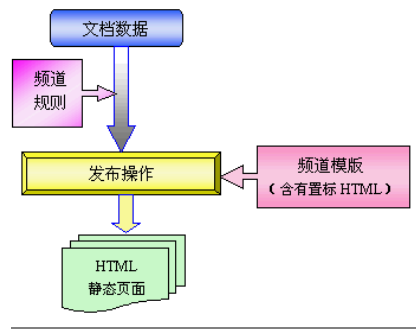


# TRS WCM 6.0 发布置标附录

## 页面发布工作原理

传统上，人们建立一个 Web 站点主要是靠手工维护 HTML 页面的方式进行的，这就要求维护人员不但要熟悉 HTML 技术，同时也需要人工维护大量的页面链接，这样不仅导致大量的死链接产生，同时也往往会严重影响网站内容的更新速度。

TRS WCM 的发布功能为解决上述问题，提供了一套强大的站点建立、管理工具，同时能够实现一定的权限控制和流程控制，保证系统的安全。在页面发布过程中，TRS WCM 的发布功能正是依靠程序自动结合 HTML 模板和数据库中的数据，生成最终的 HTML 页面的，同时可以自动生成页面的链接而不需要手工干预。其静态页面的工作原理见上图。



## HTML 和 TRS 置标

每个页面都必须遵循 Web 页面标准，才能被浏览器很好的解释和显示，TRS 置标本身也是符合 HTML 标准的附加置标方式，TRS 置标和 HTML、XML 标准兼容。含有 TRS 置标的 Web 页面，它可以被任意可视化的 Web 编辑器（如：DreamWeaver、FrontPage）或非可视化的文本编辑器编辑。在栏目定义时需要指定所使用的模板文件，分别控制概览页面和细览页面的表现。

## TRS置标说明

置标的主要作用是控制数据在模板中的表现形式，将数据与表现分开。主要思路是 TRS 置标处理数据的内容，存放的位置以及如何存放等，数据的表现形式则由标准的 html 语言描述，例如：

```
<FONT COLOR=RED>
```

```
<TRS_DOCUMENT FIELD="DOCTITLE">这里放置文档标题</TRS_DOCUMENT>
```

```
</FONT>
```

其中：第二行是 TRS 置标，其它部分是标准的 HTML 置标。TRS 置标表明要在该位置放置字段名为“标题”的字段内容。

根据系统的需要和使用方便，在 TRS WCM 系统的置标中主要采用了四种类型的置标，即：

- 概览置标

主要用于表示一个循环的属性，例如 TRS\_DOCUMENTS、TRS\_CHANNELS、TRS\_RELNEWS 等。

- TRS\_RECORD

只能在概览置标内部使用；一个概览置标内可以放置多个 TRS\_RECORD，用来控制文档不同显示风格。语法如下：

```
<TRS_DOCUMENTS [properties_list]>
  <TR>
    <TD><TRS_RECORD>(column_blocks)</TRS_RECORD></TD>
    <TD><TRS_RECORD>(column_blocks)</TRS_RECORD></TD>
  </TR>
  .....
</TRS_DOCUMENTS>
```

一般可以用来作为不同风格显示的控制标记，也可以实现特殊效果，比如实现一行多列的效果。

- 细览置标

主要用于引用某个具体的数据库字段，例如 TRS\_DOCUMENT、TRS\_CHANNEL 等。

- TRS\_TEMPLATE

用来实现模板的嵌套，嵌套模板支持多级。语法如下：

```
<TRS_TEMPLATE>(template_name)</TRS_TEMPLATE>
```

(template\_name)和置标之间不能有其他任何字符，否则会出现解析不出正确模板的问题。

TRS 置标统一以“<TRS\_”开头（置标名、属性名与大小写无关），<TRS\_XXX>与 </TRS\_XXX> 应成对出现，其中细览置标可以这样撰写 <TRS\_XXX FIELD="\*\*\*"/>。<TRS\_XXX>与</TRS\_XXX>之间的内容在生成页面时，将被数据库的输出内容所替代。每个置标都可以有多个置标属性，以实现不同的控制目的。XXX 表示当前所取的对象类型，比如文档置标就是 DOCUMENT；最后的字母为 S 表示是一个概览置标，获取的是一个循环。

其中 ID 属性表示当前置标所代表的对象从哪个地方获取；例如 TRS\_DOCUMENTS 中的 ID 表示的是从哪个栏目获取文档集合；FIELD 属性表示当前置标获取的是对象的那个属性，和数据库字段一一对应；例如<TRS\_DOCUMENT

`FIELD="DOCTITLE" />`表示获取的是当前文档的标题属性。

## 详细说明：

约定：

下面置标语法中所提及的`<column_block>`，是指需要配合细览置标使用。具体的表示方式，请参照各个细览置标说明。

属性的详细说明中的*斜体*表示自定义的项，**粗体**表示默认，普通表示可选项

## ■ TRS\_DOCUMENTS: 文档列表

### ■ 语法

```
<TRS_DOCUMENTS [ [ID="栏目名称" ] [CHILDINDEX="子栏目的序号" ]  
[CHANNELTYPE="栏目类型" ] [NUM="记录条数" ] [STARTPOS="开始位置" ]  
[PAGESIZE="分页参数" ] [WHERE="额外的检索条件" ] [ORDER="额外的  
排序条件" ] [AUTOMORE="是否自动显示更多内容" ] [BEGINMORE="放在更  
多内容前面的 HTML" ] [ENDMORE="放在更多内容后面的 HTML" ]  
[MORETEXT="更多内容的 HTML" ] [MOREEXTRA="A 元素上的额外 HTML" ]  
[MORETARGET="_blank" ] ]> <column_block> </TRS_DOCUMENTS>
```

### ■ 描述

显示文档列表，顺序与栏目“文档列表”页面相同。显示的内容由相应的细览置标`<TRS_DOCUMENT>`来定义。

### ■ 属性

属性	属性描述	类型	取值	说明
ID	表示当前的文档集合从那个栏目获取	字符串	<i>栏目名称</i>	当前站点存在的栏目名称
			PARENT	当前栏目的父栏目
			OWNER	当前栏目
CHILDINDEX	设置从指定栏目（ID）的哪个子栏目获取文档	数字	-1	表示为当前栏目
			$\geq 0$ 的整数	表示为第几个子栏目
CHANNELT	栏目类型	数字	0	普通栏目

<b>TYPE</b>			1	图片新闻，获取指定栏目的图片新闻的文档
			2	头条新闻，获取指定栏目的头条新闻的文档
<b>STARTPOS</b>	设置从哪个位置读取记录	数字	$\geq 0$ 的整数	默认为 0
<b>NUM</b>	设置显示的最大记录条数	数字	$> 0$ 的整数	默认为系统设定的最大数目
<b>PAGESIZE</b>	使用分页显示，每一页显示的记录数目	数字	$> 0$ 的整数	默认表示不进行分页显示
<b>WHERE</b>	额外的检索条件，对指定栏目的记录进行再检索	字符串	SQL 检索表达式	默认不检索
<b>ORDER</b>	额外的排序条件，对指定栏目的记录进行再排序	字符串	SQL 排序表达式	默认不排序
<b>AUTOMORE</b>	设置是否产生更多	字符串	true	如果存在更多，产生更多链接
			false	即使存在更多，不产生更多链接
<b>BEGINMORE</b>	更多链接的 HTML 开始部分 (A 元素之前)	字符串	HTML 代码	默认不产生
<b>ENDMORE</b>	更多链接的 HTML 结束部分 (A 元素之后)	字符串	HTML 代码	默认不产生
<b>MORETEXT</b>	设置更多链接的描述文字	字符串	Text	更多的描述文字，默认为：“更多内容...”
<b>MOREEXTRA</b>	链接元素上扩展的 HTML 内容	字符串	HTMLAttr	扩展的 HTML 内容
<b>MORETARGET</b>	设置更多打开后的窗口目标	字符串	Name	目标窗口的名称
			_blank	新开一个窗口
			_parent	在父窗口打开
			_self	在当前窗口打开

			_top	在顶级窗口打开
--	--	--	------	---------

### ■ 例子

[显示当前栏目前 10 条文档](#)

[显示当前栏目前 10 条文档，同时定义更多的方式](#)

[按照一页 20 条显示当前栏目前的前 500 篇文档](#)

[显示“国际新闻”栏目前 10 条文档](#)

[显示“新闻中心”第一个子栏目前的前 10 条文档](#)

[显示“新闻中心”头条新闻](#)

## ■ TRS\_RELNEWS: 相关新闻

### ■ 语法

```
<TRS_RELNEWS [[STARTPOS="开始位置"] [NUM="记录条数"]
[MODE="USERDEF|KEYWORDS|ALL"]][CHILDTYPE="栏目类型"]><column_block> </TRS_RELNEWS>
```

### ■ 描述

检索设置的关键词及手工设置的相关文档并显示相关的已发文档内容。利用 MODE 属性，可以获得三种不同的相关新闻集合，发布时，如果 MODE 没有设置，则系统默认取值为 USERDEF，表示只检索手工设置的相关文档并显示。只能应用到细览模板

### ■ 属性

属性	属性描述	类型	取值	说明
<b>NUM</b>	设置显示的最大记录条数	数字	≥0 的整数	默认为系统设定的最大数目
<b>STARTPOS</b>	设置从哪个位置读取记录	数字	≥0 的整数	默认为 0
<b>MODE</b>	相关新闻提取的方式	字符串	<b>USERDEF</b>	只提取用户手工设置的相关文档

			KEYWORDS	表示从站点所有文档中提取包含当前文档设置的关键词的所有文档
			ALL	表示提取 USERDEF 和 KEYWORDS 两种方式可以获得的所有相关新闻，并且剔除编号相同的文档

#### ■ 例子

[人工相关，必须手动指定](#)

[自动相关，按照关键词自动进行检索](#)

[获取所有相关文档](#)

## ■ TRS\_CHANNELS: 栏目列表

#### ■ 语法

```
<TRS_CHANNELS [ID="父栏目名"] [CHILDINDEX="子栏目序号"]
[STARTPOS="起始位置"] [NUM="数目"] [CHILDTYPE="所取子栏目的类型"]>
<column_block> </TRS_CHANNELS>
```

#### ■ 描述

显示指定父栏目的所有子栏目链接。需要系统能够自动产生一个栏目列表的时候，使用此项。目的在于使系统自动产生一个带有链接的栏目列表，并放置在置标所在的页面中。

#### ■ 属性

属性	属性描述	类型	取值	说明
ID	表示从那个栏目获取栏目集合	字符串	栏目名称	当前站点存在的栏目名称
			PARENT	当前栏目的父栏目
			SITE	当前站点

			OWNER	当前栏目
CHILDINDEX	设置从指定栏目 (ID) 的哪个子栏目获取栏目集合	数字	-1	表示为当前栏目 (ID)
			≥0 的整数	表示为第几个子栏目
STARTPOS	设置从哪个位置读取子栏目	数字	≥0 的整数	默认为 0
NUM	设置显示的最大子栏目数	数字	≥0 的整数	默认为系统设定的最大数目
CHILDTYPE	所取子栏目的类型	数字	-1、0、1、2	-1 为取所有类型的子栏目, 0 为普通栏目, 1 为图片新闻, 2 为头条新闻,

#### ■ 例子

[显示当前栏目的子栏目列表 \(从第 2 个子栏目开始\)](#)

## ■ TRS\_CHANNEL: 获取指定栏目信息

#### ■ 语法

```
<TRS_CHANNEL [ [ID="父栏目名" ] [CHILDINDEX="子栏目序号" ]
FIELD=" 字段名 " [NUM= " 最大长度 " ] [DATEFORMAT=" 日期格式" ]
[AUTOFORMAT=" 自动格式化" ] [AUTOFORMATTYPE=" 格式化方式" ]
[AUTOLINK=" 是否自动产生链接" ] [EXTRA="A 元素上扩展的 HTML 内容" ]
[TARGET=" 窗口目标 " ] [LINKALT=" 链接 是否显示提示 " ]
[LINKALTTEXT=" 链接提示内容" ] [ TRUNCATEDFLAG=" 栏目名称被截断后
显示内容"> <column_block> </TRS_CHANNEL>
```

#### ■ 描述

显示指定栏目 (*ID*, *ChildIndex*) 指定字段 (*Field*) 的信息。

#### ■ 属性

属性	属性描述	类型	取值	说明
----	------	----	----	----

<b>ID</b>	指定的栏目	字符串	栏目名称	当前站点存在的栏目名称
			PARENT	当前栏目的父栏目
			OWNER	当前栏目
<b>CHILDINDEX</b>	设置指定栏目的子栏目	数字	-1	表示为当前栏目
			≥0 的整数	表示为第几个子栏目
<b>FIELD</b>	栏目属性名称	字符串	栏目字段名称	在数据字典中定义，默认为 <b>CHNLDESC</b> (栏目显示名称)
			_RECURL	当前栏目路径
			UrlIsAbs	当前栏目全路径
<b>NUM</b>	设置显示的最大长度	数字	>0 的整数	默认 <b>不限制</b> ，必须大于“TRUNCATEDFLAG”字符串的长度，否则不会截断。
<b>DATEFORMAT</b>	设置日期的显示格式	字符串	日期格式字符串	参照 <a href="#">时间格式</a> ，默认为 <b>yyyy-MM-dd HH:mm:ss</b> ；仅仅对于日期属性有效
<b>AUTOFORMAT</b>	是否按照指定方式格式化当前数据	字符串	true	需要格式化
			false	不需要格式化
<b>AUTOFORMATTYPE</b>	格式化类型	字符串	HTML	按照 HTML 方式格式化代码 (回车→ ，空格→&nbsp;等等)
			HTMLValue	按照 HTML 元素属性的方式转义数据 (&→&amp; "→&quot; 等等)
			JAVASCRIPT	按照 JavaScript 数据规则转义数据 ("→\" \\→\\ 等等)
			XML	按照 XML 元素属性的方式转义数据 (&→&amp; "→&quot; 等等)
<b>AUTOLINK</b>	是否产生链接，应用于 CHNLNAME	字符串	true	FIELD 为 <b>CHNLNAME</b> 或者 <b>CHNLDESC</b> 时，缺省值是 <b>true</b>



	CHNLDESC		<b>false</b>	FIELD为其他内容时, 缺省值是 <b>false</b> 。
<b>TARGET</b>	链接打开后的窗口目标	字符串	<i>Name</i>	目标窗口的名称
			<i>_blank</i>	新开一个窗口
			<i>_parent</i>	在父窗口打开
			<i>_self</i>	在当前窗口打开
			<i>_top</i>	在顶级窗口打开
<b>EXTRA</b>	链接元素上扩展的HTML内容	字符串	<i>HTMLAttr</i>	扩展的HTML内容
<b>LINKALT</b>	是否产生链接的内容提示	字符串	<b>true</b>	产生链接内容提示
			<b>false</b>	不产生
<b>LINKALTTEXT</b>	设置链接的内容提示文字	字符串	<i>Text</i>	链接的内容提示文字, 默认为 <b>当前属性值</b>
<b>TRUNCATEDFLAG</b>	栏目名称被截断后, 附加在后面的字符串	字符串	<i>Text</i>	默认" <b>...</b> ", 即三个半角省略号

#### ■ 例子

[显示科教新闻栏目的名称, 并且产生链接](#)

[显示新闻中心第一个子栏目的信息, 并且产生链接](#)

### ■ TRS\_DOCUMENT: 文档详细信息

#### ■ 语法

```
<TRS_DOCUMENT FIELD="字段名" [NUM="最大长度"] [DATEFORMAT="日期格式"] [AUTOFORMAT="自动格式化"] [AUTOFORMATTYPE="格式化方式"] [AUTOLINK="是否自动产生链接"] [EXTRA="A 元素上扩展的 HTML 内容"] [TARGET="窗口目标"] [LINKALT="链接是否显示提示"] [LINKALTTEXT="链接提示内容"] [AUTOCOLOR="文档标题颜色"] >通用细
```

览</TRS\_DOCUMENT>

### ■ 描述

从文档提取指定属性[FIELD]，按照指定的方式生成内容替换当前的置标内容。

### ■ 属性

属性	属性描述	类型	取值	说明
<b>FIELD</b>	文档属性名称	字符串	文档字段名称	在数据字典中定义，默认为 <b>DOCTITLE</b> (文档标题)
			DOCSOURCE	文档来源 (5.2 置标)
			DOCFROM	文档来源 (5.1 置标，现兼容)
			DOCSECURITY	文档安全级别
			_RECURL	当前文档路径
			UrlIsAbs	当前文档的全路径
<b>NUM</b>	设置显示的最大长度	数字	>0 的整数	默认不限制
<b>DATEFORMAT</b>	设置日期的显示格式	字符串	日期格式字符串	参照 <a href="#">日期格式</a> ，默认为 <b>yyyy-MM-dd</b> <b>HH:mm:ss</b> ；仅仅对于日期属性有效
<b>AUTOFORMAT</b>	是否按照指定方式格式化当前数据	字符串	<b>true</b>	需要格式化
			<b>false</b>	不需要格式化
<b>AUTOFORMATTYPE</b>	格式化类型	字符串	<b>HTML</b>	按照 HTML 方式格式化代码 (回车 →  空格 →&nbsp; 等等)
			HTMLValue	按照 HTML 元素属性的方式转义数据 (&→&amp; "→&quot; 等等)

			JAVASCRIPT	按照 JavaScript 数据规则转义数据（"→\" \\→\\等等）
			XML	按照 XML 元素属性的方式转义数据（&→&amp; "→&quot;等等）
<b>AUTOLINK</b>	是否产生链接, 不允许 AUTOLINK 的字段有: DOCHTMLCON, DOCHTMLCONTENT, DOCPUBHTMLCON, DOCPUBTIME, DOCSECURITY	字符串	true	产生当前文档的链接, 对于来源为来源链接。对于 DocTitle, 默认为 <b>true</b> , 但在细览页面 AUTOLINK 无效。
			false	不产生链接; 对于其它属性默认为 <b>false</b> 。
<b>TARGET</b>	链接打开后的窗口目标	字符串	Name	目标窗口的名称
			_blank	新开一个窗口
			_parent	在父窗口打开
			_self	在当前窗口打开
			_top	在顶级窗口打开
<b>EXTRA</b>	链接元素上扩展的 HTML 内容	字符串	HTMLAttr	扩展的 HTML 内容
<b>LINKALT</b>	是否产生链接的内容提示	字符串	true	产生链接内容提示
			false	不产生
<b>LINKALTT EXT</b>	设置链接的内容提示文字, 不支持文档来源	字符串	Text	链接的内容提示文字, 默认为 <b>当前属性值</b>
<b>AUTOCOLOUR</b>	控制是否自动添加文档标题的颜色	字符串	True	只有当FIELD属性是"DOCTITLE", 才有效。文档标题的颜色是指在文档编辑器中指定的颜色。
			false	

#### 例子

[显示标题字段的内容, 不设定长度](#)

[显示标题字段的内容, 并设定长度为 50](#)

[显示日期字段的内容，并且格式化日期](#)

[显示摘要，并且格式化](#)

[显示文档来源](#)

## ■ TRS\_APPENDIX: 文档附件

### ■ 语法

```
<TRS_APPENDIX MODE=" 附件的类型 " [FIELD=" 附件属性名称 " ]  
[INDEX="序号"] [BEGINHTML="附件前面额外 HTML 属性"] [ENDHTML=" 附件后面额外 HTML 属性" ] [AUTOLINK="产生链接" ] [EXTRA="链接上额  
外的 HTML 属性" ] [MEMO="是否产生图片说明" ] [BEGINMEMO="图片说明前  
HTML 代码" ] [ENMEMO="图片说明后 HTML 代码" ] [SEPERATOR="多个附  
件分隔符" ] [UPLOAD="是否强制分发附件" ] [TARGET="窗口目标" ]>附件  
</TRS_APPENDIX>
```

### ■ 描述

在细览中显示当前文档的附件内容。

### ■ 属性

属性	属性描述	类型	取值	说明
MODE	附件类型	字符串	PIC	图片
			FILE	文件
			LINK	链接
			ALL	所有附件
FIELD	附件属性名称	字符串	<i>字段名称</i>	WCMAppendix 的字段名称
			_RECURL	当前文档附件路径
			UrlIsAbs	当前文档附件的全路径
INDEX	获取指定序号的附件	数字	-1	表示获取所有的附件
			$\geq 0$ 的整数	指定的序号
AUTOLINK	是否产生链接	字符串	true	产生当前附件的链接
			false	不产生链接
EXTRA	链接元素上扩展的 HTML 内容	字符串	HTMLAttr	扩展的 HTML 内容
BEGINHTML	附件前面额外 HTML 代码	字符串	HTMLAttr	附件前面额外 HTML 代码

<b>ENDHTML</b>	附件后面额外 HTML 代码	字符串	<i>HTMLAttr</i>	附件后面额外 HTML 代码
<b>MEMO</b>	是否产生图片说明。此属性只对图片附件有效	字符串	<b>true</b>	产生图片说明
			<b>false</b>	不产生图片说明
<b>BEGINMEMO</b>	图片说明前面额外 HTML 代码。此属性只对图片附件且 MEMO 为 true 时有效	字符串	<i>HTMLAttr</i>	图片说明前面额外 HTML 代码
<b>ENDMEMO</b>	图片说明后面额外 HTML 代码。此属性只对图片附件且 MEMO 为 true 时有效	字符串	<i>HTMLAttr</i>	图片说明后面额外 HTML 代码
<b>SEPERATOR</b>	多个附件之间的分隔符	字符串	 	如果没有设置 FIELD, 默认为 
			\n	如果设置了 Field, 默认\n
			<i>字符串</i>	用户自定义的分隔符, 支持空字符串或长度为 0 的字符串。
<b>UPLOAD</b>	是否强制分发附件。如果没有设置有效的 FIELD 属性, 一定会分发附件, 此属性无效	字符串	<b>true</b>	强制分发附件
			<b>false</b>	不分发附件。
<b>TARGET</b>	链接打开后的窗口目标	字符串	<i>Name</i>	目标窗口的名称
			_blank	新开一个窗口
			_parent	在父窗口打开
			_self	在当前窗口打开
			_top	在顶级窗口打开

## 例子

[显示文档的第一个图片附件](#)

[显示文档的第一个文件附件](#)

[显示文档的第一个链接附件](#)

## ■ TRS\_CURPAGE: 当前位置

### 语法

```
<TRS_CURPAGE [[VALUE=" 链接符 "] [ONLY=" 是否显示自己 "]  
[AUTOLINK=" 自动链接 "] [TARGET=" 窗口目标 "][EXTRA=" 额外的 HTML  
内容 "]]>当前页面</TRS_CURPAGE>
```

### 描述

显示该页面在当前站点中的地图，可以显示自己或包含自己和所有父节点的层次关系，可以应用到概览模板和细览模板中。

### 属性

属性	属性描述	类型	取值	说明
<b>VALUE</b>	路径中栏目名称之间的分隔符	字符串	<i>Separator</i>	栏目名称之间的分隔符 默认为 ">"
<b>ONLY</b>	仅仅显示当前栏目	字符串	<b>true</b>	仅仅显示自己
			<b>false</b>	显示完整的路径
<b>AUTOLINK</b>	是否产生链接	字符串	<b>true</b>	路径中每一个节点产生链接
			<b>false</b>	不产生链接
<b>TARGET</b>	链接打开后的窗口目标	字符串	<i>Name</i>	目标窗口的名称
			<b>_blank</b>	新开一个窗口
			<b>_parent</b>	在父窗口打开
			<b>_self</b>	在当前窗口打开
<b>_top</b>	在顶级窗口打开			
<b>EXTRA</b>	链接元素上	字符串	<i>HTMLAttr</i>	扩展的 HTML 内容

	扩展的 HTML 内容			
--	----------------	--	--	--

#### ■ 例子

[显示当前所在的栏目路径](#)

### ■ TRS\_DATETIME:发布日期

#### ■ 语法

```
<TRS_DATETIME [[DATEFORMAT="日期格式"]]> 发布日期  
</TRS_DATETIME>
```

#### ■ 描述

显示发布该页面的发布日期

细览置标；独立使用；可以应用到概览模板和细览模板

#### ■ 属性

DATEFORMAT

#### ■ 说明

- **DATEFORMAT** 字符串，设置日期的显示格式。  
参照 [日期格式](#)，默认为yyyy-MM-dd HH:mm:ss；  
仅仅对于日期属性有效

#### ■ 例子

[显示当前的发布时间](#)

## ■ TRS\_REPLACE: 替换内容

### ■ 语法

```
<TRS_REPLACE VALUE="替换名称">替换内容</TRS_REPLACE>
```

### ■ 描述

根据名称显示在栏目中定义替换内容的内容。独立使用，可以应用到概览模板和细览模板。

### ■ 属性

VALUE

### ■ 说明

VALUE="替换名称"属性是必须要设置的。

### ■ 例子

如果在栏目A和B中都定义了替换内容DEMO,并分别给他们赋值[www.TRS.com.cn](http://www.trs.com.cn)和[wcm.trs.com.cn](http://wcm.trs.com.cn)。同时在他们使用的共同模板C中使用<TRS\_REPLACE VALUE="DEMO">替换内容</TRS\_REPLACE>。发布后,在栏目A页面的TRS\_REPLACE置标放置的地方用[www.TRS.com.cn](http://www.trs.com.cn)替换,在栏目B页面的TRS\_REPLACE置标放置的地方用[wcm.trs.com.cn](http://wcm.trs.com.cn)替换。

[在模板中替换已定义内容](#)

## ■ TRS\_RECORD: 分组显示置标

### ■ 语法

```
<TRS_RECORD [NUM="记录数"]><column_block></TRS_RECORD>
```

### ■ 描述

获取指定记录数的数据并结合column\_block中定义的内容显示



不能独立使用；必须结合概览置标才能使用

#### ■ 属性

NUM

#### ■ 说明

- **NUM**     **数字**     设置显示的最大记录条数  
   >0 的整数，默认为 **1**

#### ■ 例子

[多行显示文档，一行显示三篇文档](#)

## ■ TRS\_TEMPLATE: 嵌套模板

#### ■ 语法

```
<TRS_TEMPLATE TEMPNAME="嵌套模板名称" [ IDENTITY="嵌套模板标识"]>注释信息</TRS_TEMPLATE>
```

#### ■ 描述

指定当前的位置使用嵌套模板的内容

#### ■ 属性

TEMPNAME, IDENTITY

#### ■ 说明

- **TEMPNAME**   **字符串**   **嵌套模板的名称**
- **IDENTITY**   **字符串**   **嵌套模板的标识**    **可选**

需要注意的是，如果模板多次嵌套实用了同一个模板，虽然置标解析机制允许平行地嵌套多个完全一样的模板，但是当需要区别出这些“TRS\_TEMPLATE”置标时，那么在这个模板中需要给予它们不同的 IDENTITY 值。例如下面的写法：

```
...  
<TRS_TEMPLATE tempname="测试嵌套模板变量" identity="测试嵌套模板变量 1"></TRS_TEMPLATE>  
<TRS_TEMPLATE tempname="测试嵌套模板变量" identity="测试嵌套模板变量 2"></TRS_TEMPLATE>
```

```
<TRS_TEMPLATE tempname="测试嵌套模板变量" identity="测试嵌套模板变量 3"></TRS_TEMPLATE>
```

...

如果模板“测试嵌套模板变量”中定义了一些模板变量（参见本手册中关于变量声明的描述），那么这样的写法，让我们可以为每个“TRS\_TEMPLATE”置标分别赋予不同的变量值。

#### 例子

[指定当前位置使用\[页面头部嵌套模板\]的内容](#)

## ■ TRS\_DATETIME: 发布时间

#### 语法

```
<TRS_DATETIME [DATEFORMAT="时间格式"]>文档发布时间</TRS_DATETIME >
```

#### 描述

显示生成文档的时间，或者说发布时间

#### 属性

DATEFORMAT

#### 说明

- **DATEFORMAT** 日期格式字符串 设置日期的显示格式  
参照 [日期格式](#)，默认为**yyyy-MM-dd**

同时对以下格式字符进行自动转换：

- \* [1] %Y --> yyyy
- \* [2] %MON --> MM
- \* [3] %m --> MM
- \* [4] %DAY --> dd
- \* [5] %d --> dd
- \* [6] %H --> HH
- \* [7] %M --> mm
- \* [8] %S --> ss

## ■ TRS\_CONDITION: 条件置标

### ■ 语法

<TRS\_CONDITION CONDITION="条件变量声明|@属性名称" [REFERENCE="参考常量|参考变量声明|@属性名称"] [OPERATOR="条件值和参考值的比较方式"] [REVERSE="是否翻转条件值和参考值"] [TYPE="条件值和参考值的数据类型"] [NOT="是否求反"]>条件执行的内容</TRS\_CONDITION>

### ■ 描述

条件执行置标。如果指定的条件为真，则继续执行解析该置标下的模板，如果指定的条件为假，则忽略其下的模板。

其中指定的条件是指一个设定的条件变量值和一个参考值是否相等，或者是指某个发布对象的某项属性值和参考值相比较是否符合特定的条件，例如属性值大于参考值。

### ■ 属性

属性	属性描述	类型	取值	说明
CONDITION	条件变量声明 @属性名 (必须)	字符串		条件值声明。可以是使用\${变量名}形式的变量，也可以是以“@”符号开始的字符串，以指定取发布对象的特定属性值作为条件值。详细语法见 <b>对象的属性名称</b> 。
REFERENCE	参考常量定义 参考变量声明 @属性名 (可选)	字符串		参考值声明，可以是常量定义，也可以是使用\${变量名}形式的变量，也可以是以“@”符号开始的字符串，以指定取发布对象的特定属性值作为参考值。详细语法见 <b>对象的属性名称</b> 。

OPERATOR	条件值和参考值比较方式 (可选)	字符串		条件值和参考值比较方式。或者叫做比较运算符,指等于还是大于等等。缺省方法是“==”,即忽略大小写的情况下比较两个字符串是否相等。其他比较方式见 <b>比较方式列表</b> 。			
REVERSE	是否翻转条件值和参考值 (可选)	字符串	<table border="1"> <tr> <td>true</td> <td rowspan="2">是否翻转条件值和参考值。即把参考值作为条件值,把条件值作为参考值,然后再进行比较。这在普通的是否相等的比较中不起作用,但是在判断一个字符串是否以另一个字符串开始或者结束时起作用。其缺省值为“false”,即不翻转。</td> </tr> <tr> <td>false</td> </tr> </table>	true	是否翻转条件值和参考值。即把参考值作为条件值,把条件值作为参考值,然后再进行比较。这在普通的是否相等的比较中不起作用,但是在判断一个字符串是否以另一个字符串开始或者结束时起作用。其缺省值为“false”,即不翻转。	false	
true	是否翻转条件值和参考值。即把参考值作为条件值,把条件值作为参考值,然后再进行比较。这在普通的是否相等的比较中不起作用,但是在判断一个字符串是否以另一个字符串开始或者结束时起作用。其缺省值为“false”,即不翻转。						
false							
TYPE	条件值和参考值的数据类型 (可选)	字符串	<table border="1"> <tr> <td>INT</td> <td rowspan="3">条件值和参考值的数据类型。因为对于整数和小数,如果也当作字符串来进行大小的比较,有可能出现错误的结果,所以需要声明进行比较的数据的数据类型,缺省数据类型是<b>STING</b>,即字符串。</td> </tr> <tr> <td>FLOAT</td> </tr> <tr> <td>STING</td> </tr> </table>	INT	条件值和参考值的数据类型。因为对于整数和小数,如果也当作字符串来进行大小的比较,有可能出现错误的结果,所以需要声明进行比较的数据的数据类型,缺省数据类型是 <b>STING</b> ,即字符串。	FLOAT	STING
INT	条件值和参考值的数据类型。因为对于整数和小数,如果也当作字符串来进行大小的比较,有可能出现错误的结果,所以需要声明进行比较的数据的数据类型,缺省数据类型是 <b>STING</b> ,即字符串。						
FLOAT							
STING							
NOT	对比较结果是否求反 (可选)	字符串	<table border="1"> <tr> <td>true</td> <td rowspan="2">对比较结果是否求反。如果设置为“true”,则在比较结果为真的情况下,整个条件置标的结果为假。缺省为“false”,即不求反。</td> </tr> <tr> <td>false</td> </tr> </table>	true	对比较结果是否求反。如果设置为“true”,则在比较结果为真的情况下,整个条件置标的结果为假。缺省为“false”,即不求反。	false	
true	对比较结果是否求反。如果设置为“true”,则在比较结果为真的情况下,整个条件置标的结果为假。缺省为“false”,即不求反。						
false							

■ 比较方式

符号	名称	大小写敏感	支持数据类型	说明
==	等于	否	STRING	忽略大小写比较字符串
=	等于	是	STRING	大小写敏感

				比较字符串
<	小于	是	INT, FLOAT, STRING	条件值小于参考值
<=	小于等于	是	INT, FLOAT, STRING	条件值小于等于参考值
>	大于	是	INT, FLOAT, STRING	条件值大于参考值
>=	大于等于	是	INT, FLOAT, STRING	条件值大于等于参考值
<b>present</b>	存在	是	STRING	不 等 于 “null”
<b>blank</b>	空值	是	STRING	为 “null” 或者零长度
<b>contain</b>	包含	是	STRING	条件值包含 参考值
<b>startsWith</b>	以...开始	是	STRING	条件值以参 考值开始
<b>endsWith</b>	以...结束	是	STRING	条件值以参 考值结束
<b>match</b>	符合模式	是	STRING	条件值符合 参考值模式
<b>L</b>	长于	否	STRING	<sup>注1</sup> 字符串长度 大于
<b>L=</b>	长于等于	否	STRING	<sup>注1</sup> 字符串长度 大于等于
<b>S</b>	短于	否	STRING	<sup>注1</sup> 字符串长度 小于
<b>S=</b>	短于等于	否	STRING	<sup>注1</sup> 字符串长度 大小等于

**注 1:** 进行字符串的长度比较时，极端情况下行为的定义。

1. 如果置标属性 **CONDITION** 指定的值为 null，则认为其长度为零。
2. 如果置标属性 **REFERENCE** 指定的值为 null，则认为参考长度为零。
3. 如果置标属性 **REFERENCE** 指定的值是非负值整数（字符串完全由阿拉伯数字构成），则使用该整数值与条件值字符串的长度相比较；如果置标属性 **REFERENCE** 指定的值不是一个非负值整数（含有阿拉伯数字以外的字符），则使用该参考值字符串的长度与条件值字符串的长度进行比较。

4. 如果置标属性 **REFERENCE** 指定的值是一个非负值整数，但是该整数已经超出了合理的范围，则仍然使用该参考值字符串的长度与条件值字符串的长度进行比较。目前合理范围的定义是小于等于 **2147483647**。
5. 如果没有显示的设置置标属性 **REFERENCE**，其缺省值为“true”，即取“true”字符串的长度 4 进行比较运算。

#### ■ 对象的属性名称

置标属性 **CONDITION** 和 **REFERENCE** 的值。除了使用变量外，新增了使用发布对象的属性值作为条件值和参考值的功能，具体的语法是：

@[[DomainName.]DomainName.]PropertyName

其中符号“@”是必须的，用以标识该置标属性是获取某个发布对象的属性值。

PropertyName 也是必须的，用以标识取发布对象的哪一个属性，其值可以是指定发布对象上任意有效的属性名称。

DomainName 是可选的，用以标识是取哪一个发布对象的属性。如果没有提供，则默认取当前发布对象的属性。其他有效的 DomainName 如下：

DomainName	说明
PAGE	取当前发布页面的相关对象的属性值。如果使用，则必须是第一个 DomainName，但是在其后可以附加其他的 DomainName。
SITE	如果跟在 PAGE 后面，则是取当前发布页面所属站点的属性，如果 PAGE 没有出现，则是当前发布对象所属站点的属性。如果使用，必须是最后一个 DomainName。
PARENT	取父对象的属性值。可以叠代，即 PARENT.PARENT 的形式，以获取父对象的父对象的属性是支持的。同样受 PAGE 指示是当前发布对象还是当前发布页面。

#### ■ 注意事项

关于日期类型的比较。目前还不支持把日期类型作为一种单独数据类型进行比较，只能是作为字符串进行比较，即要求两个日期类型的字符串必须具有的相同的格式。

#### ■ 例子

[是否显示嵌套的站点导航条](#)  
[根据属性值决定是否自动产生文档连接](#)

## ■ TRS\_ECHO: 在当前置标位置输出指定的内容

### ■ 语法

```
<TRS_ECHO VALUE="待输出的内容或者变量声明">  
    <!-- 可选的描述信息，会被置标忽略-->  
</TRS_ECHO>
```

### ■ 描述

在当前置标位置输出指定的内容，虽然也可以直接设定要输出的内容，但是只有将要输出的内容声明为变量时才有意义。

### ■ 属性

属性	属性描述	类型	取值	说明
<b>VALUE</b>	待输出的内容	字符串		直接设定的待输出内容或者变量声明

### ■ 例子

[概览页面的文档列表中每一篇文章前的前缀](#)

## ■ TRS\_NEWICON: 显示最新新闻标志

### ■ 语法

```
<TRS_NEWICON [FIELD="时间字段"] [TIME="时间差值"]>  
    <!-- HTML 代码，最新新闻标志 -->  
</TRS_NEWICON>
```

### ■ 描述

取当前对象指定的字段的时间值和发布时的时间相比较，如果其差值在指定的时间差值之内，则输出其下的最新新闻标志，否则不输出任何东西。

一些特殊的情况下。

1. 如果指定的时间字段不存在，则不输出任何内容。
2. 如果当前对象的指定属性时间发生在当前时间之后，则认为系统错误，

也不输出任何内容。

#### ■ 属性

属性	属性描述	类型	取值	说明
<b>FIELD</b>	时间字段(可选)	字符串	时间类型字段的名称	时间类型字段的名称, 缺省值: <b>CRTIME</b>
<b>TIME</b>	时间差值(可选)	字符串	数字[单位]	直接精确的设定一个时间差, 其中时间单位可以是(mhdw), 分别代表分钟、小时、天、周, 缺省的时间单位是分钟。
			day	当日新闻显示为最新
			week	当周新闻显示为最新
			month	当月新闻显示为最新

#### ■ 例子

[为当天的新闻添加最新新闻标志](#)

## ■ TRS\_VARIABLE: 变量声明

#### ■ 语法

```
<TRS_VARIABLE NAME="变量名称" [TYPE="变量类型"] [SPLIT="枚举值和枚举显示值的分割字符"] [DEFAULT="变量缺省值"] [CANCUSTOM="该变量是否允许用户直接输入值"] [ENUMERATE="变量的枚举值 1"] [ENUMERATE="变量的枚举值 2"] ...>
  <!-- 可能的嵌入式枚举值声明 -->
</TRS_VARIABLE>
```

#### ■ 描述

在使用模板变量之前, 声明变量, 如果需要同时声明变量的枚举值, 在后面的模板中使用变量时, 如果同名变量已经预先声明, 则可以简化解析工作, 直接使用已有的解析结果。

关于置标属性 **CANCUSTOM**, 如果没有声明变量枚举值, 则 **CANCUSTOM** 的



缺省值是“true”，即用户可以直接输入；如果声明了有效的变量枚举值，则其缺省值为“false”，即用户不能直接输入，只能从已经声明的枚举值中选择。

#### ■ 属性

属性	属性描述	类型	取值	说明
<b>NAME</b>	变量名称（必须）	字符串	任意名称	任意的变量名称
<b>TYPE</b>	变量类型（可选）	字符串	<b>BOOLEAN</b>	逻辑类型变量，变量的值可以是 true 或者 false，显示名称仅为“是”和“否”
			<b>INTEGER</b>	整数类型变量，变量的值只可以是整数
			<b>NUMBER</b>	数值类型变量，变量的值只可以是整数或者小数
			<b>STRING</b>	字符串类型变量，变量的值可以是字符串
<b>SPLIT</b>	分隔字符，可以多次设置（可选）	字符串	长度为 1 的字符串	用于分割枚举值和枚举显示值
<b>DEFAULT</b>	变量缺省值（可选）	字符串	任意字符串	变量缺省值和显示值
<b>ENUMERATE</b>	枚举值和枚举显示值，可以多个（可选）	字符串		变量枚举值和显示值
<b>CANCUSTOM</b>	该变量是否允许用户直接输入值（可选）	字符串	true	用户可以直接输入
			false	用户不能输入，只能从已经声明的枚举值中选择。

#### ■ 例子

[声明变量的示例 1](#)

## ■ TRS\_ENUM: 声明变量枚举值

### ■ 语法

```
<TRS_ENUM VALUE="变量枚举值" [DISPLAY="枚举显示值"]>  
  <!-- 可选的描述信息，会被置标忽略-->  
</TRS_ENUM>
```

### ■ 描述

用以简化 **TRS\_VARIABLE** 置标的变量枚举值声明。

### ■ 属性

属性	属性描述	类型	取值	说明
<b>VALUE</b>	变量枚举值 (可选)	字符串	任意字符串	变量枚举值，在应用模板发布页面时使用
<b>DISPLAY</b>	枚举显示值 (可选)	字符串	任意字符串	枚举显示值，在用户修改设置变量值时使用

### ■ 例子

[声明变量的示例 2](#)

## ■ TRS\_ROLLEDDOCUMENTS: 滚动新闻列表

### ■ 语法

```
<TRS_ROLLEDDOCUMENTS FROMCHANNELIDS=" 栏目 ID 序列 "  
  [[TIMERANGE="滚动几天前的新闻" | QUERYDAY="滚动指定天的新闻"]  
  AUTOTRIGGER="指定栏目数据更新后，当前模板所关联栏目是否自动触发发布" ] ]  
  [[NUM="记录条数"] [STARTPOS="开始位置"] [PAGE_SIZE="分页参数"]  
  [WHERE="额外的检索条件"] [ORDER="额外的排序条件"] ] >  
<column_block> </TRS_ROLLEDDOCUMENTS>
```

### ■ 描述

按照指定顺序显示指定栏目 (**FromChannelIds**) 的已发文档，顺序可以在置标中指定。显示的内容由相应的细览置标 **<TRS\_DOCUMENT>** 来定义。

■ 属性

属性	属性描述	类型	取值	说明
<b>FROMCHANNELIDS</b>	栏目 ID 序列	字符串		以逗号分隔的 ID 字符串
<b>TIMERANGE</b>	滚动几天前的新闻, 这个置标属性取的文档必须要是已发状态的	数字	$\geq 1$ 的整数	默认为 1 (以系统当前时间为基准, 每 24 小时为一天, 不是自然天数, 如果指定了有效的 <b>QUERYDAY</b> 属性则忽略该参数)
<b>QUERYDAY</b>	滚动指定天的新闻	数字	$\geq 0$ 的整数	默认为 -1 (0 为当天)
<b>AUTOTRIGGER</b>	指定栏目数据更新后, 当前模板所关联栏目是否自动触发概览页面的更新发布	字符串	true	如果指定栏目发布后, 模板关联栏目同步触发概览页面的发布
			false	默认值; 如果指定栏目发布后, 模板关联栏目不同步触发概览页面发布
<b>STARTPOS</b>	设置从哪个位置读取记录	数字	$\geq 0$ 的整数	默认为 0
<b>NUM</b>	设置显示的最大记录条数	数字	$> 0$ 的整数	默认为 20 条
<b>PAGESIZE</b>	使用分页显示, 每一页显示的记录数目	数字	$> 0$ 的整数	默认表示不进行分页显示
<b>WHERE</b>	额外的检索条件, 对指定栏目的记录进行再检索	字符串	SQL 检索表达式	默认不检索 (保留, 尚未实现)

<b>ORDER</b>	额外的排序条件,对指定栏目的记录进行再排序	字符串	SQL 排序表达式	默认不排序 (保留,尚未实现)
--------------	-----------------------	-----	-----------	--------------------

#### ■ 例子

[将ID为 84,85,104 的栏目一天前的新闻作为滚动新闻显示在页面中](#)

## ■ TRS\_CHANNELLOGO: 获取指定频道图片LOGO地址

#### ■ 语法

```
<TRS_CHANNELLOGO [[ID="频道名/父频道名"] [CHILDINDEX="子频道序号"]] [INDEX="获取第几个 LOGO"] [NAMEONLY="仅仅生成文件名"]/>
```

#### ■ 描述

获取指定频道(*ID,ChildIndex*)指定序号(*index*)的图片 LOGO 地址。

#### ■ 属性

属性	属性描述	类型	取值	说明
<b>ID</b>	指定的频道	字符串	<i>频道名称</i>	当前站点存在的频道名称
			PARENT	当前频道的父频道
			SITE	当前站点
			OWNER	当前频道
<b>CHILDINDEX</b>	设置指定频道的子频道	数字	-1	表示为当前频道
			$\geq 0$ 的整数	表示为第几个子频道
			_RECURL	当前频道路径
<b>INDEX</b>	设置获取第几个 LOGO	数字	$> 0$ 的整数	默认为 1
<b>NAMEONLY</b>	仅仅显示	字符串		默认为 false

	文件名			
--	-----	--	--	--

#### ■ 例子

获取当前频道的第一个 Logo 的地址

```
<TRS_CHANNELLOGO INDEX=1 />
```

获取当前频道的第一个 Logo 的文件名

```
<TRS_CHANNELLOGO INDEX=1 NAMEONLY="true" />
```

### ■ TRS\_DOCUMENTLOGO: 获取指定文档LOGO信息

#### ■ 语法

```
<TRS_DOCUMENTLOGO [ ID="文档 ID" ] [ INDEX="获取第几个 LOGO" ]  
[ NAMEONLY="仅仅生成文件名" ] />
```

#### ■ 描述

获取指定文档(*ID*)指定序号(*index*)的 LOGO 信息。

#### ■ 属性

属性	属性描述	类型	取值	说明
<b>ID</b>	指定文档	数字	文档 <i>ID</i>	文档 <b>ID</b> 默认为当前文档
<b>INDEX</b>	设置获取第几个 LOGO	数字	>0 的整数	默认为 <b>1</b>
<b>NAMEONLY</b>	仅仅显示文件名	字符串		默认为 <b>false</b>

#### ■ 例子

获取当前文档的第一个 Logo 的地址

```
<TRS_DOCUMENTLOGO INDEX=1 />
```

获取文档 ID 为 38 的第一个 Logo 的文件名

```
<TRS_DOCUMENTLOGO DOCID=38 INDEX=1 />
```

置标撰写样例：

显示当前栏目前 10 条文档

Code:

```
<TABLE>
  <TR>
    <TD>标题</TD>
    <TD>入库时间</TD>
  </TR>
  <TRS_DOCUMENTS NUM="10" >
  <TR>
    <TD><TRS_DOCUMENT FIELD="DOCTITLE" /></TD>
    <TD><TRS_DOCUMENT FIELD="CRTIME" /></TD>
  </TR>
</TRS_DOCUMENTS>
</TABLE>
```

发布结果：

```
<TABLE>
  <TR>
    <TD>标题</TD>
```

```

        <TD>入库时间</TD>
    </TR>
    <TR>
        <TD>文档1(标题根据实际情况而变)</TD>
        <TD>2005-01-01 13:20:31</TD>
    </TR>
    <TR>
        <TD>文档2(标题根据实际情况而变)</TD>
        <TD>2005-01-02 13:20:31</TD>
    </TR>
    <TR>
        <TD>文档3(标题根据实际情况而定)</TD>
        <TD>2005-01-03 13:20:31</TD>
    </TR>
    ...
</TABLE>

```

显示当前栏目前 10 条文档，同时定义更多的方式

Code:

```

<TABLE>
    <TR>
        <TD>标题</TD>
        <TD>入库时间</TD>
    </TR>
    <TRS_DOCUMENTS NUM="10" AUTOMORE="TRUE" BEGINMORE="<TR><TD
ColSpan=2 align=right>" ENDMORE="</TD></TR>" MORETEXT="更多新闻">
    <TR>
        <TD><TRS_DOCUMENT FIELD="DOCTITLE" /></TD>
        <TD><TRS_DOCUMENT FIELD="CRTIME" /></TD>
    </TR>

```

```
</TRS_DOCUMENTS>
</TABLE>
```

发布结果:

```
<TABLE>
  <TR>
    <TD>标题</TD>
    <TD>入库时间</TD>
  </TR>
  <TR>
    <TD>文档1 (标题根据实际情况而变)</TD>
    <TD>2005-01-01 13:20:31</TD>
  </TR>
  <TR>
    <TD>文档2 (标题根据实际情况而变)</TD>
    <TD>2005-01-02 13:20:31</TD>
  </TR>
  <TR>
    <TD>文档3 (标题根据实际情况而定)</TD>
    <TD>2005-01-03 13:20:31</TD>
  </TR>
  ...
  <TR>
    <TD colspan=2 align=right>
      <A href="." target="_blank" >更多新闻</A>
    </TD>
  </TR>
</TABLE>
```

按照一页 20 条显示当前栏目前 500 篇文档

Code:



```

<TABLE>
  <TR>
    <TD>标题</TD>
    <TD>入库时间</TD>
  </TR>
  <TRS_DOCUMENTS NUM="500" PAGESIZE=20 AUTOMORE="FALSE">
  <TR>
    <TD><TRS_DOCUMENT FIELD="DOCTITLE" /></TD>
    <TD><TRS_DOCUMENT FIELD="CRTIME" /></TD>
  </TR>
</TRS_DOCUMENTS>
</TABLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function createPageHTML(_nPageCount, _nCurrIndex, _sPageName,
_sPageExt){
  if(_nPageCount == null || _nPageCount<=1){
    return;
  }

  var nCurrIndex = _nCurrIndex || 0;
  if(nCurrIndex == 0)
    document.write("&nbsp;");
  else
    document.write("<a
href=\""+_sPageName+"."+_sPageExt+"\">1</a>&nbsp;");

  for(var i=1; i<_nPageCount; i++){
    if(nCurrIndex == i)
      document.write((i+1) + "&nbsp;");
    else
      document.write("<a href=\""+_sPageName+"_" + i +
."+_sPageExt+"\">" + (i+1) + "</a>&nbsp;");
  }
}
//WCM置标
createPageHTML(${PAGE_COUNT}, ${PAGE_INDEX}, "${PAGE_NAME}",

```

```
"${PAGE_EXT}");  
//-->  
</SCRIPT>
```

发布结果:

```
<TABLE>  
  <TR>  
    <TD>标题</TD>  
    <TD>入库时间</TD>  
  </TR>  
  <TR>  
    <TD>文档1(标题根据实际情况而变)</TD>  
    <TD>2005-01-01 13:20:31</TD>  
  </TR>  
  <TR>  
    <TD>文档2(标题根据实际情况而变)</TD>  
    <TD>2005-01-02 13:20:31</TD>  
  </TR>  
  <TR>  
    <TD>文档3(标题根据实际情况而定)</TD>  
    <TD>2005-01-03 13:20:31</TD>  
  </TR>  
  ...  
</TABLE>  
<SCRIPT LANGUAGE="JavaScript">  
  ...  
  //WCM 置标  
  createPageHTML(5(页数根据实际情况), 0, "default", "htm");  
  //-->  
</SCRIPT>
```

## 显示国际新闻前 10 条文档

Code:

```
<TABLE>
  <TR>
    <TD>标题</TD>
    <TD>入库时间</TD>
  </TR>
  <TRS_DOCUMENTS ID="国际新闻" NUM="10" AUTOMORE="FALSE">
  <TR>
    <TD><TRS_DOCUMENT FIELD="DOCTITLE" /></TD>
    <TD><TRS_DOCUMENT FIELD="CRTIME" /></TD>
  </TR>
</TRS_DOCUMENTS>
</TABLE>
```

发布结果:

```
<TABLE>
  <TR>
    <TD>标题</TD>
    <TD>入库时间</TD>
  </TR>
  <TR>
    <TD>文档1 (标题根据实际情况而变) </TD>
    <TD>2005-01-01 13:20:31</TD>
  </TR>
  <TR>
    <TD>文档2 (标题根据实际情况而变) </TD>
    <TD>2005-01-02 13:20:31</TD>
  </TR>
  <TR>
    <TD>文档3 (标题根据实际情况而定) </TD>
```

```
<TD>2005-01-03 13:20:31</TD>
</TR>
...
</TABLE>
```

显示新闻中心第一个子栏目前 10 条文档

Code:

```
<TABLE>
  <TR>
    <TD>标题</TD>
    <TD>入库时间</TD>
  </TR>
  <TRS_DOCUMENTS ID="新闻中心" CHILDINDEX=0 NUM="10"
AUTOMORE="FALSE" >
  <TR>
    <TD><TRS_DOCUMENT FIELD="DOCTITLE" /></TD>
    <TD><TRS_DOCUMENT FIELD="CRTIME" /></TD>
  </TR>
</TRS_DOCUMENTS>
</TABLE>
```

发布结果:

```
<TABLE>
  <TR>
    <TD>标题</TD>
    <TD>入库时间</TD>
  </TR>
  <TR>
```

```
<TD>文档1(标题根据实际情况而变)</TD>
<TD>2005-01-01 13:20:31</TD>
</TR>
<TR>
<TD>文档2(标题根据实际情况而变)</TD>
<TD>2005-01-02 13:20:31</TD>
</TR>
<TR>
<TD>文档3(标题根据实际情况而定)</TD>
<TD>2005-01-03 13:20:31</TD>
</TR>
...
</TABLE>
```

## 显示新闻中心头条新闻

Code:

```
<TRS_DOCUMENTS ID="新闻中心" CHANNELTYPE="2" NUM="1"
AUTOMORE="FALSE">
<TRS_DOCUMENT FIELD="DOCTITLE" AUTOLINK="true"/>
</TRS_DOCUMENTS>
```

发布结果:

```
<A href="dmtx/200507/03/t2319.html">WCM52成功发布</A>
```

## 显示科教新闻栏目的名称，并且产生链接

Code:

```
<TRS_CHANNEL ID="科教新闻" AUTOLINK="TRUE"/>
```

发布结果:

```
<a href="xwzx/kjxw/">科教新闻</a>
```

显示新闻中心第一个子栏目的信息，并且产生链接

Code:

```
<TRS_CHANNEL ID="新闻中心" CHILDINDEX=0 AUTOLINK="TRUE" />
```

发布结果:

```
<a href="xwzx/rdxw/">热点新闻</a>
```

人工相关，必须手动指定

Code:

```
<TRS_RELNEWS NUM="5" MODE="USERDEF">  
  <TRS_DOCUMENT FIELD="DOCTITLE" /><BR>  
</TRS_RELNEWS>
```

发布结果:

```
<A href="dmtx/200507/03/t2319.html">WCM52成功发布</A>
```

自动相关，按照关键词自动进行检索

Code:

```
<TRS_RELNEWS NUM="5" MODE="KEYWORDS" >
  <TRS_DOCUMENT FIELD="DOCTITLE" /><BR>
</TRS_RELNEWS>
```

发布结果:

```
<A href="dmtx/200507/03/t2319.html">WCM52成功发布</A>
```

获取所有相关文档

Code:

```
<TRS_RELNEWS NUM="5" MODE="ALL" >
  <TRS_DOCUMENT FIELD="DOCTITLE" /><BR>
</TRS_RELNEWS>
```

发布结果:

```
<A href="dmtx/200507/03/t2319.html">WCM52成功发布</A>
```

显示当前栏目的子栏目列表（从第 2 个子栏目开始）

Code:

```
<TRS_CHANNELS STARTPOS=1>
  <TRS_CHANNEL FIELD="CHNLNAME" > 栏目 </TRS_CHANNEL>
</TRS_CHANNELS>
```

发布结果:

Empty dashed box for the first example's result.

显示标题字段的内容，不设定显示长度

Code:

```
<TRS_DOCUMENT FIELD="DOCTITLE">标题</TRS_DOCUMENT>
```

发布结果:

Empty dashed box for the first example's result.

显示标题字段的内容，并设定长度为 50

Code:

```
<TRS_DOCUMENT FIELD="DOCTITLE" NUM="50">标题</TRS_DOCUMENT>
```

发布结果:

Empty dashed box for the second example's result.

显示日期字段的内容，并且格式化日期

Code:

```
<TRS_DOCUMENT FIELD="CRTIME" DATEFORMAT="yyyy-MM-DD">日期</TRS_DOCUMENT>
```

发布结果:



Empty dashed box for the first example's result.

### 显示摘要内容，并且格式化摘要信息

Code:

```
<TRS_DOCUMENT FIELD="DOCABSTRACT" AUTOFORMAT="TRUE">摘要  
</TRS_DOCUMENT>
```

发布结果:

Empty dashed box for the first example's result.

### 显示文档来源

Code:

```
<TRS_DOCUMENT FIELD="DOCSOURCE">文档来源</TRS_DOCUMENT>
```

发布结果:

Empty dashed box for the second example's result.

### 显示文档的第一个图片附件

Code:

```
<TRS_APPENDIX INDEX="0" MODE="PIC" />
```

发布结果:

Empty dashed box for the third example's result.

[Empty dashed box]

### 显示文档的第一个文件附件

Code:

```
<TRS_APPENDIX INDEX="0" MODE="FILE" />
```

发布结果:

[Empty dashed box]

### 显示文档的第一个链接附件

Code:

```
<TRS_APPENDIX INDEX="0" MODE="LINK" />
```

发布结果:

[Empty dashed box]

### 显示当前所在的栏目路径

Code:

```
<TRS_CURPAGE TARGET="_blank" AUTOLINK="TRUE">当前位置  
</TRS_CURPAGE>
```

发布结果:

## 显示当前页面的发布时间

Code:

```
<TRS_DATETIME DATEFORMAT="yyyy-MM-dd">发布日期</TRS_DATETIME >
```

发布结果:

## 在模板中替换已定义内容

Code:

```
<TRS_REPLACE VALUE="Demo">替换内容</TRS_REPLACE>
```

发布结果:

```
<a href=http://www.trs.com.cn>www.TRS.com.cn</a>
```

注: Demo = <a href=http://www.trs.com.cn>www.TRS.com.cn</a>

## 多行显示文档，一行显示三篇文档

Code:

```
<TRS_DOCUMENTS>
```

```
<TRS_RECORD NUM=3>
  <TRS_DOCUMENT FIELD="DOCTITLE" />
</TRS_RECORD>
<br>
</TRS_DOCUMENTS>
```

发布结果:

## 使用页面头部嵌套模板

Code:

```
<TRS_TEMPLATE TEMPNAME=" 页面头部嵌套模板">页面头部嵌套模板
</TRS_TEMPLATE>
```

如果[页面头部嵌套模板](#)中是这样定义的:

```
<TRS_CHANNELS> [ <TRS_CHANNEL FIELD="CHNLNAME" AUTOLINK="true" /> ]
</TRS_CHANNELS>
```

那么相当于在使用嵌套模板的地方也定义了以上类似的代码

发布结果:

## 增加分页标志

Code:

## 第一步：定义分页函数

```
<SCRIPT LANGUAGE="JavaScript">
<!--
function createPageHTML(_nPageCount, _nCurrIndex, _sPageName,
_sPageExt){
    if(_nPageCount == null || _nPageCount<=1){
        return;
    }

    var nCurrIndex = _nCurrIndex || 0;
    if(nCurrIndex == 0)
        document.write("&l&nbsp;");
    else
        document.write("<a
href=\""+_sPageName+"."+_sPageExt+"\">l</a>&nbsp;");

    for(var i=1; i<_nPageCount; i++){
        if(nCurrIndex == i)
            document.write((i+1) + "&nbsp;");
        else
            document.write("<a href=\""+_sPageName+"_" + i +
"+"._sPageExt+"\">" + (i+1) + "</a>&nbsp;");
    }
}
-->
</SCRIPT>
```

## 第二步：在需要分页的地方调用分页函数

```
<SCRIPT LANGUAGE="JavaScript">
    createPageHTML( ${PAGE_COUNT}, ${PAGE_INDEX}, "${PAGE_NAME}",
"${PAGE_EXT}");
</SCRIPT>
```

备注：函数中使用的环境变量说明

名称	描述	类型
<code>\${PAGE_COUNT}</code>	分页页码总数	数值
<code>\${PAGE_INDEX}</code>	现在所处的分页页码	数值
<code>\${PAGE_NAME}</code>	分页页面名称的共同前缀	字符串
<code>\${PAGE_EXT}</code>	分页页面的文件名后缀	字符串

用户可以直接使用这些内置的环境变量，或按自己的需要使用自定义的参数。

发布结果：

## 是否显示嵌套的站点导航条

```
<TRS_CONDITION CONDITION ="${是否显示站点导航条}">
  <TRS_TEMPLATE TEMPNAME="站点导航模板"/>
</TRS_CONDITION>
```

## 概览页面的文档列表中每一篇文档前的前缀

```
<TRS_DOCUMENTS NUM=10>
  <TRS_ECHO VALUE="${概览标题前缀}"/>
</TRS_DOCUMENTS>
```

## 为当天的新闻添加最新新闻标志

```
<TRS_DOCUMENTS NUM=10>
  <TRS_DOCUMENT />
  <TRS_NEWICON TIME="day">新</TRS_NEWICON>
</TRS_DOCUMENTS>
```

## 声明变量的示例 1

```
<TRS_VARIABLE NAME="打开方式" TYPE="S" DEFAULT="_blank`新页打  
开" ENUMERATE="_blank`新页打开" ENUMERATE="_self`本页打开"/>
```

## 声明变量的示例 2

```
<TRS_VARIABLE NAME="打开方式" TYPE="S" DEFAULT="_blank`新页  
打开" >  
  <TRS_ENUM VALUE="_blank" DISPLAY="新页打开" />  
  <TRS_ENUM VALUE="_self" DISPLAY="本页打开" />  
</TRS_VARIABLE>
```

## 使用条件置标示例 2

ISPUBLINK 是文档的一个扩展属性。

```
<TRS_DOCUMENTS>  
  <TRS_CONDITION CONDITION="@ISPUBLINK" REFERENCE="1">  
    <a href="<TRS_DOCUMENT FIELD="DOCPUBURL" />" target="_blank">  
  </TRS_CONDITION>  
  <TRS_DOCUMENT FIELD="DOCTITLE" />  
  <TRS_CONDITION CONDITION="@ISPUBLINK" REFERENCE="1">  
    </a>  
  </TRS_CONDITION>  
</TRS_DOCUMENTS>
```

将ID为 84,85,104 的栏目一天前的新闻作为滚动新闻显示在  
页面中

```
<TRS_ROLLDOCUMENTS FROMCHANNELIDS="84,85,104" TIMERANGE="1">  
.....  
  <TRS_DOCUMENT ...../>  
.....  
</TRS_ROLLDOCUMENTS>
```



## 模板示例

### 例子一：显示一个概览页面

#### HTML 源代码:

```
<HTML>
<BODY>
<FONT style="font-size:10.8pt;line-height:150%">
<TRS_CURPAGE VALUE=" -> "> 当前位置 -- </TRS_CURPAGE>
<ul>
<TRS_DOCUMENTS NUM="60" >
  <li>
    <TRS_DOCUMENT FIELD="DOCTITLE"> 标题放于此处 </TRS_DOCUMENT>
  </li><BR>
</TRS_DOCUMENTS>
</ul>
</FONT>
</BODY>
</HTML>
```

**说明：**黑色部分是HTML的置标，红色部分是TRS置标。

```
<TRS_CURPAGE VALUE=" -> "> 当前位置 </TRS_ CURPAGE>
```

这个置标说明要把页面的当前位置导航自动生成，并且放于此处。

```
<TRS_DOCUMENTS NUM="60" >
```



```
片 </TRS_APPENDIX>
<TRS_DOCUMENT FIELD="DOCHTMLCON" > 正文 </TRS_DOCUMENT>

<p>
<hr noshade size="1" color="#0000FF">
    相关新闻:
    <UL>
        <TRS_RELNEWS NUM="5">
            <li><TRS_DOCUMENT FIELD="DOCTITLE"> 标题
        </TRS_DOCUMENT></li><BR>
        </TRS_RELNEWS>
    </UL>
</FONT>
</BODY>
</HTML>
```

**说明：**黑色部分是HTML的置标，红色部分是TRS置标。其中：

```
<TRS_DOCUMENT FIELD="DOCSOURCE" AUTOLINK="TRUE">来源</TRS_DOCUMENT>
```

这个置标说明此处放置新闻的来源

```
<TRS_DATETIME">时间</TRS_DATETIME>
```

这个置标说明此处放置新闻的发布日期

```
<TRS_APPENDIX MODE="PIC" EXTRA="Border=0 width=80 align=left"> PICTURE
```

```
</TRS_APPENDIX>
```

这个置标说明此处放置新闻的细览图片

```
<TRS_RELNEWS NUM="5">
```

```
<li><TRS_DOCUMENT FIELD="DOCTITLE"> 标题 </TRS_DOCUMENT></li><BR>
```

```
</TRS_RELNEWS>
```

这个置标说明此处放置本新闻的所有相关新闻。（\*关于相关新闻的描述，请参考使用手册或者在线帮助）

### 例子三：显示一个主页页面

#### HTML 源代码：

```
<HTML>
<BODY>
.....

<TABLE border=0 cellPadding=0 cellSpacing=0 width=500>
<TBODY>
<TR>
<TD height=120 vAlign=top width=76>
</TD>
<TD vAlign=top width=106>
  <TRS_DOCUMENTS CHANNELTYPE="1" NUM="1">
    <TRS_APPENDIX INDEX=0 EXTRA="border=0 width=100" > 图片
  </TRS_APPENDIX>
  <font style="font-size:9pt">
    <TRS_DOCUMENT FIELD="DOCTITLE"> TITLE </TRS_DOCUMENT>
  </font>
</TRS_DOCUMENTS>
```

```
</TD>
<TD vAlign=top align=left width=269>
<FONT style="font-size:9pt">
  <ul>
    <TRS_DOCUMENTS CHANNELTYPE="2" NUM="8">
      <LI><TRS_DOCUMENT FIELD="DOCTITLE" NUM="10"> 标 题
    </TRS_DOCUMENT>
  </TRS_DOCUMENTS>
</ul>
</FONT>
</TD>
</TR>
<TR><TD>&nbsp;</TD>
<TD></TD>
</TR>
<TR>
<TD height=120 vAlign=top width=76>
<p align="center"><FONT color=#ffffff>
<IMG align=absBottom height=37 src="images/mid1_0601.gif"
width=24><BR>
<FONT color=#330099 size=2>焦点 </FONT></FONT></p>
</TD>
<TD vAlign=top width=375 colspan="2"><FONT size=2>
<ul>
```

```

        <TRS_DOCUMENTS ID="体育新闻" NUM=10 AUTOMORE=TRUE>
            <LI>
                <TRS_DOCUMENT FIELD="DOCTITLE" > </TRS_DOCUMENT>
            </LI><BR>
        </TRS_DOCUMENTS>
    </ul>
</FONT>
</TD>
.....
</BODY>
</HTML>

```

**说明：**黑色部分是HTML的置标，红色部分是TRS置标。其中：

```

<TRS_DOCUMENTS CHANNELTYPE="1" NUM="1">
<TRS_APPENDIX INDEX=0 EXTRA="border=0 width=100" >图片</TRS_APPENDIX>
<font style="font-size:9pt"><TRS_DOCUMENT FIELD="DOCTITLE"> TITLE
</TRS_DOCUMENT> </font>
</TRS_DOCUMENTS>

```

表示此处放置图片新闻，并只放一条

```

<TRS_DOCUMENTS CHANNELTYPE="2" NUM="8">
<LI><TRS_DOCUMENT FIELD="DOCTITLE" NUM="10">标题</TRS_DOCUMENT>
</TRS_DOCUMENTS>

```

表示此处放置头条新闻，并最多放八条

```

<TRS_DOCUMENTS ID="体育新闻" NUM=10 AUTOMORE=TRUE>
<LI><TRS_DOCUMENT FIELD="DOCTITLE" > </TRS_DOCUMENT></LI><BR>
</TRS_DOCUMENTS>

```

表示此处放置栏目名称为“体育新闻”的新闻概览，并最多放十条，如果超过十条，自动显示更多新闻。

## 附录：标准日期格式说明

字符	描述	表现形式	示例
G	纪元标志	Text	AD
y	年	Year	1996; 96
M	月	Month	July; Jul; 07
w	星期（年）	Number	27
W	星期（月）	Number	2
D	日（年）	Number	189
d	日（月）	Number	10
F	月中星期的第几天	Number	2
E	星期几	Text	Tuesday; Tue
a	上午/下午标志	Text	PM
H	小时(0-23)	Number	0
k	小时 (1-24)	Number	24
K	上/下午的小时(0-11)	Number	0
h	上/下午的小时(1-12)	Number	12
m	分	Number	30
s	秒	Number	55
S	毫秒	Number	978
z	时区	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	时区	RFC 822 time zone	-0800