

ⓒ2014重庆软航科技有限公司 【普通】

NTKO大文件上传控件

——集成开发参考【标准】V4.0

**接口手册**

重庆软航科技有限公司

编制：NTKO 日期：2014.05

# 软件最终用户许可协

重要须知 ━ 请认真阅读：本《最终用户许可协议》（以下称《协议》）是您（个人或单一实体）与 我们之间有关NTKO大文件上传控件软件产品的法律协议。本“软件产品”包括计算机软件，并可能包括相关媒体、印刷材料，和“联机”或电子文档（“软件产品”）。本“软件产品”还包括对我们提供给您的原“软件产品”的任何更新和补充资料。任何与本“软件产品”一同提供给您的并与单独一份最终用户许可证相关的软件产品是根据那份许可协议中的条款而授予您。您一旦安装、复制、下载、访问或以其它方式使用“软件产品”，即表示您同意接受本《协议》各项条款的约束。如您不同意本《协议》中的条款，请不要安装或使用“软件产品”；

NTKO大文件上传控件是受版权保护的。如果您得到的是未注册版本，您还没有获得运行授权。请在测试完成之后从您的机器中删除。如果您得到的是注册版本，您可以在授权协议许可的范围内使用本软件。

您不能分发，销售，以及非法破解本软件。也不能超出协议许可的范围使用本软件。否则，将被视为侵犯了我们的知识产权。

对反向工程 (Reverse Engineering)、反向编译 (Decompilation)、反汇编 (Disassembly) 的限制。您不得对本“软件产品”进行反向工程、反向编译 (decompile) 或反汇编 (disassemble)，但尽管有这项限制，如适用法律明示允许上述活动，则不在此限。另外，我们不对您在使用此软件的过程中对您的机器或者数据造成的损失负责。

# 功能介绍

## NTKO大文件上传控件功能简介

NTKO大文件上传控件，采用多线程技术，提供专业的大文件上传解决方案。可以轻松迅速的将几百M甚至2G以上的文件上传到WEB服务器，并支持断点续传上载。NTKO大文件上传控件是目前国内唯一支持超过2G以上大文件上传的产品。

上传到WEB服务器的文件，通过web服务器编程支持，还可以采用迅雷，FLASHGET等支持断点续传下载的工具软件进行多线程，可续传的下载。NTKO大文件上传控件可支持除了Domino服务器之外的大部分WEB服务器(因domino数据库如果保存大量大附件性能明显下降，不予支持)，并且可以在浏览器中通过JScript或者VBScript使用控件提供的属性和方法进行操作

## 详细功能

### 平台支持功能

|  |  |  |
| --- | --- | --- |
| ★ | 支持多种后台web服务器和开发语言 | 支持IIS,WebLogic,Apache等多种后台web服务器和ASP,JSP,C#,VB.NET,PHP等多种后台开发语言。LotusDomino服务器除外。 |

### 网络及协议功能支持

|  |  |  |
| --- | --- | --- |
| ★ | 多文件上传支持 | 可以一次按顺序上传多个超大文件。支持2G以上超大文件。 |
| ★ | 多线程上传，性能更高 | 控件完全采用多线程方式上传，可以更好的改善用户体验，达到更高性能。 |
| ★ | 支持UTF-8网页编码 | 通过IsUseUTF8Data属性，可以支持UTF-8编码网页的智能提交。 |
| ★ | 支持HTTP协议，HTTPS OVER SSL协议 | 对HTTPS协议的支持使得可以创建更加安全的应用。并且自动支持Session Cookie。使用当前页面的Session Cookie和服务器进行交互。 |
| ★ | 支持进度事件信息及百分比显示 | 控件通过事件触发，支持实时显示单个文件及所有文件的上传实时进度信息及百分比显示。 |

### 编程支持

|  |  |  |
| --- | --- | --- |
| ★ | 支持简洁高效的编程接口 | 可使用Javascript和VBScript对控件进行编程。 |
| ★ | 支持过滤文件大小，文件总数量，以及文件类型 | 通过控件的BeforeFileAdded事件，用户可以限定允许添加到控件的文件类型，大小，以及数量。 |

### 安全及其他支持

|  |  |  |
| --- | --- | --- |
| ★ | 安全可靠 | 系统不会自动增加未经用户授权的文件。上传文件必须经过用户许可。 |

使用NTKO大文件上传控件，可以大大缩短整个系统的开发时间，降低系统的开发成本，提高产品的质量和用户体验。  ……

## 需要的软件

正确运行本控件需要客户机安装Windows xp及以上操作系统，Internet Explorer 6.0及以上版本。

## 支持的Web服务器

支持几乎所有的支持HTTP/1.1标准的Web服务器。LotusDomino服务器除外。

## HTML Object标识符

*  注意：某些特殊版本比如OEM可能会具有不同的CLSID值。

<object id="Ntko.LargeFileUploader" classid="clsid: 882C96AC-B79E-4f50-A7A6-6E5E42431A05" codebase="NtkoLFUploader.cab#version=3,0,0,1" width="200" height="100">

<param name="BackColor" value="16744576">

<param name="ForeColor" value="16777215">

</object>

# 设计架构及协议

## 总体设计思路

### 突破HTTP上传协议限制

因HTTP上传协议本身不支持续传上载，因此，要突破这个限制，我们采用了对文件分块的形式进行上传。综合考虑各种因素并进行优化，文件块大小被固定设定为128K字节。

### 服务器端文件存储

因为是分块上传，因此，服务器需要通过数据库来记录每个文件的状态。并且我们建议如示例代码所示将每个文件分块存储以加快上传的速度和保证存储的可靠性。同时，为了避免上传相同内容的文件，我们还会实现对每个文件的内容进行散列，并提取其MD5值作为特征码。每个文件的所有块都将保存在同一个目录，目录名是文件的特征码。

### 和服务器的交互

为了更高效的实现上传，并且能支持各种web服务器，我们设计了控件和web服务器的交互协议。具体请参考后面的介绍。具体流程是：

* 控件要上传一个文件时，先提取其MD5特征值，并根据这个特征值向服务器查询这个文件在服务器上的状态（由QueryFileStatusURL属性指明接受查询的URL），服务器通过这个URL，提取MD5特征值，并判断该文件的状态，如是否存在，是否已经上传完成，如果部分完成，已经完成了哪些部分。并将这些状态返回给控件。
* 控件接收到服务器信息之后，根据需要将未完成的文件块上传到服务器URL(由控件的BeginSaveToURL方法的第一个参数指定)。
* 控件发送完所有文件块之后，会通知服务器已经完成上载（通过FinishedUploadURL属性指定的URL进行通知）。服务器应该将文件状态标识为已完成。

## 查询文件状态协议

### 作用：

由控件发送请求，查询一个文件状态的状态。该URL由控件的QueryFileStatusURL属性指定，并应该接受HTTP POST请求。

### 提交的协议域

控件通过此URL提交的POST请求包含的域如下：

* NTKOLF\_FileMd5ID：文件的md5散列值。32字节长的字符串。
* NTKOLF\_FileOriginalName：文件的原始文件名，不包含路径部分。
* NTKOLF\_FileSize:：文件的总体大小。

如：

NTKOLF\_FileMd5ID=XXXXXXXXXXXXXXXXXXXXXX&

NTKOLF\_FileOriginalName=测试大文件.exe& NTKOLF\_FileSize=15535

### 返回协议域定义

域之间以&分隔。域顺序如下：

* NTKOLF\_FileExist域：标识NTKOLF\_FileMd5ID指定的文件是否存在。

如果NTKOLF\_FileMd5ID指定的文件存在，则值是true，否则是false。都是字符串。如果该域的值是true，那么必须指定后面的NTKOLF\_FileFinished域。

比如：NTKOLF\_FileExist=true

* NTKOLF\_FileFinished域：标识NTKOLF\_FileMd5ID指定的文件是否已经上传结束。如果该域的值是false，那么必须指定后面的NTKOLF\_ExistFileParts域。

比如：NTKOLF\_FileFinished=false

* NTKOLF\_ExistFileParts域：标识服务器上已经存在的该文件的文件块的序号。

比如：NTKOLF\_ExistFileParts=0,5,7

所有返回值将被控件接收，并可以通过控件的方法获得。以便开发者进行调试。如果返回的HTTP状态码不是200 OK，控件认为查询失败。

几个可能的完整返回协议如下：

比如，如果文件不存在，可以直接返回：

NTKOLF\_FileExist=false

如果文件存在，且已经上传完成，则可以返回：

NTKOLF\_FileExist=true& NTKOLF\_FileFinished=true

如果文件存在，且上传未完成，服务器上已经有文件块0,2,3，则可以返回：

NTKOLF\_FileExist=true& NTKOLF\_FileFinished=false& NTKOLF\_ExistFileParts=0,2,3

## 上传文件块协议

### 作用：

由控件发送请求给服务器，上传一个文件块。该URL由控件的BeginSaveToURL方法的第一个参数指定。控件采用HTTP Multipart form POST的协议（也即浏览器上传文件的协议），因此要求后台能够正确处理这种协议。

### 提交的协议域

控件通过此URL提交的POST请求包含的域如下：

* NTKOLF\_FileMd5ID：文件的md5散列值。32字节长的字符串。
* NTKOLF\_FileOriginalName：文件的原始文件名，不包含路径部分。
* NTKOLF\_FileSize:：文件的总体大小。
* NTKOLF\_FileBlockSeqNo：此次上传的文件块的序号，从0开始；
* NTKOLF\_FileBlockSize：此次上传的文件块的大小。一般除最后一块，都是128K字节。即131072字节。
* 用户自定义的其他域（由控件的BeginSaveToURL方法的第3个参数指定）。
* 上传的文件块。上传块文件的域名由控件的BeginSaveToURL方法的第2个参数指定。

### 返回协议域定义

任意。可以返回任何值。但是，服务器必须返回数据以便控件接收。该返回值将被控件接收，并可以通过控件的方法获得。以便开发者进行调试。如果返回的HTTP状态码不是200 OK，控件认为上传失败。否则认为成功。

## 通知服务器完成上载协议

### 作用：

由控件发送请求，通知服务器一个文件已完成上载。该URL由控件的FinishedUploadURL属性指定，并应该接受HTTP POST请求。服务器应该标记文件状态为已完成。

### 提交的协议域

控件通过此URL提交的POST请求包含的域如下：

* NTKOLF\_FileMd5ID：文件的md5散列值。32字节长的字符串。
* NTKOLF\_FileOriginalName：文件的原始文件名，不包含路径部分。
* NTKOLF\_FileSize:：文件的总体大小。

如：

NTKOLF\_FileMd5ID=XXXXXXXXXXXXXXXXXXXXXX& NTKOLF\_FileOriginalName=测试大文件.exe& NTKOLF\_FileSize=15535

实际上，提交的协议域和查询文件状态协议是完全相同的。

### 返回协议域定义

任意。可以返回任何值。但是，服务器必须返回数据以便控件接收。该返回值将被控件接收，并可以通过控件的方法获得。以便开发者进行调试。如果返回的HTTP状态码不是200 OK，控件认为通知失败

# 控件主要属性参考

## 界面相关属性

### BackColor

BackColor：背景颜色；可读写；

设置：Obj.BackColor = OLE\_COLOR

说明：OLE\_COLOR是一个BGR (Blue, Green, Red)数值。

BGR value = (blue \* 65536) + (green \* 256) + red

### ForeColor

ForeColor：前颜色；可读写；；

设置：Obj. ForeColor= OLE\_COLOR

说明：OLE\_COLOR是一个BGR (Blue, Green, Red)数值。

BGR value = (blue \* 65536) + (green \* 256) + red

## 网络协议相关属性

### IsUseUTF8URL

IsUseUTF8URL：读取/设置是否使用UTF-8传输URL。

### IsUseControlAgent

IsUseControlAgent：读取/设置向服务器提交请求时是否使用“NTKO ATTACH MANAGE CONTROL AGENT”.此信息也可用于判断是否是控件请求的数据。默认为FALSE，使用浏览器AGENT。

### IsUseUTF8Data

IsUseUTF8Data：读取/设置是否使用UTF-8在智能提交中传输网页数据。如果网页编码设置为UTF-8，需要设置此属性，以便控件自动使用UTF-8编码进行网页数据的提交。

### IsDirectConnect

IsDirectConnect：是否使用直接连接连接服务器。【而不是默认的采用浏览器设置】。对于某些老版本的代理服务器环境，设置此选项可以避免因浏览器自动检测代理服务器引起的问题

## 行为控制属性

### QueryFileStatusURL

设定控件用于查询状态的URL。

### FinishedUploadURL

设定控件用于通知完成的URL。

### IsUploading

只读。返回控件是否正在上传文件。

### LastQueryStatusRetMes

最后一次查询状态URL，服务器返回的信息。

### LastFinishedUploadRetMes

最后一次通知服务器完成URL返回的信息。

### CancelLastCommand

CancelLastCommand：只允许写入。设定该值将取消最后的操作。可用于添加文件之前的BeforeFileAdded事件。

### FilesCount

FilesCount：只读。返回当前控件中文件的数量。

### StatusMessage

返回最后操作的结果字符串,只读.

### StatusCode

返回最后操作的结果代码,只读.

0: 成功; 1:文件错误; 2:网络错误; 3: 内存错误; 4:参数错误; 100：其他错误。

10、 ProductCaption，ProductKey

标题栏名称和密钥。如果密钥正确，将可以正常使用。否则，将显示演示版本标识，且会到期。

### MakerCaption，MakerKey

仅对开发商授权版本有效。设定开发商授权密钥。

# 控件主要方法参考

## AddLocalFile

AddLocalFile(

[in, optional] VARIANT FileName,

[in, optional] VARIANT PermitMultiSelect,

[in, optional] VARIANT FileType

)

FileName：初始文件或者目录；

PermitMultiSelect：是否允许选择多个文件；

FileType：默认的文件过滤类型。比如：”\*.gif;\*.jpg;\*.bmp”。多种类型以分号分隔。

* 注意：如果用户取消，或者控件正在上传，会抛出异常。
* 注意：此方法会触发OnStatusChange，BeforeFileAdded和/或OnLocalFileAdded事件。
* 技巧：此方法可以被js调用，用来过滤文件类型或者结合其他系统使用，为其他系统选择文件。比如，在BeforeFileAdded事件中，可以通过判断扩展名，或者当前已经存在的文件数量(FilesCount属性)，或者大小来过滤是否允许增加该文件。如果不允许增加，可以直接设定CancelLastCommand为true取消增加文件。

## BeginSaveToURL

BeginSaveToURL(

[in] VARIANT URL,

[in] VARIANT FileFieldName,

[in,optional] VARIANT CPARA

);

* 注意：如果控件正在上传，会抛出异常。
* 注意：如果参数错误，或者查询文件状态或者服务器通知URL属性没有正确设定，或者控件授权信息错误，会抛出异常。
* 注意：此方法会触发OnStatusChange，OnFileProcessStatusChange ，OnManagerStatusChange，OnOneFileUploadFinished及OnSaveToURLFinished事件。

该函数使用HTTP Multipart form POST将文件保存到URL。该调用将触发多个事件，请参考“主要事件参考”部分获得详细信息。

URL可以是相对于控件所在的网页的相对URL或者是绝对URL。

FileFieldName相当于一个HTML页面的Form中的<input type=file name=”FileFieldName”>中的FileFieldName。

可选参数CPARA是可以让用户定义没有在表单中出现的参数。该参数是以”&”分隔的参数－值对。一个可能的CPARA值为：”key=thiskey&type=word&load=mywave”。

* 注意：对于CPARA参数，因为控件接受以”&”分隔的参数－值对，所以，如果在值中，有”&”符号出现，应该替换成”%26”；相应的，所以如果在参数的值中有”%”符号出现，应该替换成”%25”。

比如下面的代码将一个值中所有的”%”符号和”&”符号进行替换，并返回替换后的字符串：

function encodeObjValue(value)

{

var t;

t = value.replace(/%/g,"%25");

return(t.replace(/&/g,"%26"));

}

## StopSaveToURL

StopSaveToURL()方法停止文件上传。

## GetFile

AttachFile GetFile([in] long index)

获取指定索引index所在的文件类。返回AttachFile对象。可以通过结合这个方法及FilesCount属性，循环获得控件中所有文件的详细信息。

## GetLastUploadRetMes

GetLastUploadRetMes([in] long index)

获取指定索引index的线程的最后上传信息。取值0-4。

## Reset

Reset();方法清除控件中已经加入的所有文件。

*  注意：如果控件正在上传，会抛出异常。

# 控件主要事件参考

## OnStatusChange

OnStatusChange([in] BSTR StatusText,[in] long StatusCode);

该事件在控件发生错误或者其他事件改变时触发。第一个参数是状态信息，字符串类型。第二个参数是状态码。数字。具体如下：

ErrorCode =0：成功

ErrorCode =2：标识网络传送错误

ErrorCode =3：标识内存错误

ErrorCode =4：标识传入控件方法/属性的参数错误

ErrorCode =5：查询文件状态或者服务器通知URL属性没有正确设定.

ErrorCode =6：标识控件因授权错误而无法执行指定操作。

ErrorCode =10：标识用户取消

ErrorCode =11：标识控件正在忙碌中。无法执行指定操作。

ErrorCode =100：标识其他未知错误

以下是代码示例：

<object id="Ntko.LargeFileUploader" classid="clsid: 882C96AC-B79E-4f50-A7A6-6E5E42431A05" codebase="NtkoLFUploader.cab#version=3,0,0,1" width="200" height="100">

<param name="BackColor" value="16744576">

<param name="ForeColor" value="16777215">

</object>

<!-- 控件状态改变事件 -->

<script language="JScript" for="Ntko.LargeFileUploader" event="OnStatusChange(mes,code)">

document.all("status").innerHTML = mes + ",code:"+code;

</script>

## BeforeFileAdded

BeforeFileAdded(FilePath, FileName, FileSize);

文件增加之前的事件。第一个参数标识文件全路径，字符串类型。第二个参数标识文件名，字符串类型。第三个参数标识文件大小，数字类型。

您可以判断这些信息是否满足需求。如果不满足，可以通过CancelLastCommand设定为true来取消对该文件的增加。

<script language="JScript" for="Ntko.LargeFileUploader" event="BeforeFileAdded(fullname,fname,fsize)">

if(fsize<1000000)

{

alert(fullname + "的大小小于1M，请选择1M以上的文件");

CancelLastCommand = true;

}

</script>

## OnLocalFileAdded

OnLocalFileAdded(FilePath, FileName, FileSize);

该事件在增加了本地文件之后被触发。第一个参数标识文件全路径，字符串类型。第二个参数标识文件名，字符串类型。第三个参数标识文件大小，数字类型。例如：

<script language="JScript" for="Ntko.LargeFileUploader" event="OnLocalFileAdded(fullname,fname,fsize)">

document.all("AllFilesInfo").innerHTML += "文件:"+fname + ",大小:"+fsize+"字节.<br>";

</script>

## OnFileProcessStatusChange

OnFileProcessStatusChange(AttachFile,StatusText,IsPersent, PersentNumber);

当单个文件上传状态发生改变时触发该事件。可以通过该事件显示文件的上传状态。

第一个参数AttachFile标识一个AttachFile类实例。请参考第7章有关该类的介绍。

第二个参数标识状态文字，字符串类型。

第三个参数是否是说明状态的百分比状态的事件。布尔类型。

第四个参数在第三个参数为true的情况下，说明处理状态的百分比。数字类型。比如：

<script language="JScript" for="Ntko.LargeFileUploader" event="OnFileProcessStatusChange(AttachFile,mes,isPersent,Persent)">

document.all("FileProcessStatus").innerHTML = mes ;

if(isPersent)

{

var totalWidth = parseInt(document.all("FilePersentDisplayTotalWidth").style.width); //因为width返回的是类似250px，有单位，所以需要转换

document.all("FilePersentDisplay").style.width = Persent\*totalWidth/100;

document.all("FilePersentDisplayText").innerHTML = Persent + "%";

}

</script>

…..

<!-- 显示文件的处理状态 -->

<span style="font-size:10pt;color:blue;">文件分析/上传状态:</span> <br>

<span id="FilePersentDisplayTotalWidth" style="background-color:#FFFFFF;color:red;width:250;border-color:red;border-width:1px;border-style:solid;">

<span id="FilePersentDisplay" style="background-color:#0000FF;color:red;width:0;borderColor:#0000FF">

</span>

</span><span id="FilePersentDisplayText" style="font-size:10pt;color:blue;"></span><br>

<div id="FileProcessStatus" style="font-size:10pt;color:black;">

</div>

## OnManagerStatusChange

OnManagerStatusChange (StatusText,IsPersent, PersentNumber);

当总体上传状态发生改变时触发该事件。可以通过该事件显示总体的上传状态。

第一个参数标识状态文字，字符串类型。

第二个参数是否是说明状态的百分比状态的事件。布尔类型。

第三个参数在第三个参数为true的情况下，说明处理状态的百分比。数字类型。比如：

<script language="JScript" for="Ntko.LargeFileUploader" event="OnManagerStatusChange(mes,isPersent,Persent)">

document.all("ManagerProcessStatus").innerHTML = mes ;

if(isPersent)

{

var totalWidth = parseInt(document.all("ManagerPersentDisplayTotalWidth").style.width);

document.all("ManagerPersentDisplay").style.width = Persent\*totalWidth/100;

document.all("ManagerPersentDisplayText").innerHTML = Persent + "%";

}

…..

<!-- 显示总体上传状态 -->

<span style="font-size:10pt;color:blue;">总体上传状态:</span> <br>

<span id="ManagerPersentDisplayTotalWidth" style="background-color:#FFFFFF;color:red;width:250;border-color:red;border-width:1px;border-style:solid;">

<span id="ManagerPersentDisplay" style="background-color:#0000FF;color:red;width:0;borderColor:#0000FF">

</span>

</span><span id="ManagerPersentDisplayText" style="font-size:10pt;color:blue;"></span><br>

<div id="ManagerProcessStatus" style="font-size:10pt;color:black;">

</div>

## OnOneFileUploadFinished

OnOneFileUploadFinished(AttachFile, IsAllUploaded);

在单个文件上传成功之后触发。

第一个参数AttachFile标识一个AttachFile类实例。请参考第7章有关该类的介绍。

第二个参数IsAllUploaded是布尔类型，说明该文件是否已上传成功。

示例代码：

<script language="JScript" for="Ntko.LargeFileUploader" event="OnOneFileUploadFinished(AttachFile,IsAllUploaded)">

document.all("PerFileUploadStatus").innerHTML += AttachFile.FileName +

":大小" + AttachFile.FileSize + "上传停止," + ((IsAllUploaded)?"全部完成.":"尚未完成.") +"<br>";

</script>

…

<span style="font-size:10pt;color:blue;">单个文件上传结束消息:</span> <br>

<div id="PerFileUploadStatus" style="font-size:10pt;color:black;">

</div>

## OnSaveToURLFinished

OnSaveToURLFinished (IsAllUploaded);

在保存到服务器结束之后触发。参数IsAllUploaded是布尔类型，说明是否所有文件已上传成功。

示例代码：

<script language="JScript" for="Ntko.LargeFileUploader" event="OnSaveToURLFinished(IsAllUploaded)">

if(IsAllUploaded)

{

document.all("SaveToURLFinishedStatus").innerHTML = "所有文件上传已成功完成";

}

else

{

document.all("SaveToURLFinishedStatus").innerHTML = "上传结束，不是所有文件上传都已成功";

}

</script>

…..

<span style="font-size:10pt;color:blue;">全部上传结束消息:</span> <br>

<div id="SaveToURLFinishedStatus" style="font-size:10pt;color:black;">

</div>

## AttachFile类介绍

### 说明

在介绍控件的主要方法之前，首先介绍AttachFile类。该类代表控件中的一个文件。可以通过控件的GetFile(index)方法获得一个AttachFile的类实例。或者，在某些事件中，控件会传入一个该类对象。

### AttachFile类属性

1、 FilePath

FilePath：只读。返回文件的全路径。

2、 FileName

FileName：只读。返回文件的名称部分。

3、 FileSize

FileSize：只读。返回文件的大小。

4、 MD5ID

MD5ID：只读，返回文件的MD5ID。该属性只有在文件上传之后才能获得。

# 常见问题（FAQs）

## 如果控件不能运行怎么办？

请检查Internet安全选项。应该设置为中或者以下。

## 如何在<object>的<param>部分设置属性？

在控件的<param>部分设置参数时要注意，在ActiveX控件中，-1表示TRUE，0表示FALSE，这和传统习惯不同。比如下面的设置将IsUseUTF8URL属性设置为TRUE：

<param name="IsUseUTF8URL" value="-1">

## 如何输入产品密钥？

在控件的<param>部分输入密钥。

对于单套授权版本，采用如下方式输入密钥：

<param name="ProductCaption" value="最终用户公司">

<param name="ProductKey" value="99999999999999999999999999999999">

单套授权版本的密钥是我们颁发的。

对开发商授权版本，采用如下方式输入密钥：

<param name="ProductCaption" value="自定义标题">

<param name="ProductKey" value="99999999999999999999999999999999">

<param name="MakerCaption" value="贵公司名称">

<param name="MakerKey" value="99999999999999999999999999999999">

其中，ProductCaption和ProductKey是用我们提供的工具产生；MakerCaption是固定的，MakerKey由我们颁发。

如果您还有其它的问题，请通过电子邮件saleinfo@ntko.com与我们联系

# 软航™， NTKO™商标说明

软航™及NTKO™均为重庆软航科技有限公司的商标，未经商标所有人许可，任何组织或者个人不得非法使用。

【完】