbarVisible() {
        if(ctl)
            ctl.ToolbarVisible = document.getElementById("chkToolbarVisible").checked
    }
</script>
```

方法的调用示例

```
<script type="text/javascript">
    //放大操作
    function f_zoomIn() {
        if(ctl)
            ctl.ZoomIn();
    }
</script>
```

事件的捕捉示例，兼容所有浏览器：

```
<script type="text/javascript" event="OnError(errorCode)" for="pdfCtl">
    f_onError(errorCode);
</script>
<script type="text/javascript">
    function f_onError(errorCode) {
        document.getElementById("info").innerHTML = "OnError事件，文档加载失败，错误码： "
        + errorCode;
    }
</script>
```

2.4 服务器文档加密

使用优道科技提供的文档加密工具 UdPdfTools.exe 对文档进行加密，加密后的文件会在原扩展名前加上 en 前缀。加密工具支持命令行方式。

加密时可以指定加密密码，加密后需要将密码的哈希值填写到控件的 HashKey 参数当中，将加密后的文件上传到服务器上，在控件页面根据所传递的参数将需要的加密文件读取并进行输出

若不想别人通过页面获取到参数的配置情况，可以使用此工具对属性配置内容进行加密，原配置字符串使用"&"分隔的键值对格式，多个参数之间使用"|"进行分隔，支持相应的属性或方法名称

也可在服务器上调用加密命令行工具直接完成普通文档文件到加密文件的转换，命令行使用说明：

参数说明(参数名之后接空格和要传入的参数值内容)：

参数名称	参数说明
-input	要进行加密保护的 PDF 文件名
-output	指定加密后的文件输出目录
-password	设置加密使用的密码, 会返回加密过的哈希码, 可放在 PDF 控件的 HashKey 属性当中
-encrypt	创建使用"&"进行分隔的加密字符串, 可用在 PDF 控件的 EncryptConfig 属性和 EncryptExecute 方法中使用, 也可对要传入的密码参数进行加密。
-help	显示 UdPdfTools.exe 的使用帮助信息

使用示例：

```
UdPdfTools -input c:\test.pdf -output "c:\hello world" -password 123abc
```

以下是对属性配置进行加密, 加密后的字符串可放在 EncryptConfig 属性当中

```
UdPdfTools -encrypt "EnableCopy=true&EnablePrint=false&ToolbarVisible=false"
```

以下是对方法进行加密, 支持"&"分隔的多个参数, 加密后的字符串可以用 EncryptExecute(string) 函数来执行

```
UdPdfTools -encrypt "LoadFromWeb&sample.asp|123456"
```

如果是对密码参数进行加密, 传入密码参数时需要添加 "[Encrypt]" 前缀。

例如: UdPdfTools.exe -encrypt 123456

The encrypt confing string is:

4019DA9B613B82F72186203AA9ABE0C7

然后可以添加 "[Encrypt]" 前缀使用加密密码的方式打开 pdf 文档:

```
ctl.LoadFromWeb("getfile.aspx", "[Encrypt]4019DA9B613B82F72186203AA9ABE0C7", "post");
```

ActiveX 控件的 FileId 会被提交回服务器, 可以根据该值输出不同的文件, 以下是 C# 的服务器 ASP.NET 示例代码, 也可方便的用其它语言编写:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request.Form["uudoc_getfile"] == "1") //如果是提交请求文件
    {
```

```
//可以通过HashKey的设置进行HTTP头安全校验
if (Request.Headers["UdPdf-HashKey"] != "3CE59E850E2C590AFD6E4D3516BA3F1B")
    return;
//根据文件ID获取不同的加密文件或原始文件，需要根据实际情况编写自己的业务逻辑
if (Request.Form["uudoc_encrypted"] == "1") //是否加密读取方式
{
    //读取加密过的PDF文件
    Response.WriteFile(Server.MapPath("upload/" + Request.Form["uudoc_fileid"] + ".enpdf"));
}
else
{
    //读取原始PDF文件
    Response.WriteFile(Server.MapPath("upload/" + Request.Form["uudoc_fileid"] + ".pdf"));
}
Response.End();
}
```

可向服务器提交的参数有：

参数名称	参数说明
uudoc_getfile	1表示开始从后台读取文件
uudoc_fileid	文件 Id 值，用来判断要读取的文件名称，对应前端控件设置的 FileId 属性值
uudoc_encrypted	1表示采用的是加密文件读取方式

2.5 服务器端版权控制

优道 PDF 控件支持服务器端的属性设置，在服务器端进行属性设置后客户端的部分版权控制功能将不再有效，从而更安全的保护了 PDF 文档的安全性。

服务器端使用 http 头的方式对控件属性进行设置，支持的 http 头设置的名称为 UdPdf-Config 和 UdPdf-EncryptConfig。

UdPdf-Config 配置 的参数值为使用 “&” 分隔 的字符串，例如 "EnableCopy=true&EnablePrint=false&ToolbarVisible=false"。

UdPdf-EncryptConfig 配置 的参数值为以上加密后的字符串内容，具体加密方法请

参看上一节中 UdPdfTools.exe 工具的使用

在 C#/VB.NET 中可以使用 Response.AppendHeader 进行 http 头的设置，例如：

```
Response.AppendHeader("UdPdf-Config", "EnableCopy=false&EnablePrint=false");
```

或者：

```
Response.AppendHeader("UdPdf-EncryptConfig",  
"74321C08CA9D4B127A06C1B07A78A8B1344B4A88E7F2517D90A04FFEB46F7A9FF99E3DD7B5  
2282441B6A3FCC6E9BA168");
```

在 Java 语言中可以使用 response.setHeader 进行 http 头的设置，例如：

```
response.setHeader("UdPdf-Config ", "EnableCopy=false&EnablePrint=false");
```

或者：

```
response.setHeader("UdPdf-EncryptConfig ", "***加密后的字符串***");
```

使用 http 头进行属性设置后下列和版权保护相关的属性在客户端的设置将不再有效：

服务器端设置影响的属性	参数说明
HashKey	加密密码 Hash 值，该值同时会被以 http 头的方式提交回服务器
EnableCopy	是否允许复制
EnablePrint	是否允许打印
EnableCapture	是否允许屏幕截图
ToolbarVisible	是否显示工具栏
RemoveWaterMark	移除水印的方法，不允许再移除水印

ActiveX 控件向服务器请求文件的时候同时会以 http 头的方式提交下列值：

UdPdf-Version: 优道 PDF 阅读控件的版本号

UdPdf-HashKey: HashKey 属性，可用作请求合法性校验

UdPdf-UrlType: 如果使用相对路径来请求的这里值为 0，使用绝对路径值为 1，可以用来控制是否只允许相对路径来获取文件

在 C#/VB.NET 语言中可以使用 Request.Headers["UdPdf-UrlType"]来获取到相关的 http 头的值，Java 中可以使用 request.getHeader("UdPdf-UrlType")来获取。

2.6 控件属性

属性名称	类型	属性说明
Width	整数	控件区域宽度
Height	整数	控件区域高度
License	字符	获取或设置控件的注册码信息
HashKey	字符	获取或设置密码的 MD5 Hash 值，可以用 UdPdfTools.exe 命令行工具加 -password <password> 获取
FileId	字符	获取或设置文件的 ID 值，该值会提交回服务器端
Encrypted	布尔	获取或设置是否采用加密方式加载文档，默认为 TRUE，加载加密过的文档；若设置为 FALSE，则加载原始未加密的文档文件
EnableCapture	布尔	获取或设置是否允许进行屏幕截图，默认为 TRUE
EnableCopy	布尔	获取或设置是否允许复制文档内容，默认为 TRUE
EnablePrint	布尔	获取或设置是否允许打印文档，默认为 TRUE
EnablePrint	布尔	获取或设置是否允许打印文档，默认为 TRUE
ToolbarVisible	布尔	获取或设置是否显示文档工具栏，默认为 TRUE
HasBookmark	布尔	获取文档是否含有书签
BookmarkVisible	布尔	获取或设置是否显示书签栏，默认为 TRUE
DocBgColor	颜色	获取或设置文档的背景颜色
ShowContextMenu	布尔	获取或设置是否显示右键上下文菜单，默认为 TRUE
TotalPage	整数	只读属性，获取文档的总页码
CurrentPage	整数	只读属性，获取文档的当前页码
DefaultZoom	浮点	获取或设置文档打开时的默认缩放比例，默认为 -1，表示适合内容，-2 表示适合宽度，100 表示实际大小显示
DefaultDisplayMode	整数	获取或设置文档打开时的默认显示模式，默认为 0，1 为单页显示，4 为连续显示

UILanguage	字符	获取或设置控件所使用的界面语言，en 为英文、cn 为中文简体、tw 为中文繁体，留空为根据操作系统自动判断所使用的语言
EncryptConfig	字符	获取或设置文档打开时的加密配置字符串，加密需要使用 UdPdfTools.exe 命令行工具，参看 UdPdfTools.exe 相关说明
EnableShowAbout	布尔	获取或设置是否按 Ctrl+Shift+A 时显示关于对话框，默认为 true 显示
IsReady	布尔	只读属性，获取文档是否已加载完成，状态已就绪

2.7 控件方法

方法名称	参数说明	方法说明
void Zoom(float zoom)	zoom:缩放比例, 实际大小为100.00	对文档进行指定比例缩放
void ZoomIn()	无	放大文档视图
void ZoomOut()	无	缩小文档视图
void ZoomFitWidth()	无	以适应宽度方式显示文档
void ZoomFitPage()	无	以适应页面大小方式显示文档
void ZoomActual()	无	以实际大小显示文档
float GetZoom()	无	获取文档的缩放比例
void RotateLeft()	无	左旋转 90°, 即逆时针旋转
void RotateRight()	无	右旋转 90°, 即顺时针旋转
void FindNext(string findText)	findText:要搜索的内容	查找下一个字符
void FindPrev(string findText)	findText:要搜索的内容	查找上一个字符
void ZoomIn()	无	放大文档视图
void GoToPage()	无	进行换页操作
void GoToPageNo(int no)	no:要转到的页码	转到指定的页码
void GoToNextPage()	无	转到下一页
void GoToPrevPage()	无	转到上一页
void GoToFirstPage()	无	转到第一页
void GoToLastPage()	无	转到最后一页
void ToggleFullScreen()	无	全屏幕显示模式转换, 进入或退出全屏状态
void Print()	无	打印当前文档
void LoadFromWeb(string url, string password, string method)	加载地址, 如果为空字符串表示当前页, 以 http 开头则为绝对路径, 否则为相对路径 password:PDF 文件的密码 (可选) method:请求方式, Post 或 Get, 默认为 Get (可选)	向 WEB 页面提交 POST 请求加载文档文件, 提交参数为 uudoc_getfile=1 和 uudoc_fileid Web 网络版支持该函数

void LoadFromData(Byte[] pData, string password)	pData: 字节数组 password:PDF 文件的密码 (可选)	从字节数组中加载文档 桌面控件版支持该函数
void LoadFromFile(string fileName, string password)	fileName: 文件路径全名 password:PDF 文件的密码 (可选)	根据文件名加载指定文档
void SetServerUser(string username, string password)	username: 服务器身份验证使用用户名 password: 服务器用户登录密码	如果需要, 设置服务器身份验证的用户名和密码
int GetErrorCode()	无	获取异常的错误码。 详见错误码说明
int GetFileSize()	无	获取文档文件的字节大小
void EncryptExecute()	无	执行加密字符定义的命令, 具体参见 UdpdfTools 相关说明
void Close()	无	关闭打开的文档
string GetVersion()	无	获取控件的版本号
string Hash(string str)	要加密转换的字符串	获取字符串的 Md5 Hash 码, 可用于 HashKey 属性
void ScrollTo(int x,int y)	无	滚动文档显示位置到指定像素值
void SetWaterFont(string fontName, long fontSize, long fontColor, bool bold, bool italic, bool underline, bool strikeOut)	fontName:字体名称 fontSize: 字体大小, 单位 pt fontColor:字体颜色值 bold:是否粗体显示(可选) italic:是否斜体显示(可选) underline:是否添加下划线(可选) strikeOut:是否添加删除线(可选)	设置水印文本的字体格式
long AddWaterText(string text, long x, long y, long transparent, long rotate, long repeatX, long repeatY)	text:水印文字 x: 水印位置的横坐标 y: 水印位置的纵坐标 transparent: 透明度 (0-100), 100 为完全透明, 默认为 0 不透明(可选) rotate:逆时针旋转角度, 默认 0(可选) repeatX:横轴重复显示的间距, 默认为 0 不重复(可选) repeatY:纵轴重复显示的间距, 默认为 0 不重复(可选)	添加文字水印, 返回值为所添加水印的索引号
long RemoveWaterMark(long index)	index:要删除的水印的索引号	删除指定的水印, 成功返回 0

在需要传入密码的方法中,例如 LoadFromWeb, LoadFromFile, SetServerUser 等方法,可以先使用 UdPdfTools.exe 工具对密码进行加密,然后使用 “[Encrypt]” 前缀(不分大小写)支持传入加密过的密码,防止别人获取到密码直接对 PDF 文档进行解密。

例如某个 pdf 文档的默认加密密码是: 123456, 使用命令行工具:

```
UdPdfTools.exe -encrypt 123456
```

获取到加密后的密码是: 4019DA9B613B82F72186203AA9ABE0C7

然后可以添加[Encrypt]前缀使用加密密码的方式打开 pdf 文档:

```
ctl.LoadFromWeb(“getfile.aspx”, “[Encrypt]4019DA9B613B82F72186203AA9ABE0C7”, “post”);
```

2.8 控件事件

事件名称	参数说明	事件说明
OnReady()	无	文档加载完成触发该事件
OnDownload(int totalSize, int downSize)	totalSize: 总共要下载的文件大小; downSize: 已经下载的字节大小	文档下载过程中触发该事件
OnPageChanged(int currentPage, int totalPage)	currentPage: 当前页码; totalPage: 总页码	换页时触发该事件
OnZoom(float zoomScale)	zoom: 缩放比例	缩放操作时候触发该事件
OnError(int errorCode)	errorCode: 错误码。详见以下错误码说明	出现异常时触发该事件
OnPrint	无	打印时触发该事件
OnHScrolled(int x)	x: 水平移动到的位置	水平移动文档查看区域后触发该事件
OnVScrolled(int y)	y: 垂直移动到的位置	垂直移动文档查看区域后触发该事件
OnClosed	无	文档关闭后触发该事件

2.9 错误码说明

错误码值	异常说明
0	一切正常
1	在虚拟机或远程桌面中打开文档
2	使用非 HTTP 访问模式
3	加载文档安全组件失败
4	文件格式错误
5	加载文件失败
6	请求服务器文件失败

如果您还有其它的问题，请通过以下方式和我们联系，我们竭诚为您服务：

服务热线：0756-2132796

电子邮件：support@uudoc.com

技术 QQ：13063574

商务 QQ：81909064

珠海市优道信息科技有限公司保留所有权利。