
Справочник по TypoScript

Ключ расширения: doc_core_tsref

Язык: ru

Версия: 4.5.0

Ключевые слова: forAdmins, forIntermediates

Авторские права 2000-2011, Documentation Team, <documentation@typo3.org>

Этот документ публикуется под Open Content License

доступной на <http://www.opencontent.org/opl.shtml>

Содержимое этого документа относится к TYPO3

- GNU/GPL CMS/Framework доступной на www.typo3.org

Официальная документация

Этот документ является частью официальной документации TYPO3. Одобрение было получено после экспертной оценки командой по документации TYPO3. Читатель вправе ожидать точной информации в этом документе, о любом несоответствии сообщайте в команду по документации (documentation@typo3.org). Официальные документы сохраняются по возможности актуальными командой по документации.

Перевод официальной документации

Этот документ является переводом официальной документации TYPO3. Перевод может быть не столь актуальным, как оригинал. В случае вопросов, обращайтесь к английской версии.

Руководство по ядру системы

Этот документ является руководством по ядру системы. Такие документы описывают встроенную функциональность TYPO3 и служат для обеспечения читателя подробной информацией. Каждый из документов, посвященных ядру системы, адресован конкретному процессу или функции, и их реализации в исходном коде ядра TYPO3. Это может быть информация по доступным API, определенным параметрам настроек и т.п.

Руководства по ядру системы написаны в виде справочников. Читатель должен сверяться с содержанием для определения конкретного, нужного ему раздела, лучшим образом подходящего под решаемую задачу.

Содержание

Справочник по TypoScript.....	1
Введение.....	
Об этом документе.....	5
Нововведения.....	5
Составители.....	5
Обратная связь.....	5
Общая информация.....	6
Типы данных.....	
Введение.....	7
Справочник по типам данных.....	7
Объекты и свойства.....	
Введение.....	16
Условия.....	
Справочник по условиям.....	21
Функции.....	
stdWrap.....	36
imgResource.....	47
imageLinkWrap.....	49
numRows.....	51
select.....	51
split.....	53
if.....	54
typolink.....	56
textStyle.....	62
encapsLines.....	64
tableStyle.....	66
addParams.....	67
filelink.....	68
numberFormat.....	70
parseFunc.....	71
makelinks.....	74
tags.....	75
HTMLparser.....	76
HTMLparser_tags.....	77
Настройка.....	
	79
Top-level objects (основные объекты).....	
"дополнение" TLO.....	81
"CONFIG".....	82
"CONSTANTS".....	111
"PAGE".....	111
"FE_DATA".....	118
"FE_TABLE".....	118
"FRAMESET".....	120
"FRAME".....	120
"META".....	121
"CARRAY".....	122
Объекты содержимого (cObject).....	
HTML.....	125
TEXT.....	125
COBJ_ARRAY (COA, COA_INT).....	125
FILE.....	126
В /typo3/sysext/cms/tslib/class.tslib_content.php он определен как функция PHP fileResource().	126
IMAGE.....	127
IMG_RESOURCE.....	128
CLEARGIF.....	128
CONTENT.....	129
RECORDS.....	131
HMENU.....	132
CTABLE.....	146
OTABLE.....	147
Создается простая таблица. Можно настроить смещение и несколько других параметров тега table.....	147
COLUMNS.....	147
Вставляет таблицу с несколькими столбцами. Размер и стилизация тега table может быть указана в соответствующих параметрах....	147
HRULER.....	148
IMGTEXT.....	149
CASE.....	154

LOAD_REGISTER.....	155	IMGMENU.....	211
RESTORE_REGISTER.....	155	IMGMENUITEM.....	213
FORM.....	155	JSMENU.....	213
SEARCHRESULT.....	164	JSMENUITEM.....	214
Объект может использоваться для отображения результатов поиска.....	164	Приложение А – дополнения media/scripts/ .216	
USER и USER_INT.....	167	media/scripts/ в общем.....	216
TEMPLATE.....	168	fe_adminLib.inc.....	216
FLUIDTEMPLATE.....	173	tipafriendLib.inc.....	230
MULTIMEDIA.....	174	plaintextLib.inc.....	231
SVG.....	175	Приложение В – стандартные шаблоны.....234	
EDITPANEL.....	176	static_template.....	234
GIFBUILDER.....179		Media.....	234
GIFBUILDER.....	179	Приложение С – включение сценариев PHP. .235	
Названия объектов в этом разделе.....	181	Введение.....	235
He GifBuilderObj.....	191	Настройки TypoScript.....	235
Объекты MENU.....192		Включение собственного сценария.....	238
Общие свойства.....	192	Пример.....	240
Общие состояния элементов для TMENU, GMENU и IMGMENU:.....	196	Сохранение данных пользователя или сессии	242
[menuObj].sectionIndex.....	197	Использование встроенной "покупательской корзины"	243
GMENU.....	198	Приложение D – index.php.....245	
GMENU_LAYERS / TMENU_LAYERS.....	201	Введение.....	245
GMENU_FOLDOUT.....	205	Передача данных в index.php.....	245
TMENU.....	208	Поиск.....	246
TMENUITEM.....	209	Почтовые формы.....	246
		Передача данных в базу данных.....	246

Введение

Об этом документе

Этот документ является полным справочником по всем объектам и свойствам TypoScript, применительно к шаблонам в TYPO3 (а не к настройкам Tsconfig).

За информацией по самому TypoScript, обратитесь к документы «Синтаксис TypoScript и его всестороннее изучение».

Эта версия обновлена для версии TYPO3 4.5.

Нововведения

Основные изменения — это включение новых объектов содержимого FLUIDTEMPLATE и SVG, новое свойство stdWrap "numberFormat" и новые свойства для imageLinkWrap для упрощенной интеграции открывающегося по ссылке окна (lightbox). Кроме того, к справочнику по условиям было добавлено новое условие для текущей страницы, и некоторые улучшения по определению новых браузеров и операционных систем. Теперь почти все свойства TypoScript поддерживают функции stdWrap. Был добавлен новый объект GIFBUILDER "ELLIPSE" и исправлены ошибки в документации многих свойств GIFBUILDER.

Было внесено множество исправлений. Обновлены все иллюстрации.

Детальную информацию по изменениям в различных версиях TYPO3 можно найти по следующим ссылкам.



Дополнительная информация по изменениям в свойствах

можно найти в списках изменений для предыдущих версий TYPO3 в wiki:

http://wiki.typo3.org/Documentation_changes_in_4.2

http://wiki.typo3.org/Documentation_changes_in_4.3

http://wiki.typo3.org/Documentation_changes_in_4.4_and_4.5

Составители

Изначально руководство было написано Каспером Скарджи (Kasper Skårhøj). В течение многих лет оно поддерживалось и обновлялось Майклом Стаки (Michael Stucki), Франсуа Сатером (François Suter) и Кристофером Стилмасзиком (Christopher Stelmaszyk).

Обратная связь

По общим вопросам о документации, пишите на documentation@typo3.org.

Если найдена ошибка в этом руководстве, опишите проблему по данному руководству в системе отслеживания ошибок: http://forge.typo3.org/projects/typo3v4-doc_core_tsref/issues

Поддержка качественной документации является тяжелой работой, и команде по документации всегда нужны добровольцы. Если Вы желаете помочь, присоединяйтесь к списку рассылок (typo3.projects.documentation на lists.typo3.org).

Общая информация

Чувствительность к регистру

Все названия и ссылки в TypoScript **чувствительны к регистру!** Это важно помнить. Это значит, что:

```
myObject = HTML
myObject.value = <strong> HTML - code </strong>
```

не тоже самое, что:

```
myObject = html
myObject.Value = <strong> HTML - code </strong>
```

В то время как первое будет распознано как объект содержимого "HTML" и будет выведено ожидаемым образом, второе не будет распознано, и ничего не будет выведено. Даже если во втором примере написать "HTML" в верхнем регистре, он не будет работать, так как свойство "value" все еще написано неверно.

Всегда держите в голове: в данном руководстве важно название объектов.

Типы данных

Введение

The values you assign to properties in TypoScript are often of a specific format. The following table describes these formats.

E.g. if a value is defined as the type "<tag>", you're supposed to supply HTML-code. If it is of the type "resource", it's a reference to a file from the resource-field in the template. If the type is "GraphicColor" a color-definition is expected and you should supply an HTML-valid color-code or RGB-values comma-separated.

Значения, определяемые для свойств, зачастую имеют определенный формат в TypoScript. Эта таблица — справочник по данным форматам.

Например, если значение определено как тип "<tag>", вы можете предположить поддержку кода HTML. Если это тип "resource", то это ссылка на файл из поля ресурсов шаблона. Если тип — "GraphicColor", ожидается определение цвета и, возможно, соответствующие HTML-коды цвета или значения RGB через запятую.

Справочник по типам данных

Тип данных:	Примеры:	Комментарии:	По умолчанию:
<tag>	<BODY bgcolor="red">		
align	right	right / left / center Определяет выравнивание, обычно в тегах HTML	left
VHalign	Гор.выр. = вправо и Верт.выр. = по центру: r , c	Пара значений через запятую. Первое значение отвечает за выравнивание по горизонтали, второе — по вертикали. Возможные значения: r/c/l , t/c/b Значения для выравнивания по горизонтали: вправо, по центру, влево. Значения для выравнивания по вертикали: по верху / по центру / по низу.	l , t
resource	<i>Из поля ресурсов: toplogo*.gif</i> <i>Относительно файловой системы: fileadmin/picture.gif</i>	<ol style="list-style-type: none"> Ссылка на файл из поля ресурсов в шаблоне. Можно написать имя точно или использовать шаблон-звездочку (*) для подстановки. Рекомендуется перед расширением использовать звездочку ** (пример слева). Это гарантирует корректную ссылку на файл даже при копировании шаблона и добавлении номера к названию файла!!! Если значение содержит "/", значит используется ссылка (абсолютная или относительная) 	

Тип данных:	Примеры:	Комментарии:	По умолчанию:
		к файлу в файловой системе, вместо поля ресурсов. Знаки для шаблона не поддерживаются.	
imgResource	<pre>Здесь "file" – imgResource: file = toplogo*.gif file.width = 200</pre> <p>GIFBUILDER:</p> <pre>file = GIFBUILDER file { ... (GIFBUILDER- properties here) }</pre>	<ol style="list-style-type: none"> "resource" (выше) + свойства imgResource (пример слева и ссылка на объект ниже). Типы файлов могут быть любыми из разрешенных в переменной настроек \$TYPO3_CONF_VARS['GFX'] ['imagefile_ext']. Обычно pdf, gif, jpg, jpeg, tif, bmp, ai, pcx, tga, png. Объект GIFBUILDER. 	
HTML-код	Полужирный текст	Чистый код HTML.	
target	_top _blank content	Цель для тега <A>. Обычно то же значение, что и название основного объекта, определяющего фрейм.	
imageExtension	jpg web (gif или jpg ..)	Расширения изображений могут быть любыми из разрешенных в глобальной переменной настроек \$TYPO3_CONF_VARS['GFX'] ['imagefile_ext']. Обычно pdf, gif, jpg, jpeg, tif, bmp, ai, pcx, tga, png. Специальное значение "web". Просто обеспечивает преобразование изображения в формат для сети (gif или jpg), если этого еще не было сделано!	
degree		От -90 до 90, целое число.	
posint / int+		Положительное целое.	
int		Целое. (иногда используется вместо других, более уместных значений, вроде "pixels").	
str / string / value		Строка. (иногда используется вместо других, более уместных значений, вроде "align")	
boolean	1	Булева переменная. Не пустая строка (но не ноль) воспринимается как "истина".	
rotation		Целое, градусы от 0 до 360.	
x,y,w,h	10,10,5,5	x,y – смещение от верхнего левого угла. w,h – ширина и высота.	
HTML-цвет	red #ffeecc	Коды HTML цветов: Black = "#000000" Silver = "#C0C0C0" Gray = "#808080" White = "#FFFFFF" Maroon = "#800000"	

Тип данных:	Примеры:	Комментарии:	По умолчанию:
		Red = "#FF0000" Purple = "#800080" Fuchsia = "#FF00FF" Green = "#008000" Lime = "#00FF00" Olive = "#808000" Yellow = "#FFFF00" Navy = "#000080" Blue = "#0000FF" Teal = "#008080" Aqua = "#00FFFF"	
GraphicColor	red #ffeecc 255,0,255 Дополнительно: red : *0.8 ("red" с фактором затемнения 0.8) #ffeecc : +16 ("ffeecc" преобразуется в #ffedc, так как добавлено 16).	Цвет может быть задан как HTML-цвета или в виде списка RGB-значений (целых) через запятую. Можно добавить дополнительные параметры, делающие математические преобразования: Синтаксис: [цвет] : [параметр], где параметр может быть и целое, которое добавляется/вычитается из трех RGB-каналов или число с плавающей точкой с префиксом "*", которое служит множителем.	
page_id	this 34	Id страницы (целое) или "this" (=id текущей страницы).	
pixels	345	Расстояние в пикселях.	
list	item,item2,item3	Список значений.	
margins	Левый отступ установлен на 10, а отступ снизу — на 5. Отступы сверху и справа не установлены (ноль): 10,0,0,5	l,t,r,b слева, сверху, справа, снизу.	
wrap	Значение будет заключено в тег font с красным цветом шрифта: 	<...> </...> Используется для "оборачивания" чего-либо. Части справа и слева от вертикальной черты помещаются, соответственно, справа и слева от значения к которому применяется выражение.	
linkWrap	При этом создается ссылка на корень сайта: 	<.. {x}.> </...> {x}; x целое (0-9) и указатель на ключ в массиве PHP корневой линии. Ключ означает уровень текущей страницы относительно корня сайта. Если ключ существует, то вместо {x} подставляется uid из уровня на который указывается . Таким образом можно вставить id страниц предыдущего уровня.	
case	upper	"upper" / "lower" Преобразование регистра символов.	
space	5 5	"до после"	

Тип данных:	Примеры:	Комментарии:	По умолчанию:
		Используется для установки отступов до после содержимого.	
date-conf	d-m-y (<i>формат dd-mm-yy</i>)	<p>Смотрите PHP функцию Date()!</p> <p>а - "am" или "pm" A - "AM" или "PM" d – день месяца, число, 2 знака (с нулем) D – день недели, текст, 3 знака; напр. "Fri" F – месяц, текст, длинное; напр. "January" h - час, число, 12 часовой формат H - час, число, 24 часовой формат i – минуты, число j – день месяца, число, без нуля l (нижний регистр 'L') – день недели, текст, длинное; напр. "Friday" m – месяц, число M – месяц, текст, 3 знака; напр. "Jan" s – секунды, число S – английский порядковый суффикс, текст, 2 знака; напр. "th", "nd" U – секунды с начала эпохи Y – год, число, 4 знака w – день недели, число, 0 означает воскресенье y – год, число, 2 знака z – день года, число; напр. "299"</p>	
strftime-conf	Дата "ДД-ММ-ГГ" = %e:%m:%y Время "НН:ММ:СС" = %H:%M:%S или просто %T	<p>%a – аббревиатура дня недели, в соответствии с региональными настройками %A – полное название дня недели, в соответствии с региональными настройками %b – аббревиатура названия месяца, в соответствии с региональными настройками %B – полное название месяца, в соответствии с региональными настройками %c – предопределенная установка даты и времени для текущего региона %C – номер столетия (год, деленный на 100, и усеченный до целого, в диапазоне от 00 до 99) %d – день месяца, как десятичное число (от 00 до 31) %D – то же, что %m/%d/%y %e – день месяца, как десятичное число, однозначные цифры предополнены пробелом (диапазон от ' 1' до '31') %h – то же, что %b %H – часы, как число в 24-часовом диапазоне (от 00 до 23) %I – часы, как число в 12-часовом диапазоне (от 01 до 12)</p>	

Тип данных:	Примеры:	Комментарии:	По умолчанию:
		<p>%j – день года в виде числа (от 001 до 366)</p> <p>%m – месяц, как число (от 01 до 12)</p> <p>%M – минуты, как число</p> <p>%n – знак перехода строки</p> <p>%r - либо 'am', либо 'pm' в соответствии с заданным значением времени или соответствующим строкам для данного региона</p> <p>%g – время, с а.м. и р.м.</p> <p>%R – время в 24 часовом формате</p> <p>%S – секунды, как число</p> <p>%t – знак табуляции</p> <p>%T – текущее время, то же, что %H:%M:%S</p> <p>%u – день недели, как число [1,7], 1 представляет понедельник</p> <p>%U – номер недели текущего года, как число, считая первое воскресенье как первый день первой недели</p> <p>%V – номер недели текущего года в формате ISO 8601:1988, диапазон от 01 до 53, где 1 — первая неделя, имеющая хотя бы 4 дня в текущем году, и понедельник считается первым днем недели.</p> <p>%W – номер недели текущего года, как число, начиная с первого понедельника, как первого дня первой недели.</p> <p>%w – день недели, как число, воскресенье - 0</p> <p>%x – предопределенная дата представляющая текущие региональные установки, без времени</p> <p>%X - предопределенное время представляющее текущие региональные установки, без даты</p> <p>%y – год, как число без столетия (от 00 до 99)</p> <p>%Y – год, как число, включая столетие</p> <p>%Z – временная зона, название или аббревиатура</p> <p>%% - знак '%'. </p>	
UNIX-time	<i>Секунды от 07/04 2000 23:58: 955144722</i>	Секунды, начиная с 1/1 1970...	
path	<i>fileadmin/stuff/</i>	Путь относительно директории с которой работаем.	
<tag>-data	<i><frameset>-data: row может быть '150, *'</i>		
<tag>-параметры	<i><frameset>-параметры может быть 'border="0" framespacing="0"'</i>		
getText		Возвращает значение из массива PHP, определенного в типе. Синтаксис "тип :	

Тип данных:	Примеры:	Комментарии:	По умолчанию:
	<p>= field : header получение содержимого из <code>\$_cObj->data-array[header]</code></p> <p>= parameters : color получение содержимого из <code>\$_cObj->parameters-array[color]</code></p> <p>= register : color получение содержимого из <code>\$GLOBALS['TSFE']->register[color]</code></p> <p>= leveltitle : 1 получение заголовка страницы первого уровня корневой линии</p> <p>= leveltitle : -2 , slide получение заголовка страницы, уровня сразу перед текущей страницей <i>И</i>, если ее нет, ниже по корневой линии, пока не обнаружится заголовок</p> <p>= leveluid : 0 получение <i>id</i> корневой страницы сайта (нулевого уровня)</p> <p>= levelfield : -1 , user_myExtField , slide получение значения пользователяского поля "user_myExtField" в корневой линии (требуется дополнительная настройка в <code>\$TYPO3_CONF_VARS</code> для включения поля!)</p> <p>= global : HTTP_COOKIE_VARS some_cookie получение любой переменной <code>\$HTTP_COOKIE_VARS[some_cookie]</code></p> <p>= date : d-m-y получение текущего времени в формате <i>дд-мм-гг</i></p> <p>= page : title получение заголовка текущей страницы</p>	<p>указатель". Значение для типа чувствительно к регистру.</p> <p>field: [название поля из текущего <code>\$_cObj->data</code> массива данных в <code>cObj</code>] По умолчанию <code>\$_cObj->data</code> массив данных, это <code>\$GLOBALS['TSFE']->page</code> (запись текущей страницы!). В TMENU: <code>\$_cObj->data</code> установлены на запись страницы для каждого элемента меню. В CONTENT/RECORDS <code>\$_cObj->data</code> установлены на фактическую запись. В GIFBUILDER <code>\$_cObj->data</code> установлены на поддерживаемые GIFBUILDER данные.</p> <p>parameters: [название поля из текущего <code>\$_cObj->parameters</code> массива параметров в <code>cObj</code>]. Смотрите <code>->parseFunc</code>!</p> <p>register: [название поля из <code>\$GLOBALS['TSFE']->register</code>]. Смотрите <code>cObject "LOAD_REGISTER"</code></p> <p>leveltitle, leveluid, levelmedia: [<code>levelTitle</code>, <code>uid</code> или <code>media</code> в корневой линии, 0- , negative = назад, " , <code>slide</code>" параметра принудительно перемещается вниз по корневой линии, пока не будет возвращено "правильное" значение. Полезно для порядковых медиа].</p> <p>levelfield: вроде "leveltitle" и т.п., но второй параметр — нужное для получения значения поле в корневой линии. Синтаксис: [указатель, целое], [название поля], ["slide"]</p> <p>global: [GLOBAL-переменная, разделяемая по если нужно получение значения из массива! УСТАРЕЛО, используйте GP, TSFE или <code>getenv</code>]</p> <p>date: [настройка даты]</p> <p>page: [запись текущей страницы]</p>	

Тип данных:	Примеры:	Комментарии:	По умолчанию:
	<p>= current : 1 получение текущего значения</p> <p>= level : 1 получение уровня текущей страницы в корневой линии</p> <p>= GP : stuff получение введенного значения из строки запроса, (&stuff=)</p> <p>= GP : stuff key получение введенного значения из строки запроса, (&stuff[key]=)</p> <p>= getenv : HTTP_REFERER получение любой переменной HTTP_REFERER</p> <p>= getIndpEnv : REMOTE_ADDR получение IP клиента</p> <p>= DB : tt_content:234:header получение значение заголовка записи с uid 234 из таблицы tt_content</p> <p>= fullRootLine : -1, title получение заголовка страницы, сразу до начальной страницы текущего сайта</p> <p>= LLL:EXT:css_styled_content/pi1/locallang.x:login.logout получение локализованной метки для кнопки выхода</p>	<p>current: 1 (получение "текущего" значения)</p> <p>level: 1 (получение уровня текущей страницы в корневой линии)</p> <p>GP: значение из методов GET или POST. Используйте вместо глобальной GPvar: использование "GPvar" осуждается. Используйте вместо этого "GP".</p> <p>getenv: значение из переменных окружения</p> <p>getIndpEnv: значение из t3lib_div::getIndpEnv()</p> <p>DB: значение из базы данных, синтаксис: [название таблицы] : [uid] : [поле]. Здесь можно выбрать любые записи из таблицы, определенной в TCA. Записи, отмеченные для удаления не могут быть получены таким образом.</p> <p>fullRootLine: синтакси [указатель, целое], [название поля], ["slide"] Это свойство можно использовать для получения значений "выше" корня текущей страницы. В приведенном ниже дереве страниц, текущее местоположение – страница "Вы здесь!". Используя описанное выше свойство "levelfield", возможно перейти вверх лишь до страницы "Корень сайта", так как это корень нового (под) сайта. Используя "fullRootLine" возможно перейти вплоть до корня дерева страниц. Числа в квадратных скобках говорят, на какую страницу будут указывать соответствующие значения:</p> <ul style="list-style-type: none"> - Корень дерева страниц [-2] - 1. страница перед [-1] - Корень сайта (здесь шаблон сайта!) [0] - Вы здесь! [1] <p>Параметр "slide" можно добавлять также, как для свойств levelfield до этого.</p> <p>LLL: ссылка на locallang (php или xml) метку label. Ссылка состоит из [ссылки на файл]:[ключа метки]</p>	

Тип данных:	Примеры:	Комментарии:	По умолчанию:
	<p>= path:EXT:ie7/js/ie7-standard.js получение пути к файлу, относительно корня сайта, можно помещать в расширении</p> <p>= cObj : parentRecordNumber получение номера записи текущего cObject</p> <p>= debug : rootLine визуальное представление текущей корневой линии в HTML</p>	<p>path: путь к файлу, возможно находящемуся в расширении, возвращает пустое значение, если файла не существует.</p> <p>cObj: [внутренняя переменная из списка: "parentRecordNumber"]: для CONTENT и RECORDS cObjects запрашивается через select, что возвращает номер строки (1,2,3,...) текущей записи cObject.</p> <p>debug: возвращает содержимое переменной PHP, определенной по ключевому слову, в формате HTML. Доступны ключи "rootLine", "fullRootLine", "data".</p> <p>Получение элементов массива/объекта. Можно выбрать значение массива/объекта, разделяя их вертикальной чертой " ". Пример: = TSFE:fe_user user username</p> <p>Получение нескольких значений. При разделении значений getText знаком "//" (двойной слеш), getText выбирает первое значение. Если оно пусто (""" или ноль), выбирается следующее значение и т.д. Пример: = field:header // field:title // field:uid Выбирается "title" если "header" пустое. Если же "title" тоже пуст, выбирается поле "uid".</p>	
dir	<p><i>Возвращает список всех pdf, gif и jpg файлов из fileadmin/files/ отсортированный по названиям в обратном порядке и с полным путем (с префиксом "fileadmin/files/")</i></p> <p>fileadmin/files/ pdf,gif,jpg name r true</p>	<p>[путь, относительно корня сайта] [список расширений] [сортировка: name, size, ext, date] [обратная: "r"] [возвратить полный путь: булева переменная] Соответствующие файлы возвращаются в строке, через запятую.</p> <p>Замечание: Значение настройки "lockFilePath" должно соответствовать первой части пути к файлам. Иначе путь к файлам будет заблокирован.</p>	
function name	<p>Функция: user_reverseString</p> <p>Метод в классе: user_stringReversing ->reverseString</p>	<p>Указание для вызова функции или метода в классе. Дополнительные сведения содержатся в USER cObject. В зависимости от реализации, названия класса или функции (но не метода) возможно должны быть содержать префикс "user_". Это можно изменить через настройку TYPO3_CONF_VARS. Кроме того, функция / метод обычно вызывается с двумя параметрами, typو</p>	

Тип данных:	Примеры:	Комментарии:	По умолчанию:
		<p>\$conf (TS настройка) и \$content (некое содержимое для обработки и возвращения).</p> <p>Также, при вызове метода в классе, проверяется (при использовании объектов USER/USER_INT) наличие класса с тем же именем, но с префиксом "ux_", и если он существует, создается экземпляр этого класса. За дополнительной информацией по классам обратитесь к документу "Inside TYPO3"!</p>	

[tsref:(datatypes)]

Тип данных: Object types

Это некоторые "типы данных", которые могут встретиться, а также соответствующие их значения:

Тип данных:	Комментарии:
cObject	"cObjects", также называется "Объект содержимого". Смотрите раздел "Объекты содержимого" далее в этом руководстве.
	Примеры: HTML / TEXT / IMAGE
frameObj	FRAMESET / FRAME
menuObj	Смотрите раздел "Объекты меню" далее в этом руководстве.
	Примеры: GMENU / TMENU / IMGMENU / JSMENU
GifBuilderObj	Смотрите раздел "GIFBUILDER" далее в этом руководстве.
	Примеры: TEXT / SHADOW / OUTLINE / EMBOS / BOX / IMAGE / EFFECT

Объекты и свойства

Введение

Ссылка на объекты

Везде, где в таблицах встречается `->[название_объекта]`, имеется ввиду, что свойство — это "название_объекта" со всеми своими свойствами для объекта `название_объекта`. Тип объекта не нужно определять.

Вычисляемые значения (+calc)

Иногда тип данных установлен как "что-то +calc". "+calc", это означает вычисление значения посредством "+/*". *Будьте внимательны, при вычислениях не учитывается "вес" операторов.* Вычисления проводятся по мере следования — справа налево.

Пример:

`45 + 34 * 2 = 158` (что в обычной арифметике записывается как: $(45+34)*2=158$)

"... /stdWrap"

Если тип данных установлен как "type /stdWrap" значит значение пропускается через функцию stdWrap со значениями свойств в качестве параметров.

Пример:

`pixels /stdWrap:` то есть, значение должно быть установлено в пикселях и пропускается через stdWrap.

Фактически в реальном приложении можно сделать так:

```
.pixels.field = imagewidth  
.pixels.intval = 1
```

В этом примере, импортируются значения из поля "imagewidth" текущего \$cObj->массива данных. Но здесь не нужно проверять, чтобы значение было целым числом, так как оно анализируется функцией intval().

optionSplit

optionSplit — очень хитрая функция. Прежде всего, она используется в объектах меню, когда свойства целой ветви элементов устанавливаются за раз. При этом, значения свойств анализируются этой функцией, и, в зависимости от настроек, можно, например, выделить последний пункт меню другим цветом.

Синтаксис примерно такой:

- |*| - разделение значений на части *первая, средняя, последняя*.
 - || - разделитель каждой из *первой, средней, последней* частей на подразделы.
1. Приоритет — *последняя, первая, средняя*.
 2. Если *среднее* значение пустое (""), повторяется последняя часть первого значения.
 3. Если *первое* или *среднее* значение пустое, повторяется первая часть последнего значения перед последним значением.

4. *Среднее* значение повторяется.

Например: `первое1 || первое2 |*| среднее1 || среднее2 || среднее3 |*| последнее1 || последнее2`

Примеры:

Все это очень сложно, и можно подумать, что слишком далеко забежали. Но это действительно полезно.

Рассмотрим меню из пяти элементов:

Введение
О нас
Услуги
Контакты
Ссылки

... и подобную настройку (взята из образца кода первых страниц):

```
temp.topmenu.1.NO {  
    backColor = red  
    ...  
}
```

Если взглянуть в этом справочнике (ниже) на свойство `linkWrap` объекта `GMENU`, то видно, что все свойства `.NO` анализируются функцией `optionSplit`. Это значит, что перед формированием каждого пункта меню, его свойства разделяются этой функцией. Давайте рассмотрим несколько примеров:

Подчасти ||

Пример:

Для всех элементов берется тоже самое значение. Определена только *первая* часть, которая повторяется для всех элементов.

TS: `backColor = red`

Введение	(red)
О нас	(red)
Услуги	(red)
Контакты	(red)
Ссылки	(red)

Пример:

Здесь *первая* часть разделена на подразделы. Третий подраздел повторяется, так как в

меню пять элементов.

TS: `backColor = red || yellow || green`

Введение	(red)	первая, подраздел 1
О нас	(yellow)	первая, подраздел 2
Услуги	(green)	первая, подраздел 3
Контакты	(green)	первая, подраздел 3 (повтор)
Ссылки	(green)	первая, подраздел 3 (повтор)

Части `|*`

Пример:

Теперь определено и *среднее значение* ("white"). Это значит, что после первых двух элементов меню, используется *среднее значение*.

TS: `backColor = red || yellow |*| white`

Введение	(red)	первая, подраздел 1
О нас	(yellow)	первая, подраздел 2
Услуги	(white)	средняя
Контакты	(white)	средняя
Ссылки	(white)	средняя

Пример:

Теперь также определено *последнее значение* ("blue || olive"). Это значит, что после первых двух элементов меню используется *среднее значение*.

TS: `backColor = red || yellow |*| white |*| blue || olive`

Введение	(red)	первая, подраздел 1
О нас	(yellow)	первая, подраздел 2
Услуги	(white)	средняя
Контакты	(blue)	последняя, подраздел 1
Ссылки	(olive)	последняя, подраздел 2

...немного увеличим меню (повторяется *среднее значение*!)

Введение	(red)	первая, подраздел 1
О нас	(yellow)	первая, подраздел 2
Услуги	(white)	средняя
...	(white)	средняя
...	(white)	средняя

...	(white)	средняя
...	(white)	средняя
Контакты	(blue)	последняя, подраздел 1
Ссылки	(olive)	последняя, подраздел 2

...а если урезать меню до четырех элементов (*среднее* значение не используется, ввиду его низкого приоритета)

Введение	(red)	первая, подраздел 1
О нас	(yellow)	первая, подраздел 2
Контакты	(blue)	последняя, подраздел 1
Ссылки	(olive)	последняя, подраздел 2

...а теперь урежем меню до трех элементов (последний подраздел *первого* значения не используется, так как его приоритет ниже *последнего* значения)

Введение	(red)	первая, подраздел 1
Контакты	(blue)	последняя, подраздел 1
Ссылки	(olive)	последняя, подраздел 2

"1: Приоритет последняя, первая, средняя"

В последних двух примерах показано, что *последнее* значение имеет высший приоритет, затем идет *первое* значение, а в конце — *среднее*.

"2: Если среднее значение пустое, повторяется последняя часть первого значения"

Пример:

Здесь *среднее* значение отсутствует. Поэтому повторяется второй подраздел *первого* значения. *Заметьте, что между двумя |*||*| нет никаких пробелов!*

TS: backColor = red || yellow |*||*| blue || olive

Введение	(red)	первая, подраздел 1
О нас	(yellow)	первая, подраздел 2
Услуги	(yellow)	первая, подраздел 2 (повтор)
Контакты	(blue)	последняя, подраздел 1
Ссылки	(olive)	последняя, подраздел 2

"3: Если первое или последнее значения пустые, перед последним значением повторяется первая часть последнего значения"

Пример:

Теперь отсутствуют первое и последнее значения. Поэтому повторяется первый подраздел последнего значения. Заметьте, что между двумя `|*||*` нет никаких пробелов!

TS: `backColor = |*||*| blue || olive`

Введение	(blue)	последняя, подраздел 1 (повтор)
О нас	(blue)	последняя, подраздел 1 (повтор)
Услуги	(blue)	последняя, подраздел 1 (повтор)
Контакты	(blue)	последняя, подраздел 1
Ссылки	(olive)	последняя, подраздел 2

"4: Среднее значение повторяется"

Пример:

TS: `backColor = red |*| yellow || green |*`

Введение	(red)	первая
О нас	(yellow)	средняя, подраздел 1
Услуги	(green)	средняя, подраздел 2
...	(yellow)	средняя, подраздел 1
....	(green)	средняя, подраздел 2
...	(yellow)	средняя, подраздел 1
....	(green)	средняя, подраздел 2
Контакты	(yellow)	средняя, подраздел 1
Ссылки	(green)	средняя, подраздел 2

Условия

Справочник по условиям

Общий синтаксис

Каждое условие заключается в квадратные скобки. Список доступных условий приведен ниже.

"[ELSE]" доступен в качестве оператора иначе (else). Это условие возвращает TRUE, если предыдущие условия вернули FALSE.

Каждый блок условия должен заканчиваться "[GLOBAL]".

Пример:

```
[browser = msie]
    # Код TypoScript для пользователя с Internet Explorer.
[ELSE]
    # Код TypoScript для пользователей с другими браузерами.
[GLOBAL]
```

Общие замечания

Обычно значения обрезаются по пробелу перед сравнением, поэтому пробелы не берутся во внимание.

Можно комбинировать несколько условий посредством двух операторов: && (и), || (или).

Вместо "&&" и "||" можно использовать "AND" и "OR". Оператор AND имеет приоритет над ИЛИ. Если операторов не используется, подразумевается оператор ИЛИ.

Обратите внимание, что условия нельзя использовать внутри фигурных скобок.

Полная информация об условиях содержится в документе «Синтаксис TypoScript и его всестороннее изучение».

Примеры:

Это условие проверяет, открыт ли сайт в браузере Internet Explorer на Windows (но не на Mac):

```
[browser = msie] && [system = win]
```

Это будет соответствовать браузерам Opera или Firefox:

```
[browser = opera] || [browser = firefox]
```

Это соответствие Firefox или Internet Explorer. В случае Internet Explorer, версия должна быть больше или равна 6.

```
[browser = firefox] || [browser = msie] && [version => 6]
```

browser

Синтаксис:

```
[browser = browser1,browser2,...]
```

Значение и условия:

Браузер:	Обозначение:
Amaya	amaya

Браузер:	Обозначение:
AOL	aol
Avant	avant
Camino	camino
Google Chrome	chrome
Mozilla Firefox	firefox
Flock	flock
Gecko	gecko
Konqueror	konqueror
Lynx	lynx
NCSC Mosaic	mosaic
Microsoft Internet Explorer	msie
Navigator	navigator
Netscape Communicator	netscape
OmniWeb	omniweb
Opera	opera
Safari	safari
SeaMonkey	seamonkey
Webkit	webkit
?? (если ничего из приведенного выше не было найдено в user agent)	unknown

Условие работает со строкой user agent. Она анализируется регулярным выражением, ищущим соответствие строки приведенным выше идентификаторам. Если соответствие множественное, используется расположение правее, так как в основном оно и является правильным.

Пример user agent может быть таким:

Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.2.15) Gecko/20110303 Firefox/3.6.15

В этой строке имеются идентификаторы "Gecko" и "Firefox". Условие

[browser = firefox]

возвратит истину.



Ранние версии TYPO3

До TYPO3 4.2 user agent определялся по другому: каждое значение сравнивалось с (\$browsername.\$browsersversion, например, "netscape4.72"), используя `strstr()`.

Таким образом, если значение было "netscape" или даже "scape", либо "net" — то ему соответствовали все браузеры netscape. Если значение было "netscape4", то ему соответствовали все браузеры Netscape 4.xx. Если любое значение из списка соответствует текущему браузеру, возвращалась верное условие.

В TYPO3 версии 4.2 и старше не определяются все перечисленные выше браузеры.

Примеры:

Соответствует браузерам Chrome и Opera-browsers:

```
[browser = chrome, opera]
```

version**Синтаксис:**

```
[version = value1, >value2, =value3, <value4, ...]
```

Условие:

Значения — цифры с плавающей запятой, в качестве разделителя используется точка ".".

Значения могут использовать три оператора:

Оператор:	Функция:
[nothing]	Значение должно быть частью начала строки версии. То есть, версии "4.72" соответствуют значения "4" или "4.7", а значение "4.73" не соответствует. Пример синтаксиса: "значение1"
=	Значение должно точно соответствовать. Версии "4.72" соответствует лишь значение "4.72".
>	Версия должна быть больше значения.
<	Версия должна быть меньше значения.

Примеры:

Это значение соответствует лишь точному номеру версии браузеров "4.03":

```
[version= =4.03]
```

Это значение соответствует всем браузерам, с версией выше 4 и браузерам netscape 3:

```
[version= >4][browser= netscape3]
```

system**Синтаксис:**

```
[system= system1,system2]
```

Значения и сравнение:

Система:	Обозначение:
Linux	linux
Android	android
OpenBSD/NetBSD/FreeBSD	unix_bsd
SGI / IRIX	unix_sgi
SunOS	unix_sun
HP-UX	unix_hp
Chrome OS	chrome
iOS	iOS
Macintosh	mac

Система:	Обозначение:
Windows 7	win7
Windows Vista	winVista
Windows XP	winXP
Windows 2000	win2k
Windows NT	winNT
Windows 98	win98
Windows 95	win95
Windows 3.11	win311
Amiga	amiga

Сравнение с операционной системой, используемой посетителями. Значение системы извлекается из строки useragent.

Значения — это строки и совпадением считается вхождение указанных строк в первую часть идентификатора системы.

Например, значение "win9" соответствует системе "win95" и "win98".

Примеры:

Это значение соответствует только системам windows и mac:

[system= win,mac]



Обратная совместимость с ранними версиями TYPO3

TYPO3 версии 4.4 или выше не обнаруживали все указанные выше системы.

Для обратной совместимости, некоторым системам даются более расплывчатые строки.

Рекомендуется использовать приведенные выше идентификаторы, хотя следующее тоже будет верным:

Система:	Общий идентификатор:
Android	linux
Chrome OS	linux
iOS	mac
Windows 7	winNT
Windows Vista	winNT
Windows XP	winNT
Windows 2000	winNT

device

Синтаксис:

[device= device1, device2]

Значения и сравнение:

Устройство:	Обозначение:
КПК	pda
WAP телефоны	wap
Грабберы:	grabber
Индексирующие роботы:	robot

Значения — это строки и совпадением считается вхождение указанных строк в первую часть идентификатора устройства.

Примеры:

Это значение соответствует WAP-телефонами КПК:

```
[device = wap, pda]
```

useragent

Синтаксис:

```
[useragent = agent]
```

Значения и сравнение:

Это прямое сравнение со строкой пользовательского агента из `getenv("HTTP_USER_AGENT")`

Имеется возможность вставить "*" в начало и/или в конец значения *agent* для соответствия этому шаблону!

Примеры:

Если `HTTP_USER_AGENT` - "Mozilla/4.0 (совместимый; Lotus-Notes/5.0; Windows-NT)", то соответствующее значение:

```
[useragent = Mozilla/4.0 (compatible; Lotus-Notes/5.0; Windows-NT)]
```

Это значение также будет соответствовать:

```
[useragent = *Lotus-Notes*]
```

...но это значение будет соответствовать и пользовательскому агенту вроде этого: "Lotus-Notes/4.5 (Windows-NT)"

Краткий список строк значений пользовательских агентов и условий соответствия им:

HTTP_USER_AGENT:	Описание агента:	Условия соответствия:
Nokia7110/1.0+(04.77)	Nokia 7110 WAP телефон	[useragent= Nokia7110*]
Lotus-Notes/4.5 (Windows-NT)	Браузер Lotus-Notes	[useragent= Lotus-Notes*]
Mozilla/3.0 (compatible; AvantGo 3.2)	Браузер AvantGo	[useragent= *AvantGo*]
Mozilla/3.0 (compatible; WebCapture 1.0; Auto; Windows)	Adobe Acrobat 4.0	[useragent= *WebCapture*]

WAP-агенты:

Это некоторые из известных WAP агентов:

HTTP_USER_AGENT:	HTTP_USER_AGENT (продолжение):
ALAV UP/4.0.7	PLM's WapBrowser
Alcatel-BE3/1.0 UP/4.0.6c	QWAPPER/1.0
AUR PALM WAPPER	R380 2.0 WAP1.1
Device V1.12	SIE-IC35/1.0
EricssonR320/R1A	SIE-P35/1.0 UP/4.1.2a
fetchpage.cgi/0.53	SIE-P35/1.0 UP/4.1.2a
Java1.1.8	UP.Browser/3.01-IG01
Java1.2.2	UP.Browser/3.01-QC31
m-crawler/1.0 WAP	UP.Browser/3.02-MC01
Materna-WAPPreview/1.1.3	UP.Browser/3.02-SY01
MC218 2.0 WAP1.1	UP.Browser/3.1-UPG1
Mitsu/1.1.A	UP.Browser/4.1.2a-XXXX
MOT-CB/0.0.19 UP/4.0.5j	UPG1 UP/4.0.7
MOT-CB/0.0.21 UP/4.0.5m	Wapalizer/1.0
Nokia-WAP-Toolkit/1.2	Wapalizer/1.1
Nokia-WAP-Toolkit/1.3beta	WapIDE-SDK/2.0; (R320s (Arial))
Nokia7110/1.0 ()	WAPJAG Virtual WAP
Nokia7110/1.0 (04.67)	WAPJAG Virtual WAP
Nokia7110/1.0 (04.67)	WAPman Version 1.1 beta:Build
Nokia7110/1.0 (04.69)	W2000020401
Nokia7110/1.0 (04.70)	WAPman Version 1.1
Nokia7110/1.0 (04.71)	Waptor 1.0
Nokia7110/1.0 (04.73)	WapView 0.00
Nokia7110/1.0 (04.74)	WapView 0.20371
Nokia7110/1.0 (04.76)	WapView 0.28
Nokia7110/1.0 (04.77)	WapView 0.37
Nokia7110/1.0 (04.80)	WapView 0.46
Nokia7110/1.0 (30.05)	WapView 0.47
Nokia7110/1.0	WinWAP 2.2 WML 1.1
	wmlb
	YourWap/0.91
	YourWap/1.16
	Zetor

language

Синтаксис:

```
[language = lang1, lang2, ...]
```

Сравнение:

Значения должны строго соответствовать значению getenv("HTTP_ACCEPT_LANGUAGE") из PHP. Также, если значение заключено в "*" (например "*en-us*"), то все языки, найденные в строке HTTP_ACCEPT_LANGUAGE будут разбиты и будет произведена попытка сравнения с любой из частей этой строки. Так, обычно строка выглядит примерно так: "de,en-us;q=0.7,en;q=0.3", и значение "*en-us*" будет ей соответствовать.

IP

Синтаксис:

```
[IP = ipaddress1, ipaddress2, ...]
```

Сравнение:

Значение сравнивается с `getenv("REMOTE_ADDR")` из PHP.

Возможно включать "*" вместо одной из частей значения. Также можно включать в список первую, вторую или третью части адреса, при этом сравниваться будут лишь они.

Примеры:

В этом примере строке соответствуют лишь IP-адреса, начинающиеся с "123":

`[IP = 123.*.*.*]`

или

`[IP = 123]`

В этом примере строке соответствуют IP-адреса, оканчивающиеся на "123" или адрес "192.168.1.34":

`[IP = *.*.*.123] [IP = 192.168.1.34]`

hostname**Синтаксис:**

`[hostname = hostname1, hostname2, ...]`

Сравнение:

Значение сравнивается с полноценным названием хоста `getenv("REMOTE_ADDR")`, найденным в PHP.

Значение — список названий соответствующих доменов через запятую. Возможно использование знака звездочки *, но она не может использоваться в качестве части строки, а должна заменять полное название хоста (например, `myhost.*.com` => правильно, `myhost.*domain.com` => не правильно)

hour**Синтаксис:**

`[hour = hour1, > hour2, < hour3, ...]`

Замечание: первый знак "=", сразу после слова "hour", обязателен, и не является оператором. После него идет оператор, а потом — час.

Сравнение:

Возможные значения — от 0 до 23 (в 24-часовом формате). Значение в виде числа с плавающей точкой сравнивается с текущим часом сервера.

Как указано в разделе "Синтаксис" выше, можно разделять несколько значений в одном условии при помощи запятой. Запятая связывает условия в виде логического сложения (ИЛИ), то есть все условие будет верным, если хотя бы один из операндов верен.

Оператор:	Функция:
[none]	Требуется точное соответствие.
>	Час должен быть больше значения.
<	Час должен быть меньше значения.

Оператор:	Функция:
<code><=</code>	Час должен быть меньше или равен значению.
<code>>=</code>	Час должен быть больше или равен значению.
<code>!=</code>	Час не должен быть равен значению.

Примеры:

Будет верным, если значение между 9 и 10 часами (в соответствии с временем на сервере):

`[hour = 9]`

Будет верным, если значение меньше 7 часов:

`[hour = < 7]`

Будет верным, если значение меньше 15 часов:

`[hour = <= 14]`

Следующий пример показывает использование запятой внутри условия:

Будет верным, если значение между 8 и 9 часами (час равен 8) или после 16 часов (час больше или равен 16):

`[hour = 8, >= 16]`

Будет верным, если значение между 16 и 8 часами (помните, что запятая выступает в роли оператора ИЛИ):

`[hour = > 15, < 8]`

И наоборот, условие верно, если менее 8 и более 16 часов:

`[hour = > 7] && [hour = < 16]`

minute

Тот же синтаксис, что и у "Hour" выше!

Синтаксис:

`[minute = ...]`

Сравнение:

Минуты часа, возможный интервал значений 0-59.

Помимо этого используется тот же способ сравнения, что и для часа.

month

Тот же синтаксис, что и у "Hour" выше!

Синтаксис:

`[month = ...]`

Сравнение:

Месяц, начиная с января — 1 и до декабря — 12.

Помимо этого используется тот же способ сравнения, что и для часа.

year

Тот же синтаксис, что и у "Hour" выше! За дополнительной информацией обратитесь к руководству по PHP, формату строки `Y` функции `date()`.

Синтаксис:

`[year = ...]`

Сравнение:

Год, как 4-значное число.

Помимо этого используется тот же способ сравнения, что и для часа.

dayofweek

Тот же синтаксис, что и у "Hour" выше!

Синтаксис:

`[dayofweek = ...]`

Сравнение:

День недели, начиная с воскресенья — 0 и до субботы — 6.

Помимо этого используется тот же способ сравнения, что и для часа.

dayofmonth

Тот же синтаксис, что и у "Hour" выше!

Синтаксис:

`[dayofmonth = ...]`

Сравнение:

День месяца, возможный интервал значений 1-31

Помимо этого используется тот же способ сравнения, что и для часа.

dayofyear

Тот же синтаксис, что и у "Hour" выше! За дополнительной информацией обратитесь к руководству по PHP, формату строки `z` функции `date()`.

Синтаксис:

`[dayofyear = ...]`

Сравнение:

День года, возможный интервал значений 0-364 (или 365 для високосных лет). Учитывая, что условие использует 0 для первого дня, окажется, что `[dayofyear = 7]` будет верным для 6^{го} Января.

Помимо этого используется тот же способ сравнения, что и для часа.

usergroup

Синтаксис:

```
[usergroup = group1-uid, group2-uid, ...]
```

Сравнение:

Сравнение может возвращать истину лишь при не пустом списке групп (глобальная переменная "gr_list").

Значения должны либо существовать в списке групп ИЛИ значение должно быть "*".

Пример:

Соответствует всем авторизациям:

```
[usergroup = *]
```

Это значение соответствует регистрации пользователей из групп с uid 1 и/или 2:

```
[usergroup = 1,2]
```

loginUser

Синтаксис:

```
[loginUser = fe_users-uid, fe_users-uid, ...]
```

Сравнение:

Соответствует uid авторизованного fe_user. Работает также, как "usergroup" выше, включая знак звездочки * для выбора ЛЮБОГО пользователя.

Пример:

Соответствует любой авторизации (вместо "[usergroup = *]" для соответствия авторизации пользователю!):

```
[loginUser = *]
```

Кроме того, возможно проверить, авторизованы ли внешние пользователи вообще.

Пример:

Соответствует, если нет авторизованных пользователей:

```
[loginUser = ]
```

page

Синтаксис:

```
[page|field = value]
```

Сравнение:

Это условие проверяет значения текущей записи страницы. Хотя то же самое можно сделать при помощи условий TSFE:[field] во внешнем интерфейсе, это условие остается верным и во внешнем, и во внутреннем интерфейсах.

Пример:

Условие верно, если поле layout установлено на 1:

```
[page|layout = 1]
```

treeLevel

Синтаксис:

```
[treeLevel = levelnumber, levelnumber, ...]
```

Сравнение:

Проверка наличия последнего элемента корневой линии на одном из соответствующих уровней. Уровень = 0 – "корень" сайта. Уровень = 1 — элемент меню первого уровня.

Пример:

Так можно поменять что-либо в шаблоне, если просматриваемая страница находится либо на 0 уровне (основном), либо на 2 уровне.

```
[treeLevel = 0,2]
```

PIDinRootline

Синтаксис:

```
[PIDinRootline = pages-uid, pages-uid, ...]
```

Сравнение:

Проверка нахождения PID (uid страниц) на соответствующем уровне корневой линии.

Пример:

Так можно поменять что-либо в шаблоне, если в корневой линии просматриваемой страницы имеются страницы 34 или 36.

```
[PIDinRootline = 34,36]
```

PIDupinRootline

Синтаксис:

```
[PIDupinRootline = pages-uid, pages-uid, ...]
```

Сравнение:

Делает тоже, что и PIDinRootline, за исключением того, что uid текущей страницы исключается из проверки.

compatVersion

Синтаксис:

```
[compatVersion = x.y.z]
```

Сравнение:

Условие соответствия версии. Эта версия не обязательно эквивалентна версии TYPO3, это значение из настроек, и оно может быть изменено из Инструмента установки (Install Tool).

"compatVersion" особенно полезно, если нужно предоставить новый параметр настроек по умолчанию, и оставить обратную совместимость со старой версией TYPO3.

globalVar

Синтаксис:

```
[globalVar = var1 = value1, var2 > value2, var3 < value3, var4 <= value4, var5 >= value5,
var6 != value6, ...]
```

Сравнение:

Цифры с плавающей запятой, сравниваются с глобальными переменными "var1", "var1", "var2" ... выше.

Можно разделять несколько значений в одном условии при помощи запятой. Запятая связывает условия в виде логического сложения (ИЛИ), то есть все условие будет верным, если хотя бы один из операндов верен.

Оператор:	Функция:
=	Строгое соответствие
>	Переменная должна быть больше значения
<	Переменная должна быть меньше значения
<=	Переменная должна быть меньше или равна значению
>=	Переменная должна быть больше или равна значению
!=	Переменная не должна равняться значению

Примеры:

Соответствует ли url, подобному "...&print=1":

```
[globalVar = GP:print > 0]
```

Выполняется, если id страницы выше или равно 10:

```
[globalVar = TSFE:id >= 10]
```

Выполняется, если id страницы не равно 316:

```
[globalVar = TSFE:id != 316]
```

Выполняется при отсутствии GPvar style:

```
[globalVar = GP:style = ]
```

Выполняется, если переменная GET/POST "L" равна 8 или переменная GET/POST "M" равна 2, либо все вместе:

```
[globalVar = GP:L = 8, GP:M = 2]
```

Соответствует страницам, с полем Шаблон, установленным на "Layout 1":

```
[globalVar = TSFE:page|layout = 1]
```

Если переменная {\$constant_to_turnSomethingOn} равна "1", то условие выполняется:

```
[globalVar = LIT:1 = {$constant_to_turnSomethingOn}]
```

globalString

Синтаксис:

```
[globalString = var1=value, var2= *value2, var3= *value3*, ...]
```

Сравнение:

Прямое сравнение с глобальными строками.

Возможно добавление в значение знака звездочки "*" в качестве подстановки или регулярного выражения в стиле PCRE (должно быть заключено в "/").

Примеры:

Соответствие HTTP_HOST "www.typo3.com":

```
[globalString = IENV:HTTP_HOST = www.typo3.com]
```

Соответствие будет и при этом выражении:

```
[globalString = IENV:HTTP_HOST = *typo3.com]
```

...но при этом соответствие HTTP_HOST будет и при этом значении: "demo.typo3.com".

ВАЖНО ЗАМЕЧАНИЕ ПО globalVar и globalString:

Можно использовать значения из глобальных массивов и объектов, разделяя названия переменных знаком "|" (вертикальная черта).

Примеры:

глобальные переменные \$HTTP_POST_VARS["key"]["levels"] можно запросить так "HTTP_POST_VARS|key|levels"

Также помните, что рекомендуется программировать свои сценарии в соответствие с настройками php.ini-оптимизации. Просмотрите этот файл (из своего дистрибутива).

Это значит, что нужно получать значения, типа HTTP_HOST через getenv(), возможно сделать запрос GET/POST значений через t3lib_div::GP(). И наконец, полезны многие значения из объекта TSFE. Чтобы получить эти значения для соответствия условиям в "globalVar" и "globalString", используйте префикс в названии переменных - "IENV"/"ENV:", "GP:", "TSFE:" или "LIT:", соответственно. Для разделения ключей в массивах и/или объектах может использоваться разделитель "|". "LIT" означает "literal", а строка после ":" обрезается и возвращается в качестве значениями (без разделителя "|" и чего-бы то ни было)

Примечание: использование префикса "IENV:" крайне рекомендуется для получения переменных сервера/окружения, зависимых от системы. В основном значение будет возвращено через вызов функции t3lib_div::getIndpEnv(). Через "ENV:" будет получен прямой вывод из getenv(), который НЕ всегда одинаковый на разных системах!

Примеры:

Соответствие начала адреса удаленного компьютера "192.168."

```
[globalString = IENV:REMOTE_ADDR = 192.168.*]
```

Соответствие имени пользователя значению "test":

```
[globalString = TSFE:fe_user|user|username = test]
```

userFunc

Синтаксис:

```
[userFunc = user_match(checkLocalIP)]
```

Сравнение:

Вызывается функция “user_match” с первым параметром “checkLocalIP”. Вы сами пишите эту функцию, Вы же решаете, что она проверяет. Результат функции должен быть истина/ложь.

Пример:

Вставьте эту функцию в свой файл localconf.php:

```
function user_match($cmd) {
    switch($cmd) {
        case 'checkLocalIP':
            if (strstr(getenv('REMOTE_ADDR'), '192.168')) {
                return TRUE;
            }
        break;
        case 'checkSomethingElse':
            // ....
        break;
    }
}
```

Условия будут выполнены, если адрес удаленного компьютера содержит “192.168” - это и выясняет функция.

```
[userFunc = user_match(checkLocalIP)]
```

Функции

stdWrap

Эта функция зачастую добавляется в качестве свойств к значениям в TypoScript.

Пример с объектом содержимого "HTML":

```
10 = HTML
10.value = текст
10.value.case = upper
```

Здесь значение объекта "10" перед выводом переводится в верхний регистр.

Свойства stdWrap применяются в порядке их следования в приведенной ниже таблице.

Если нужны дополнительные сведения, обратитесь к

typo3/sysext/cms/tslib/class.tslib_content.php, где можно найти функцию stdWrap() и массив \$stdWrapOrder, в котором и представлен точный порядок выполнения.

Свойства stdWrap относящиеся к содержимому

Свойства анализируются в порядке их следования в таблице. Свойства "data", "field", "current", "cObject" (в этом порядке!) — особенные, так как они используются для импорта содержимого из переменных или массивов. Приведенный выше пример может быть переписан так:

```
10 = HTML
10.value = текст
10.value.case = upper
10.value.field = header
```

При этом строка "10.value = текст" не используется, потому что вместо нее значение "импортируется" из поля под названием "header" из \$cObj->массива данных.

Свойство:	Тип данных:	Описание:	По умолчанию:
Получение данных:			
setContentToCurrent	boolean	Устанавливает текущее значение, как входящее значение для функции.	
setCurrent	string /stdWrap	Установка "текущего" значения. Обычно устанавливается из какой-либо внешней подпрограммы, так что будьте внимательны. Но это-то и удобно с другой стороны.	
lang	Массив языковых ключей	<p>Используется для определения определенных значений для языков.</p> <p>Если ключ для глобального языка установлен через свойство -> config, то .language, имеется в этом массиве, и используется его значение, вместо определенного по умолчанию входного значения в stdWrap.</p> <p>Пример:</p> <pre>config.language = de page.10 = TEXT page.10.value = I am a Berliner! page.10.lang.de = Ich bin ein Berliner!</pre> <p>Будет выведено значение "Ich bin..." вместо "I am..."</p>	
data	getText		

Свойство:	Тип данных:	Описание:	По умолчанию:
field	название поля	<p>Устанавливает значение содержимого \$cObj->data[field]</p> <p>Пример: установить для содержимого значение поля "title": ".field = title" \$cObj->изменение данных. Смотри описание для типа данных "getText"/field!</p> <p>Замечание: возможно разделять названия полей знаком "//". Скажем, вы установили в качестве значения "nav_title // title", при этом будет возвращено значение поля nav_title, если оно не пустое, при возвращении значения поля title.</p>	
current	boolean	Установка "текущего" значения для содержимого (смотрите ->split).	
cObject	cObject	Загрузка содержимого из объекта-содержимого.	
numRows	->numRows	Возвращает число строк из выборки.	
filelist	dir /stdWrap	Читает директорию и возвращает список файлов. Значение разделяется знаком " " на параметры: 1: Путь 2: Список расширений через запятую (без пробелов); если пуст, то используются все расширения. 3: сортировка: name, size, ext, date, mdate (дата изменения) 4: обратный порядок: устанавливается через "r" 5: fullPath_flag: при установке, список файлов возвращается с полным путем, а не просто как названия файлов	
preUserFunc	Название функции	Вызов PHP-функции или метода в классе для анализа текущего содержимого, передаваемого в первом параметре, а свойства передаются как второй параметр. Смотрите <i>postUserFuncCalling</i> .	
Переназначение / условия:			
override	string /stdWrap	Если "override" возвращает что-либо кроме "" или нуля (урезается), то в содержимое загружается ее результат!	
prefEmptyListNum	(то же, что и "listNum" ниже)	(то же, что и "listNum" ниже)	
ifEmpty	string /stdWrap	Если к этому моменту содержимое пусто (урезается), загружается содержимое из "ifEmpty". Нули рассматриваются как пустые значения!	
ifBlank	string /stdWrap	То же, что и "ifEmpty", но делается проверка через strlen().	

Свойство:	Тип данных:	Описание:	По умолчанию:
listNum	int +calc +"last"	<p>Разбивает содержимое по "," (запятой), а содержимое устанавливается в item[value].</p> <p>Специальный ключ: "last" — последний элемент массива!</p> <p>.splitChar (строка): Определяется строка, использующаяся в качестве разделителя. Если splitChar - целое число, используется знак с этим кодом (например "10" в качестве разделителя строк...). По умолчанию: " " (запятая)</p> <p>.stdWrap (свойства stdWrap): stdWrap свойства для listNum...</p> <p>Примеры: Имеются значение "item 1, item 2, item 3, item 4": При этом будет возвращено значение "item 3": .listNum = last - 1</p>	
trim		PHP-функция trim(); удаляет пустое место вокруг значения.	
stdWrap	->stdWrap	Рекурсивный вызов функции stdWrap.	
required	boolean	<p>Этот флагозначает, что требуется установка какого-либо значения для содержимого после любого производимого импорта или обработки содержимого (data, field, current, listNum, trim). Ноль НЕ расценивается как пустое значение! Вместо этого используйте "if"!</p> <p>Для пустого содержимого, сразу же возвращается "".</p>	
if	->if	Если объект if возвращает ложь, stdWrap сразу же возвращает "".	
fieldRequired	название поля	Для этого поля ТРЕБУЕТСЯ значение.	
Обработка данных:			
csConv	string	Преобразование кодировки символов строки из заданной в качестве значения кодировки, в кодировку вывода для внешнего интерфейса (renderCharset).	
parseFunc	object path reference / ->parseFunc	<p>Инструкция обработки содержимого.</p> <p>Примечание: при вводе в качестве значения строки, она будет использована как ссылка на путь объекта глобально в дереве объектов TypoScript. Это будет основной настройкой для parseFunc объединенной с любыми свойствами, которые вы здесь добавите. Это работает точно так же, как ссылки для элементов содержимого.</p> <p>Пример:</p> <pre>parseFunc = < lib.parseFunc_RTE parseFunc.tags.myTag = TEXT parseFunc.tags.myTag.value = This will be inserted when &lt;myTag&gt; is found!</pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
HTMLparser	boolean / ->HTMLparser	Этот объект позволяет обрабатывать HTML-содержимое и по всякому фильтровать содержимое. Свойства и нужное значение те же, что и в ->HTMLparser. (Смотрите параметры для ->HTMLparser в "Core API").	
split	->split		
prioriCalc	boolean	Вычисление значения, используя операторы $-+*/%^$ плюс поддержка приоритета операторов + и - и уровней скобок (). . (точка) десятичный разделитель. Возвращает значение double. Если .prioriCalc установлен на "intval" возвращается целое число. Проверка ошибок, деления на ноль и других неверных значений не проводится. Используйте правильный синтаксис, так как в дальнейшем функция может быть расширена поддержкой дополнительных операторов и особенностей. Примеры: <code>100%7 = 2 -5*-4 = 20 +6^2 = 36 6 ^(1+1) = 36 -5*-4+6^2-100%7 = 54 -5 * (-4+6) ^ 2 - 100%7 = 98 -5 * ((-4+6) ^ 2) - 100%7 = -22</code>	
char	int	Значение содержимого устанавливается в <code>chr(value)</code> . <code>PHP: \$content = chr(intval(\$conf['char']));</code>	
intval	boolean	PHP функция <code>intval()</code> ; возвращает целое число. <code>PHP: \$content = intval(\$content);</code>	
numberFormat	->numberFormat	Формат значения с плавающей запятой для любого количества символов (полезно, например, для цен).	
date	date-conf	Содержимое должно быть типа "UNIX-time". Возвращаемое значение содержимого форматируется как дата. <code>\$content=Date(\$conf["date"], \$content);</code> Пример в котором импортируется <code>timestamp</code> : <code>.value.field = tstamp .value.date =</code>	
strftime	strftime-conf	То же, что и "date" выше. Смотрите код или тип данных "strftime-conf" в руководстве по PHP (<code>strftime</code>). Это форматирование полезно, если региональные установки сделаны заранее в объекте CONFIG. Ищите здесь. Свойства: .charset : возможно установить кодировки выводимой строки, если ее нужно преобразовать через <code>renderCharset</code> . По умолчанию разумно предположить кодировку из <code>t3lib_cs</code> .	

Свойство:	Тип данных:	Описание:	По умолчанию:
age	boolean or string	<p>При включении через "1" (целое число) содержимое будет видно как дата (UNIX-time) и разница во времени в содержимом и настоящем времени возвращается в одном из восьми вариантов: "xx min", "xx hrs", "xx days", "xx yrs", "xx min", "xx hour", "xx day" или "year".</p> <p>Переход на использование следующего шаблона - 60 минут, 24 часа, 365 дней.</p> <p>ЗАМЕЧАНИЕ: Если установить это свойство не как целое число, то оно должно использовать формат восьми частей. Первые четыре значения для множественного числа, оставшиеся — для единственного. Стока по умолчанию: " min hrs days yrs min hour day year"</p> <p>Установите другую строку, если нужно изменить единицы. Возможно включить символы ". ". Они в любом случае будут удалены, но при этом пробелы между числом и единицами сохраняются.</p> <p>Пример: <pre>lib.ageFormat = TEXT lib.ageFormat.data = page:tstamp lib.ageFormat.age = " Minuten Stunden Tage Jahre Minute Stunde Tag Jahr"</pre> </p>	
case	case	<p>Преобразование регистра</p> <p>Для этой операции используется "renderCharset".</p>	
bytes	boolean	<p>Форматирование вводимого (целого числа) в виде байтов: bytes, kb, mb</p> <p>Добавив значения для свойства "labels", можно использовать отличающиеся от принятых по умолчанию суффиксы. Метки для байт, кило-, мега- и гига- разделяются вертикальной чертой () и могут быть заключены в кавычки "". Например: " K M G" (значение по умолчанию).</p> <p>Полное написание:</p> <pre>bytes.labels = " K M G"</pre>	
substring	[p1], [p2]	<p>Возвращает подстроку с [p1] и [p2] посылаемыми как второй и третий параметры в функцию PHP <code>substring</code>.</p> <p>Для этой операции используется "renderCharset".</p>	
removeBadHT ML	boolean	Удаляется "плохой" HTML код, считающийся опасным для ошибок XSS, на основе шаблона, отфильтровывающего HTML.	

Свойство:	Тип данных:	Описание:	По умолчанию:
cropHTML		<p>Обрезает содержимое до нужной длины. В отличие от stdWrap.crop, оставляет теги HTML. Не обрезает по содержимому внутри тегов и закрывает открытые теги. Объекты (вроде ">") считаются одним символом. Смотрите ниже stdWrap.crop относительно синтаксиса и примеров.</p> <p>Обратите внимание, что stdWrap.crop не должно использоваться, если уже применялось stdWrap.cropHTML.</p>	
stripHtml	boolean	Очищает все html-теги.	
crop		<p>Обрезает содержимое до нужной длины.</p> <p>Синтаксис: +/- (знаков) = слева / справа [строка] [булева переменная: сохранять целые слова]</p> <p>Примеры:</p> <p>20 ... => максимум 20 знаков. Если больше, значение будет обрезано до первых 20 знаков с добавлением "..."</p> <p>-20 ... => максимум 20 знаков. Если больше, значение будет обрезано до последних 20 знаков с добавлением "..."</p> <p>20 ... 1 => максимум 20 знаков. Если больше, значение будет обрезано до последних 20 знаков с добавлением "...". Если образка приходится на середину слова, удаляется все слово.</p> <p>Для этой операции используется "renderCharset".</p>	
rawUrlEncode	boolean	Пропуск содержимого через PHP функцию rawurlencode().	
htmlSpecialChars	boolean	Пропуск содержимого через PHP функцию htmlspecialchars(). Дополнительное свойство ".preserveEntities" сохраняет все значащие символы.	
doubleBrTag	string	Заменяет все символы двойного перевода строки на данное значение.	
br	boolean	PHP функция nl2br(); преобразует концы строк в теги .	
brTag	string	Все коды ASCII "10" (CR) заменяются на данное значение.	
encapsLines	->encapsLines	Позволяет разбить содержимое по chr(10) и независимо обработать каждую строку. Использует формат содержимого, сделанного в RTE.	
keywords	boolean	Разделяет содержимое по знакам "," ";" и chr(10) (разрыв строки), обрезая каждое значение и возвращает список значений через запятую.	
innerWrap	wrap /stdWrap	"Обертка" содержимого	
innerWrap2	wrap /stdWrap	То же, что .innerWrap (но смотрите порядок их выполнения)	
fontTag	wrap		

Свойство:	Тип данных:	Описание:	По умолчанию:
addParams	->addParams	Позволяет добавить параметры тега к содержимому, если содержимое является тегом!	
textStyle	->textStyle	Заключает содержимое в теги font.	
tableStyle	->tableStyle	Заключает содержимое в теги table.	
filelink	->filelink	Используется для создания списков ссылок на файлы.	
preCObject	cObject	cObject предваряющий содержимое.	
postCObject	cObject	Cobject добавляемый к содержимому.	
wrapAlign	align /stdWrap	Заключает содержимое в теги <div style="text-align: [value];"> </div>, если установлено выравнивание (align).	
typolink	->typolink	Заключает содержимое в тег-ссылку.	
TCASelectItem	Массив свойств	<p>Преобразование списка значений через запятую в представление элементов в ТСА.</p> <p>.table (string): просматриваемая таблица .field (string): поле для вставки .delimiter (string): разделитель для объединения нескольких элементов.</p> <p>Примечание: на данный момент это работает лишь с полями ТСА типа "select", не участвующих в связях базы данных.</p>	
spaceBefore	int /stdWrap	Отступ в пикселях до. При помощи clear-gif; 	
spaceAfter	int /stdWrap	Отступ в пикселях после. При помощи clear-gif; 	
space	space /stdWrap	<p>[пробел до] [пробел после]</p> <p>Дополнительное свойство: .useDiv = 1 При установке, clear gif не используется вместо тега <div> с атрибутом style с настройками для высоты. (относится и к spaceBefore и spaceAfter).</p>	
wrap	wrap /+.splitChar	.splitChar определяет альтернативный символ-разделитель (по умолчанию " " - вертикальная черта)	
noTrimWrap	"special" wrap	Значение заключается между val1 и val2, в приведенном примере — включая пробелы! Без обрезки значения. Помните, что этот тип "обертки" требует символов " " в конце и начале.	
wrap2	wrap /+.splitChar	Пример: val1 val2 <i>To же, что и .wrap (смотрите порядок их применения)</i>	

Свойство:	Тип данных:	Описание:	По умолчанию:
dataWrap		<p>Содержимое раздела {...} анализируется и в содержимое {...} типа getText подставляется результат getText.</p> <p>Пример: В результате, содержимое помещается в тег с атрибутом, содержащим номер текущей страницы:</p> <pre><div id="{tsfe : id}"> </div></pre>	
prepend	cObject	cObject предваряющий содержимое (перед).	
append	cObject	cObject добавляемый к содержимому (после).	
wrap3	wrap /+.splitChar	<i>То же, что и .wrap (смотрите порядок их применения).</i>	
outerWrap	wrap /stdWrap	<i>Обертка для всего содержимого.</i>	
insertData	boolean	<p>При установке, строка содержимого анализируется наподобие .dataWrap выше.</p> <p>Пример: Отображение заголовка страницы:</p> <pre>10 = TEXT 10.value = This is the page title: {page:title} 10.insertData = 1</pre>	
offsetWrap	x,y	<p>Входное значение заключается в таблицу с колонкой слева и сверху, при этом содержимое смещается на значения x,y. В основе лежит cObject OTABLE.</p> <p>.tableParams / .tdParams /stdWrap - используется для управления смещением tableParams/tdParams (по умолчанию width=99%) . По умолчанию: смотрите OTABLE.</p> <p>.stdWrap - stdWrap свойства для обертки результата offsetWrap.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
postUserFunc	название функции	<p>Вызов PHP функции или метода в классе для анализа текущего содержимого, передаваемого в первом параметре, а любые свойства передаются в виде второго параметра. За дополнительной информацией обратитесь к описанию cObject USER.</p> <p>Пример: Можно вставить этот пример непосредственно в новую запись шаблона.</p> <pre>page = PAGE page.typeNum=0 includeLibs.something = media/scripts/example_callfunction.php page.10 = TEXT page.10 { value = Hello World postUserFunc = user_reverseString postUserFunc.uppercase = 1 } page.20 = TEXT page.20 { value = Hello World postUserFunc = user_various- >reverseString postUserFunc.uppercase = 1 postUserFunc.typolink = 11 }</pre>	
postUserFuncIn t	название функции	<p>Вызов PHP функции или метода в классе для анализа текущего содержимого, передаваемого в первом параметре, а любые свойства передаются в виде второго параметра. Результат не будет кеширован и выведен вне основного формирования страницы. За дополнительной информацией обратитесь к описанию cObject USER_INT и PHP_SCRIPT_INT.</p> <p>Предоставлено Jens Ellerbrock</p>	
prefixComment	string	<p>Предваряет содержимое HTML комментарием из второй части введенной строки (разделяется символом " "), первая часть — количество табуляций перед комментарием на новой строке. Содержимое анализируется посредством insertData.</p> <p>Пример: <code>prefixComment = 2 CONTENT ELEMENT, uid: {field:uid}/{field:CType}</code></p> <p>Отступ для комментария будет в одну табуляцию (а для следующей строки на 2+1 табуляции) (добавлено в TYPO3 >3.6.0RC1)</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
editIcons	string	<p>Если не пустое, то вставляется значок, связанный с typo3/alt_doc.php с параметрами для вставки формы редактирования некоторых полей для внутреннего пользователя.</p> <p>Значение этого свойства — список редактируемых полей из таблицы. Предполагается, что текущая запись — это запись редактируемого объекта cObj.</p> <p>Синтаксис: <i>возможно название таблицы : список полей через запятую[список названий полей формы, разделенных символом]</i></p> <p>.beforeLastTag (1,0,-1): если (1), значок будет вставлен перед последним тегом HTML в содержимом. Если -1, значок будет предварять содержимое. Если ноль (0), значок добавляется в конец содержимого.</p> <p>.styleAttribute (строка): добавляет атрибуты style к изображению значка. Например, можно указать "position:absolute", если нужна неразрушающая вставка значка. Примечание: для общей стилизации, все значки редактирования имеют класс "frontEndEditIcons", который можно использовать в таблице стилей для сайта.</p> <p>.iconTitle (строка): атрибут title тега image.</p> <p>.iconImg (HTML): альтернативный HTML код, используемый вместо отображаемого по умолчанию значка. Возможно использовать для установки другого значка для редактирования (например — красная точка или что-то еще... :-)</p> <p>Пример: вставляется значок редактирования, связанный с формой, в которой отображаются и редактируются заголовок и основной текст (если у пользователя имеется доступ!). <code>editIcons = tt_content : header, bodytext</code> так можно вставить поля header_align и date в "форму", это значит, что они отобразятся на одной строке, ниже поля header. Что сохранить немного места. <code>editIcons = header[header_align date], bodytext</code></p>	
editPanel	boolean / editPanel	Смотрите cObject EDITPANEL.	
debug	boolean	Выводит содержимое с HTMLSpecialChars() и <PRE></PRE>: полезно для отладки, для просмотра реальных значений stdWrap при разработке сайта, используя TypoScript. Должно использоваться только для разработки.	

Свойство:	Тип данных:	Описание:	По умолчанию:
debugFunc	boolean	Выводит содержимое непосредственно в браузер, используя функцию debug(). Должно использоваться только для разработки. Установка значения "2" приведет к выводу содержимого в таблицу, что лучше выглядит.	
debugData	boolean	Вывод текущего массива данных, \$cObj->data, непосредственно в браузер. Так получает данные ".field". Должно использоваться только для разработки.	

[tsref:->stdWrap]

imgResource

imgResource содержит свойства, используемые с типом данных imgResource.

Пример:

Масштабирование изображения toplogo.gif до ширины 200 пикселей.

```
file = toplogo.gif
file.width = 200
```

Свойство:	Тип данных:	Описание:	По умолчанию:
ext	imageExtension /stdWrap		web
width	pixels /stdWrap	<p>Если установлены и ширина, и высота, а к одному из значений добавлено "m", будут соблюдены пропорции при масштабировании, ввиду чего будет использовано максимальное значение для ширина/высоты изображения. Т.е. изображение будет вписано в прямоугольник с размерами ширины/высоты.</p> <p>Если установлены и ширина, и высота, а к одному из значений добавлено "c", будет произведено масштабирование с подрезкой. Это значит, что будут соблюдены пропорции, а изображение будет масштабировано до прямоугольника с размерами ширины/высоты. После чего изображение центрируется и выступающие за рамку масштабирования части изображения обрезаются. Параметр "c" может иметь добавочное процентное значение (-100 ... +100), означающее смещение рамки обрезки от центра к краю.</p> <p>Помните, что можно использовать что-то одно, "m" или "c"!</p> <p>Примеры: Масштабирование и обрезка от центра изображения до 120x80 пикс.: <pre>.width = 120c .height = 80c</pre> Обрезка до 100x100 пикс.; для альбомного изображения — слева, а для портретного — с центра: <pre>.width = 100c-100 .height = 100c</pre> Обрезка до 100x100 пикс.; для альбомного изображения — немного правее, а для портретного — немного выше центра: <pre>.width = 100c+30 .height = 100c-25</pre> </p>	
height	pixels /stdWrap	Смотрите ".width"	
params	string	!Командная строка ImageMagick: fx. "-rotate 90" или "-negate"	
sample	boolean	Если имеется, для масштабирования изображения используется -sample, вместо -geometry. Sample не использует сглаживание, и поэтому работает значительно быстрее.	

Свойство:	Тип данных:	Описание:	По умолчанию:
noScale	boolean /stdWrap	<p>Если установлено, само изображение никогда не будет масштабировано. Вычисляется лишь ширина или высота, в соответствие с другими свойствами, таким образом изображение <i>отображается</i> с измененными размерами, но используется оригинальный файл. Можно использовать для создания PDF или печати страниц, когда оригинальный файл имеет гораздо лучшее качество, чем масштабированный.</p> <p>Пример:</p> <pre>// test.jpg could e.g. have 1600 x 1200 pixels file = test.jpg file.width = 240m file.height = 240m file.noScale = 1</pre> <p>Результатом будет следующий тег image. Обратите внимание, что в src="test.jpg" используется <i>оригинальный</i> файл:</p> <pre></pre>	0
alternativeTempPath	string	Альтернативный путь для хранения временных изображений. Должен входить в список TYPO3_CONF_VARS[FE] [allowedTempPaths]	
frame	int	Выбор фрейма в анимированном gif или страницы в pdf файлах. "" = первый фрейм (ноль)	
import	path /stdWrap	<p>Значение должно быть установлено в путь файла, используя stdWrap название файла можно получить из массива данных.</p> <p>Пример:</p> <p>Это вернет первое изображение в поле "image" из массива данных:</p> <pre>.import = uploads/pics/ .import.field = image .import.listNum = 0</pre>	
maxW	pixels /stdWrap	Максимальная ширина.	
maxH	pixels /stdWrap	Максимальная высота.	
minW	pixels /stdWrap	Минимальная ширина (переназначает maxW/maxH).	
minH	pixels /stdWrap	Минимальная высота (переназначает maxW/maxH).	
stripProfile	boolean	<p>Если установлено, команда IM будет использовать команду stripProfile, сжимающую формируемый эскизы. За деталями обратитесь к Install Tool.</p> <p>Если установлено im_useStripProfileByDefault в install tool, можно отключить это, установив stripProfile=0.</p> <p>Пример:</p> <pre>10 = IMAGE 10.file = fileadmin/images/image1.jpg 10.file.stripProfile = 1</pre>	
Маскирование: (черное скрывается, белое показывается)			

Свойство:	Тип данных:	Описание:	По умолчанию:
m.mask	imgResource	Маскирование, при котором маскируется "m.bglImg". "m.mask" и "m.bglImg" масштабируются до размера изображения imgResource! ЗАМЕЧАНИЕ: "m.mask" и "m.bglImg" должны быть изображениями.	
m.bglImg	imgResource	ЗАМЕЧАНИЕ: "m.mask" и "m.bglImg" должны быть изображениями.	
m.bottomImg	imgResource	Изображение "m.bglImg" маскируется посредством "m.bottomImg_mask", перед тем, как imgResources маскируется посредством "m.mask". "m.bottomImg" и "m.bottomImg_mask" масштабируются до размера изображения imgResource! Это чаще всего используется для создания подложки для imgResource. ЗАМЕЧАНИЕ: "m.bottomImg" и "m.bottomImg_mask" должны быть изображениями.	
m.bottomImg_mask	imgResource	(дополнительно) ЗАМЕЧАНИЕ: "m.bottomImg" и "m.bottomImg_mask" должны быть изображениями.	

[tsref:->imgResource]

imageLinkWrap

Этот объект заключает входящее (изображение) в ссылку, подготовленную для вызова сценария eID "tx_cms_showpic" с параметрами, определяющими размер изображения, цвет фона нового окна и т.п.

Формируются md5-хеш параметры. Хеш также формируется и в сценарии "tx_cms_showpic", и они ДОЛЖНЫ совпадать для показываемых изображений. Это служит защитой от изменения параметров url пользователем.

Начиная с TYPO3 4.5 стало возможным отображать изображение во всплывающем окне, вместо использования showpic.php. С краткой инструкцией можно ознакомиться в свойстве "linkParams" ниже.

Свойство:	Тип данных:	Описание:	По умолчанию:
file	stdWrap	Переназначение пути отображаемого изображения.	
width	int (1-1000) /stdWrap	Добавление "m" к ширине или высоте, приведет к соблюдению пропорций, а width/height сработают для большего из измерений	
height	int (1-1000) /stdWrap	смотрите ".width"	

Свойство:	Тип данных:	Описание:	По умолчанию:
effects	Смотрите эффекты GIFBUILDER / . (из библиотеки stdgraphics) /stdWrap	Пример: gamma=1.3 sharpen=80 solarize=70	
sample	boolean /stdWrap	Если имеется, для масштабирования изображения используется -sample, вместо -geometry. Sample не использует сглаживание, и поэтому работает значительно быстрее.	
alternativeTempPath	path /stdWrap	Альтернативный путь для хранения временных изображений. Должен входить в список \$TYPO3_CONF_VARS['FE']['allowedTempPaths'].	
title	string /stdWrap	Заголовок нового окна (HTML).	
bodyTag	<tag> /stdWrap	тег body нового окна	
wrap	wrap /stdWrap	Обертка изображения, выводящаяся между тегами body	
target	<A>-data:target /stdWrap	ЗАМЕЧАНИЕ: только если установлено ".JSwindow"	
JSwindow	boolean /stdWrap	Если установлено "1", изображение будет открыто в новом окне, соответствующем по размерам изображению! Здесь можно использовать stdWrap.	
JSwindow.expand	x,y /stdWrap	х и у добавляется к размерам окна.	
JSwindow.newWindow	boolean /stdWrap	Каждая картинка будет открываться в новом окне!	
JSwindow.altUrl	string /stdWrap	Если что-либо будет возвращено, URL, показываемый в JS-окне, будет НЕ из tx_cms_showpic, а введенный здесь!	
JSwindow.altUrl_noDefaultParams	boolean	Если включено, параметры изображения не добавляются к altUrl автоматически. Полезно, если нужно создать их через пользовательскую функцию.	
typolink	->typolink	ЗАМЕЧАНИЕ: если что-либо будет возвращено, то imageLinkWrap будет переписан заданным здесь!!!	
directImageLink	boolean /stdWrap	Если включено, то будет возвращена прямая ссылка на сформированный файл изображения (это значит, что showpic.php не будет использоваться).	0

Свойство:	Тип данных:	Описание:	По умолчанию:
linkParams	->typolink	<p>Возможность управления формированием typolink, если не используется JSwindow.</p> <p>Пример:</p> <pre>JSwindow = 0 directImageLink = 1 linkParams.ATagParams.dataWrap = class="{{\$styles.content.imgtext.linkWrap.lightboxCssClass}}" rel="{{\$styles.content.imgtext.linkWrap.lightboxRelAttribute}}"</pre> <p>Используя этот параметр, запросто можно включить использование всплывающего окна для отображения полноразмерного изображения во внешнем интерфейсе: нужно лишь интегрировать всплывающее окно, включив для него файлы JS и CSS, и активировать по определенной ссылке (например для ссылок с классом "lightbox").</p>	
stdWrap	->stdWrap	Включение stdWrap для изображения.	
enable	boolean /stdWrap	Изображение имеет ссылки ТОЛЬКО если это истинно!!!	0

[tsref:->imageLinkWrap]

Пример:

```
1.imageLinkWrap = 1
1.imageLinkWrap {
    enable = 1
    bodyTag = <BODY bgColor=black>
    wrap = <A href="javascript:close();"> | </A>
    width = 800m
    height = 600

    JSwindow = 1
    JSwindow.newWindow = 1
    JSwindow.expand = 17,20
}
```

numRows

Этот объект возвращает количество строк.

Свойство:	Тип данных:	Описание:	По умолчанию:
table	Название таблицы		
select	->select	<p>Запрос на выборку для операции.</p> <p>Свойство "selectFields" внутренне переписывается на "count(*)".</p>	

[tsref:->numRows]

select

Этот объект формирует запрос SQL-select для выбора записей из базы данных.

Некоторые записи скрыты или ограничены во времени. Они автоматически добавляются к SQL-select при поиске в массиве tables.php (доступные поля).

Кроме того, если используется "pidInList", любая страница в списке pid, не видимая для пользователя сайта УДАЛЯЕТСЯ из списка pid. Поэтому записи из скрытых, ограниченных по времени или недоступных страниц не могут быть выбраны! Также, как и записи из корзин.

Свойство:	Тип данных:	Описание:	По умолчанию:
uidInList	<i>список page_id</i>		
pidInList	<i>список page_id /stdWrap</i>		this
recursive	int	Рекурсивные уровни для pidInList	0
orderBy	<i>SQL-orderBy /stdWrap</i>	Без "order by"! Например "sorting, title"	
groupBy	<i>SQL-groupBy /stdWrap</i>	Без "group by"! Например "CType"	
max	int +calc +"total"	Максимальное количество записей Специальные ключи: "total" заменяются на count(*)	
begin	int +calc +"total"	Начиная с записи под номером <i>значения</i> Специальные ключи: "total" заменяются на count(*)	
where	<i>SQL-where</i>	Без "where"!, Например " (title LIKE '%SOMETHING%' AND NOT doktype) "	
andWhere	<i>SQL-where /stdWrap</i>	Без "AND"!, Например "NOT doktype".	
languageField	string	Если установлено, это указывает на поле записи, в котором содержатся ссылка на запись в таблице sys_language. И, если установлено, записи, возвращенные функцией выбора будут выбраны только если значение этого поля соответствуют \$GLOBALS["TSFE"]->sys_language_uid (устанавливается через параметр config.sys_language_uid).	
selectFields	string	Список полей для выбора, или "count(*)". Если записи нуждаются в локализации, включите соответствующие поля локализации (uid,pid,languageField,transOrigPointerField). Иначе внутренняя локализация TYPO3 не будет соблюдена.	*
join leftjoin rightjoin	string	Введите название таблиц для JOIN , LEFT OUTER JOIN и RIGHT OUTER JOIN соответственно.	

Свойство:	Тип данных:	Описание:	По умолчанию:
markers	Массив маркеров	<p>Определенные в этом разделе маркеры можно использовать обычным образом <code>###название_маркера###</code> в любых других свойствах для <code>select</code>. Каждое значение очищается (escaped) и заключается в кавычки для предотвращения проблем с инъекциями SQL. Это дает возможность безопасного использования внешних данных (например, полей базы данных, параметров GET/POST) в запросе.</p> <p><code><markername>.value (value)</code> Прямое задание значения.</p> <p><code><markername>.commaSeparatedList (bool)</code> Если установлено, то значение расценивается как список значений через запятую. Каждое значение из списка очищается (escaped) и заключается в кавычки.</p> <p>(свойства <code>stdWrap ...</code>) Для каждого из названий маркеров можно использовать все свойства <code>stdWrap</code>.</p> <p>Пример:</p> <pre>page.60 = CONTENT page.60 { table = tt_content select { pidInList = 73 where = header != ###whatever### orderBy = ###sortfield### markers { whatever.data = GP:first sortfield.value = sor sortfield.wrap = ting } } }</pre>	

[tsref:->select]

split

Этот объект используется для разбиения введенного значения по символам и анализа их определенными функциями.

Для каждой итерации, индекс разбиения начинается с 0 (нуля), сохраняется в ключе регистра `SPLIT_COUNT`.

Пример:

Это пример кода TypoScript для импорта содержимого поля "bodytext" из `$cObj->массива` данных (строка 2). Содержимое разбивается по знаку перехода строки (строка 4). Все элементы должны быть обработаны `stdWrap` (строка 5), в которую импортируется значение элемента (строка 6). Это значение заключается в строку таблицы, в первом столбце которой содержится маркер-`gif` (строка 7). И наконец все содержимое заключается в соответствующие теги `table` (строка 9).

```

1      20 = TEXT
2      20.field = bodytext
3      20.split {
4          token.char = 10
5          cObjNum = 1

```

```

6      1.current = 1
7      1.wrap = <TR><TD valign="top"><IMG src="dot.gif"></TD><TD valign="top"> |
</TD></TR>
8      }
9      20.wrap = <TABLE border="0" cellpadding="0" cellspacing="3" width="368"> |
</TABLE><BR>

```

Свойство:	Тип данных:	Описание:	По умолчанию:
token	str /stdWrap	Строка или символ (маркер) для разделения значений.	
max	int /stdWrap	Максимальное количество частей.	
min	int /stdWrap	Минимальное количество частей.	
returnKey	int /stdWrap	Вместо анализа, из результата разбиения просто возвращается элемент с соответствующим индексом.	
cObjNum	cObjNum +optionSplit /stdWrap	Это указатель на массив объекта this ("1,2,3,4"), с элементами разбиения.	
1,2,3,4	->CARRAY /stdWrap	<p>Объект, относящийся к значению.</p> <p>ЗАМЕЧАНИЕ: значение "current" устанавливается в значение текущего элемента при вызове объекта. Смотрите "stdWrap" / current.</p> <p>Пример (stdWrap used):</p> <pre> 1.current = 1 1.wrap = </pre> <p>Пример (CARRAY used):</p> <pre> 1 { 10 = TEXT 10.current = 1 10.wrap = 20 = CLEARGIF 20.height = 20 } </pre>	
wrap	wrap +optionSplit /stdWrap	Определение обертки для каждого элемента.	

[tsref:->split]

if

Эта функция возвращает истину, если соблюдены ВСЕ присутствующие условия (они объединяются оператором И). Если хотя бы одно условие не выполняется, возвращается ложь.

Возвращаемое значение можно обратить, установив свойство ".negate".

Свойство:	Тип данных:	Описание:	По умолчанию:
isTrue	str /stdWrap	Если значение "истинно".... (не пустая строка и не ноль)	
isFalse	str /stdWrap	Если значение "ложно".... (пустое или ноль)	
isPositive	int /stdWrap + calc	Возвращается ложь, если содержимое не позитивно	

Свойство:	Тип данных:	Описание:	По умолчанию:
isGreaterThan	value /stdWrap	Возвращается ложь, если содержимое не больше ".value"	
isLessThan	value /stdWrap	Возвращается ложь, если содержимое не меньше ".value"	
equals	value /stdWrap	Возвращается ложь, если содержимое не равно ".value"	
isInList	value /stdWrap	Возвращается ложь, если содержимое не находится в разделенном запятой списке ".value". Значения в списке ".value" не должны иметь пробелов!!	
value	value /stdWrap	".value" (значение для сравнения, упоминаемое выше)	
negate	boolean	Обращает ранее возвращенное значение. То есть, если ранее была возвращена истина, то при этом будет возвращена ложь!!	
directReturn	boolean	Если это свойство существует, то будет возвращено значение истина/ложь. Можно использовать для установки истины/ложи для констант TypoScript.	

[tsref:->if]

Объяснение

Функция "if" возвращает истину / ложь не вполне ясным способом! Будьте внимательны!

"if" обычно используется для решения, выводить ли объект или возвращать ли значение (смотрите cObjects и stdWrap)

Как она работает:

Функция возвращает истину или ложь. Что она возвращает — определяется свойствами этой функции. Скажем, если установить "isTrue = 1", то результат будет истина. Если установить "isTrue.field = header", функция возвратит истину, если поле "header" в \$cObj->data заполнено!

Для сравнения значений, нужно загрузить базовое значение в свойство ".value". Пример:

```
.value = 10
.isTrue = 11
```

При этом будет возвращена истина, так как значение ".isGreaterThan" больше 10, являющегося базовым значением.

Сложнее:

```
.value = 10
.isTrue = 11
.isTrue.field = header
.negate = 1
```

Здесь два условия - isGreaterThan и isTrue. Если они оба истинны (И), в общем будет возвращена истина, НО (!) при этом общий результат будет ложь, так как установлен флаг ".negate" обращающий предыдущий результат!

Пример:

Это объект GIBUILDER пишущий "НОВЫЙ!" на элементе меню, если поле "newUntil" имеет дату, меньше текущей!

```
...
30 = TEXT
30.text = НОВЫЙ!
30.смещение = 10,10
30.if {
    value.data = date: U
    isLessThan.field = newUntil
    negate = 1
}
...
```

typolink

Превращает значение в ссылку.

Если используется из parseFunc, загружается \$cObj->массив_параметров с параметрами ссылки (в нижнем регистре)!

Свойство:	Тип данных:	Описание:	По умолчанию:
extTarget	target /stdWrap	Цель для внешней ссылки.	_top
fileTarget	target /stdWrap	Цель для ссылок на файлы.	
target	target /stdWrap	Цель для внутренних ссылок.	
no_cache	boolean /stdWrap	Добавление параметра "&no_cache=1" к ссылке.	
useCacheHash	boolean	Если включено, список additionalParams отбрасывается и преобразуется в хеш-строку, добавляемую к url, вроде "&cHash=ae83fd7s87". Когда механизм кеширования замечает это значение, он вычисляет то же значение на сервере, полагаясь на значения в HTTP_GET_VARS, исключая значения id,type,no_cache,ftu,cHash,MP. Если значение cHash соответствует вычисленному, страница может кешироваться на основе этого. [SYS][encryptionKey] включается в хеш для его уникальности для сервера и непредсказуемости.	
additionalParams	string /stdWrap	<p>Этот параметр добавляется к концу url. Должно быть кодом, готовым для вставки после последнего параметра.</p> <p>Пример: '&print=1' '&sword_list[]="word1&sword_list[]="word2'</p> <p>Applications: Очень полезно при прикреплении к страницам из результата поиска. Искомые слова сохраняются в ключе регистра SWORD_PARAMS и могут быть вставлены подобным образом: .additionalParams.data = register:SWORD_PARAMS</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		ЗАМЕЧАНИЕ: это работает только для внутренних ссылок!	
addQueryString	boolean	<p>К началу ссылки добавляется QUERY_STRING. Помните, что проверки дублирования параметров не проводится! Хотя это не проблема (будет рассмотрен лишь последний параметр с тем же названием), но если этого нужно избежать, включите "config.uniqueLinkVars".</p> <p>.method: если установлен GET или POST, будут обрабатываться запрашиваемые аргументы (данные GET или POST). Эти установки полезны, если используется обработка URL в расширениях, вроде Real URL, преобразующие часть пути в очередь аргументов.</p> <p>Возможно получить сразу и POST, и GET данные, установив "POST,GET" или "GET,POST". Последний метод в этой последовательности имеет приоритет и он перезаписывает части, присутствующие в первом методе.</p> <p>.exclude: список исключаемых из ссылки параметров запроса (например L или cHash).</p>	
jumpurl	boolean	<p>Говорит, должна ли ссылка вызывать сценарий с параметром jumpurl для регистрации каких-либо переходов в статистике.</p> <p>Работает так же, как и "filelink.jumpurl" – за деталями обратитесь к его описанию.</p> <p>Пример:</p> <pre>lib.parseFunc_RTE.tags.link { typolink.jumpurl = 1 typolink.jumpurl.secure = 1 typolink.jumpurl.secure.mimeTypes = pdf=application/pdf, doc=application/msword, png=image/png, gif=image/gif, jpg=image/jpg }</pre> <p>Такие настройки в шаблоне TS сделают любую ссылку на внутренний файл, вставленную в RTE, безопасной для скачивания файла.</p>	0
wrap	wrap /stdWrap	Обертки для ссылок.	
ATagBeforeWrap	boolean	Если включено, ссылка сначала заключается во ".wrap", а затем в тег <A>.	
parameter	string /stdWrap	<p>Это основные данные, которые используются для создания ссылки. Это может быть id страницы, URL внешней страницы, адрес e-mail или ссылка на файл на сервере. Кроме того, может даваться дополнительная информация, определяющая цель, класс и заголовок. Ниже приведено несколько примеров с пояснениями.</p> <p>Примеры:</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>parameter = 51 <i>Наиболее простой. Будет создана ссылка на страницу.</i></p> <p>parameter = 51 _blank specialLink "Very important information" <i>Полный пример. Ссылка на страницу 51, которая откроется в новом окне. Ссылка будет иметь атрибут class со значением "specialLink" и атрибут title "Very important information". В результате появится следующее:</i></p> <pre></pre> <p>parameter = http://typo3.org/ - specialLink <i>Внешняя ссылка с атрибутом class. Обратите внимание на дефис (-), заменяющий второй параметр (цель). Это дает возможность определения класса (class — третий параметр), не определяя цели.</i></p> <p>parameter = info@typo3.org - - "Send a mail to main TYPO3 contact" <i>Создание ссылки mailto с атрибутом title (но без цели и класса).</i></p> <p>Как видно из примеров, каждая значимая часть параметров отделяется пробелом. Сами значения, содержащие пробелы, должны быть помещены в двойные кавычки. Каждое из значений подробно описывается далее.</p> <p>Цель Первым параметром определяется цель для ссылки. Если присутствует знак @, то значение расценивается как адрес email и будет создана ссылка mailto. Если значение содержит точку (.) перед первым слешем (/), или двойной слеш (/ /), или внутри находится схема (вроде http), ссылка расценивается как внешняя. Если же имеется слеш, но без точки перед ним, это будет расценено как путь к файлу, и ссылка будет установлена на него (даже если он не существует, это может быть расценено, как понятный URL). В других случаях подразумевается, что значение либо id страницы, либо ее псевдоним и создается ссылка на эту страницу, если она существует.</p> <p>В случае ссылки на страницу, значение может быть более сложным, чем просто номер или псевдоним. Может быть три "значения", разделенные запятой. Пример:</p> <pre>typolink.parameter = 51,100,&test=1 - - "RSS Feed"</pre> <p>Первое значение — это id страницы, второе — ее тип, третье переназначает свойство "additionalParams". Также возможно определить section переназначающее свойство section. Если меткой section является целое число, оно расценивается как указание на запись в</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>tt_content. Если нет, то используется как есть. Если имеется только метка в section, создается ссылка на текущую страницу. Примеры:</p> <pre>typolink.parameter = 51#345 Создается ссылка на страницу 51 с якорем на элемент tt_content с номером 345</pre> <pre>typolink.parameter = #top Создается ссылка на текущую страницу, с якорем "top".</pre> <p>Также возможно настроить typolink на использование своей функции ("обработчика ссылок") для построения ссылки. Детально это описано после этой таблицы.</p> <p>Цель или настройка всплывающего окна</p> <p>Обычно цели определяются описанными выше свойствами (extTarget, fileTarget и target), но возможно переназначить их прямым указанием в свойстве parameter. Для пропуска какого-либо значения, если нужно определить, например, третью или четвертое свойства без определения цели (как в приведенном выше примере), можно использовать дефис (-). Вместо цели, второй параметр можно использовать для определения параметров всплывающего окна JavaScript, в котором будет открыта ссылка (используя window.open). Можно определить ширину и высоту окна, также как и дополнительные параметры, передаваемые в функцию JavaScript. Также смотрите свойство "JSwindow". Примеры:</p> <pre>typolink.parameter = 51 400x300 Страница 51 будет открыта во всплывающем окне с размерами 400 на 300 пикселей.</pre> <pre>typolink.parameter = 51 400x300:resizable=0,location=1 То же, что и выше, но размер окна не меняется и не будет показана область с адресом.</pre> <p>Class</p> <p>Третье значение можно использовать для определения названия класса для тега link. Этот класс вставляется в тег перед любыми другими значениями из свойства "AtagParams". Остерегайтесь конфликта атрибутов класса. Возможно применять дефис (-) для пропуска определения этого значения, для определения четвертого свойства (смотрите приведенный пример).</p> <p>Title</p> <p>Обычно атрибут title для ссылки link определяется через свойство "title" или даже через свойство "AtagParams". Но можно и установить четвертый параметр здесь, имеющий преимущество над остальными сделанными установками. Обратите внимание, что title должен быть заключен в двойные кавычки (""), если в нем имеются пробелы.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>ЗАМЕЧАНИЕ: при использовании из parseFunc, значения не нужно явно определять, а нужно импортировать через:</p> <pre>typolink.parameter.data = parameters : allParams</pre>	
forceAbsoluteUrl	boolean	<p>Принудительная установка абсолютных ссылок на внутренние страницы, таким образом соблюдается схема URL с добавлением названия домена.</p> <p>Дополнительное свойство: .scheme: определение используемой схемы URL (https или http). По умолчанию используется http.</p> <p>Пример:</p> <pre>typolink { parameter = 13 forceAbsoluteUrl = 1 forceAbsoluteUrl.scheme = https }</pre>	0
title	string /stdWrap	Установка параметра заголовка для тега A.	

Свойство:	Тип данных:	Описание:	По умолчанию:
JSwindow_params	string	Предустановка значений для открытия окна. Это пример списка с почти всеми возможными атрибутами: status=1,menubar=1,scrollbars=1,resizable=1,location=1, directories=1,toolbar=1	
returnLast	string	Если установлено "url", то будет возвращен URL ссылки (\$this->lastTypoLinkUrl). Если установлено "target", будет возвращена цель для ссылки. В этих двух случаях вы не получите значение в "обертке", а непосредственно значения url или цели!	
section	string /stdWrap	Если это значение присутствует, то к любой внутренней ссылке добавляется "#". Это полезно для создания ссылок с одной страницы непосредственно к какой-либо части другой страницы.	
ATagParams	параметры <A> /stdWrap	Дополнительные параметры Пример: class="board"	
linkAccessRestrictedPages	boolean	Если установлено, то typolink-и, указывающие на ограниченные в доступе страницы, все еще будут присутствовать, несмотря на недоступность этих страниц.	
userFunc	название функции	Данные ссылки, сформированные функцией typolink передаются в определенную пользователем функцию для окончательной обработки. Переменная \$content? передаваемая в пользовательскую функцию (первый параметр) является массивом с ключами "TYPE", "TAG", "url", "targetParams" и "ATagParams". TYPE указывает на вид ссылки: mailto, url, file, pageю TAG — полный тег <A>, сформированный и подготовленный в функции typolink. Последние три объединяются в значении "TAG" по этой формуле: <pre></pre> Пользовательская функция должна возвращать тег <A>.	

[tsref:->typolink]

Использование обработчиков ссылок

Имеется возможность (добавлена в TYPO3 4.1) зарегистрировать по ключевому слову обработчик ссылок. Например, можно сослаться на страницу с id 34 через "<link 34>", вставленное в обычное поле для основного текста, при этом теги <link> преобразуются в "->typolink". Но что, если имеется расширение "pressrelease", и нужно связать элементы пресс-релиза, выводимые расширением на какой-то странице, которую вы не помните? При помощи этой возможности возможно создать для этого расширения такую логику его работы.

В поле ссылки (значение “parameter” для `->typolink`) нужно ввести “pressrelease:123”:



Обычно некий TypoScript преобразует это значение в атрибут “parameter” при вызове `->typolink`. При вводе “pressrelease:123” в качестве “parameter” для `->typolink`, будет проверено, не является ли “pressrelease” ключевым словом, с которым связан обработчик ссылок, и если это так, то этот обработчик будет вызван для создания ссылки.

Регистрация обработчика для ключевого слова “pressrelease” делается так:

```
$TYPO3_CONF_VARS['SC_OPTIONS']['tslib/class.tslib_content.php']['typolinkLinkHandler']['pressrelease'] = 'EXT:pressrelease/class.linkHandler.php:&tx_linkHandler';
```

Файл класса “pressrelease/class.linkHandler.php” содержит класс “tx_linkHandler”, который может выглядеть так:

```
class tx_linkHandler {
    function main($linktxt, $conf, $linkHandlerKeyword, $linkHandlerValue, $link_param,
    &$p0bj) {
        $lconf = array();
        $lconf['useCacheHash'] = 1;
        $lconf['parameter'] = 34;
        $lconf['additionalParams'] =
        '&tx_pressrelease[showUid]='.$rawurlencode($linkHandlerValue);

        return $p0bj->typoLink($linktxt, $lconf);
    }
}
```

В этой функции, значение части после ключевого слова установлено как значение параметра GET, “`&tx_pressrelease[showUid]`”, а значение “parameter” новой `->typolink` устанавливается на “34”, что подразумевает, что на странице с ID 34 установлено расширение, отображающее `pressrelease` 123, при вызове через `&tx_pressrelease[showUid]=123`. Дополнительно использован атрибут “useCacheHash” для функции `typolink`, для кеширования отображаемого.

Результирующая ссылка будет выглядеть примерно так:

```
<a href="index.php?id=34&tx_pressrelease[showUid]=123%3A456&cHash=c0551fead6" >
```

Ссылка будет кодироваться через RealURL и соответствующей config.linkVars, покуда для формирования ссылок используется `->typolink`.

textStyle

Используется для стилизации текста через набор стандартных параметров + определенные специфики сайта.

Свойство:	Тип данных:	Описание:	По умолчанию:
align.field	align	Установка в поле из \$cObj->массива данных	

Свойство:	Тип данных:	Описание:	По умолчанию:
face.field	string	Установка в поле из \$cObj->массива данных [1] = "Times New Roman"; [2] = "Verdana,Arial,Helvetica,Sans serif"; [3] = "Arial,Helvetica,Sans serif";	
face.default	string /stdWrap	[default] = User defined	
size.field	string	Установка в поле из \$cObj->массива данных [1] = 1; [2] = 2; [3] = 3; [10] = "+1"; [11] = "-1";	
size.default	string /stdWrap	[default] = определенная пользователем	
color.field	string	Установка в поле из \$cObj->массива данных смотрите доступные цвета в "content.php"	
color.default	string /stdWrap	[default] = определенный пользователем	
color.1 color.2	string	[1],[2] = определенный пользователем	
properties.field	int	Установка в поле из \$cObj->массива данных Значения свойства выглядят так: bit 0: <K> bit 1: <I> bit 2: <U> bit 3: (верхний регистр) Так, значение 5 приведет к жирному, подчеркнутому тексту	
properties.default	int /stdWrap	[default] = определенный пользователем (Это значение будет использоваться, если ".field" - ложь!)	
altWrap	wrap	Если установлено это значение, обертка в тег font, с параметрами шрифта, размера и цвета НЕ делается. Скорее элемент заключается в обертку с этим значением. Используется для назначения таблиц стилей, например: <div class="text"> </div>	

[tsref:->textStyle]

encapsLines

Свойство:	Тип данных:	Описание:	По умолчанию:
encapsTagList	список строк	<p>Список тегов, расцениваемых формирующими. Должны быть в нижнем регистре.</p> <p>Пример:</p> <pre>encapsTagList = div, p</pre> <p>При такой настройке, красные строки будет распознаны в качестве формирующих:</p> <pre>Первая строка текста Какой-то <div>текст</div> <p>Какой-то текст</p> <div>Какой-то текст</div> Какой-то текст</pre>	
remapTag. [tagname]	string	<p>Введите здесь новое название для формирующего тега, если нужно привести все формирующие теги к единому названию.</p> <p>Например, установка значения "remapTags.P=DIV", приведет к преобразованию:</p> <pre><p>Какой-то текст</p> <div>Какой-то текст</div></pre> <p>в</p> <pre><div>Какой-то текст</div> <div>Какой-то текст</div></pre> <p>([tagname] в верхнем регистре).</p>	
addAttributes. [tagname]	массив строк	<p>Атрибуты для формирующих тегов.</p> <p>Пример:</p> <pre>addAttributes.P { style=padding-bottom:0px; margin-top:1px; margin-bottom:1px; align=center }</pre> <p>([tagname] в верхнем регистре).</p> <pre>.setOnly = exists : значение будет установлено, ТОЛЬКО если свойства еще имеются blank : значение будет установлено, ТОЛЬКО если свойства еще имеются ИЛИ пустое ("")</pre> <p>По умолчанию значения атрибутов всегда устанавливаются/переназначаются.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
removeWrapping	boolean	<p>Если установлено, все существующие "обертки" строк будут удалены.</p> <p>Это:</p> <pre>Первая строка текста Какой-то <div>текст</div> <p>Какой-то текст</p> <div>Какой-то текст</div> Какой-то текст</pre> <p>станет этим:</p> <pre>Первая строка текста Какой-то <div>текст</div> Какой-то текст Какой-то текст Какой-то текст</pre>	
wrapNonWrappedLines	wrap	<p>Обертка не формирующих строк</p> <p>Пример:</p> <pre>.wrapNonWrappedLines = <P> </P></pre> <p>Это:</p> <pre>Первая строка текста <p>Какой-то текст</p></pre> <p>превратиться в это:</p> <pre><P>Первая строка текста</P> <p>Какой-то текст</p></pre>	
innerStdWrap_all	->stdWrap	Обертка содержимого внутри всех строк, независимо, формирующие они или нет.	
encapsLinesStdWrap.[tagname]	->stdWrap	Обертка содержимого внутри всех формирующих строк. ([tagname] в верхнем регистре).	
defaultAlign	string /stdWrap	Если установлено, то это значение "выравнивания" используется по умолчанию для тегов .encapsTagList, .bypassEncapsTagList и .nonWrappedTag	
nonWrappedTag	tagname	Для всех не заключенных в обертку строк здесь можно установить, какой тег использовать для обертки. Примером может быть тег "P". Это альтернатива .wrapNonWrappedLines и имеет преимущество, что возможно установить атрибуты через .addAttributes и defaultAlign. Так возможно легко установить теги используемые для строк с оберткой и без обертки.	

[tsref:->encapsLines]

Пример:

```
encapsLines {
  encapsTagList = div,p
  remapTag.DIV = P
  wrapNonWrappedLines = <P>|</P>
  innerStdWrap_all.isEmpty = &nbsp;
}
```

Этот пример показывает, как можно обработать содержимое посредством TYPO3 и таблиц стилей, когда используется тег `<P>` для формирования каждой строки.

Скажем это содержимое сделано в Rich Text Editor:

```
Строка №1
[Выше имеется пустая строка! ]
<DIV align=right>Эта строка выравнивается по правому краю</DIV>
```

После обработки `encapsLines` с приведенной настройкой, содержимое будет таким:

```
<P>Строка №1</P>
<P>&nbsp;</P>
<P>[Выше имеется пустая строка! ] </P>
<P align="right">Эта строка выравнивается по правому краю</P>
```

Каждая строка заключена в тег `<P>`. Стока из базы данных, уже заключенная (но в теги `<DIV>`), была преобразована в `<P>`, но с сохранением своего выравнивания. В целом, обратите внимание, что Rich Text Editor ЛИШЬ сохранил строку, которая фактически была выравнена вправо, все остальные строки RTE сохранил без каких-либо тегов, чтобы содержимое в базе данных оставалось по возможности удобочитаемым.

Пример:

```
# Убедитесь, что операторы nonTypoTagStdWrap не содержатся вне <typolist> и <typohead>:
tt_content.text.20.parseFunc.tags.typolist.breakoutTypoTagContent = 1
tt_content.text.20.parseFunc.tags.typohead.breakoutTypoTagContent = 1
# ... и нет <BR> перед typohead.
tt_content.text.20.parseFunc.tags.typohead.stdWrap.wrap >
# Настройка nonTypoTagStdWrap на обертку текста тегами P
tt_content.text.20.parseFunc.nonTypoTagStdWrap >
tt_content.text.20.parseFunc.nonTypoTagStdWrap.encapsLines {
    encapsTagList = div,p
    remapTag.DIV = P
    wrapNonWrappedLines = <P style="margin:0 0 0;">|</P>

    # Назначение атрибутов в любые формирующие теги
    addAttributes.P {
        style=margin:0 0 0;
    }
    innerStdWrap_all.isEmpty = &nbsp;
    innerStdWrap_all.textStyle < tt_content.text.20.textStyle
}
# в конце удаляем старый стиль текста во всем основном тексте.
tt_content.text.20.textStyle >
# ... и тег <BR> больше не нужен после содержимого...
tt_content.text.20.wrap >
```

Это пример заключения обычного текста содержимого `tt_content` в теги `<P>`, установки пустой строки через тег `
`, но совместимость с функционалом RTE, вроде выравнивания и классов для параграфов, все еще остается.

tableStyle

Используется для стилизации тега `table`. Исходное значение будет вставлено в тег `table`.

Свойство:	Тип данных:	Описание:	По умолчанию:
align	align /stdWrap		
border	int /stdWrap		

Свойство:	Тип данных:	Описание:	По умолчанию:
cellspacing	int /stdWrap		
cellpadding	int /stdWrap		
color.field	string	Установка в поле из \$cObj->массива данных	
color.default color.1 color.2	string	[default],[1],[2] = определенное пользователем	
params	<TABLE>-параметры		

[tsref:->tableStyle]

Пример:

```
styles.content.tableStyle {
    align.field = text_align
    border.field = table_border
    cellspacing.field = table_cellspacing
    cellpadding = 1

    color.field = table_bgColor
    color.default = {$styles.content.tableStyle.color}
    color.1 = {$styles.content.tableStyle.color1}
    color.2 = {$styles.content.tableStyle.color2}
}
```

addParams

Свойство:	Тип данных:	Описание:	По умолчанию:
_смещение	int	Используйте, чтобы определить, каким тегом нужно управлять. 1 — первый тег из введенного значения, 2 — второй, -1 — последний, -2 — второй с конца.	1
(массив строк)	string /stdWrap	Определяет содержимое каждого добавляемого свойства тега. Если свойство тега с таким названием уже было (чувствительно к регистру!), то оно будет переназначено! Если возвращается пустая строка (но не ноль!), то существующее (если имеется) свойство не переназначается.	

[tsref:->addParams]

Пример:

```
page.13 = HTML
page.13.value = <tr><td valign=top>
page.13.value.addParams.bgcolor = {$menuCol.bgColor}
page.13.value.addParams._смещение = -1
```

Результат примера:

```
<tr><td valign="top" bgcolor="white">
```

В этом примере добавляется свойство "bgColor" к значению HTML cObject, если содержимое не "" (ноль здесь рассматривается как значение!).

filelink

Исходное — название файла по пути "path".

Значок, размер и файл обрабатываются в указанном порядке.

Свойство:	Тип данных:	Описание:	По умолчанию:
path	path /stdWrap	Пример: "uploads/media/"	
icon	boolean /stdWrap	Нужно ли отображать значок.	
icon_image_ext_list	список расширений изображений /stdWrap	Расширения файлов, которые должны выводиться в виде эскизов вместо значков.	
icon_thumbSize	string /stdWrap	<p>Определение размера значка в пикселях.</p> <p>Чтобы свойство сработало, "icon" должен быть включен, а расширение файла изображение должно входить в список "icon_image_ext_list".</p> <p>Можно установить одно из двух значений, согласно примеру. Если установить два значения, первое будет определять максимальную ширину, а второе — максимальную высоту. Пропорции оригинального изображения будут соблюдены.</p> <p>Примеры:</p> <pre>icon_thumbSize = 150 icon_thumbSize = 40x40</pre>	
iconCObject	cObject	Введите cObject для альтернативного использования для значков, например типа IMAGE. Если установлено, то это будет подставлено в сценарий эскизов для отображения эскизов.	
icon_link	boolean	Должны ли значки быть ссылками.	
labelStdWrap	->stdWrap	Параметры stdWrap для меток (по умолчанию меткой является название файла), перед тем, как они помещаются в теги A. Используйте для, например, импорта других меток из поля базы данных и т.п.	
wrap	wrap /stdWrap	Обертка для ссылок.	
ATagBeforeWrap	boolean	Если установлено, то ссылка сначала оборачивается через ".wrap", а затем помещается в тег <A>.	
file	->stdWrap	stdWrap метки (по умолчанию меткой является название файла) после помещения в тег A!	
size	boolean /stdWrap	Нужно ли показывать размер.	

Свойство:	Тип данных:	Описание:	По умолчанию:
jumpurl	boolean	<p>Должна ли ссылка вызывать сценарий с параметрами jumpurl для регистрации любых переходов в статистике. Плюсы в том, что любые щелчки по файлу будут зарегистрированы в статистике.</p> <p>Минус в том, что пользователь не сможет щелкнуть по ссылке правой кнопкой мыши и выбрать "Сохранить объект как" в браузере.</p> <p>Дополнительные свойства:</p> <p>.secure = [boolean]; если установлено, то НЕ происходит перенаправление к файлу, на который указывает jumpurl, файл читается и возвращаются корректные заголовки.</p> <p>Этот параметр добавляет хеш и locationData к url, и к записи ДОЛЖЕН иметься доступ для скачки файла. Кроме того, если местоположение файла на сервере защищено через .htaccess, то к файлу не будет НИКАКОГО доступа и получится защищенная загрузка!</p> <p>.secure.mimeTypes = список mimetypes, синтаксис [ext] = [mimetype]</p> <p>.parameter = [string/stdWrap]</p> <p>По умолчанию ссылка jumpurl использует текущий pid и typeNum.</p> <p>Если нужна альтернатива значений (например, для записи в журнал), можно определить их здесь.</p> <p>Параметры смотрите в typolink.parameter.</p> <p>Пример:</p> <pre>jumpurl.secure = 1 jumpurl.secure.mimeTypes = pdf=application/pdf, doc=application/msword</pre> <p>Пример:</p> <pre>jumpurl.secure = 1 jumpurl.secure.mimeTypes = pdf=application/pdf, doc=application/msword</pre>	
target	target		
stdWrap	->stdWrap		
ATagParams	Параметры <A> /stdWrap	<p>Дополнительные параметры.</p> <p>Пример:</p> <pre>class="board"</pre>	
removePrependNumbers	boolean	Если установлено, из имени файла удаляются любые 2 предварительно установленные цифры (например _23).	
altText titleText	string /stdWrap	<p>Для значков (изображения сформированные через "iconCObject" должны иметь свои свойства)</p> <p>Если titleText не определен, используется altText.</p> <p>Если не определен altText, используется пустой altText.</p>	
emptyTitleHandling	string /stdWrap	Значение может быть "keepEmpty" для сохранения пустого атрибута title, или "useAlt" для использования содержимого атрибута alt.	useAlt

Свойство:	Тип данных:	Описание:	По умолчанию:
longdescURL	string /stdWrap	Для значков (изображения сформированные через "iconCObject" должны иметь свои свойства) Атрибут "longdesc" (URL документа с деталями об изображении).	

[tsref:->filelink]

Пример:

```
1. filelink {
  path = uploads/media/
  icon = 1
  icon.wrap = <td> | </td>
  size = 1
  size.wrap = <td> | </td>
  file.fontTag = {$styles.content.uploads.wrap}
  file.wrap = <td> | </td>
  jumpurl = 1
  target = _blank
  stdWrap = <tr> | </tr>
}
```

numberFormat

Воспользовавшись этим свойством, можно отформатировать значения с плавающей запятой, например для указания цен. Это обертка для функции `number_format()` из PHP.

Можно определить количество знаков после запятой и разделитель десятичных знаков и тысяч.

Так как в конечной итоге свойства используются в функции PHP `number_format()`, необходимо убедиться в передаче правильных параметров в функцию. Если не уверены, наведите справки в руководстве по PHP.

Свойство:	Тип данных:	Описание:	По умолчанию:
decimals	integer /stdWrap	Количество знаков после запятой в отформатированном числе. По умолчанию 0, при этом исходное значение будет округляться вверх или вниз до целого числа.	0
dec_point	string /stdWrap	Символ-разделитель десятичных знаков. По умолчанию точка ".".	.
thousands_sep	string /stdWrap	Символ-разделитель тысяч. По умолчанию запятая ","; установите пустое значение, если разделитель тысяч не нужен.	,

[tsref:->numberFormat]

Примеры:

```
lib.myPrice = TEXT
lib.myPrice {
  value = 0.8
  numberFormat {
    decimals = 2
    dec_point.cObject = TEXT
    dec_point.cObject {
      value = .
      lang.ru = ,
    }
  }
}
```

```

        noTrimWrap = || &euro;|
    }
# По умолчанию будет выводиться "0.80 €", а для русского языка "0,80 €".

lib.carViews = CONTENT
lib.carViews {
    table = tx_mycaresh_car
    select.pidInList = 42
    renderObj = TEXT
    renderObj {
        field = views
        # По умолчанию используется 3 знака после запятой или
        # количество, заданное в переменной Get/Post precisionLevel, если задано.
        numberFormat.decimals = 3
        numberFormat.decimals.override.data = GP:precisionLevel
        numberFormat.dec_point = ,
        numberFormat.thousands_sep = .
    }
}
# Результатом будет что-то подобное "9.586,007".

```

parseFunc

Этот объект используется для анализа некоторого содержимого на предмет специальных типов тегов, в вещей типа "makeLinks" и т.п...

Пример:

В этом примере берется содержимое поля "bodytext" и анализируется при помощи функций makelinks, с подстановкой всех тегов <LINK> и <TYPOLIST> с чем-либо еще.

```

tt_content.text.default {
    20 = TEXT
    20.field = bodytext
    20.wrap = | <BR>
    20.brTag = <br>
    20.parseFunc {
        makelinks = 1
        makelinks.http.keep = path
        makelinks.http.extTarget = _blank
        makelinks.mailto.keep = path
        tags {
            link = TEXT
            link {
                current = 1
                typolink.extTarget = _blank
                typolink.target={$cLinkTagTarget}
                typolink.wrap = <B><FONT color=red>|</FONT></B>
                typolink.parameter.data = parameters : allParams
            }
        }
        typolist < tt_content.bullets.default.20
        typolist.trim = 1
        typolist.field >
        typolist.current = 1
    }
}

```

Свойство:	Тип данных:	Описание:	По умолчанию:
externalBlocks	список названий тегов/ +properties	<p>Позволяет предварительно разбить содержимое, так, чтобы в parseFunc передавалось для анализа лишь содержимое блока вне заданных тегов.</p> <p>Дополнительные свойства:</p> <p>.[tagname] {</p> <p>callRecursive = [boolean]; если задано, то содержимое блока снова направляется в parseFunc. То есть, содержимое не</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>обрабатывается ничем иным, кроме stdWrap (см. ниже).</p> <p>callRecursive.dontWrapSelf = [boolean]; если установлено, то содержимое, возвращенное из parseFunc не заключается в теги блока.</p> <p>callRecursive.alternativeWrap = альтернативные теги для помещения содержимого, вместо оригинальных тегов.</p> <p>callRecursive.tagStdWrap = ->stdWrap обработка тегов блока.</p> <p>stdWrap = ->stdWrap обработка целого блока (независимо от установки callRecursive).</p> <p>stripNLprev = [boolean]; удаление последнего перехода строки предыдущего внешнего блока.</p> <p>stripNLnext = [boolean]; удаление первого перехода строки следующего внешнего блока.</p> <p>stripNL = [boolean]: проделывает две предыдущие операции.</p> <p>HTMLtableCells = [boolean]; если установлено, содержимое проверяется на наличие таблицы и разбивается каждая ячейка.</p> <p># ниже default — все ячейки, а 1,2,3... определяют конкретные столбцы.</p> <pre>HTMLtableCells.[default/1/2/3/...] { callRecursive = [boolean]; содержимое обрабатывается текущей parseFunc. stdWrap = ->stdWrap обрабатывает содержимое ячейки. tagStdWrap = -> тег <TD> обрабатываемый ->stdWrap } HTMLtableCells.addChr10BetweenParagraphs = [boolean]; если установлено, то все вхождения </P><P> дополняются chr(10), вставленным между ними. }</pre> <p>Пример:</p> <p>Этот пример используется для разбивки обычного текста содержимого так, чтобы таблицы и цитаты обрабатывались корректно. Цитаты снова возвращаются в parseFunc (рекурсивно), где их верхние/нижние отступы устанавливаются на 0 (чтобы не были видны переводы строк).</p> <p>Также таблицы отображаются с рядом переназначенных свойств для ячеек.</p> <pre>tt_content.text.20.parseFunc.externalBlocks { blockquote.callRecursive=1 blockquote.callRecursive.tagStdWrap.HTMLparser = 1 blockquote.callRecursive.tagStdWrap.HTMLparser { tags.blockquote.fixAttrib.style.list = margin- bottom:0; margin-top:0; tags.blockquote.fixAttrib.style.always=1 } blockquote.stripNLprev=1 blockquote.stripNLnext=1 table.stripNL=1 table.stdWrap.HTMLparser = 1 table.stdWrap.HTMLparser { tags.table.overrideAttribs = border=0 cellpadding=2 cellspacing=1 style="margin-top: } }</pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<pre>10px; margin-bottom: 10px;" tags.tr.allowedAttribs=0 tags.td.overrideAttribs = valign="top" bgcolor="#eeeeee" style="font-family: Verdana, Geneva, Arial, Helvetica, sans-serif; font-size: 10px;" }</pre>	
constants	boolean	<p>Константы, определенные на основном уровне подставляются в текст. Название констант заключается в "####".</p> <p>Пример: constants.EMAIL = email@email.com (ЗАМЕЧАНИЕ: это TypoScript основного уровня!) Все вхождения строки ####EMAIL### будут замещены. Константы определены как объекты основного уровня.</p>	
short	массив строк	<p>Тоже, что выше, но на локальном уровне.</p> <p>Пример: Все встреченные "T3" заменяются на "TYPO3 CMS", а "T3web" - на ссылку на typo3.com.</p> <pre>short { T3 = TYPO3 CMS T3web = typo3 }</pre>	
plainTextStdWrap	->stdWrap	Это свойства stdWrap для всего содержимого, кроме тегов.	
userFunc	название функции	Все содержимое, кроме тегов, передается в выбранную функцию. Похоже, например, на .postUserFunc в stdWrap. Помните, что название функции должно по возможности быть с префиксом "user_".	
nonTypoTagStdWrap	->stdWrap	Наподобие .plainTextStdWrap. Разница: .plainTextStdWrap работает во ВСЕХ местах текста без тегов. .nonTypoTagStdWrap последовательно обрабатывает весь текст (включая теги) между специальными TypoTag (если .breakoutTypoTagContent не установлен для TypoTag)	
nonTypoTagUserFunc	название функции	Наподобие .userFunc. Разница в том (как и в nonTypoTagStdWrap), что производится последовательная обработка всех кусков содержимого вокруг TypoTag-ов, пока .userFunc ни обработает все содержимое вне тегов. (Примечание: .breakoutTypoTagContent должно быть установлено для TypoTag, если оно исключено из nonTypoTagContent)	
sword	wrap	<p>Отмечает любые слова из метода GET, посыпая массив sword_list[] в текст. Слово ДОЛЖНО состоять хотя бы из двух символов!</p> <p>ЗАМЕЧАНИЕ: работает лишь при \$GLOBALS["TSFE"]->no_cache==1</p>	
makelinks	boolean / ->makelinks	Преобразует адреса web, начинающиеся с "http://" и адреса почты, начинающиеся с "mailto:" в ссылки.	
tags	->tags	Здесь возможно определить свои теги , которые будут анализировать содержимое каким-либо образом.	

Свойство:	Тип данных:	Описание:	По умолчанию:
allowTags	список строк	Список тегов, которые могут присутствовать в коде! Самый высокий приоритет: если тег присутствует в allowTags, то denyTags игнорируется!!!	
denyTags	список строк	Список тегов, которые НЕ могут присутствовать в коде! (используйте "*" для тек.) Приоритет ниже: если тег НЕ находится в allowTags, то используется denyTags. Если denyTags не "*" и тега нет в списке, то тег может пропускается! Пример: Позволены только теги , <I>, <A> и .allowTags = b,i,a,img .denyTags = *	
if	->if	Если "if" возвращает ложь, то входное значение не обрабатывается, а возвращается как есть.	

[tsref:->parseFunc]

makelinks

makelinks преобразует все вхождения

<http://www.webaddress.rld>

<mailto:name@email.rld>

...в настоящие теги ссылок.

Свойство:	Тип данных:	Описание:	По умолчанию:
http.extTarget	target	Цель ссылки.	_top
http.wrap	wrap /stdWrap	Во что помещается ссылка.	
http.ATagBeforeWrap	boolean	Если включено, то ссылка сначала помещается в <i>http.wrap</i> , а затем в тег <A>.	
http.keep	list: "scheme", "path", "query"	По умолчанию, текстом ссылки будет полное имя домена ссылки. Примеры: <code>http://www.webaddress.rld/test/doc.php?id=3 "": www.webaddress.rld "scheme": http://www.webaddress.rld "scheme,path": http://www.webaddress.rld/test/doc.php "scheme,path,query": http://www.webaddress.rld/test/doc.php?id=3</code>	
http.ATagParams	параметры <A> /stdWrap	Дополнительные параметры. Пример: <code>class="board"</code>	
mailto.wrap	wrap /stdWrap	Обертка ссылки.	
mailto.ATagBeforeWrap	boolean	Если включено, то ссылка сначала помещается в <i>mailto.wrap</i> , а затем в тег <A>.	

Свойство:	Тип данных:	Описание:	По умолчанию:
mailto.ATagParams	параметры <A> /stdWrap	Дополнительные параметры. Пример: class="board"	

[tsref:->makelinks]

tags

Используется для создания собственных тегов и описания их работы. Используется вместе с *parseFunc*.

Свойство:	Тип данных:	Описание:	По умолчанию:
Массив...	cObject +stripNL + breakoutTypoTagContent	<p>Каждая запись <i>Массива...</i> относится к тегу, который будет анализироваться. Элементы ДОЛЖНЫ быть в нижнем регистре.</p> <p>Каждая запись должна соотноситься с объектом содержимого.</p> <p>"current" относится к содержимому тега, например <TAG>содержимое</TAG>: здесь "current" соотносится с "содержимое".</p> <p>Параметры:</p> <p>Параметры тега устанавливаются в \$cObj-> parameters (ключ в нижнем регистре):</p> <pre><TAG COLOR="red">содержимое</TAG> => \$cObj->parameters[color] = red</pre> <p>Специальные добавочные свойства в объекте содержимого:</p> <p>\$cObj->parameters[allParams]: при этом автоматически устанавливается целая строка параметров тега, например "color="red""</p> <p>[cObject].stripNL: булева переменная, говорящая <i>parseFunc</i>, что новая строка перед и после содержимого удаляется.</p> <p>[cObject].breakoutTypoTagContent: булева переменная, говорящая <i>parseFunc</i>, что этот блок содержимого разбивает содержимое без TypoTag, и что содержимое после этого должно быть заново "обернуто".</p> <p>Примеры:</p> <pre>tags.bold = TEXT tags.bold { current = 1 wrap = } tags.bold.stripNL = 1</pre>	

[tsref:->tags]

Пример:

В этом примере создаются 4 тега. The <LINK>-, <TYPOLIST>-, <GRAFIX>- и <PIC>.

<LINK> превращается в typolink и предоставляет простую возможность создания ссылок в тексте.

<TYPOLIST> используется для создания маркированных списков.

<GRAFIX> создает gif-файл, размером 90x10 пикселей, с текстом, содержащимся в теге.

<PIC> позволяет вставлять изображение в текст. Содержимое тега должно быть ссылкой на изображение в "fileadmin/".

```

tags {
  link = TEXT
  link {
    current = 1
    typolink.extTarget = _blank
    typolink.target={$cLinkTagTarget}
    typolink.wrap = <B><FONT color=red>|</FONT></B>
    typolink.parameter.data = parameters : allParams
  }

  typolist < tt_content.bullets.default.20
  typolist.trim = 1
  typolist.field >
  typolist.current = 1

  grafix = IMAGE
  grafix {
    file = GIFBUILDER
    file {
      XY = 90,10
      100 = TEXT
      100.text.current = 1
      100.смещение = 5,10
      100.nicetext = 1
    }
  }
  pic = IMAGE
  pic.file.import = fileadmin/
  pic.file.import.current = 1
}

```

HTMLparser

Свойство:	Тип данных:	Описание:
allowTags	list of tags	Дозволенные по умолчанию теги
tags.[tagname]	boolean/->HTMLparser_tags	Можно установить это свойство на 0 или 1, что позволит или запретить тег. Если ввести свойство ->HTMLparser_tags, автоматически выключит этот параметр и он станет ненужным. [tagname] в нижнем регистре.
localNesting	список тегов, должен входить в сохраняемые теги	Список тегов (среди уже установленных), что принудительно установит флагок nesting.
globalNesting	(ibid)	Список тегов (среди уже установленных), что принудительно установит флагок nesting на "global"
rmTagIfNoAttrib	(ibid)	Список тегов (среди уже установленных), что принудительно установит rmTagIfNoAttrib на истину.
noAttrib	(ibid)	Список тегов (среди уже установленных), что принудительно установит allowedAttribs равным нулю (то есть удалит все атрибуты).
removeTags	(ibid)	Список тегов (среди уже установленных), которые будут настроены для удаления.

Свойство:	Тип данных:	Описание:
keepNonMatchedTags	boolean / "protect"	Если включено (истина=1), то все теги, сохраненные независимо от тегов, присутствующих как ключи массива \$tags. Если "protect", то теги сохраняются с преобразованием <> в < и > По умолчанию, все недозволенные теги УДАЛЯЮТСЯ! Поэтому Вы, вероятно, захотите установить это значение!
htmlSpecialChars	-1 / 0 / 1 / 2	Это оценивает все содержимое, НЕ являющееся тегами: "0" значит "отключено" — ничего не происходит. "1" значит, что содержимое вне тегов передается в htmlspecialchars() (функцию PHP преобразующая &"<> в &...;) "2" тоже, что "1", но значения, вроде "&" или "ê" не затрагиваются. "-1" противоположное "1" - преобразует < в <, > в >, " в " etc.
xhtml_cleaning	boolean	Очищает содержимое для соответствия XHTML. Все еще экспериментальная функция и поддерживает лишь некоторые операции очистки (вроде перевода тегов и атрибутов в нижний регистр).

[page:->HTMLparser; tsref:->HTMLparser]

HTMLparser_tags

Свойство:	Тип данных:	Описание:
overrideAttribs	string	Если установлено, эта строка присутствует в качестве атрибутов тега.
allowedAttribs		'0' (ноль) = атрибутов нет, "[список атрибутов через запятую]" = дозволенные атрибуты. Если пусто/не установлено, позволены все атрибуты.
fixAttrib.[attribute].set	string	Переназначает значение атрибута на заданное.
fixAttrib.[attribute].unset	boolean	Если включено, то атрибут отключается.
fixAttrib.[attribute].default	string	Если атрибутов с таким названием нет, это значение используется по умолчанию (если оно не пустое).
fixAttrib.[attribute].always	boolean	Если установлено, то атрибут всегда обрабатывается. Обычно атрибут обрабатывается только если он существует.
fixAttrib.[attribute].trim fixAttrib.[attribute].intval fixAttrib.[attribute].upper fixAttrib.[attribute].lower	boolean	Если установлены любые из этих ключей, то значения обрабатываются соответствующей функцией PHP.
fixAttrib.[attribute].range	[low],[high]	Установка интервала (целые значения).
fixAttrib.[attribute].list	Список значений, очищенный	Значения атрибута должны находиться в этом списке. Если нет, то используется первое значение.
fixAttrib.[attribute].removeIfFalse	boolean/"blank" string	Если установлено, то "ошибочный" атрибут удаляется. Если это значение "blank", то значение может быть пустой строкой (то есть "нулевое" значение не удаляется).
fixAttrib.[attribute].removeIfEquals	string	Если значение атрибута соответствует установленному здесь, то оно удаляется.

Свойство:	Тип данных:	Описание:
fixAttrib. [attribute].casesensitiveCom p	boolean	Если установлено, то сравнение в .removeIfEquals и .list будут зависеть от регистра. Здесь не чувствительно.
fixAttrib. [attribute].prefixLocalAnch ors	integer	<p>Если первый символ - "#" (якорь тегов <a>), то значение будет использоваться как префикс относительных или локальных путей.</p> <p>Если значение - "1" то получите абсолютный путь (t3lib_div::getIndpEnv('TYPO3_REQUEST_URL'))</p> <p>Если значение - "2" то получите относительный путь (удалением t3lib_div::getIndpEnv('TYPO3_SITE_URL'))</p> <p>Пример:</p> <pre>...fixAttrib.href.prefixLocalAnchors = 1</pre>
fixAttrib. [attribute].prefixRelPathWith	string	<p>Если значение атрибута - относительный URL ("http" "/" в качестве первого символа не используются), то значение этого свойства будет служить префиксом к атрибуту.</p> <p>Пример:</p> <pre>...fixAttrib.href.prefixRelPathWith = http://192.168.230.3/typo3/32/dummy/</pre>
fixAttrib.[attribute].userFunc	function reference	<p>Пользовательская функция для обработки атрибута.</p> <p>Пример:</p> <pre>...fixAttrib.href.userFunc = tx_realurl->test_urlProc</pre>
protect	boolean	Если установлено, то знаки тегов <> преобразуются в < и >
remap	string	Если установлено, название тега переопределяется на это.
rmTagIfNoAttrib	boolean	Если установлено, то тег удаляется, если не имеет атрибутов.
nesting		<p>Если истина, то этот тег должен иметь начальные и конечные теги. Любые неверные теги будут удалены. Это "<I></I>" будет преобразовано в это: "<I></I>".</p> <p>Если это значение "глобальное", то истина для nesting относительно других тегов, назначенных для "глобального" контроля вложенности сохраняется. Это значит, что, если и <I> установлены как глобальные nesting, то строка '<I></I>' превратится в ''.</p>

[page:->HTMLparser_tags; tsref:->HTMLparser_tags]

Настройка

Top-level objects (основные объекты)

Свойство:	Тип данных:	Описание:	По умолчанию:
types	readonly	Типы (внутренние) type=99 зарезервирован для отображения простого текста.	
resources	readonly	Ресурсы в списке (внутренние).	
sitetitle	readonly	Заголовок сайта (внутренний).	
config	->CONFIG	Глобальная настройка. Эти значения сохраняются с кешируемыми страницами, это значит, что они доступны при получении кешируемых страниц.	
constants	->CONSTANTS	Константы, относящиеся к сайту, например общий адрес email. Эти константы могут быть подставлены в текст на страницах. Подстановка осуществляется через parseFunc. (Параметр: constants=1)	
FEData	->FE_DATA	Здесь можно настроить обработку данных, подставленных из внешнего интерфейса, скрипты и т.п.	
includeLibs	Массив строк	Здесь можно включить файлы php с библиотеками функций, для использования в своих включенных сценариях в TYPO3. Смотрите объект PAGE, имеющий те же свойства.	
Другие зарезервированные TLO: plugin tt_* temp styles lib _GIFBUILDER		Эти названия объектов верхнего уровня зарезервированы. Это значит, что Вы рискуете, используя их в статических шаблонах: “ plugin ” используется для формирования специального содержимого, вроде досок, решений ecommerce, гостевых книг и т.п. Обычно устанавливаемых через static_template. <i>Смотрите отдельные описания ниже!</i> “ tt_* ”, например tt_content (из “content (default)”) используется для формирования содержимого из таблиц. “ temp ” и “ styles ” используются для библиотек условий, копирующихся во время анализа, но не сохраняемых в кеше вместе с шаблоном. “temp” / “styles” убираются перед кешированием шаблона! Поэтому, эти названия используйте для хранения временных данных. “ lib ” может использоваться для “библиотеки” или кода, на который можно сослаться в TypoScript (в отличие от “ styles ” которые не сохраняются)	

Свойство:	Тип данных:	Описание:	По умолчанию:
...	->PAGE	<p>Начало новой страницы.</p> <p>Пример:</p> <pre>page = PAGE page.typeNum = 1</pre> <p>Руководство:</p> <p>Хорошо использовать общие названия объектов PAGE: <code>page</code> для основной страницы с содержимым <code>frameset, frameset2</code> для фреймов. <code>top, left, menu, right, bottom, border</code> для верхних и меню и т.п. фреймов.</p> <p>Это просто рекомендации. Особенно распространено название "page" для основной страницы с содержимым.</p>	
...	(что-нибудь)	<p>Если объект основного уровня — не объект PAGE, он может использоваться в качестве временного хранилища для настройки. В этом случае нужно использовать объекты "temp" или "styles".</p> <p>"tt_..." обычно используется для определения настроек записей содержимого. Например, "tt_content" используется по умолчанию для таблицы tt_content. Смотрите cObject "CONTENT".</p>	

[tsref:(TLO)]

"дополнение" TLO

Используется в TYPO3 для расширений, установленных как дополнения внешнего интерфейса. Обычно здесь можно установить настройки для расширений. Скажем, имеется расширение с ключом "tx_myext", а в нем имеется внешнее дополнение с названием "tx_myext_pi1", тогда его настройки через TypoScript можно найти в дереве объектов в разделе "plugin.tx_myext_pi1"!

Большинство расширений — это объекты USER или USER_INT, то есть они имеют хотя бы 1 или 2 свойства. Кроме того, приведенная таблица обрисовывает некоторые свойства по умолчанию. Обычно, системные свойства содержат префикс с подчеркиванием:

Свойство:	Тип данных:	Описание:	По умолчанию:
<code>userFunc</code>		<i>Свойства для настроек объекта USER / USER_INT расширения</i>	
<code>includeLibs</code>		<i>Свойства для настроек объекта USER / USER_INT расширения</i>	
<code>_CSS_DEFAULT_STYLE</code>	string	Используйте для создания некоторых CSS по умолчанию, вставленных в раздел заголовка документа. Скорее всего, это будет представлять наиболее приемлемое отображение для расширения, но в идеале, оно должно быть перемещено во внешнюю таблицу стилей. Это значение для всех расширений читается сценарием pagegen при формировании заголовка документа.	
<code>_DEFAULT_PI_VARS.</code> [<code>piVar-key</code>]	string	Позволяет установить значения по умолчанию для массива piVars, использующихся (и должны использоваться) в большинстве расширений для обмена данными между собой. Работает только при вызове расширения через <code>\$this->pi_setPiVarDefaults()</code> .	
<code>_LOCAL_LANG.[lang-key].[label-key]</code>	string	Может использоваться для переназначения меток локализации расширения по умолчанию.	

[tsref:plugin]

"CONFIG"

В typo3/sysext/cms/tslib/ это известно как `$GLOBALS["TSFE"]->config["config"]`, ввиду чего, свойство "debug" (отладка) доступно как `$GLOBALS["TSFE"]->config["config"]["debug"]`.

Свойство:	Тип данных:	Описание:	По умолчанию:
defaultGetVars	array	<p>Позволяет установить значения по умолчанию для параметров GET. Значения по умолчанию берутся только если параметры GET не определены.</p> <p>Обозначения в массиве делаются точками, например: <code>test[var1]</code> будет записано как <code>test.var1</code></p> <p>Пример:</p> <pre>config.defaultGetVars { test.var1.var2.p3 = 15 L = 3 }</pre>	
linkVars	list	<p>HTTP_GET_VARS, что должно передаваться вместе с ссылками в TYPO3. Это компилируется в строку, хранящуюся в <code>\$GLOBALS["TSFE"]->linkVars</code></p> <p>Значения rawurlencode в PHP.</p> <p>Можно определить диапазон правильных значений, добавив () после каждого значения. Если диапазон не верный, переменная не добавляется к ссылкам. Это очень важно для предотвращения забивания системы кеширования подделанными значениями.</p> <p>Диапазон может содержать одно из следующих значений:</p> <ul style="list-style-type: none"> • [a]-[b] – диапазон целых значений; • int – дозволены лишь целые значения; • [a][b][c] – список допустимых строк (пробелы удаляются); • /[regex]/ – соответствие регулярному выражению (стиль PCRE) <p>Пример:</p> <pre>config.linkVars = L, print</pre> <p>Все ссылки в TYPO3 будут дополнены "&L=[значение L]&print=[значение print]."</p> <pre>config.linkVars = L(1-3), print</pre> <p>Тоже, что выше, но "&L=[значение L]" будет добавлено только если текущее значение равно 1, 2 или 3.</p> <p>Замечание: не включайте параметр "type" в список linkVars, так как это приведет к неожиданным результатам.</p>	
uniqueLinkVars	boolean	Может получиться, что TYPO3 формирует ссылки с одинаковыми параметрами дважды и более раз. Это не проблема, так как используются лишь последние параметры, это нужно лишь для опрятности.	1

Свойство:	Тип данных:	Описание:	По умолчанию:
MP_defaults	string	<p>Позволяет установить список номеров id страниц, к которым всегда добавляется соответствующий "&MP=..." параметр.</p> <p>Синтаксис: [id],[id],... : [MP-var] [id],[id],... : [MP-var] ...</p> <p>Пример: <code>config.MP_defaults = 36,37,48 : 2-207</code></p> <p>Это добавляет по умолчанию "&MP=2-207" ко всем ссылкам, указывающим на страницы 36,37 и 48</p>	
MP_mapRootPoints	список PID/string	<p>Определяет список номеров ID, начиная с которого в ветви автоматически вычисляются переменные MP. Результат используется также, как и MP_defaults, для поиска переменных MP, если ни один не был указан до вызова в t3lib_tstemplate::linkData().</p> <p>Возможно определить "root" в качестве специального ключа в списке ID, при этом будет создано карта-дерево для целого сайта (но это ОЧЕНЬ сильно сказывается на производительности при обработке большого количества страниц!).</p> <p>Порядок указанных ID может иметь значение; любые, уже обработанные ID в ветви (указанием ID предыдущей корневой точки) снова не обрабатываются.</p>	
MP_disableTypolinkClosesMPvalue	boolean	Если установлено, функция typolink не будет пытаться найти ближайшее значение MP для id.	
renderCharset	string	<p>Кодировка, используемая для формирования внутреннего содержимого страниц. Крайне рекомендуется, чтобы это значение было таким же, как и кодировка содержимого в основном источнике данных (например в базе данных). Ввиду чего не придется делать никаких преобразований.</p> <p>Все строки из locallang файлов и локальных строк преобразуются в "renderCharset" в процессе формирования.</p> <p>Если для вывода нужна отличная от render charset кодировка, смотрите ниже - "metaCharset".</p> <p>Если установлено значение TYPO3_CONF_VARS['BE']['forceCharset'], то оно по умолчанию используется для "renderCharset". Для многоязычных сайтов крайне рекомендуется использование TYPO3_CONF_VARS['BE']['forceCharset'] в TYPO3. Установив его, не нужно будет заботиться о renderCharset и metaCharset – кодировка во всей системе будет одна и та же.</p>	\$TYPO3_CONF_VARS['BE']['forceCharset'] если что-то отличное от "iso-8859-1"

Свойство:	Тип данных:	Описание:	По умолчанию:
metaCharset	string	<p>Кодировка, используемая для вывода документа. Например, в мета теге:</p> <pre><meta http-equiv="Content-Type" content="text/html; charset=...></pre> <p>Используется для а) HTML тега meta, б) HTTP заголовка (если не заблокировано через .disableCharsetHeader) и с) xhtml prologues (если доступно)</p> <p>Если renderCharset и metaCharset отличаются, выводимое содержимое автоматически преобразуется в metaCharset перед выводом и наоборот, значения, отсылаемые из страницы, преобразуются из metaCharset в renderCharset для внутренней обработки. Конечно же, это преобразование занимает некоторое время, ввиду чего лучше везде использовать одинаковую кодировку.</p>	значение ".renderCharset"
disableCharsetHeader	boolean	По умолчанию отсылается заголовок "content-type:text/html; charset...". Этот параметр отключает его отсылку.	

Свойство:	Тип данных:	Описание:	По умолчанию:
sendCacheHeaders	boolean	<p>Если установлено, TYPO3 выведет заголовки управления кешем для клиента, полагаясь на внутреннее кеширование страницы. Это возможность позволяет браузерам клиентов и/или прокси серверам управлять загрузкой сайтов через TYPO3.</p> <p>Условия для позволения клиентского кеширования:</p> <ul style="list-style-type: none">● страница была в кеше● На странице нет *_INT или *_EXT объектов (например USER_INT)● Не зарегистрирован внешний пользователь● Не зарегистрирован внутренний пользователь <p>Если эти условия выполняются, высылаются заголовки:</p> <ul style="list-style-type: none">● С последними изменениями [SYS_LASTCHANGED для id страницы]● Истекшие [время, истекшее со времени кеширования страницы]● Etag [md5 содержимого]● Cache-Control: max-age: [секунды со времени expiretime]● Pragma: public <p>Если кеширование запрещено, отправляются такие заголовки, для запрещения кеширования:</p> <ul style="list-style-type: none">● Cache-Control: private <p>Обратите внимание, что включение кеширования браузером, значит согласование в написании файлов журналов. Так как, при кешировании страницы у клиента, она не будет запрашивать сервер, поэтому запрос не будет зафиксирован в журнале. Поэтому должны быть способы для обхода этой проблемы, но они в любом случае находятся вне компетенции TYPO3.</p> <p>Совет: включение заголовков управления кешем может запутать редакторов сайтов, видящих устаревшее содержимое, взятое из кеша браузера. "Shift-перезагрузка" решает как проблему кеширования браузера, так и прокси, и даже может заставить TYPO3 заново создать содержимое страницы. Запомните эту уловку!</p> <p>Благодарим Ole Tange, www.forbrug.dk за соавторство в этой возможности.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
sendCacheHeaders_onlyWhenLoginDeniedInBranch	boolean	<p>Если включено, то заголовки управления кешем позволяют клиентское кеширование, только если регистрации пользователей отключены для ветви. Эта функция облегчает управление клиентским кешированием на сайтах со смесью статических страниц и динамических разделов для зарегистрированных пользователей.</p> <p>Подоплека проблемы вот в чем: в TYPO3 тот же URL может отобразить различное содержимое, в зависимости от того, зарегистрирован ли пользователь, или нет. Если пользователь зарегистрирован, заголовки кеширования никогда не позволяют клиентское кеширование. Но если тот же URL посещается до регистрации (кеширование разрешено), то пользователь после регистрации все еще будет видеть страницу из кеша до регистрации (и будет думать, что он не зарегистрировался)! Существует лишь один способ для предотвращения такой ситуации - различные URL для страниц, когда пользователь зарегистрирован в системе (что может сделать расширение "realurl").</p> <p>Другой путь разрешения этой проблемы — использование этого параметра в комбинации с разрешением и запретом регистрации в различных частях сайта. В записи страниц (вкладка "доступ", параметр "режим входа в систему") можно отключить регистрацию внешних пользователей для ветви дерева страниц. Так как регистрация на многих сайтах требуется лишь для некоторых страниц в дереве, отключение возможности регистрации на остальных страницах упростит использование заголовков управления кешем в комбинации с регистрацией; заголовки для кеша будут посыпаться, когда регистрация пользователей отключена и ни в коем случае не посыпаются при дозволенной регистрации! При этом ситуация с одинаковым URL при регистрации возникать просто не будет.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
additionalHeaders	Строки, разделенные знаком " "	<p>Свойство можно использовать для указания расширенных заголовков HTTP. Каждый заголовок отделяется вертикальной чертой " ".</p> <p>Примеры:</p> <p>Content-type: text/vnd.wap.wml (будет отослан заголовок содержимого для WAP-сайта).</p> <p>Content-type: image/gif Expires: Mon, 25 Jul 2011 05:00:00 GMT (будет отослан заголовок содержимого для файла GIF и заголовок Expires).</p> <p>Location: www.typo3.com (перенаправление страницы на www.typo3.com)</p> <p>По умолчанию TYPO3 отсылает заголовки "Content-Type" в кодировке по умолчанию, если не отключено через config.disableCharsetHeader (см. выше). Затем отсылаются кешированные заголовки, если настроено (см. выше). Затем настает очередь расширенных заголовков, плюс финальный заголовок "Content-Length", если включено (см. ниже).</p>	
enableContentLengthHeader	boolean	<p>Если установлено, посыпается заголовок "content-length: [bytes of content]".</p> <p>Если на странице обнаружен объект PHP_SCRIPT_EXT, или если зарегистрирован внутренний пользователь, этот параметр отключается. Причина в том, что величина содержимого этих объектов не может быть включена в content-length, а некоторые браузеры обрезают документ до обозначенной длины.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
doctype	string	<p>Если установлено, то будет сформировано объявление типа документа (и вводная часть XML). Значение может быть либо полным doctype или одним из следующих ключевых слов:</p> <p>"xhtml_trans" для XHTML 1.0 переходный тип документа. "**xhtml_frames**" для XHTML 1.0 тип документа с набором фреймов. "**xhtml_strict**" для XHTML 1.0 строгий тип документа doctype. "**xhtml_basic**" для XHTML основной тип документа. "**xhtml_11**" для XHTML 1.1 типа документа. "**xhtml+rdfa_10**" для XHTML+RDFa 1.0 типа документа. "**xhtml_2**" для XHTML 2 типа документа. "**html5**" для HTML5 типа документа. "**none**" БЕЗ типа документа вообще.</p> <p>Замечание: В TYPO3 4.4 ключевое слово для HTML5 – "html_5". Это написание устарело начиная с TYPO3 4.5.</p> <p>Помните, что ключевые слова заставляют TYPO3 изменить способ формирования некоторых тегов XHTML для соответствия стандартам XML. Если установлен строгий тип документа, то нужно также установить config.xhtmlDoctype (см. далее).</p> <p>За детальной информацией по эффектам в теге html обратитесь к "config.htmlTag_setParams" и "config.htmlTag_langKey".</p> <p>По умолчанию используется переходный HTML 4 тип документа:</p> <pre><!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"></pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
doctypeSwitch	boolean / string	<p>Если установлено, порядок <?xml...> и <!DOCTYPE...> будет изменен. Это необходимо для MSIE для соответствия стандартам XHTML.</p> <p>Что происходит: По умолчанию TYPO3 выводит XML/DOCTYPE в соответствии со стандартами XHTML. Тем не менее, браузеры, вроде MSIE, будут работать в режиме совместимости, если теги <?xml> и <DOCTYPE> расположены в обратном порядке. И это прервет проверку правильности CSS... Посредством этого параметра, дизайнеры могут определиться в своих желаниях.</p> <p>Если нужно проверить режим совместимости своего браузера, то сделать это можно при помощи простого JavaScript, который может быть вставлен на страницу TYPO3, например так:</p> <pre>page.headerData.1 = TEXT page.headerData.1.value = <script>alert(document.compatMode);</script></pre> <p>Если браузер правильно определит DOCTYPE, то отобразится "CSS1Compat". Если же режим совместимости не будет соблюдаться, будут получены другие сообщения. Например MSIE выведет "BackCompat" - это значит, что он работает в режиме совместимости, имитируя старые движки.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
xhtmlDoctype	string	<p>Установка типа документа формируемых страниц для версии XHTML.</p> <p>Если установлена строка configdoctype, то config.xhtmlDoctype должен быть установлен на одно из этих ключевых слов:</p> <p>"xhtml_trans" для XHTML 1.0 переходный тип документа.</p> <p>"xhtml_frames" для XHTML 1.0 тип документа с набором фреймов.</p> <p>"xhtml_strict" для XHTML 1.0 строгий тип документа doctype.</p> <p>"xhtml_basic" для XHTML основной тип документа.</p> <p>"xhtml_11" для XHTML 1.1 типа документа.</p> <p>"xhtml_2" для XHTML 2 типа документа.</p> <p>Это пример использования MathML 2.0 в документе XHTML 1.1:</p> <pre>configdoctype (<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN" "http://www.w3.org/Math/DTD/mathml2/xhtml- math11-f.dtd">) config.xhtmlDoctype = xhtml_11</pre> <p>По умолчанию: то же, что и configdoctype, если используется ключевое слово.</p>	
xmlprologue	string	<p>Если пусто (не установлено), то по умолчанию используется вводная часть XML 1.0, если тип документа установлен на известные ключевые слова (например xhtml_11):</p> <pre><?xml version="1.0" encoding="[config.renderCharset]"></pre> <p>Если установлено на одно из ключевых слов, то будет использована стандартная вводная часть:</p> <p>"xml_10" XML 1.0 вводная часть (выше)</p> <p>"xml_11" XML 1.1 вводная часть</p> <p>Если установлено "none", то по умолчанию вводная часть XML не используется.</p> <p>Любые другие строки используются сами по себе в качестве вводной части XML.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
htmlTag_setParams	string	<p>Установка атрибутов для тега <html> страницы. Если "config.doctype" установлен на ключевые слова с XHTML, то некоторые атрибуты уже установлены. Это свойство позволяет переназначить любые предустановленные свойства на необходимые.</p> <p>Особенность: если установить "none", то не атрибуты не будут установлены вообще.</p> <p>Пример: <code>config.htmlTag_setParams = xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-US"</code></p>	
namespaces	массив строк	<p>Это свойство дает возможность добавить область имен xml (namespaces, xmlns) в тег <html>. Это особенно полезно при добавлении RDFa или микроформатов в свой html.</p> <p>Пример: <code>config.namespaces.dc = http://purl.org/dc/elements/1.1/ config.namespaces.foaf = http://xmlns.com/foaf/0.1/</code></p> <p>Результатом такой настройки будет тег <html> вроде:</p> <pre><html xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:foaf="http://xmlns.com/foaf/0.1/"></pre>	
htmlTag_langKey	string	<p>Позволяет установить значение языка для атрибутов "xml:lang" и "lang" в теге <html> (при использовании "config.doctype = xhtml*").</p> <p>Значение должно соответствовать формату, определенному в IETF RFC 3066</p> <p>Пример: <code>config.htmlTag_langKey = en-US</code></p>	en
htmlTag_dir	string	<p>Установка направления написания для всего документа (полезно для отображения арабского, еврейского языка).</p> <p>В основном используется как значение атрибута "dir" тега <html>.</p> <p>Значения: rtl = с права налево (арабский / еврейский) ltr = с лева направо (по умолчанию для всех языков)</p> <p>Пример: <code>config.htmlTag_dir = rtl</code></p>	
disableImgBorderAttr	boolean	Возвращает атрибут 'border' для тега , только если doctype не xhtml_strict, xhtml_11 или xhtml_2, либо если не установлен параметр настроек 'disableImgBorderAttr'.	

Свойство:	Тип данных:	Описание:	По умолчанию:
ATagParams	<i>параметры <A></i>	<p>Дополнительные параметры для всех ссылок в TYPO3 (за исключением ссылок в меню)</p> <p>Пример: Для размытия ссылок, вставьте: <code>onFocus="blurLink(this)"</code></p>	
setJS_openPic	boolean	Если включено, принудительно включается JavaScript функция openPic/	
setJS_mouseOver	boolean	Если включено, принудительно включаются JavaScript функции over() и out().	
removeDefaultJS	boolean / string	<p>Если установлено, JavaScript в заголовке по умолчанию будет удален. JavaScript по умолчанию — это функция blurLink и переменные определения браузера.</p> <p>Особый случай: если используется значение "external", то JavaScript по умолчанию записывается во временный файл и включается из этого файла. Смотрите "" далее.</p> <p>Зависит от режима совместимости (см. Сервис управления>Установка>Update wizard): <i>compatibility mode < 4.0: 0</i> <i>compatibility mode >= 4.0: 1</i></p> <p>Примеры: <code>config.removeDefaultJS = external</code> <code>config.removeDefaultJS = 1</code></p>	
minifyJS	boolean	<p>Если установлено inline или externalized (см. removeDefaultJS выше) JavaScript будет минимизирован. Минификация заключается в удалении всех пробелов, при этом страницы загружаются быстрее. Вместе с removeDefaultJS = external это должно сократить объем сайта.</p> <p>В зависимости от режима совместимости (см. Сервис управления>Установка>Update wizard): <i>compatibility mode < 4.0: 0</i> <i>compatibility mode >= 4.0: 1</i></p> <p>Пример: <code>config.minifyJS = 1</code></p> <p>Замечание: JavaScript во внешних файлах внешнего интерфейса будет минимизирован только если обработчик компрессии зарегистрирован через <code>\$GLOBALS['TYPO3_CONF_VARS']['FE']['jsCompressHandler']</code>.</p> <p>Пример: <code>\$GLOBALS['TYPO3_CONF_VARS']['FE']['jsCompressHandler'] =</code> <code>t3lib_extMgm::extPath(\$_EXTKEY) .</code> <code>'Classes/class.tx_myext_jsCompressHandler.php</code> <code>:tx_myext_jsCompressHandler->minifyJs';</code></p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
minifyCSS	boolean	<p>Параметр включает минимизацию CSS.</p> <p>Пример: <code>config.minifyCSS = 1</code></p> <p>Замечание: CSS во внешних файлах внешнего интерфейса будет минимизирован только если обработчик компрессии зарегистрирован через <code>\$GLOBALS['TYPO3_CONF_VARS']['FE']['cssCompressHandler']</code>.</p> <p>Пример: <code>\$GLOBALS['TYPO3_CONF_VARS']['FE']['cssCompressHandler'] = t3lib_extMgm::extPath(\$_EXTKEY) . 'Classes/class.tx_myext_cssCompressHandler.php:tx_myext_cssCompressHandler->minifyCss';</code></p>	
concatenateJsAndCss	boolean	<p>Настройка config.concatenateJsAndCss объединяет JS и CSS файлы внешнего интерфейса.</p> <p>Пример: <code>config.concatenateJsAndCss = 1</code></p> <p>Замечание: нет обработчика объединения по умолчанию. Можно указать собственный обработчик, зарегистрировав его через <code>\$GLOBALS['TYPO3_CONF_VARS']['FE']['concatenateHandler']</code>.</p> <p>Пример: <code>\$GLOBALS['TYPO3_CONF_VARS']['FE']['concatenateHandler'] = t3lib_extMgm::extPath(\$_EXTKEY) . 'Classes/class.tx_myext_concatenateHandler.php:tx_myext_concatenateHandler->concatenateFiles';</code></p>	0
inlineStyle2TempFile	boolean	<p>Если установлено, встроенные стили управления ядра TYPO3 записываются в файл, <code>typo3temp/stylesheet_[hashstring].css</code>, а заголовок содержит лишь ссылку на таблицу стилей.</p> <p>Хеш файла зависит исключительно от содержимого стилей.</p> <p>Зависит от режима совместимости (см. Сервис управления>Установка>Update wizard): <code>compatibility mode < 4.0: 0</code> <code>compatibility mode >= 4.0: 1</code></p> <p>Пример: <code>config.inlineStyle2TempFile = 1</code></p>	
meaningfulTempFilePrefix	integer	<p>Если > 0, то TYPO3 попытается создать значащий префикс заданной длины для временных файлов изображений.</p> <p>Работает с файлами GIFBUILDER (используя содержимое из объектов GIFBUILDER TEXT в качестве основы для префикса), меню (берется название элемента меню) и масштабированными изображениями (на основе оригинального названия).</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
ftu	boolean	<p>Если установлено, вставляется "&ftu=...." идентификатор GET-восстановления. "&ftu=[hash]" всегда вставляется в ссылки первой посещенной пользователем страницы. Если при следующем обращении оказывается, что пользователем разрешены cookies, эта переменная более не устанавливается, вместо этого срабатывают cookies. Если cookies не установлены, "ftu" остается установленной для всех ссылок сайта, поэтому возможно отслеживать перемещения пользователя.</p> <p>Этот параметр не должен быть установлен, если на Ваш сайт заходят пауки, вроде Teleport! Не нужно устанавливать этот параметр, если нужно, чтобы поисковые движки проиндексировали Ваш сайт (вместе с параметром simulateStaticDocuments!)</p> <p>Также можно пропустить эту возможность, если уверены, что пользователи сайта будут использовать cookies. "ftu" означает fe_type_user ("fe" это "внешний интерфейс").</p>	false
mainScript	string	Это позволяет определить альтернативный "mainScript", т.е. документ ожидаемый TYPO3, как документ по умолчанию. Это используется в тегах form и других местах, где TYPO3 должен ссылаться непосредственно на основной сценарий приложения.	index.php
pageGenScript	resource	<p>Альтернативный сценарий формирования страниц для приложений, использующих index_ts.php для инициализации, кеширования, статистики и т.п. Этот сценарий включается в глобальное пространство index_ts.php-script, поэтому сюда возможно включать библиотеки. Для библиотек всегда используйте include_once.</p> <p>Помните, что ничего не должно выводиться из такого включенного сценария. Все содержимое должно устанавливаться в \$TSFE->content. Взгляните на typo3/sysext/cms/tslib/pagegen.php</p> <p>ЗАМЕЧАНИЕ: этот параметр игнорируется, если в localconf.php установлено</p> <pre>\$TYPO3_CONF_VARS['FE']['noPHPscriptInclude'] = 1;</pre>	typo3/sysext/cms/tslib/pagegen.php
debug	boolean	Если установлено, то выводиться отладочная информация в коде TypoScript. На данный момент применяется лишь для объектов меню.	

Свойство:	Тип данных:	Описание:	По умолчанию:
message_page_is_being_generated	string	<p>Альтернативное HTML сообщение, появляющееся при формировании страницы.</p> <p>Обычно, после формирования страницы, ее временная копия сохраняется в буферной таблице на 30 секунд.</p> <p>Возможно использовать ключевые слова, заменяющиеся соответствующими значениями.</p> <p>Возможные ключевые слова: ###TITLE###, ###REQUEST_URI###</p>	
message_preview	string	Альтернативное сообщение в HTML, появляющееся при активации функции предпросмотра!	
message_preview_workspace	string	<p>Альтернативное сообщение в HTML, появляющееся при активации функции предпросмотра в черновом рабочем пространстве. Можно использовать sprintf() метку для названия пространства (первое) и номер (второе).</p> <p>Примеры:</p> <pre>config.message_preview_workspace = <div class="previewbox"> Выводиться пространство с названием "%s" (number %s)!</div> config.message_preview_workspace = <div class="previewbox"> Номер пространства %2\$s название "%1\$s"!</div></pre>	
disablePreviewNotification	boolean	Полное отключение блока сообщений о предварительном просмотре.	0
locale_all	string	<p>PHP: setlocale("LC_ALL", [значение]);</p> <p>примеры значений: deutsch, de_DE, danish, portuguese, spanish, french, norwegian, italian. За остальными значениями обратитесь на www.php.net. То же в linux, ищите в at /usr/share/locale/.</p> <p>TSFE->localeCharset разумно установить на строку кодировки для соответствующего региона. Это используется в stdWrap.strftime для преобразования локальных строк в renderCharset внешнего интерфейса.</p> <p>Пример:</p> <p>Это сформирует данные из stdWrap/strftime так, как принято в Дании:</p> <pre>locale_all = danish locale_all = da_DK</pre>	
sword_standalone	boolean	<p>Используется в parseFunc-подстановки искомых слов (sword):</p> <p>Если установлено, слова ДОЛЖНЫ быть окружены пробелами для разметки.</p>	
sword_noMixedCase	boolean	<p>Используется в parseFunc-подстановки искомых слов (sword):</p> <p>Если установлено, слова ДОЛЖНЫ быть точно в том же регистре, как и искомое слово.</p>	
intTarget	target	Внутренняя цель по умолчанию. Используется в typolink, если цель не установлена.	

Свойство:	Тип данных:	Описание:	По умолчанию:
extTarget	target	Внешняя цель по умолчанию. Используется в typolink, если extTarget не установлена.	_top
fileTarget	target	Цель для ссылок на файлы по умолчанию. Используется в typolink, если fileTarget не установлена.	
spamProtectEmailAddresses	"ascii" / -10 to 10	<p>Если установлено, то адреса email в typolink будут зашифрованы от спама, и боты не смогут их определить.</p> <p>Если установить число в качестве значения, то шифрование осуществляется простым смещением кодов символов. Если установить значение, как "-2", то все символы изменят свой ASCII код на "-2". Для осуществления этого, к каждой странице добавляется маленький JavaScript код!</p> <p>(Рекомендуется устанавливать это значение в диапазоне от -5 до 1, так как установка значения ≥ 2 означает преобразование "z" в " ", которое является специальным символом в синтаксисе TYPO3 и может привести к ошибкам)</p> <p>Альтернативно, возможно установить как ключ "ascii". При этом каждый символ в адресе "mailto:" будет переведен в Unicode HTML</p> <p>примечание. Взгляните на пример, чтобы понять, как это работает.</p> <p>Пример: <code>mailto:a@b.c</code> будет преобразовано в <code>mailto:&#97;&#64;&#46;&#99;</code></p> <p>Большое преимущество этого метода то, что не требуется JavaScript!</p>	
spamProtectEmailAddresses_atSubst	string	Метка, подставляемая вместо символа (@).	(at)
spamProtectEmailAddresses_lastDotSubst	string	Метка, подставляемая вместо последней точки email адресах. Пример: (dot)	По умолчанию: . (<= обычная точка)
forceTypeValue	int	Принудительное значение &type для всех ссылок, формируемых TYPO3 (за исключением тех, которые локально переназначены значениями forceTypeValue). Полезно при использовании шаблона с определенным содержимым, скажем &type=95, но при этом цель не должна быть затронута. При этом цели остаются пустыми, а значение типа вводится по желанию.	

Свойство:	Тип данных:	Описание:	По умолчанию:
frameReloadIfNotInFrameset	boolean	Если установлено, то текущая страница проверяется, не существует ли для названия объекта страницы (например "page" или "frameset") родительского объекта — "родитель.[название]" (например "parent.page"), и, если нет, страница загружается в top frame. Это страховка от того, чтобы ссылки из поисковых машин, указывающие на содержимое фреймов, загружали весь набор фреймов. Работает лишь со значениями type отличных от нуля.	
jumpurl_enable	boolean	JumpUrl — понятие, когда внешние ссылки перенаправляются из сценария index_ts.php, который сначала регистрирует URL. Эта возможность интересна лишь при использовании "config.sys_stat".	0
jumpurl_mailto_disable	boolean	Отключение использования jumpUrl для ссылок на email-адреса.	0
compensateFieldWidth	double	<p>Это значение в виде числа с плавающей точкой используется в FORMS cObject для компенсации длины текстового поля формы и полей ввода. Параметр полезен, если установлен параметр для страницы "smallFormFields". В этом случае Netscape формирует немного более длинные поля для форм, чем IE. Если нужно, чтобы оба браузера отображали поля форм одинакового размера, используйте значение, примерно "0.6" для браузера netscape.</p> <p>Пример:</p> <pre>[browser = netscape] config.compensateFieldWidth = 0.6 [global]</pre> <p>Этот параметр может быть переназначен в FORMS-cObject.</p>	
includeLibrary	resource	Включение файла PHP.	
incT3Lib_htmlmail	boolean	Включение t3lib/class.t3lib_htmlmail.php.	
lockFilePath	string	Используется для блокировки путей "внутри" данного пути. Используется в "filelist" из stdWrap	fileadmin/
noScaleUp	boolean	Обычно изображения масштабируются до размера, определенного через TypoScript. При этом маленькие изображения принудительно увеличиваются до большего размера, что не всегда хорошо. Если установлено это свойство, изображения не будут увеличиваться. Этот параметр очищает переменную \$this->mayScaleUp класса t3lib_stdgraphics (зачастую "gifbuilder").	
USERNAME_substToken	string	Маркер, использующийся на странице для замены именем текущего пользователя ЕСЛИ он зарегистрировался! Если он не зарегистрировался, замены не происходит.	<!-- ###USERNAME### -->

Свойство:	Тип данных:	Описание:	По умолчанию:
USERID_substToken	string	Маркер, использующийся на странице для замены UID текущего пользователя ЕСЛИ он зарегистрировался! Если он не зарегистрировался, замены не происходит. Для этого параметра нет значения по умолчанию и замена происходит только если это значение определить.	
cache_period	int, seconds	Количество секунд, в течение которых страница хранится в кеше. Это значение переназначается значением, установленным для записи страницы (поле="cache_timeout"), если оно больше нуля.	86400 (=24ч)
cache_clearAtMidnight	boolean	Установка этого параметра приводит к обновлению кеша каждую полночь.	false
no_cache	boolean	Если это истина, то страница не кешируется. Если ложь, то игнорируется. Другие параметры могут установить его в истину по другим причинам.	-
disableAllHeaderCode	boolean	Если установлено, то ни один из параметров объекта PAGE не обрабатывается и содержимым страницы станет результирующий массив cObject (1,2,3,4...) объекта PAGE. Это значит, что результирующий cObject будет включать все из <HTML> </HTML> тега!!! Используйте эту возможность в шаблонах, представляющих отличные от HTML типы содержимого. Это могут быть изображения или WAP-страницы!	false
disablePageExternalUrl	boolean	Если установлено, то для страниц с типом документа "External Url" не будут включены jumpUrl в TSFE. Это поможет со внешними url, которые открываются внутри набора фреймов.	
stat	boolean	Включение журналирования статистики для всего.	true
stat_typeNumList	int/list	Список типов страниц, которые должны регистрироваться в статистической таблице, sys_stat. Если типы не перечислены, регистрируются все типы. По умолчанию "0,1", что обычно регистрирует все посещения в наборе фреймов и посещения страниц с содержимым. Конечно, это зависит от дизайна шаблона.	0,1
stat_excludeBEuserHits	boolean	Если установлено, посещение страницы не регистрируется при регистрации пользователя в TYPO3.	false
stat_excludeIPList	список строк	Если REMOTE_ADDR находится в списке IP-адресов, он также не регистрируется. Можно использовать шаблон, например "192.168.1.*"	
stat_mysql	boolean	Включение регистрации в таблицу MySQL sys_stat.	false
stat_apache	boolean	Включение регистрации в файл "stat_apache_logfile"	false

Свойство:	Тип данных:	Описание:	По умолчанию:
stat_apache_logfile	filename	<p>Определение название файла журнала, куда TYPO3 записывает статистику в стиле Apache. Расположение директории определяется в \$TYPO3_CONF_VARS["FE"]["logfile_dir"], она должна существовать и быть открытой для записи. Путь может быть относительным (к PATH_site) или абсолютным, но, в любом случае, он должен находиться внутри дозволенных путей для TYPO3 (для абсолютного пути это значит, что должно быть внутри переменной "lockRootPath" из \$TYPO3_CONF_VARS).</p> <p>Также в названиях файлов возможно использовать маркеры времени, как они описаны в функции PHP strftime(). Это обеспечит естественный оборот файлов статистики.</p> <p>Пример:</p> <pre>config.stat_apache_logfile = typo3_%Y%m%d.log</pre> <p>Создается ежедневный файл статистики (напр. typo3_20060321.log).</p>	
stat_apache_pagename	string	<p>"pagename" смоделированная для apache.</p> <p>По умолчанию: "[path][title]--[uid].html"</p> <p>Коды:</p> <ul style="list-style-type: none"> [title] = вставляет заголовок, без спецсимволов и урезанный до 30 символов. [uid] = id [alias] = любой псевдоним [type] = тип (typeNum) [path] = путь к странице [request_uri] = вставляет серверное значение REQUEST_URI (полезно, например для RealUrl) 	
stat_apache_notExtended	boolean	Если истина, то файл статистики НЕ записывается в расширенном формате Apache.	
stat_apache_noHost	boolean	Если истина, HTTP_HOST, если доступна, НЕ вставляется вместо IP-адреса.	
stat_apache_niceTitle	boolean / string	<p>Если установлено, URL будет переведен из renderCharset в ASCII (например ä => ae, à => a, &#945; "alpha" => a), что делает хорошо читаемые заголовки страниц в журнале. Все не ASCII символы, которые не могут быть преобразованы заменяются на подчеркивания.</p> <p>Если установлено в "utf-8", заголовки страниц будут преобразованы в UTF-8, что в результате даст даже лучший результат, только если программа анализа статистики поддерживает эту кодировку.</p>	
stat_apache_noRoot	boolean	Если установлено, корневая часть (уровень 0) пути будет удалена. Названия становятся короче, если имеется избыточная часть, вроде "дом" или "мой сайт".	
stat_titleLen	int 1-100	Длина названий страниц в пути, записываемом в журнал/базу данных.	20

Свойство:	Тип данных:	Описание:	По умолчанию:
stat_pageLen	int 1-100	Длина названия страниц (в конце пути) записываемом в журнал/базу данных.	30
simulateStaticDocuments	boolean / string	<p>Если установлено, TYPO3 делает ссылки не так как обычно. Может использоваться с Apache компилированным с mod_rewrite и настройками в httpd.conf для использования этого в файлах ".htaccess".</p> <p>Включите это в файл .htaccess:</p> <pre>RewriteEngine On RewriteRule ^[^\.]*\.html\$ index.php</pre> <p>Это значит, что любые "*.html" документы должны быть обработаны index.php.</p> <p>Если это так, TYPO3 будет интерпретировать url html документа так:</p> <pre>[title].[id].[type].html</pre> <p>Заголовок не обязателен и полезен лишь для файла журнала apache. Можно не использовать и [title], и [type], но если заголовок присутствует, то должен быть и тип!.</p> <p>Пример: TYPO3 интерпретирует это как страницу с uid=23 и type=1 : <code>Startpage.23.1.html</code></p> <p>TYPO3 интерпретирует это как страницу с alias = "start" и type ноль (по умолчанию): <code>start.html</code></p> <p>Альтернативный параметр (PATH_INFO): Вместо использования модуля rewrite в apache (например в Windows!) возможно использовать переменную PATH_INFO из PHP. Это очень просто. Просто установите simulateStaticDocuments равным "PATH_INFO" и все заработает!</p> <p>Дополнительно: смотрите далее, .absRefPrefix</p> <p>Пример (вставьте в поле Setup в вашем шаблоне): <code>config.simulateStaticDocuments = PATH_INFO</code></p>	<p>По умолчанию определяется параметром настроек в localconf.php. \$TYPO3_CONF_VARS["FE"]["simulateStaticDocuments"] = 1; затрагивает все сайты в базе данных. Также возможно установить это значение к строке "PATH_INFO"</p>
simulateStaticDocuments_addTitle	int	<p>Если не ноль, TYPO3 формирует url с названиями, обрезанными до первых [simulateStaticDocuments_addTitle] количества символов.</p> <p>Пример: <code>Startpage.23.1.html</code> вместо по умолчанию, "23.1.html", без заголовка.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
simulateStaticDocuments_noTypeIfNoTitle	boolean	<p>Если установлено, то значение type не устанавливается в моделируемом названии файла, если значение type равно нулю. Тем не менее, название файла должно быть без заголовка.</p> <p>Пример: “Startpage.23.0.html” останется “Startpage.23.0.html” “23.0.html” станет “23.html” (то же, что и было, но без нуля) “23.1.html” останется “23.1.html”</p>	
simulateStaticDocuments_replacementChar	string	<p>Разделитель слов для URL сформированных через simulateStaticDocuments. Если установлено на дефис, этот параметр позволяет поисковым движкам индексировать ключевые слова в URL. До TYPO3 4.0 этот символ был жестко закодирован на подчеркивание.</p> <p>В зависимости от режима совместимости (см. Сервис управления>Установка>Update wizard): <i>compatibility mode < 4.0:</i> подчеркивание “_” <i>compatibility mode >= 4.0:</i> дефис “-”</p>	
simulateStaticDocuments_dontRedirectPathInfoError	boolean	<p>Зависит от режима PATH_INFO:</p> <p>Когда страница запрашивается методом “PATH_INFO”, он должен быть настроен для правильной работы. Если PATH_INFO не настроен, сценарий index_ts.php отсылает заголовок расположения правильной страницы. В любом случае, если Вы предпочитаете сообщения об ошибках, просто установите этот параметр.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
simulateStaticDocuments_pEnc	string	<p>Позволяет закодировать дополнительные параметры в моделируемом названии файла.</p> <p>Пример: Имеется новостное расширение. Url главной страницы "Page_1.228.0.html", но при щелчке по элементу новостей, url должен быть "Page_1.228.0.html?&tx_mininews_pi1[showUid]=2&cHash=b8d239c224". Кроме того, этот URL не должен индексироваться сторонними поисковыми механизмами, так как имеется строка запроса (все после знака "?"). Это свойство позволяет решить эту проблему через добавление параметров:</p> <p>Установить значение "base64": При этом название файла будет преобразовано так: "Page_1.228+B6JnR4X21pbmluZXdzX3BpMVtzaG93VWIkXT0yJmNIYXNoPWl4ZDlzOWMyMjQ_.html". Стока запроса была просто закодирована в base64 (и что-то еще...) и добавлена к названию HTML файла (теперь внешние поисковые системы обнаружат его!). По настоящему здорово то, что в названии файла содержатся параметры. Расплата за это — очень длинное название файла.</p> <p>Установить значение "md5": При этом название файла будет преобразовано так: "Page_1.228+M57867201f4a.0.html". Здесь название короче! При этом все параметры хешированы в 10-символьную строку, добавленную в название файла. В то же время, элемент был добавлен в таблицу кеша в базу данных. Поэтому, когда запрос на это имя файла будет отправлен во внешнем интерфейсе, в базе данных будет найдена ДЕЙСТВИТЕЛЬНАЯ строка параметров! И главное — название файла осталось довольно короткой (в отличие от метода base64). Недостаток в том, что ЕСЛИ очистить таблицу кеша базы данных, эти URL не сработают, пока страница с ссылками не будет снова сформирована (и в базу данных будет снова занесен список параметров).</p> <p>ЗАМЕЧАНИЕ: начиная с TYPO3 3.6.0 кодирование работает лишь с параметрами, вручную занесенными в список через .simulateStaticDocuments_pEnc_onlyP (см. следующее) или с теми, которые могут быть разрешены различными расширениями. Это позволяет уменьшить риск при объединении многих параметров.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
simulateStaticDocuments_pEnc_onlyP	string	<p>Список переменных, которые могут быть частью виртуального названия файла simulate_static_document, закодированного с помощью md5/base64 (см. свойство выше).</p> <p>Пример:</p> <pre>simulateStaticDocuments_pEnc_onlyP = tx_maillisttofaq_pil[pointer], L, print</pre> <p>-> здесь включен параметр "pointer" для расширения "maillisttofaq" (в дополнение к любым переменным, установленным самим расширением), а также и параметры "L" (возможно выбор языка) и "print" (возможно версия для печати).</p>	
content_from_pid_allowOutsideDomain	boolean	<p>Возможность "Показать содержимое этой страницы" позволяет отобразить содержимое страниц лишь текущего домена. Включение этого параметра позволяет показывать содержимое любой страницы из дерева страниц!</p>	
absRefPrefix	string	<p>Если установлено это значение, то все относительные ссылки в TypoScript будут содержать эту строку, как префикс. Используется для преобразования относительных путей в абсолютные.</p> <p>Замечание: эти значения автоматически устанавливаются на название директории, содержащей сценарий index.php, если simulateStaticDocuments установлен на "PATH_INFO". Если вы работаете на сервере, к которому есть и внутренний и внешний доступ, то полезно установить absRefPrefix на url и путь этого сайта, например http://www.typo3.com/. Если это не сделать, то имеется риск возвращения страниц из кеша внутренней сети, при этом изображения и ссылки будут недоступны извне.</p>	
pageRendererTemplateFile	string	<p>Установка шаблона для класса, формирующего страницу (t3lib_PageRenderer).</p> <p>Пример:</p> <pre>pageRendererTemplateFile = fileadmin/test_pagerender.html</pre>	
noPageTitle	integer	<p>Если нужно лишь название сайта (из записи шаблона) в теге <title>, установите здесь 1. При значении 2 тег <title> совсем не выводится.</p> <p>Обратите внимание, что этот тег нужен для совместимости с XHTML, поэтому выключить его можно только если он формируется вручную.</p>	0
pageTitleFirst	boolean	<p>По умолчанию TYPO3 выводит тег title в формате "сайт: заголовок страницы".</p> <p>Если установлено (и выводится заголовок страницы), то заголовок страницы будут выводится ПЕРЕД заголовком шаблона. Это будет выглядеть так: "заголовок страницы: сайт".</p>	0

Свойство:	Тип данных:	Описание:	По умолчанию:
titleTagFunction	название функции	Передает содержимое тега <title> в эту функцию. Никакие параметры typoScript не передаются.	
moveJsFromHeaderToFooter	boolean	Если установлено, то весь Javascript (включенный и содержащийся в коде) перемещается вниз документа HTML, после содержимого и перед закрывающим тегом body.	
headerComment	string	Содержимое добавляется перед комментарием "TYPO3 Content Management Framework" в разделе <head> страницы. Используется для вставки примечаний, вроде "Сделано в моем агентстве".	
language	string	<p>Языковой ключ. Дополнительную информацию см. Выбор между: English (default) = [empty] Danish = dk German = de Norwegian = no Italian = it и т.п...</p> <p>Значение должно соответствовать ключу, используемому для языка внутреннего интерфейса, если он есть. Смотрите в config_default.php или ищите официальные 2-байтовые ключи для нужных языков на TYPO3.org. Помните, что выбор официальных ключей важен, если нужно использовать корректные языковые метки из "locallang" файлов.</p> <p>Если нужный язык пока еще не является системным языком в TYPO3 можно использовать собственную искусственную строку и брать для нее значения через шаблон TypoScript, в котором свойство "_LOCAL_LANG" в большинстве расширений служит для перезаписи / добавления значений меток.</p> <p>Используемые ключи нужно искать в locallang файлах используемых расширений.</p>	
language_alt	string	Если используется "config.language" (выше), здесь возможно установить ключ другого языка, используемого для меток, если метки для основного языка не найдены. Например, бразильский сайт на португальском может иметь "pt" в качестве альтернативного языка, при этом метки будут показаны на португальском, если они не будут найдены на бразильском наречии португальского языка. Эта возможность важна, если один из языков не до конца переведен, но имеется перевод на другом, сходном языке.	

Свойство:	Тип данных:	Описание:	По умолчанию:
sys_language_uid	int	<p>Это значение указывает на uid записи из таблицы "sys_language" и, если установлено, то различные части кода, отображаемого во внешнем интерфейсе, будут выбирать записи, связанные с этим языком. Смотрите ->SELECT.</p> <p>Кроме того, значение зависит от того, могут ли быть найдены записи страницы на альтернативном языке. Если нет, значение по умолчанию будет нулевым (язык по умолчанию), если значение "sys_language_mode" не установлено на что-то вроде "content_fallback".</p>	
sys_language_mode	string	<p>Настройка режимов обработки локализации. Синтаксис "[ключ] : [значение]".</p> <p>Возможные ключи:</p> <p>[default] – система ищет переводы страницы (в таблице "Alternative Page Language") и, если не находит, возвращает для отображения язык по умолчанию.</p> <p>content_fallback - [рекомендуется] система всегда будет работать с выбранным языком, даже если для страницы нет записей переводов. При этом меню и т.п. останутся переведенными. В любом случае, <i>содержимое</i> на странице может быть выведено на другом языке, определяемым значением этого ключа, например "content_fallback ; 1,0" для вывода не переведенного содержимого на языке sys_language_uid 1, и если оно не представлено, то на языке по умолчанию (0)</p> <p>strict – система сообщит об ошибке, если запрашиваемый перевод не найден. Обычно это значит, что все страницы с серым фоном в модуле Веб->Инфо / Обзор локализации будут с ошибкой (другими словами, они будут сброшены на язык по умолчанию так или иначе)</p> <p>ignore – система останется на выбранном языке, даже если страница не переведена и содержимое на выбранном языке недоступно, после чего эту ситуацию возможно обработать самостоятельно.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
sys_language_overlay	boolean / keyword	<p>Если установлено, записи из определенных таблиц, выбранных через CONTENT cObject используя установку "languageField" выберут язык по умолчанию (0) вместо любых языков, установленных в sys_language_uid / sys_language_mode. Дополнительно система будет искать перевод выбранных записей и перекрывать настроенные поля.</p> <p>Для этого требуется, чтобы таблица была настроена с "languageField" и "transOrigPointerField" в разделе [ctrl] \$TCA. Кроме того, исключение определенных полей может быть сделано через директиву "l10n_mode" в настройке полей \$TCA.</p> <p>Для управления внутренним интерфейсом требуется настройка модуля "Веб>Страница" для отображения соответствующих элементов; то есть будет показан элемент по умолчанию и рядом с ним любой найденный перевод. Эта настройка может быть сделана через Tsconfig страницы для раздела сайта, используя путь объекта "mod.web_layout.defLangBinding = 1".</p> <p>Ключ:</p> <p>hideNonTranslated : при использовании этого ключа, запись без перевода не будет показана. По умолчанию запись без перевода отображается на языке по умолчанию.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
sys_language_softMergeIfNotBlank	string	<p>Установка дополнительных полей "mergeIfNotBlank" из TypoScript.</p> <p>Объяснение: В ТСА возможно настроить "l10n_mode", модель локализации, для каждого поля. Два параметра отвечают за отображение содержимого во внешнем интерфейсе; это "exclude" и "mergeIfNotBlank" (за деталями обратитесь к документы "TYPO3 Core API"). Первый ("exclude") означает, что поле найденное в переводе записи не будет перекрывать значение записи поля по умолчанию. Второй ("mergeIfNotBlank") значит, что оно будет перекрываться, <i>только если</i> имеется не пустое значение в переводе. Так как практически установить поля "mergeIfNotBlank" для каждого сайта, этот параметр позволяет переназначить дополнительные поля из таблиц.</p> <p>Синтаксис: [таблица]:[поле], [таблица]:[поле], [таблица]:[поле], ...</p> <p>Пример: <code>config.sys_language_softMergeIfNotBlank = tt_content:image , tt_content:header</code></p> <p>Эта настройка означает, что поля header и image элемента содержимого будут браться из перевода, только если в них содержится не пустое значение. Для поля image это очень практично, т.к. будет(ут) использоваться изображение(я) из перевода по умолчанию если не вставлены другие изображения!</p>	
sys_language_softExclude	string	<p>Установка дополнительного "exclude" флага l10n_mode в ТСА для вывода во внешний интерфейс. Работает также, как sys_language_softMergeIfNotBlank (детали содержится там, синтаксис тот же!).</p> <p>Свойства для полей, установленные этим свойством будут сброшены, если для тех же полей установлено "sys_language_softMergeIfNotBlank".</p>	
typolinkCheckRootline	boolean	<p>Если установлено, то каждая "typolink" проверяется на связь со страницей в корневой линии сайта. Если связи нет, то TYPO3 ищет первую запись домена (без перенаправления) в этой корневой линии. Если найден (другой домен), то этот домен приставляется к ссылке и используется внешняя цель, а ссылка ведет на страницу с правильным доменом.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
typolinkEnableLinksAcrossDomains	boolean	<p>Этот параметр включает создание ссылок между доменами, используя схему ссылок текущего домена.</p> <p>Если этот параметр не установлен, то все междоменные ссылки будут формироваться так: "http://domain.tld/index.php?id=12345" (где 12345 – id страницы). Если этот параметр включен и текущий сайт использует, например, simulateStatic, то ссылки будут формироваться так: "http://domain.tld/PageTitle.12345.html" (включая RTE ссылки). Для этого параметра требуется, чтобы домены, на которые ссылаются со страниц, имели те же настройки для:</p> <ul style="list-style-type: none"> - схемы ссылок (т.е. везде должны использоваться simulateStatic или RealURL или CoolURI но не их смесь) - все домены имели идентичные параметры локализации (config.sys_language_XXX директивы) - все домены имели те же определенные языки <p>Этот параметр подразумевает "config.typolinkCheckRootline=1", который включается автоматически. Установка параметра "config.TypolinkCheckRootline" в TS шаблоне не будет иметь эффекта.</p> <p>Оговорка: должно быть понятно, что ссылка на другой домен формируется в контексте текущего домена. Побочные эффекты этого не наблюдались на момент написания этой документации, но они могут существовать. Если они будут найдены, документация будет обновлена соответствующим образом.</p>	0
typolinkLinkAccessRestrictedPages	integer (id страницы) / ключ "NONE"	<p>Если установлено, то typolink, указывающие на страницы с запрещенным доступом останутся, даже если страница не будет доступна. Если значение этой страницы будет целым числом, то оно рассматривается как id страницы, на которую будет вести такая ссылка.</p> <p>Если значение "NONE" оригинальная ссылка на страницу будет сохранена, хотя она сгенерирует ситуацию страница-не-найдена (которая, конечно же, должна быть правильно обработана).</p> <p>Смотрите также "showAccessRestrictedPages" для объектов меню (та же возможность для меню).</p> <p>Пример:</p> <pre>config.typolinkLinkAccessRestrictedPages = 29 config.typolinkLinkAccessRestrictedPages_addParams = &return_url=###RETURN_URL###&pageId=###PAGE_ID###</pre> <p>Создается ссылка на страницу id 29 и добавляются параметры GET с обратным URL и id оригинальной страницы.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
typolinkLinkAccessRestrictedPages_addParams	string	Смотрите “typolinkLinkAccessRestrictedPages” выше.	
notification_email_url_mode	string	<p>Этот параметр позволяет обрабатывать URL в простых текстах email, так чтобы длинные URL, длиной более 76 символов не обрезались. Этот параметр может быть либо пустым, либо “76”, либо “all”.</p> <p>Если строка пустая, все ссылки в простом тексте email не затрагиваются.</p> <p>Если установлено 76, то все ссылки длиннее 76 символов сохраняются в базе данных, а хеш в переменной GET ?RDCT=[md5/20] посыпается вReinstated PHP_SCRIPT сценарий index.php, который ищет соответствующую ссылку в базе данных, получая заголовок местоположения (перенаправление).</p> <p>Если установлено “all”, тогда преобразуются ВСЕ “http://” ссылки в сообщении.</p>	
notification_email_encoding	string	<p>Настройка кодировки обычных текстовых email (уведомлений). По умолчанию кодировка “quoted-printable”. Но установкой, например “base64”, можно закодировать содержимое через base64.</p> <p>Возможные значения: base64 quoted-printable 8bit</p>	
notification_email_charset	string	Альтернативная кодировка для уведомительных сообщений.	ISO-8859-1
admPanel	boolean	<p>Если установлено, внизу страниц появляется панель управления.</p> <p>ЗАМЕЧАНИЕ: дополнительно панель должна быть включена для пользователя через TSconfig пользователя! Смотрите документацию по TSconfig относительно дополнительных свойств панели администрирования.</p>	
beLoginLinkIPList	[IP-номер]	<p>Если установлено и REMOTE_ADDR соответствует списку IP-адресов (поддерживаются шаблоны, *), то ссылка на сценарий typo3/ login будет перенаправлена на отображаемую страницу.</p> <p>ЗАМЕЧАНИЕ: beLoginLinkIPList_login и/или beLoginLinkIPList_logout (см. далее) должны быть определены, если ссылка должна быть показана!</p>	
beLoginLinkIPList_login	HTML	HTML код, заключаемый в ссылку на регистрацию в системе, смотрите 'beLoginLinkIPList'	
		Пример: <hr />АВТОРИЗАЦИЯ	
beLoginLinkIPList_logout	HTML	HTML код, заключаемый в ссылку выхода из системы, смотрите выше.	

Свойство:	Тип данных:	Описание:	По умолчанию:
index_enable	boolean	Включение индексации кешируемых страниц.	
index externals	boolean	Если включено, внешние медиа, связанные со страницей, также индексируются.	
index_descrLgd	int	Здесь указывается, сколько символов сохранять в качестве описания индексированных страниц. Можно использовать для отображения результатов поиска.	200
index_metatags	boolean	Позволяет включить или выключить индексацию метатегов. По умолчанию включено.	true
xhtml_cleaning	string	<p>Попытка очистки выходного кода, сделать его совместимым с XHTML и немного уменьшить. ПОКА ЕЩЕ НЕ ЗАВЕРШЕНО, но можно предварительно посмотреть, нужно ли это.</p> <p>Что сделано к этому времени:</p> <ul style="list-style-type: none"> - все теги (img,br,hr) оканчиваются на "/> - что-то еще?; - элементы и атрибуты в нижнем регистре; - все атрибуты в кавычках; - добавление атрибута "alt" к тегам img, если их там не было. <p>Что пока НЕ сделано, согласно спецификации XHTML:</p> <ul style="list-style-type: none"> - Wellformedness: вложенность НЕ проверяется; - использование атрибутов name/id не отслеживается. - не позволена определенная вложенность элементов. Интереснее, что <PRE> не может содержать img, big,small,sub,sup ...; - обертка сценариев и стилевых элементов содержания в CDATA, или в качестве альтернативы, они должны быть преобразованы в литеры (entities); - настройки кодировок может поместить специальные требования в XML declaration/ meta-http-equiv. (C.9); - UTF-8 кодировка фактически ожидается в XML!!! - элементы стиля и названия атрибутов НЕ должны преобразовываться в нижний регистр; - амперсанды (и все остальные литеры (entities), как мне кажется) ДОЛЖНЫ быть преобразованы в коды символов! (&amps;). Это может означать дальнейшее преобразование содержимого вне тегов перед выводом страницы. Может быть связано с проблемой кодировки в целом. - Не разрешены сокращения значений- нужно сделать следующее: selected="selected" <p>За деталями обратитесь к классу t3lib_parsehtml Можно задействовать эту функцию со следующими значениями:</p> <p>all = содержимое всегда обрабатывается перед сохранением в кеше;</p> <p>cached = обрабатываются только страницы, помещаемые в кеш;</p> <p>output = только выводимый код перед его отображением.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
prefixLocalAnchors	string keyword	Если установлен один из ключей, содержимое будет иметь локальные привязки для ссылок в виде путей для сценария. В основном это значит, что <code></code> будет преобразовано в <code></code> . Эта процедура необходима если тег <code><base></code> установлен на сценарий (например, если используется расширение "realurl" для понятных URL). Ключи те же, что и в "xhtml_cleaning", смотрите выше.	
disablePrefixComment	boolean	Если установлено, свойство "prefixComment" в stdWrap отключено, тем самым предотвращается все объясняющие и потребляющие место комментарии в коде HTML.	
baseURL	string	В заголовок документа прописывается тег <code><base></code> . Установите его значение на соответствующий URL, и добавьте в конце строки "/".	
Пример: <code>config.baseURL = http://typo3.org/sub_dir/</code>			
tx_[extension key with no underscores]_[*]	-	Место настройки для дополнений. Можно использовать, например, для расширения в случае необходимости настройки его через TypoScript, но это фактически ничего не отображает во внешнем интерфейсе (например, невозможно получить настройки в виде аргументов из внешнего интерфейса в процессе формирования).	
Пример: <code>config.tx_realurl_enable = 1</code>			

[tsref:config/->CONFIG]

"CONSTANTS"

Свойство:	Тип данных:	Описание:	По умолчанию:
Массив...	string	Константы. Примеры: <code>.EMAIL = email@email.com</code> При этом, если parseFunc где либо будет настроена с constants=1, то все вхождения строки <code>###EMAIL###</code> будут заменены текстом. смотрите ->parseFunc	

[tsref:constants]

"PAGE"

Страница описывается двумя основными значениями — "id" и "type".

Страница описывается двумя основными значениями — "id" и "type".

"**Id**" указание uid страницы (или псевдонима). Так ищется страница.

"**Type**" определяет способ обработки страницы. В основном используется с фреймами.

Набор фреймов обычно имеет type=0 (или он не установлен), а документы в фрейме должны быть определены другими типами, например type=1 для страницы с содержимым.

Вы должны изучить наборы фреймов для TYPO3 сайтов. Кроме того, изучите стандартные шаблоны для набора фреймов.

Хорошая привычка, использовать type=1 для основной страницы сайта с фреймами. Для сайтов без фреймов, обычно тип равен нулю.

Еще одна хорошая привычка - использование "page" в качестве названия объекта основного уровня (toplevel) для страницы содержимого на сайте.

Большая часть этого кода находится в сценарии PHP
typo3/sysext/cms/tslib/class.tslib_pagegen.php.

Свойство:	Тип данных:	Описание:	По умолчанию:
typeNum	int	Это typelid страницы. Значение по умолчанию – 0 для первого найденного объекта PAGE, но ДОЛЖНО быть определено и быть уникальным, если используется более одного такого объекта (повнимательнее, если используются фреймы)!	0
1,2,3,4...	cObject		
wrap	wrap	Обертка содержимого массива cObject.	
stdWrap	->stdWrap	Обертки содержимого массива cObject с параметрами stdWrap.	
bodyTagCObject	cObject	Значение по умолчанию для bodytag переназначенное ".bodyTag", если оно установлено.	
bodyTag	<tag>	Тег body страницы. Пример: <body bgcolor="{\$bgCol}">	<body bgcolor="#FFFF FFFF">
headTag	<tag>	Тег head, если нужна альтернатива.	<head>
bodyTagMargins	int	Отступы в теге body. Свойство: .useCSS = 1 (boolean) – установит строку "BODY {margin: ...}" при объявлении стилей документа, для совместимости с XHTML. Пример: значение 4 добавит leftmargin="4" topmargin="4" marginwidth="4" marginheight="4" в тег body.	
bodyTagAdd	string	Содержимое, добавляемое в конец тега body.	
bgImg	imgResource	Фоновое изображение для страницы. Автоматически добавляется в тег body.	
frameSet	->FRAMESET	Если здесь установить любое свойство, страница будет выводится в наборе фреймов.	
meta	->META		
shortcutIcon	resource	Favicon для страницы. Создайте ссылку на файл icon! Браузеры, поддерживающие отображение favicons,	

Свойство:	Тип данных:	Описание:	По умолчанию:
		показывают ее в адресной строке, в списке закладок рядом с названием сайта и на вкладках, рядом с названием страницы.	
		<p>Примечание: Это должен быть настоящий файл ".ico" (iconfile)</p>	
headerData	->CARRAY	Вставляет содержимое в раздел header. Это могут быть JavaScript, теги meta, ссылки на другие таблицы стилей. Вставляется после определений стилей.	
footerData	->CARRAY	То же, что и headerData выше, за исключением того, что этот блок включается внизу страницы (прямо перед закрывающим тегом body).	
config	->CONFIG	Настройка страницы. Значения переназначают те же значения в объекте основного уровня "config".	
includeLibs	массив строк	<p>Так возможно включить файлы php. То же самое, что делает "includeLibrary" в ->CONFIG, но так возможно включить несколько файлов. Эти файлы включаются <i>после</i> файла includeLibrary.</p> <p>ЗАМЕЧАНИЕ: Объект основного уровня "includeLibs" и сценарии, определенные в этом свойстве, добавляются друг за другом. Ключи сценария (значение "массив строк", как ниже в "tx_myext") из этого свойства перезаписывают любые ключи сценариев из свойства объекта основного уровня "includeLibs"! Названия файлов сценариев имеют тип "resource".</p> <p>Пример: <code>includeLibs.tx_myext = lib_filename.php</code></p>	
JavaScript:			
javascriptLibs	массив строк	<p>Так можно подключить библиотеки JavaScript, которые распространяются с ядром TYPO3.</p> <pre>javascriptLibs { # включить prototype Prototype = 1 # включить Scriptaculous Scriptaculous = 1 # добавление модулей dragdrop и controls в Scriptaculous Scriptaculous.modules = dragdrop,controls # включить ExtCore ExtCore = 1 # включить файл отладки ExtCore(без сжатия) ExtCore.debug = 1 # includes ExtJS ExtJs = 1 # включить ext-all.css ExtJs.css = 1 # включить тему по умолчанию ExtJs.theme = 1 # загрузка специальных адаптеров(jquery </pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<pre>prototype yui) ExtJs.adapter = ... # инициализация QuickTips ExtJs.quickTips = 1 # включение файла отладки ExtJS(без сжатия) ExtJs.debug = 1 # включить библиотеку SVG SVG = 1 # включить файл отладки SVG SVG.debug = 1 # принудительное формирование с флеш SVG.forceFlash = 1 }</pre> <p>Замечание: если сразу запрошены ExtCore и ExtJS, будет загружена лишь ExtJS. Это отразиться на всех установленных параметрах. Они затронут лишь ExtJS.</p> <p>Замечание: в TYPO3 4.5.2 и старше необходимо запрашивать что-то одно — ExtJS или ExtCore, но не вместе. Попытка их одновременной загрузки приведет к ошибкам.</p>	
inlineLanguageLabel	массив строк	<p>ExtJS specific, adds language labels to the page.</p> <p>Пример:</p> <pre>inlineLanguageLabel { label1 = 123 label2 = 456 }</pre> <p>Выведет следующий код:</p> <pre>TYP03.lang = {"label1":"123", "label2":"456"};</pre>	
inlineSettings	массив строк	<p>Относиться к ExtJS, на страницу добавляются языковые метки.</p> <p>Пример:</p> <pre>page.inlineSettings { setting1 = Hello setting2 = GoOnTop }</pre> <p>Выведет следующий код:</p> <pre>TYP03.settings = {"TS": {"setting1":"Hello", "setting2":"GoOnTop"}};</pre>	
extOnReady	->CARRAY	<p>Относиться к ExtJS, добавляется встроенный javascript, обернутый в Ext.onReady.ExtJS specific.</p> <p>Пример:</p> <pre>page.extOnReady { 10 = TEXT 10.value = Ext.Msg.alert("TypoScript Message","Hello World!"); }</pre> <p>Выведет следующий код:</p> <pre>Ext.onReady(function() {Ext.Msg.alert("TypoScript Message","Hello World!");});</pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
includeJSlibs. [array]	resource	<p>Добавление файлов JS библиотеки в заголовок страницы.</p> <p>Определенный файл должен быть настоящим типом "resource", иначе ничего не будет вставлено. Это значит, что на файлы нельзя ссылаться (например, через "http://..."), за исключением использования свойства ".external".</p> <p>Каждый файл имеет <i>необязательные свойства</i>:</p> <ul style="list-style-type: none"> .allWrap – обертка для всего тега, полезно для условных комментариев. .external – если установлено, существование файла не проверяется. Полезно для включения внешних файлов. <p>Пример:</p> <pre>includeJSlibs.twitter = http://twitter.com/javascripts/blogger.js includeJSlibs.twitter.external = 1</pre>	
includeJSFooterIbs. [array]	resource	То же, что и includeJSlibs выше, за исключением того, что блок включается внизу страницы (прямо перед закрывающим тегом body).	
includeJS. [array]	resource	<p>Вставка одного или нескольких (Java)Scripts в тег <script>.</p> <p>Определенный файл должен быть настоящим типом "resource", иначе ничего не будет вставлено. Это значит, что на файлы нельзя ссылаться (например, через "http://..."), за исключением использования свойства ".external".</p> <p>Каждый файл имеет <i>необязательные свойства</i>:</p> <ul style="list-style-type: none"> .type - установка MIME типа сценария (по умолчанию: text/javascript) .compressed – флаг. Используется внешними компрессорами, чтобы определить, был ли файл предварительно сжат. .forceOnTop – флаг. Если установлен, то файл будет добавлен выше остальных. .allWrap – обертка для всего тега, полезно для условных комментариев. .external – если установлено, существование файла не проверяется. Полезно для включения внешних файлов. <p>Пример:</p> <pre>includeJS { file1 = fileadmin/helloworld.js file1.type = application/x-javascript file2 = javascript_uploaded_to_template*.js }</pre>	
includeJSFooter. [array]	resource	То же, что и includeJS выше, за исключением того, что блок включается внизу страницы (прямо перед закрывающим тегом body).	

Свойство:	Тип данных:	Описание:	По умолчанию:
jsInline	->CARRAY	<p>Использование cObjects для создания встроенного JavaScript.</p> <p>Пример:</p> <pre>page.jsInline { 10 = TEXT 10.dataWrap = var pageId = {TSFE:id}; }</pre> <p>Замечание: через config.removeDefaultJS = external, inlineJS помещается во внешний файл. Через config.minifyJS = 1, jsInline будет по возможности сокращен.</p>	
jsFooterInline	->CARRAY	<p>То же, что и jsInline выше, за исключением того, что блок включается внизу страницы (прямо перед закрывающим тегом body).</p>	
inlineJS	->CARRAY	<p>Включение Javascript в раздел header. Не используете теги script, так как они добавляются TYPO3.</p> <p>Пример:</p> <pre>page.inlineJS.10 = TEXT page.inlineJS.10.value = function a(val) { alert(val); }</pre> <p>Через config.removeDefaultJS = external, inlineJS перемещается во внешний файл. Через config.minifyJS = 1, inlineJS будет по возможности сокращен.</p> <p>Замечание: это устаревшая функция, которая была удалена в TYPO3 4.3. Вместо нее используйте jsInline instead.</p>	

Таблицы стилей CSS:

stylesheet	resource	Вставляет таблицу стилей в раздел <HEAD> страницы; <link rel="stylesheet" href="[resource]">	
includeCSS. [array]	resource	<p>Вставка таблицы стилей (как и свойство .stylesheet), позволяет включить несколько таблиц стилей, введя названия файлов в массив.</p> <p>Определенные файлы должны иметь правильный тип "resource", иначе они не будут вставлены.</p> <p>Каждый файл имеет <i>необязательные свойства</i>:</p> <ul style="list-style-type: none"> .media – установка атрибута media тега <style>; .title – установка заголовка тега <style>; .alternate – если включено (boolean), атрибут rel будет "alternate stylesheet"; .import – если установлено (boolean), используется возможность @import для включения таблицы стилей, вместо <link>; .allWrap – обертка всего тега, полезна для условных 	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>комментариев.</p> <p>.external – если установлено, не делается проверки существования файла. Полезно для включения внешних файлов.</p> <p>Пример:</p> <pre>includeCSS { file1 = fileadmin/mystylesheet1.css file2 = stylesheet_uploaded_to_template*.css file2.title = High contrast file2.media = print ie6Style = fileadmin/css/style3.css ie6Style.allWrap = <!--[if lte IE 7]> <! [endif]--> cooliris = http://www.cooliris.com/shared/ resources/css/global.css cooliris.external = 1 }</pre>	
cssInline	->CARRAY	<p>Используется cObjects для создания встроенного CSS.</p> <p>Пример:</p> <pre>cssInline { 10 = TEXT 10.value = h1 {margin:15px;} 20 = TEXT 20.value = h1 span {color: blue;} }</pre>	
CSS_inlineStyle	string	Значение просто передаваемое как встроенная css (встроенная в документ css, заключенная в теги <style>).	
Другое:			
insertClassesFromRTE	boolean	<p>Если установлено, классы для Rich Text Editor, настраиваемые в TSconfig страницы, вставляются в начало раздела Style, сразу после настройки таблицы стилей.</p> <p>.add_mainStyleOverrideDefs = [* / список тегов] – будут добавлены все настроенные теги “RTE.default.mainStyleOverride_add”.</p> <p><i>Скоро может устареть. Наиболее вероятно, что RTE нужно будет настраивать через таблицы стилей. Все еще в разработке...</i></p>	
noLinkUnderline	boolean	<p>Отключение подчеркивания ссылок. Используются встроенные в документ стили.</p> <p><i>Устарело. Вместо этого используйте таблицы стилей.</i></p>	
hover	HTML-цвет	<p>Цвет ссылок, когда мышка находится над ними! (только MSIE). Используются встроенные в документ стили.</p> <p><i>Устарело. Вместо этого используйте таблицы стилей.</i></p>	
hoverStyle	string	Дополнительная информация по стилям для hover-цвета.	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>Пример: <code>page.hoverStyle = font: bold; text-decoration: none;</code> <i>Устарело. Вместо этого используйте таблицы стилей.</i></p>	
smallFormFields	boolean	<p>Выводит поля форм, типа текстовых, полей ввода, блоки выбора с текстом "verdana size 1". Используются встроенные в документ стили.</p> <p>Совет: Используйте вместе с параметром настройки "compensateFieldWidth", установленным на "0.6" для браузера netscape для получения маленьких полей форм одинаковой ширины!</p> <p><i>Устарело. Вместо этого используйте таблицы стилей.</i></p>	
adminPanelStyles	boolean	Будут включены стили CSS для панели управления.	

[tsref:(page)]

"FE_DATA"

Свойство:	Тип данных:	Описание:	По умолчанию:
массив названий таблиц	->FE_TABLE		

[tsref:FEData]

"FE_TABLE"

Свойство:	Тип данных:	Описание:	По умолчанию:
default.[field]	string	<p>Это свойство отвечает за свойства таблиц по умолчанию:</p> <p>Пример: Определение значений для новых записей. Они будут переназначены любыми подставленными вместо них значениями (если подстановка разрешена через "allowNew")</p> <pre>default { subject = This is the default subject value! hidden = 1 parent = 0 }</pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
allowNew.[field]	string	<p>Это свойство отвечает за разрешение переназначения полей из внешнего интерфейса.</p> <p>Пример: Здесь определено, что поле <code>subject</code>, требуется для внешнего интерфейса, если значение не получено, в объект вставляется значение по умолчанию (смотрите выше). С другой стороны, поле <code>"hidden"</code> не может быть изменено во внешнем интерфейсе. Оно получает значение из определения его по умолчанию (смотрите выше). Установка для поля <code>"0"</code> (ноль) равнозначно не определению его в массиве.</p> <pre>allowNew { subject = 1 hidden = 0 }</pre>	
allowEdit.[field]	string	<p>Тоже, что и выше (<code>"allowNew"</code>), но отвечает за то, что записывается при обновлении записи (а не новое представление). Обратите внимание на свойство ниже (<code>"overrideEdit"</code>)</p>	
overrideEdit.[field]	string	<p>Работает подобно значению по умолчанию выше, но эти значения вставляются после отправки, вместо значений не прошедших обработку. Это означает обратное значениям по умолчанию, перезаписываемым при отправке значениями, прошедшими проверку.</p> <p>Пример: Здесь <code>overrideEdit</code> служит для защиты при обновлении пользователем своей записи (если он ее "владелец") поле <code>"hidden"</code> будет установлено не зависимо ни от чего.</p> <pre>overrideEdit { hidden = 1 }</pre>	
userIdColumn	string (field)	<p>Это строка, указывающая на столбец записи, в который должен вставляться id текущего fe_user пользователя. Этот fe_user uid вставляется/обновляется и как <code>"new"</code>, и как <code>"edit"</code></p>	
autoInsertPID	boolean	<p>Работает с новыми записями: автоматически вставляет PID страницы, на которой были отправлены данные. Все <code>"pid"</code> представленные из отправленных данных будут перезаписаны. Для удобства.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
processScript	resource	<p>Сценарий, включаемый для обработки введенных в таблицу данных. Сценарий включается из функции класса tslib_fetce.</p> <p>Очень важный параметр, так как будете ли вы использовать или нет возможность "cleaning"/"authorization", полностью зависит от написания сценария для обработки данных и помещения их в базу данных.</p> <p>Очень хороший пример приведен на "media/scripts/guest_submit.inc", включен из static_template "plugin.tt_guest" (Используется для функционала гостевой книги по умолчанию).</p>	
separator	string	Символ-разделитель, используемый, если данные получаются в виде массива из, например многоэлементного блока выбора.	chr(10) (окончание строки)
doublePostCheck	string (field name)	Определение поля (целое) в которое вставляется целочисленный хеш переданных данных. Если поле установлено, то проверяется, нет ли другой записи с таким значением. Если есть, то запись не вставляется, так как она проверяется на "дублирование" (отправка тех же данных повторно).	

[tsref:FEData.(tablename)/->FE_TABLE]

"FRAMESET"

Свойство:	Тип данных:	Описание:	По умолчанию:
1,2,3,4...	frameObj	Настройка фреймов и вложенных наборов фреймов.	
cols	<frameset>-data:cols	Колонки	
rows	<frameset>-data:rows	Строки	
params	<frameset>-параметры	Пример: border="0" framespacing="0" frameborder="NO"	

[tsref:(page).frameSet/->FRAMESET]

"FRAME"

Свойство:	Тип данных:	Описание:	По умолчанию:
obj	указатель на название объекта основного уровня	<p>Название объекта основного уровня PAGE / FRAMESET.</p> <p>Пример: "left", "page", "frameset"</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
options	<i>url</i> -параметры	Пример: print=1&othervar=anotherthing добавит '&print=1&othervar=anotherthing' к содержимому ".src" (если ".src" не установлен вручную!!!).	
params	<frame>-параметры	Пример: scrolling="AUTO" noresize frameborder="NO"	
name	<frame>-data:name	Вручную установленное название фрейма. ЗАМЕЧАНИЕ: устанавливается автоматически и не должно переписываться при обычных условиях!	значение ".obj"
src	<frame>-data:src /stdWrap	Вместо адресации в "obj", можно определить определенный источник для фрейма через этот параметр. При этом будет переназначено поведение по умолчанию для использования параметров "obj"!	typolink к id=[currentId &type=[obj- >typeNum]

[tsref:(page).frameSet.(number)/->FRAMESET.(number)]

Пример простого набора фреймов с topframe и content-frame:

```

frameset = PAGE
frameset.typeNum = 0

page = PAGE
page.typeNum = 1

top = PAGE
top.typeNum = 3

frameset.frameSet.rows = 150,*
frameset.frameSet.params = border="0" framespacing="0" frameborder="NO"
frameset.frameSet {
    1 = FRAME
    1.obj = top
    1.params = scrolling="NO" noresize frameborder="NO" marginwidth="0" marginheight="0"
    2 = FRAME
    2.obj = page
    2.params = scrolling="AUTO" noresize frameborder="NO"
}

```

"META"

Свойство:	Тип данных:	Описание:	По умолчанию:
Массив...	string /stdWrap	Теги meta Если значение пустое (после обрезки), метатег не формируется. Если "ключ" (например "REFRESH" или "DESCRIPTION") — "REFRESH" (не чувствителен к регистру), то в метатеге используется атрибут "http-equiv" вместо "name". Примеры: .REFRESH = [sec]; [url, остается пустым для некоторых страниц] .DESCRIPTION = Описание содержимого этого документа. .KEYWORDS = Ключевые слова...	

[tsref:->META]

"CARRAY"

Свойство:	Тип данных:	Описание:	По умолчанию:
1,2,3,4...	cObject	Числовой "массив" объектов содержимого (cObjects). Порядок определения объектов не важен, так как массив будет отсортирован перед его анализом!	
Отдельные свойства:			
(свойства stdWrap...)		ЗАМЕЧАНИЕ: применяется ТОЛЬКО если "CARRAY /stdWrap" установлен как тип данных Если определить для CARRAY любые свойства без чисел, stdWrap будет вызван для всех свойств CARRAY. Пример: Будет возвращено 'Это формируется перед "10"тестом' 10 = TEXT 10.value = тестом 5 = HTML 5.value = Это формируется перед "10" wrap = 	
(TDParams)	<TD>- параметры	ЗАМЕЧАНИЕ: применяется ТОЛЬКО если "CARRAY +TDParams" установлен как тип данных Это свойство используется лишь в некоторых случаях использования CARRAY. Ищите примечания об этом в случае использования.	

[tsref:->CARRAY]

Объекты содержимого (cObject)

PHP информация:

Объектами содержимого (cObjects) в основном управляет сценарий PHP "typo3/sysext/cms/tslib/content.php". Класс PHP под названием "tslib_cObj" и часто они являются также именами переменных объектов (\$cObj)

У \$cObj в PHP имеется массив, \$this->data, содержащий записи различных типов. Смотрите тип данных "getText".

Эта запись обычно "загружается" с записью из таблицы, в зависимости от ситуации. Скажем, при создании меню она зачастую загружается с записью страницы текущего элемента меню, или, относительно формирования содержимого — с записью этого содержимого.

ПОВТОРНОЕ ИСПОЛЬЗОВАНИЕ cOBJECTS:

При работе с "cObjects", возможно использовать специальный синтаксис для повторного использования cObjects без фактического создания копии. Это уменьшает размер шаблона в буфере. Но, с другой стороны, не дает гибкости при переназначении значений.

Этот пример покажет, как это работает:

```
#  
# Определение временных объектов:  
#  
lib.stdheader = COA  
lib.stdheader {  
    stdWrap.wrapAlign.field = header_position  
    stdWrap.typolink.parameter.field = header_link  
    stdWrap.fieldRequired = header  
  
    1 = TEXT  
    1.current = 1  
    1.fontTag = {$content.wrap.header1}  
  
    stdWrap.space = {$content.headerSpace}  
}  
  
#  
# CType: header  
#  
tt_content.header = COA  
tt_content.header {  
    10 < lib.stdheader  
    10.stdWrap.space >  
  
    20 = TEXT  
    20.field = subheader  
    20.fontTag = {$content.wrap.subheader1}  
}  
  
#  
# CType: bullet  
#  
tt_content.bullets = COA  
tt_content.bullets {  
    10 = < lib.stdheader  
    20 < styles.content.bulletlist_gr  
}
```

сначала определяется lib.stdheader. Это (и должен быть) cObject ! (в этом случае - COA).

Затем `lib.stdheader` копируется в `tt_content.header.10` посредством оператора "`<`". Это значит, что фактически копия `lib.stdheader` создается *во время анализа*.

Но не в случае `tt_content.bullets.10`. Здесь `lib.stdheader` просто указатель, а использоваться `lib.stdheader` как `cObject` будет *во время выполнения*.

Причина копирования `lib.stdheader` в первом случае — необходимость сброса `".stdWrap.space"` внутри `cObject` ("`10.stdWrap.space >`"). Это НЕВОЗМОЖНО сделать во втором случае, когда создается лишь указатель.

ЗАМЕЧАНИЕ:

Если `lib.stdheader` был бы `temp.stdheader`, указатель не будет работать! Это связано с тем фактом, что ссылка на `"temp."` во время выполнения ничего там не обнаружит, так как он освобождается перед тем, как шаблон сохраняется в кеше!

Это справедливо для `"temp."` и `"styles."` (смотрите описание объекта основного уровня).

Варианты перезаписи значений:

Несмотря на то, что в TypoScript невозможно перезаписать значения (через операторы и т.п.), свойства объекта со ссылками объединяются в настройками ссылки.

Пример:

```
page.10 = TEXT
page.10.value = kasper
page.10.case = upper

page.20 = < page.10
page.20.case = lower
page.20.value >
page.20.field = pages
```

Результат этой настройки:

value	kasper
case	lower
field	pages

Помните, что `.value` не очищается (красная строка), это просто объединение двух массивов:

value	kasper
case	upper

case	lower
field	pages

Поэтому красная строка в примере выше бесполезна.

HTML

Объект содержимого "HTML" можно использовать для вывода статического текста или html. stdWrap доступен как для самих cObject, так и для свойства "value". Смотрите примеры.

Свойство:	Тип данных:	Описание:	По умолчанию:
value	HTML /stdWrap	HTML-код.	
stdWrap	->stdWrap	(выполняется после stdWrap для свойства ".value".)	

[tsref:(cObject).HTML]

Пример:

```
10 = HTML
10.value = Текст в верхнем регистре
10.value.case = upper
```

Пример:

```
10 = HTML
10.value.field = bodytext
10.value.br = 1
```

Пример:

```
10 = HTML
10.stdWrap.field = title
10.stdWrap.wrap = <strong>|</strong>
```

TEXT

TEXT очень схож с cObject "HTML". Но stdWrap находится в самом корне объекта. Это нестандартно. Посмотрите примеры.

Свойство:	Тип данных:	Описание:	По умолчанию:
value	value /stdWrap	Текст, заключенный в свойства stdWrap.	
(свойства stdWrap...)			

[tsref:(cObject).TEXT]

Пример:

```
10 = TEXT
10.value = Текст в верхнем регистре
10.case = upper
```

Пример:

```
10 = TEXT
10.field = bodytext
10.br = 1
```

COBJ_ARRAY (COA, COA_INT)

Это cObject, в который можно помещать несколько других cObject, используя цифры для их упорядочивания.

Этот cObject имеет псевдоним COA. Можно его использовать вместо COBJ_ARRAY.

Это объект также может быть создан как COA_INT, в этом случае он работает точно также, как объект USER_INT: выполняется без кеширования! COA_INT предлагает использовать эту возможность не только с объектами USER cObjects, но и с любыми cObject.

Свойство:	Тип данных:	Описание:	По умолчанию:
1,2,3,4...	cObject		
if	->if	Если "if" возвращает ложь COA НЕ выполняется	
wrap	wrap /stdWrap		
stdWrap	->stdWrap		
includeLibs	список ресурсов /stdWrap	<p>(Это свойство используется только для объекта COA_INT!, смотрите Введение.)</p> <p>Список ресурсов через запятую, которые включаются как сценарии PHP (через функцию include_once()) при включении этого сценария.</p> <p>Это возможно сделать, так как любые включаемые файлы известны перед включением сценария. И это не так с обычными PHP_SCRIPT cObject.</p>	

[tsref:(cObject).COA/(cObject).COA_INT/(cObject).COBJ_ARRAY]

Пример:

```
temp.menuTable = COBJ_ARRAY
temp.menuTable {
    10 = HTML
    10.value = <table border=0 cellpadding=0 cellspacing=0>

    20 = HMENU
    20.entryLevel = 0
    20.1 = GMENU
    20.1.NO {
        wrap = <tr><td> | </td></tr>
        XY = {$menuXY}
        backColor = {$bgCol}
        20 = TEXT
        20 {
            text.field = title
            fontFile = media/fonts/hatten.ttf
            fontSize = 23
            fontColor = {$menuCol}
            смещение = |*| 5,18 || 25,18
        }
    }

    30 = HTML
    30.value = </table>
}
```

FILE

Этот объект возвращает содержимое файла, установленного в свойстве "file".

В /typo3/sysext/cms/tslib/class.tslib_content.php он определен как функция PHP fileResource().

Свойство:	Тип данных:	Описание:	По умолчанию:
file	resource /stdWrap	Если ресурс является изображением jpg,gif,jpeg,png , то изображение вставляется с тегом image. Файлы остальных форматов читаются и вставляются в код HTML. Максимальный размер читаемого файла внутренне установлен на 1024 кБайт!	
linkWrap	linkWrap /stdWrap	(Выполняется перед ".wrap" и ".stdWrap".)	
wrap	wrap /StdWrap	(Выполняется после ".linkWrap" но перед ".stdWrap".)	
stdWrap	->stdWrap	(Выполняется после ".linkWrap" и ".wrap".)	
altText titleText	string /stdWrap	Только для вывода ! Если не определен titltext, будет использоваться Если не определен alttext, будет использоваться пустой alttext	
emptyTitleHandling	string /stdWrap	Значение может быть "keepEmpty" для сохранения пустого атрибута title, или "useAlt" для использования содержимого атрибута alt.	useAlt
longdescURL	string /stdWrap	Только для вывода ! Атрибут "longdesc" (URL указывающий на документ с дополнительными деталями об изображении).	

[tsref:(cObject).FILE]

Пример:

Это пример определения страницы, но содержимое между тегами body берется из файла "gs.html":

```
page.10 = FILE
page.10.file = fileadmin/gs/gs.html
```

IMAGE

Возвращает тег image с файлом, указанным в свойстве "file", и набором параметров, указанных через соответствующие свойства.

В /typo3/sysext/cms/tslib/class.tslib_content.php он определен как функция PHP `clImage()`.

Массив `$GLOBALS["TSFE"]->lastImageInfo` устанавливается, как массив информации о возвращаемом изображении (если имеется) и содержит ширину, высоту и т.п.

Свойство:	Тип данных:	Описание:	По умолчанию:
file	imgResource		
params	- параметры / stdWrap		
border	integer	Значение атрибута "border" тега image.	0

Свойство:	Тип данных:	Описание:	По умолчанию:
altText titleText (alttext)	string /stdWrap	Если не определен alttext, будет использоваться пустой alttext ("alttext" старая проверка правописания этого атрибута. Она будет использовать только если "altText" не определен в значениях или свойствах)	
emptyTitleHandling	string /stdWrap	Значение может быть "keepEmpty" для сохранения пустого атрибута title, или "useAlt" для использования содержимого атрибута alt.	useAlt
longdescURL	string /stdWrap	Атрибут "longdesc" (URL указывающий на документ с дополнительными деталями об изображении).	
linkWrap	linkWrap /stdWrap	(перед ".wrap")	
imageLinkWrap	boolean/ - >imageLinkW rap	ЗАМЕЧАНИЕ: используется ТОЛЬКО если НЕ установлен linkWrap, а файл НЕ GIBUILDER (так как он работает лишь с оригинальными файлами изображений)	
if	->if	Если "if" возвращает ложь, изображение не отображается!	
wrap	wrap /stdWrap		
stdWrap	->stdWrap		

[tsref:(cObject).IMAGE]

Пример:

```
10 = IMAGE
10.file = toplogo*.gif
10.params = hspace=5
10.wrap = |<BR>
```

IMG_RESOURCE

Возвращается лишь ссылка на изображение, возможно заключенная в stdWrap. Может использоваться для фоновых изображений для таблиц или строк таблиц, либо для импорта изображения в свои включаемые сценарии.

Массив \$GLOBALS["TSFE"]->lastImgResourceInfo устанавливается, как массив информации о возвращаемом изображении (если имеется) и содержит ширину, высоту и т.п.

Свойство:	Тип данных:	Описание:	По умолчанию:
file	imgResource		
stdWrap	->stdWrap		

[tsref:(cObject).IMG_RESOURCE]

CLEARGIF

Вставляется прозрачный файл gif.

Свойство:	Тип данных:	Описание:	По умолчанию:
height	- data:height /stdWrap	Высота изображения.	1
width	- data:width /stdWrap	Ширина изображения.	1
wrap	wrap /stdWrap		
stdWrap	->stdWrap	(Выполняется после ".wrap".)	

[tsref:(cObject).CLEARGIF]

Пример:

```
20 = CLEARGIF
20.height = 20
```

CONTENT

Этот объект разработан для формирования содержимого, делая возможным отбор и обработку записей.

Ключ регистра SYS_LASTCHANGED обновляется на основании поля tstamp выбранных записей, имеющих более высокое по сравнению с текущим значение.

Свойство:	Тип данных:	Описание:	По умолчанию:
select	->select	Здесь устанавливается SQL-оператор!	
table	TableName /stdWrap	Таблица, из которой должно браться содержимое. В стандартной конфигурации – это "tt_content" ЗАМЕЧАНИЕ: позволены лишь таблицы "pages" или таблицы с одним из префиксов: "tt_", "tx_", "txx_", "fe_", "user_" или "static_".	
renderObj	cObject		< [tablename]

Свойство:	Тип данных:	Описание:	По умолчанию:
slide	integer /stdWrap	<p>Если установлено, и элементы содержимого не найдены через команду select, то будет исследоваться корневая линия пока не будет найдено содержимое.</p> <p>Возможные значения “-1” (переход обратно до корня сайта), “1” (только на текущем уровне) и “2” (вверх на один уровень).</p> <p>Используйте -1 в комбинации с collect.</p> <p>.collect (integer): если включено, будут собраны все элементы содержимого, найденные на текущей или родительских страницах. Иначе поиск остановился бы после первого найденного значения. Установите в качестве значения нужное число уровней для сбора элементов или “-1” для сбора всех элементов до корня сайта.</p> <p>.collectFuzzy (boolean): полезно лишь для режима collect. Если никаких элементов на определенной глубине не найдено в режиме collect, поискдвигается далее, пока не будет найдена хотя бы одно соответствие.</p> <p>.collectReverse (boolean): изменяет порядок элементов режима collect. Если включено, элементы текущей страницы будут отображены внизу.</p>	
wrap	wrap /stdWrap	Обертка всего содержимого полностью...	
stdWrap	->stdWrap	(Выполняется после ".wrap".)	

[tsref:(cObject).CONTENT]

Пример (объект CONTENT):

```
1 = CONTENT
1.table = tt_content
1.select {
    pidInList = this
    orderBy = sorting
}
```

Пример (записи renderObj):

```
// Настройка для записей со значением typeField (часто "CType") установленным в "header"
tt_content.header.default {
    10 = TEXT
    10.field = header
    ....
}

// Настройка для записей со значением typeField (часто "CType") установленным в "bullets"
// Если в поле "layout" установлено "1" или "2" используется специальная настройка, иначе – по
умолчанию
tt_content.bullets.subTypeField = layout
tt_content.bullets.default {
    ....
}
tt_content.bullets.1 {
    ....
}
tt_content.bullets.2 {
    ....
}

// А это происходит, если значение typeField не соответствует ничему упомянутому выше
tt_content.default.default {
    ....
}
```

RECORDS

Этот объект предназначен для вывода списков записей из различных таблиц. В отличие от объекта CONTENT, здесь невозможен уточненный отбор записей (отсутствует свойство "select").

Ключ регистра SYS_LASTCHANGED обновляется на основании поля tstamp выбранных записей, имеющих более высокое по сравнению с текущим значение.

ЗАМЕЧАНИЕ: записи с родительскими id (pid) недоступных страниц (скрытых, с ограниченным доступом или ограниченными по времени) обычно не выбираются. Страницы могут иметь любой тип, за исключением корзины. Отключение проверки производится через параметр "dontCheckPid".

Свойство:	Тип данных:	Описание:	По умолчанию:
source	список записей /stdWrap	Список id записей, возможно с префиксом названий таблиц. Пример: <code>source = tt_content_34, 45, tt_links_56</code>	

Свойство:	Тип данных:	Описание:	По умолчанию:
tables	<i>list of tables</i> /stdWrap	Список воспринимаемых таблиц. Если любой элемент в списке ".source" не имеет префикса с названием таблицы, то подразумевается первая таблица в этом списке. Настройка названий таблиц возможно и в .conf. Пример: <code>tables = tt_content, tt_address, tt_links conf.tx_myexttable = TEXT conf.tx_myexttable.value = Hello world</code> Добавляются таблицы tt_content, tt_address, tt_links, tx_myexttable.	
conf. [tablename]	cObject	Массив настроек представляющий записи из таблицы <i>tablename</i>	Если НЕ определено, то представление записей осуществляется через объект основного уровня [tablename] – то же самое, что cObject CONTENT!
dontCheckPid	boolean /stdWrap	Обычно записи невозможно выбрать, если их родительская страница (pid) недоступна для пользователя сайта. Этот параметр отключает такую проверку.	
wrap	wrap /stdWrap		
stdWrap	->stdWrap	(Выполняется после ".wrap".)	

[tsref:(cObject).RECORDS]

Пример:

```
20 = RECORDS
20.source.field = records
20.tables = tt_address
20.conf.tt_address < tt_address.default
```

HMENU

Формирует иерархическое меню.

Свойство:	Тип данных:	Описание:	По умолчанию:
(1 / 2 / 3 /...)	menuObj	Требуется! Определяет, как menuObj должен представлять элементы меню различных уровней. 1 — первый уровень, 2 — второй уровень, 3 — третий уровень... Пример: <code>temp.sidemenu = HMENU temp.sidemenu.1 = GMENU</code>	(по menu)

Свойство:	Тип данных:	Описание:	По умолчанию:
cache_period	int	Число секунд, в течение которых меню может оставаться в кеше. Если значение не определено, будет использовано первое доступное значение из следующих: 1) cache_timeout текущей страницы; 2) config.cache_period определенный глобально; 3) 86400 (= 1 день)	
entryLevel	int /stdWrap	Определяет, с какого уровня корневой линии должно начинаться меню. По умолчанию — "0", что означает самую первую страницу сайта. Если значение < 0, entryLevel выбирается "вне" корневой линии. Так, "-1" — меню с элементами из внешнего уровня, "-2" — на 2 уровня дальше вовне...	0
special	"directory" / "list" / "updated" / "browse" / "rootline" / "keywords" / "language"	<i>Обратитесь к разделу "Свойство .special" и соответствующим таблицам.</i>	
special.value	список uid страниц /stdWrap	Смотрите выше	
minItems	int	Минимальное количество элементов в меню. Если количество страниц меньше этого числа, добавляется фиктивная страница с заголовком "..." и uid=[текущая страница_id]. Примечание: затрагивает и подменю. Для установки индивидуального значения для каждого уровня меню, установите свойства в объектах меню (смотрите таблицу "Общие свойства").	
maxItems	int	Максимальное количество элементов меню. Избыточные элементы игнорируются. Примечание: затрагивает и подменю. (смотрите примечания для "minItems").	
begin	int +calc	Первый элемент в меню. Пример: В результате получается меню, в котором пропущены два первых элемента, а меню начинается с третьего: begin = 3 Примечание: затрагивает и подменю. (смотрите примечания для "minItems").	

Свойство:	Тип данных:	Описание:	По умолчанию:
excludeUidList	список целых чисел /stdWrap	<p>Список uid страниц через запятую, исключаемых при построении меню. Можно добавить "current", чтобы текущая страница исключалась из меню.</p> <p>Пример: Страницы с этими uid НЕ появятся в меню!!! В дополнение, всегда исключается текущая страница. <code>excludeUidList = 34,2,current</code></p>	
excludeDoktypes	список целых чисел	Список типов страниц (doktype), исключаемых из меню. По умолчанию — это "не в меню" (5) и помеченных, только для доступа внутренних пользователей (6).	5,6
includeNotInMenu	boolean	Если включено, страницы типа "не в меню", будут включены в меню.	
alwaysActivePIdList	список целых чисел /stdWrap	Список UID страниц, которые всегда будут расцениваться, как активные пункты меню и автоматически открываться не зависимо от корневой линии.	
protectLvar	boolean / keyword	<p>Если установлено, то каждая страница в меню будет проверена на наличие записи с альтернативным языком, для языка, определенного в "config.sys_language_uid" (обычно определенной через &L). Если такого нет, и в "Настройках локализации" страниц установлен параметр "спрятать страницу, если нет перевода для текущего языка", то элемент меню обычно связан с недоступной страницей, при выборе которой появляется сообщение об ошибке. Установка этого параметра предотвратит эту ситуацию, простым добавлением "&L=0" для таких страниц. Это значит, что они переключаться на язык по умолчанию, вместо использования выбранного языка.</p> <p>Проверка выполняется лишь при запросе перевода ("config.sys_language_uid" не ноль).</p> <p>Ключевое слово: "all" Если установлено "all", то выполняется та же проверка, но поиск не будет осуществляться, если установлено "спрятать страницу, если нет перевода для текущего языка", страница всегда возвращается на язык по умолчанию, если перевод не найден.</p> <p>Эти параметры имеет смысл использовать, когда "config.sys_language_mode" не установлено на "content_fallback".</p>	
addQueryString	string	<p><i>смотрите typolink.addQueryString</i></p> <p>Примечание: работает только с <code>special=language</code>.</p>	
if	->if	Если "if" возвращает ложь, меню не формируется	

Свойство:	Тип данных:	Описание:	По умолчанию:
wrap	wrap /stdWrap		
stdWrap	->stdWrap	(Выполняется после ".wrap".)	

[tsref:(cObject).HMENU]

Пример:

```
temp.sidemenu = HMENU
temp.sidemenu.entryLevel = 1
temp.sidemenu.1 = TMENU
temp.sidemenu.1 {
    target = page
    NO.afterImg = media/bullets/dots2.gif /*||*/_
    NO.afterImgTagParams = hspace="4"
    NO.linkWrap = {$fontTag}
    NO.ATagBeforeWrap = 1

    ACT < .NO
    ACT = 1
    ACT.linkWrap = <b>{$fontTag}</b>
}
```

Свойство .special

Это свойство делает возможность создания меню, которые не отображают напрямую текущую структуру страниц, а создают меню в виде ссылок на страницы, типа "следующая/предыдущая", "последняя измененная", "страницы на определенной странице" и т.п.

Замечание: .не устанавливает .entryLevel для HMENU при использовании этого параметра! Существует лишь одно исключение — special.keywords.

Кроме того, удостоверьтесь, что выбираются страницы для первого уровня меню. Подменю в menuProjects 2+ будут создаваться как обычно.

special.directory

HMENU типа special = directory позволяет создавать меню с подстраницами одной или нескольких родительских страниц. Родительские страницы определяются свойством ".value". Обычно используется для карт сайта.

Поддерживаются точки доступа.

Свойство:	Тип данных:	Описание:	По умолчанию:
value	список id страниц /stdWrap	Будет сформировано меню всех страниц с pid = 35 и pid = 56. 20 = HMENU 20.special = directory 20.special.value = 35, 56	id текущей страницы

[tsref:(cObject).HMENU.special.directory]

special.list

HMENU типа special = list позволяет создавать меню, со списком страниц, определенным в свойстве ".value".

Поддерживаются точки доступа.

Свойство:	Тип данных:	Описание:	По умолчанию:
value	список id страниц /stdWrap	<p>Будет сформировано меню из двух страниц (uid=35 и uid=36):</p> <pre>20 = HMENU 20.special = list 20.special.value = 35, 56</pre> <p>Если .value не установлено, то используется uid 0, и будет показана лишь домашняя страница.</p>	0

[tsref:(cObject).HMENU.special.list]

special.updated

HMENU со свойством special = updated позволяет создать меню наиболее часто обновляемых страниц.

Замечания по формированию: порядок сортировки по умолчанию обратный (desc) по полю, определенному через "mode", но установка "alternativeSortingField" для объекта меню (например GMENU, смотрите позже) переназначит его.

Поддерживаются точки доступа.

Свойство:	Тип данных:	Описание:	По умолчанию:
value	список id страниц /stdWrap	<p>Будет сформировано меню наиболее часто обновляемых страниц из ветвей дерева, начиная с заданных uid (uid=35 и uid=36).</p> <pre>20 = HMENU 20.special = updated 20.special.value = 35, 56</pre>	
mode	string	<p>Поле в базе данных, содержащее информацию о последних обновлениях.</p> <p>Возможны следующие значения:</p> <ul style="list-style-type: none"> - SYS_LASTCHANGED: обновляется на последнее время обновления содержимого страницы. - crdate: используется поле "crdate" из записи страницы. - tstamp: используется поле "crdate" из записи страницы, оно автоматически обновляется при сохранении страницы. - manual или lastUpdated: использование поля "lastUpdated", которое в записи страницы можно установить вручную. - starttime: использование поля starttime. <p>Поля с пустыми значениями обычно не выбираются.</p>	SYS_LASTCHANGED
depth	int	<p>Определение глубины уровней в дереве страниц. Возможный диапазон 1-20. Глубина 1 означает лишь стартовую страницу, глубина 2 — стартовый id + первый уровень.</p> <p>Примечание: "глубина" сходно с "beginAtLevel".</p>	20

Свойство:	Тип данных:	Описание:	По умолчанию:
beginAtLevel	int	Определение начального уровня в дереве страниц, сформированном на основе .value и .depth. 0 по умолчанию, включает стартовый id. 1 начиная с первого уровня подстраниц, 2 начиная со второго уровня подстраниц. Примечание: "глубина" сходна с этим свойством.	0
maxAge	int (seconds) +calc	Страницы с датами обновления старше текущего времени минус это число секунд не будут отображаться в меню. По умолчанию "не используется". Для вычислений можно использовать +-*!/.	
limit	int	Максимальное количество элементов в меню. По умолчанию - 10, максимум - 100.	10
excludeNoSearchPages	boolean	Если установлено, страницы, обозначенные "Вне поиска" не включаются в специальное меню.	0

[tsref:(cObject).HMENU.special.updated]

Пример for special = updated:

The following example will generate a menu of the most recently updated pages from the branches in the tree starting with the uid's (uid=35 and uid=56) listed. Furthermore the field "tstamp" is used (default is SYS_LASTCHANGED) and the tree depth is 2 levels. Also a maximum of 8 pages will be shown and they must have been updated within the last three days (3600*24*3):

```
20 = HMENU
20.special = updated
20.special.value = 35, 56
20.special {
    mode = tstamp
    depth = 2
    maxAge = 3600*24*3
    limit = 8
}
```

special.rootline

Меню страниц корневой линии — это меню, показывающее череду выстроенных в иерархическом порядке страниц.

HMENU со свойством special = rootline создает меню корневой линии (также известное как "breadcrumbs menu" - "хлебные крошки") и может выглядеть так:

Страница уровня 1 > Страница уровня 2 > Страница уровня 3 > *Текущая страница*

Такое визуальное представление местоположения упрощает ориентацию пользователей на сайте и ускоряет переход на различные уровни страниц.

Поддерживаются точки доступа.

Свойство:	Тип данных:	Описание:	По умолчанию:
range	string /stdWrap	[begin-level] [end-level] (также, как определялся .entryLevel для HMENU). В следующем примере начальный уровень 1 и не отображается текущая страница: temp.breadcrumbs = HMENU temp.breadcrumbs.special = rootline temp.breadcrumbs.special.range = 1 -2	
reverseOrder	boolean	Если установлено, то порядок элементов меню корневой линии будет обратным.	false
targets.[level number]	string	Для набора фреймов. Возможно определить цель по умолчанию и цель для каждого уровня, используя номер уровня в качестве дополнительного свойства. Пример: Здесь ссылки на страницы уровня 3 в качестве цели используют target="page", в то время, как остальные уровни используют цель target="_top" посредством свойства TMENU .target. page.2 = HMENU page.2.special = rootline page.2.special.range = 1 -2 page.2.special.targets.3 = page page.2.1 = TMENU page.2.1.target = _top page.2.1.wrap = <HR> <HR> page.2.1.NO { linkWrap = > }	

[tsref:(cObject).HMENU.special.rootline]

Пример для special = rootline:

Следующий пример формирует доступное меню корневой линии в виде неупорядоченного списка. Первая страница меню — страница уровня 1, то есть уровнем ниже корневой страницы сайта. Последняя страница меню — текущая.

Здесь после каждой ссылки вставляется изображение, в виде небольшой стрелки.

Текущая страница не имеет ссылки, но помещена в теги. К ней не добавляется изображение.

```
20 = HMENU
20.wrap = <ul>|</ul>
20.special = rootline
20.special.range = 1|-1

20 {
    1 = TMENU
    1.noBlur = 1

    1.NO.wrapItemAndSub = <li>|</li>
    1.NO.ATagTitle.field = description // subtitle // title
    1.NO.afterImg = fileadmin/arrow.jpg

    1.CUR = 1
    1.CUR < .1.NO
    1.CUR.doNotLinkIt = 1
    1.CUR.wrapItemAndSub = <li><em>|</em></li>
    1.CUR.afterImg >
}
```

special.browse

Внимание: страницы крепления не поддерживаются!

Это меню состоит из страниц, позволяя пользователю перемещаться на следующую / предыдущую страницу, страницу с содержимым и т.п. Меню строится из элементов, заданных в свойстве ".items".

Свойство:	Тип данных:	Описание:	По умолчанию:
value	int /stdWrap	По умолчанию – id текущей страницы. Конечно возможно перезаписать это значение другим uid страницы, которая будет взята за базу для меню и предопределенных элементов.	id текущей страницы
items	список названий элементов, разделенных " "	<p>Каждый элемент в списке (разделитель " ") – либо зарезервированное название элемента (смотрите список) с предопределенной функцией, либо определенное название, которое можно ассоциировать с ссылкой на любую страницу. Помните, что текущая страница не может быть корневой страницей сайта.</p> <p><i>Зарезервированные названия элементов:</i></p> <p>next / prev: ссылки на следующую / предыдущую страницы. Следующая и предыдущая страницы от того же "pid", что и текущий id страницы (или "value") – то есть следующий элемент меню текущей страницы. Также говорят как о текущем уровне.</p> <p>Если установлено ".prevnextToSection", то next/prev будут связаны с первой страницей следующего раздела / последней страницей предыдущего раздела.</p> <p>nextsection / prevsection: ссылки на следующий / предыдущий раздел. Раздел определяется как подстраницы страниц того же уровня, что и родительская (pid) страница текущей страницы. Не сработает, если родительская страница текущей страницы — корневая страница сайта.</p> <p>nextsection_last / prevsection_last: здесь ссылки nextsection/prevsection связаны с первыми страницами разделов, а эти — с последними. Если в разделе лишь одна страница, она будет и первой, и последней. Не сработает, если родительская страница текущей страницы — корневая страница сайта.</p> <p>first / last: первая / последняя страница текущего уровня. Если в разделе лишь одна страница, она будет и первой, и последней.</p> <p>up: ссылка на родительскую страницу (pid) для текущей страницы. (вверх на 1 уровень) Всегда доступна.</p> <p>index: ссылки на родительскую страницу для родительской страницы текущей страницы (вверх на 2 уровня). Может быть недоступна, если страница находится вне диапазона корневой линии.</p> <p>Примеры:</p> 	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>Если id=20 — текущая страница, то: 21= prev и first, 19 = next, 18 = last, 17 = up, 1=index, 10 = nextsection, 11 = nextsection_last</p> <p>prevsection и prevsection_last не представлены, так как id=3 не имеет подстраниц!</p> <p>TypoScript (лишь часть "browse", необходимо еще TMENU/GMENU):</p> <pre>xxx = HMENU xxx.special = browse xxx.special { items = index up next prev items.prevnextToSection = 1 index.target = _blank index.fields.title = INDEX index.uid = 8 }</pre>	
items.prevnextToSection	boolean	Если установлено, "prev" и "next" будут перемещать на следующий раздел при достижении конечной страницы текущего раздела. Таким образом "prev" и "next" будут выступать также ссылкой на первую страницу следующего раздела / последнюю страницу предыдущего раздела.	
[itemname].target	string	Опциональная/альтернативная цель для элементов.	
[itemname].uid	int	(uid страницы) – опциональная/альтернативная ссылка на uid страницы.	
[itemname].fields. [field name]	string	Переопределение поля "fieldname" в записи страницы.	
		<p>Пример: так ссылка на предыдущую страницу получает текст "« назад".</p> <pre>prev.fields.title = « назад</pre>	

[tsref:(cObject).HMENU.special/browse]

special.keywords

Формирование меню страниц, с одним или несколькими ключевыми словами, найденными на текущей странице.

По умолчанию порядок реверсивный (desc) по полю, определенному в "mode", но установка "alternativeSortingField" для объекта меню (например, для GMENU,смотрите далее) будет иметь преимущество.

Поддерживаются точки доступа.

Свойство:	Тип данных:	Описание:	По умолчанию:
value	int /stdWrap	<p>Страница с ключевыми словами для поиска схожих страниц.</p> <p>Пример:</p> <pre>20 = HMENU 20.special = keywords 20.special {</pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<pre>value.data = TSFE:id entryLevel = 1 mode = manual } 20.1 = TMENU 20.1.NO { ... }</pre>	
mode	string	<p>Поле с таблице страниц используемое для сортировки.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> SYS_LASTCHANGED: обновляется на последнее время обновления содержимого страницы. - manual или lastUpdated: использование поля "lastUpdated", которое в записи страницы можно установить вручную. - tstamp: используется поле "crdate" из записи страницы, оно автоматически обновляется при сохранении страницы. - crdate: используется поле "crdate" из записи страницы. - starttime: использование поля starttime. 	SYS_LASTCHANGED
entryLevel	int	Место начала поиска в корневой линии. <i>Смотрите свойство entryLevel в разделе "HMENU" выше.</i>	
depth	int	(также, как в разделе "special.updated")	20
limit	int	(также, как в разделе "special.updated")	10
excludeNoSearchPages	boolean	(также, как в разделе "special.updated")	
begin	boolean	(также, как в разделе "special.updated")	
setKeywords	string /stdWrap	Позволяет вручную определить ключевые слова в виде списка слов через запятую. Если определено это свойство, оно переопределяет настройки по умолчанию, использующие ключевые слова текущей страницы.	
keywordsField	string	Определение полей таблицы страниц, в которых будет производиться поиск ключевых слов. По умолчанию поле "keyword". Проверка существования поля не производится, поэтому вводите существующее поле. Хорошо?!	keywords
keywordsField.sourceField	string	Определение поля текущей страницы, из которого будут браться ключевые слова. По умолчанию поле "keyword" (помните, что ".keywordsField" только установка поля записи страницы для поиска в нем!).	keywords

[tsref:(cObject).HMENU.special.keywords]

special.language

Формирование меню выбора языка. Обычно делается в виде меню с флагами, представляющих языки, на которые переведена текущая страница. При щелчке по любому из элементов текущая страница обновляется, но с измененным параметром "&L" в URL.

Тип “language” формирует элементы меню на основе записи текущей страницы, но с языковой записью для каждого доступного языка. Все элементы ссылаются на текущий id страницы, но с разными параметрами “&L”.

Замечания по состояниям элементов:

Когда “TSFE->sys_language_uid” соответствует sys_language uid для элемента, устанавливается состояние “ACT”, в других случаях - “NO”. Однако, если страница недоступна ввиду “настроек локализации” для нее (они могут быть отключены переводчиком), или если запись Альтернативного языка страницы не найдена (может быть отключена через “.normalWhenNoLanguage”, см. далее), устанавливается состояние “USERDEF1” для неактивных элементов, и “USERDEF2” для активных. Итого — четыре состояния для разработки дизайна. Рекомендуется отключать ссылки для элементов меню в состояниях “USERDEF1” и “USERDEF2”, при этом они отключаются также, как если бы страница на этих языках не существовала и попытка вывода таких страниц приведет к ошибке (в зависимости от настроек сайта).

Свойство:	Тип данных:	Описание:	По умолчанию:
value	список sys_language uid через запятую /stdWrap	Количество элементов в этом списке определяет количество элементов меню.	
normalWhenNoLanguage	boolean	Если установлена 1, кнопка для языка будет сформирована не как не отключенная, даже если перевода на данный язык не найдено.	

[tsref:(cObject).HMENU.special.language]

Пример:

Формирование языкового меню с флагами (помните, что некоторые строки перенесены):



```

lib.langMenu = HMENU
lib.langMenu.special = language
lib.langMenu.special.value = 0,1,2
lib.langMenu.1 = GMENU
lib.langMenu.1.NO {
    XY = [5.w]+4, [5.h]+4
    backColor = white
    5 = IMAGE
    5.file = typo3/sysext/cms/tslib/media/flags/flag_uk.gif ||
    typo3/sysext/cms/tslib/media/flags/flag_fr.gif ||
    typo3/sysext/cms/tslib/media/flags/flag_es.gif
    5.смещение = 2,2
}

lib.langMenu.1.ACT < lib.langMenu.1.NO
lib.langMenu.1.ACT = 1
lib.langMenu.1.ACT.backColor = black

lib.langMenu.1.USERDEF1 < lib.langMenu.1.NO
lib.langMenu.1.USERDEF1 = 1
lib.langMenu.1.USERDEF1.5.file = typo3/sysext/cms/tslib/media/flags/flag_uk_d.gif ||
    typo3/sysext/cms/tslib/media/flags/flag_fr_d.gif ||
    typo3/sysext/cms/tslib/media/flags/flag_es_d.gif
lib.langMenu.1.USERDEF1.noLink = 1

```

special.userdefined

Позволяет написать свой маленький сценарий PHP, формирующий массив элементов

меню.

Свойство:	Тип данных:	Описание:	По умолчанию:
file	resource	Название включаемого php файла (также, как cObject PHP_SCRIPT).	
[any other key]		Переменные в вашем сценарии. Все они доступны в массиве \$conf вашего сценария.	

[tsref:(cObject).HMENU.special.userdefined]

Руководство:

Нужно заполнить массив \$menuItemsArray записями страниц для элементов меню.

Это делается так::

```
$menuItemsArray[] = pageRow1;
$menuItemsArray[] = pageRow2;
$menuItemsArray[] = pageRow3;
...
```

“pageRow” - запись из таблицы “pages” со всеми выбранными полями (SELECT * FROM...).

Если нужно изменить название страницы, удостоверьтесь, что добавили хотя бы значения из полей “title” и “uid”.

Замечание:

Если вы работаете с точками доступа, возможно установить параметр MP, который должен быть установлен для страницы, посредством внутреннего поля “_MP_PARAM” в записи страницы (xxx-xxx).

Переназначения URL:

Также возможно использовать внутреннее поле “_OVERRIDE_HREF” для установки своего значения href (например “<http://www.typo3.org>”), которое будет использоваться взамен ссылки на страницу, предоставленную самой страницей. Если используется “_OVERRIDE_HREF”, то может использоваться и “_OVERRIDE_TARGET” для переназначения значения цели (смотрите пример ниже).

Другие зарезервированные ключи:

“_ADD_GETVARS” возможно использовать для добавления параметров get к URL, например “&L=xxx”.

“_SAFE” возможно использовать для защиты элементов от возможной фильтрации.

Создание подменю:

Возможно создавать подменю следующего уровня простым добавлением массива элементов меню во внутреннее поле “_SUB_MENU” (смотрите пример ниже).

Предустановка состояния элементов

Если нужно, чтобы элемент распознавался в режиме элемента SPC, IFSUB, ACT, CUR или USR, то можно установить одно из этих значений в ключе “ITEM_STATE” записи страницы. Этот параметр переназначает естественное поведение.

special.userfunction

Вызов пользовательской функции/метода в классе, который должен (также, как “userdefined” выше) возвращать массив с записями страниц для меню.

Свойство:	Тип данных:	Описание:	По умолчанию:
userFunc	string	Название функции	

[tsref:(cObject).HMENU.special.userfunction]

Пример: создание иерархического меню из определенных ссылок

По умолчанию, объект HMENU был разработан для создания меню страниц в TYPO3. Такие страницы представлены содержимым своих записей для страниц. Обычно поле "title" используется для названия, а поле "uid" — для создания ссылки на страницу в меню.

Но объекты HMENU и sub-menu настолько сильны, что их удобно использовать для создания меню со ссылками, не относящимися к id страниц в TYPO3. Это может быть меню, отражающее структуру расширения, в котором каждая ссылка может быть той же ссылкой на id страницы в TYPO3, но с установкой различных параметров.

Это легко осуществить, используя специальный тип "userdefined" (смотрите таблицу выше), в которую можно передать массив элементов меню, созданный собственным сценарием PHP.

Сначала, этот код создает трехуровневое меню, первые два уровня которого — графические элементы:

```

0: # ****
1: # МЕНЮ СЛЕВА
2: # ****
3: lib.leftmenu.20 = HMENU
4: lib.leftmenu.20.special = userfunction
5: lib.leftmenu.20.special.userFunc = user_3dsplm_pi2->makeMenuArray
6: lib.leftmenu.20.1 = GMENU
7: lib.leftmenu.20.1.NO {
8:   wrap = <tr><td>|</td></tr><tr><td class="bckgdgrey1" height="1"></td></tr>
9:   XY = 163,19
10:  backColor = white
11:  10 = TEXT
12:  10.text.field = title
13:  10.text.case = upper
14:  10.fontColor = red
15:  10.fontFile = fileadmin/fonts/ARIALNB.TTF
16:  10.niceText = 1
17:  10.смещение = 14,12
18:  10.fontSize = 10
19: }
20: lib.leftmenu.20.2 = GMENU
21: lib.leftmenu.20.2.wrap = | <tr><td class="bckgdwhite" height="4"></td></tr><tr><td
class="bckgdgrey1" height="1"></td></tr>
22: lib.leftmenu.20.2.NO {
23:   wrap = <tr><td class="bckgdwhite" height="4"></td></tr><tr><td>|</td></tr>
24:   XY = 163,16
25:   backColor = white
26:   10 = TEXT
27:   10.text.field = title
28:   10.text.case = upper
29:   10.fontColor = #666666
30:   10.fontFile = fileadmin/fonts/ARIALNB.TTF
31:   10.niceText = 1
32:   10.смещение = 14,12
33:   10.fontSize = 11
34: }
35: lib.leftmenu.20.2.R0 < lib.leftmenu.20.2.NO
36: lib.leftmenu.20.2.R0 = 1
37: lib.leftmenu.20.2.R0.backColor = #eeeeee
38: lib.leftmenu.20.2.ACT < lib.leftmenu.20.2.NO
39: lib.leftmenu.20.2.ACT = 1
40: lib.leftmenu.20.2.ACT.10.fontColor = red
41: lib.leftmenu.20.3 = TMENU
42: lib.leftmenu.20.3.NO {
43:   allWrap = <tr><td>|</td></tr>

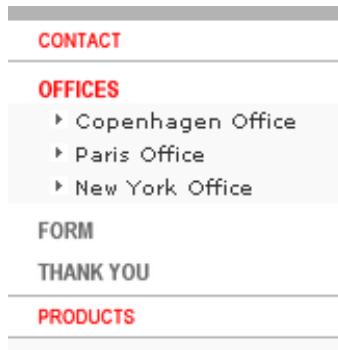
```

```

44:     linkWrap (
45:         <table border="0" cellpadding="0" cellspacing="0" style="margin: 2px; 0px; 2px;
0px;">
46:             <tr>
47:                 <td></td>
48:                 <td></td>
49:             <td>|</td>
50:         </tr>
51:     </table>
52: )
53: }

```

Меню выглядит так, как на этой странице:



Приведенный выше код TypoScript формирует это меню, но элементы элементы не указывают на страницы, как обычно, напрямую. Так как *все* меню формируется из массива, возвращаемого функцией "menuMenuArray" вызываемой в TypoScript в строках 4+5.

```

1:     function makeMenuArray($content, $conf) {
2:         return array(
3:             array(
4:                 'title' => 'Contact',
5:                 '_OVERRIDE_HREF' => 'index.php?id=10',
6:                 '_SUB_MENU' => array(
7:                     array(
8:                         'title' => 'Offices',
9:                         '_OVERRIDE_HREF' => 'index.php?id=11',
10:                        '_OVERRIDE_TARGET' => '_top',
11:                        'ITEM_STATE' => 'ACT',
12:                        '_SUB_MENU' => array(
13:                            array(
14:                                'title' => 'Copenhagen Office',
15:                                '_OVERRIDE_HREF' => 'index.php?id=11&officeId=cph',
16:                            ),
17:                            array(
18:                                'title' => 'Paris Office',
19:                                '_OVERRIDE_HREF' => 'index.php?id=11&officeId=paris',
20:                            ),
21:                            array(
22:                                'title' => 'New York Office',
23:                                '_OVERRIDE_HREF' => 'http://www.newyork-office.com',
24:                                '_OVERRIDE_TARGET' => '_blank',
25:                            )
26:                        ),
27:                    ),
28:                    array(
29:                        'title' => 'Form',
30:                        '_OVERRIDE_HREF' => 'index.php?id=10&cmd=showform',
31:                    ),
32:                    array(
33:                        'title' => 'Thank you',
34:                        '_OVERRIDE_HREF' => 'index.php?id=10&cmd=thankyou',
35:                    ),
36:                ),
37:            ),
38:            array(
39:                'title' => 'Products',
40:                '_OVERRIDE_HREF' => 'index.php?id=14',
41:            )

```

```
42:           );
43: }
```

Обратите внимание, что массив содержит "поддельные" записи страниц, в которых *нет* поля uid, а имеется лишь поля "title" и "_ OVERRIDE_HREF", как необходимые, и несколько полей, как их дополнения.

- Первый уровень с элементами "Contact" и "Products" содержат поля "title" и "_ OVERRIDE_HREF", но "Contact" расширен через массив "_SUB_MENU", содержащий схожий массив элементов.
- Первый элемент второго уровня, "Offices", содержит поле "_ OVERRIDE_TARGET". Далее, элементу устанавливается состояние "ACT", то есть он будет обрабатываться как "активный" элемент (вы должны будете определить это самостоятельно, до формирования меню для реальных страниц). И наконец, имеется еще один подуровень элементов меню.

CTABLE

Создаются стандартные таблицы, в которых возможно определить содержимое различных ячеек.

CTABLE несколько более функционален, чем простой OTABLE. Функционал касается столбцов содержимого и четырех окружающих столбцов, полезных для внесение в них меню.

Свойство:	Тип данных:	Описание:	По умолчанию:
смещение	x,y /stdWrap	Смещение от верхнего левого угла.	0,0
tm	->CARRAY +TDParams /stdWrap	TopMenu Значение по умолчанию для TDParams: valign="top". Для свойства TDParams доступно свойство stdWrap.	
lm	->CARRAY +TDParams /stdWrap	LeftMenu Значение по умолчанию для TDParams: valign="top". Для свойства TDParams доступно свойство stdWrap.	
rm	->CARRAY +TDParams /stdWrap	RightMenu Значение по умолчанию для TDParams: valign="top". Для свойства TDParams доступно свойство stdWrap.	
bm	->CARRAY +TDParams /stdWrap	BottomMenu Значение по умолчанию для TDParams: valign="top". Для свойства TDParams доступно свойство stdWrap.	
c	->CARRAY +TDParams /stdWrap	Content-cell Значение по умолчанию для TDParams: valign="top". Для свойства TDParams доступно свойство stdWrap.	
cMargins	margins /stdWrap	Расстояние вокруг содержимого ячейки "c".	0,0,0,0

Свойство:	Тип данных:	Описание:	По умолчанию:
cWidth	pixels /stdWrap	Ширина ячейки содержимого "c".	
tableParams	<TABLE>-параметры /stdWrap	Атрибуты тега table.	border="0" cellspacing="0" cellpadding="0"
stdWrap	->stdWrap		

[tsref:(cObject).CTABLE]

Пример:

```
page.10 = CTABLE
page.10 {
    смещение = 5, 0
    tableParams = border="0" width="400"
    cWidth = 400
    c.1 = CONTENT
    c.1.table = tt_content
    c.1.select {
        pidInList = this
        orderBy = sorting
    }

    tm.10 < temp.sidemenu
    tm.TDParams = valign=top

    stdWrap.wrap = <div id="mytable">|</div>
}
```

OTABLE

Создается простая таблица. Можно настроить смещение и несколько других параметров тега table.

Свойство:	Тип данных:	Описание:	По умолчанию:
смещение	x,y /stdWrap	Смещение от верхнего левого угла. Примечание: Фактически тип данных - "x,y,r,b,w,h": x,y — смещение от верхнего левого угла; r,b — смещение (отступ) вправо и внизу; w — требуемая ширина поля содержимого; h — требуемая высота поля содержимого. Все измеряется в пикселях.	
1,2,3,4...	cObject		
tableParams	<TABLE>-параметры /stdWrap	Атрибуты тега table.	border="0" cellspacing="0" cellpadding="0"
stdWrap	->stdWrap		

[tsref:(cObject).OTABLE]

Пример:

```
top.100 = OTABLE
top.100.смещение = 310,8
top.100.tableParams = border="1" cellpadding="0" cellspacing="0"
top.100.1 < temp.topmenu
```

COLUMNS

Вставляет таблицу с несколькими столбцами. Размер и стилизация тега table может быть

указана в соответствующих параметрах.

Свойство:	Тип данных:	Описание:	По умолчанию:
tableParams	<TABLE>-параметры /stdWrap	Атрибуты тега table.	border="0" cellspacing="0" cellpadding="0"
TDparams	<TD>-параметры /stdWrap	Атрибуты тегов td.	valign="top"
rows	integer (Диапазон: 2-20) /stdWrap	Количество строк в столбце.	2
totalWidth	integer /stdWrap	Общая ширина колонка+зазоры.	
gapWidth	integer /stdWrap +optionSplit	Ширина зазоров между столбцами. 0 = без зазоров.	
gapBgCol	HTML-цвет /stdWrap +optionSplit	Цвет фона для зазоров ячеек таблицы.	
gapLineThickness	integer /stdWrap +optionSplit	Толщина разделительных линий между ячейками. 0 = линии отсутствуют.	
gapLineColor	HTML-цвет /stdWrap +optionSplit	Line color of the divider line.	black
[column-number] 1,2,3,4...	cObject	Цвет линий.	
after	cObject	Это объект содержимого для каждого столбца!!!	
if	->if	Это cObject, помещаемый после столбцов таблицы!!!	
stdWrap	->stdWrap		

[tsref:(cObject).COLUMNS]

HRULER

Объект вставляет тег table, который можно использовать в качестве горизонтального разделителя.

Свойство:	Тип данных:	Описание:	По умолчанию:
lineThickness	integer /stdWrap	Диапазон: 1-50	1
lineColor	HTML-цвет /stdWrap	Цвет линии.	black
spaceLeft	pixels /stdWrap	Зазор перед линией (слева)	
spaceRight	pixels /stdWrap	Зазор после линии (справа)	
tableWidth	string /stdWrap	Ширина линии (атрибут "width" в таблице)	99%
stdWrap	->stdWrap		

[tsref:(cObject).HRULER]

IMGTEXT

Этот объект разработан для выравнивания изображений и текста. Обычно используется для вывода записей текста/изображений из таблицы tt_content.

Изображение(я) помещается в таблицу, а таблица помещается перед, за, слева/справа от текста.

Смотрите примеры в коде.

Свойство:	Тип данных:	Описание:	По умолчанию:
text	->CARRAY /stdWrap	Используется для импорта / формирования содержимого, обтекающего блок изображения.	
textPos	int /stdWrap	<p>Позиционирование текста: bit[0-2]: 000 = по центру, 001 = справа, 010 =слева bit[3-5]: 000 = над, 001 = под, 010 в тексте.</p> <p>0 — По верху: По центру; 1 - По верху: Вправо; 2 - По верху: Влево; 8 - По низу: По центру; 9 - По низу: Вправо; 10 - По низу: Влево; 17 - В тексте: Вправо; 18 - В тексте: Влево; 25 - В тексте: Вправо (без обертки); 26 - В тексте: Влево (без обертки).</p>	
textMargin	pixels /stdWrap	Расстояние от изображения до содержимого.	
textMargin_outOfText	boolean	<p>Если установлено, расстояние textMargin будет добавлено, даже если изображение помещается выше или ниже текста.</p> <p>Этот параметр используется лишь для обратной совместимости, так как это было расценено как ошибка, и на данный момент ликвидирована. Поэтому, если у кого-либо проявляется эта "ошибка", ее можно устранить при помощи этого параметра.</p>	
imgList	список файлов изображений / stdWrap	<p>Список изображений из ".imgPath".</p> <p>Пример: Список изображений импортируется из поля image tt_content. <imglist.field =="" image<="" p=""> </imglist.field></p>	
imgPath	path /stdWrap	Путь к изображениям.	
		Пример: "uploads/pics/"	
imgMax	int /stdWrap	Максимальное количество изображений.	
imgStart	int /stdWrap	Начальный номер изображения ".imgStart".	
imgObjNum	<i>imgObjNum</i> +optionSplit	Определение объекта IMAGE-cObjects из массива "1,2,3,4..." в объекте current, который должен формировать изображения. "current" установлен на название файла изображения.	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>Пример: "imgObjNum = 1 * * 2": Будут обработаны первые два изображения с "1. ..." и последнее изображение с "2. ...", при условии, что ".imgList" содержит 3 изображения.</p>	
1,2,3,4	->IMAGE (cObject)	<p>Формирование изображений. Регистр "IMAGE_NUM" установлен на количество изображений, обрабатываемых при каждой обработке объекта image. Начинается с нуля. Объект image не должен быть типа GIFBUILDER!</p> <p>Важно: "file.import.current = 1" получает название изображения!</p>	
caption	->CARRAY /stdWrap	Заголовок.	
captionAlign	align /stdWrap	Выравнивание заголовка.	default = ".textPos"
captionSplit	boolean	<p>Если установлено, текст заголовка разбивается на символы (или строки) из свойства ".token", и каждый элемент отображается как блок над изображением.</p> <p>.token = (string /stdWrap) символ, по которому разбиваются элементы заголовка (по умолчанию — chr(10))</p> <p>.cObject = cObject, используется для выбора разбиваемого заголовка.</p> <p>.stdWrap = свойства stdWrap, используемые для обработки заголовка.</p>	
altText titleText	string /stdWrap	<p>По умолчанию, если не имеется альтернативы, altText/titleText берется из ->IMAGE cObjects</p> <p>Если не определен alttext, вместо него используется пустой alttext.</p>	
emptyTitleHandling	string /stdWrap	Значение может быть "keepEmpty" для сохранения пустого атрибута title, или "useAlt" для использования содержимого атрибута alt.	useAlt
longdescURL	string /stdWrap	<p>По умолчанию, если не имеется альтернативы, longdescURL берется из ->IMAGE cObjects</p> <p>Атрибут "longdesc" (URL, указывающий на документ с дополнительными сведениями об изображении).</p>	
border	boolean /stdWrap	Если истина, вокруг изображения создается рамка.	
borderCol	HTML-цвет /stdWrap	Цвет рамки, при установленном свойстве ".border".	black
borderThick	pixels /stdWrap	Ширина рамки вокруг изображения.	1
cols	int /stdWrap	Столбцы.	
rows	int /stdWrap	Строки (приоритет выше, чем "cols")	
noRows	boolean /stdWrap	Если установлено, столбцы не разбиваются столбцами таблицы. При этом изображения, если они сильно	

Свойство:	Тип данных:	Описание:	По умолчанию:
		различаются по высоте (обычно ширина одинаковая), выглядят лучше.	
noCols	boolean /stdWrap	Если установлено, столбцы в таблице не делаются. Все изображения помещаются в один ряд, разделяемые clear giffile файлом. Если установлен noRows, noCols будет сброшено. Они не могут использоваться вместе.	
colSpace	int /stdWrap	Промежуток между столбцами.	
rowSpace	int /stdWrap	Промежуток между строками.	
spaceBelowAbove	int /stdWrap	Расстояние в пикселях между содержимым и изображением, если изображение расположено ниже или выше текста (но не в тексте).	
tableStdWrap	->stdWrap	Весь окончательный <table> для блока изображений передается в функцию stdWrap.	
maxW	int /stdWrap	Максимальная ширина таблицы изображений. При этом изображения масштабируются неправильно! Принимается во внимание количество столбцов! ЗАМЕЧАНИЕ: работает ТОЛЬКО ЕСЛИ объект IMAGE - НЕ GIFBUILDER.	
maxWInText	int /stdWrap	Максимальная ширина таблицы изображений, если текст обтекает таблицу изображений (слева или справа). При этом изображения масштабируются неправильно! Принимается во внимание количество столбцов! ЗАМЕЧАНИЕ: работает ТОЛЬКО ЕСЛИ объект IMAGE - НЕ GIFBUILDER.	50% от maxW
equalH	int /stdWrap	Если значение больше нуля, то изображения в строке будут иметь одинаковую высоту. Ширина будет вычислена автоматически. Если общая ширина изображений больше значения "maxW" для таблицы, то высота всех изображений будет уменьшена, так, чтобы все изображения влезали в totalWidth. Помните, что это значение переопределяет свойства "width", "maxH", "maxW", "minW", "minH" формируемых объектов IMAGE. Кроме того, свойство "noRows" тоже будет переопределено, и будет сформирована таблица без столбцов!	
colRelations	string /stdWrap	Это значение определяет отношение изображений в столбцах IMGTEXT. Синтаксис - "[int] : [int] : [int] : ..." для каждого столбца. Если столбцов изображений больше, чем цифр в этом значении, оно игнорируется. Если отношения между двумя из этих чисел более 10, эта функция игнорируется. Это работает только если все изображения уменьшены до их определенных maxW. Пример: Если 6 изображений помещены в три колонки, а их ширина каждого из них достаточна для	

Свойство:	Тип данных:	Описание:	По умолчанию:
		масштабирования, то эти значения масштабируют изображения, например до 100, 200 и 300 пикселей слева направо: 1 : 2 : 3	
image_compression	int /stdWrap	<p>Сжатие изображений:</p> <p>0= По умолчанию;</p> <p>1= Не изменять! (все параметры для объекта image удаляются!!!)</p> <p>(добавляется расширение gif и команда преобразования цвета)</p> <p>10= GIF/256</p> <p>11= GIF/128</p> <p>12= GIF/64</p> <p>13= GIF/32</p> <p>14= GIF/16</p> <p>15= GIF/8</p> <p>(добавляется расширение jpg и команда преобразования качества)</p> <p>20= IM: -quality 100</p> <p>21= IM: -quality 90 <=> Photoshop 60 (JPG/Very High)</p> <p>22= IM: -quality 80 (JPG/High)</p> <p>23= IM: -quality 70</p> <p>24= IM: -quality 60 <=> Photoshop 30 (JPG/Medium)</p> <p>25= IM: -quality 50</p> <p>26= IM: -quality 40 (JPG/Low)</p> <p>27= IM: -quality 30 <=> Photoshop 10</p> <p>28= IM: -quality 20 (JPG/Very Low)</p> <p>(добавляется расширение png и команда преобразования цвета)</p> <p>30= PNG/256</p> <p>31= PNG/128</p> <p>32= PNG/64</p> <p>33= PNG/32</p> <p>34= PNG/16</p> <p>35= PNG/8</p> <p>39= PNG</p> <p>По умолчанию, качество в ImageMagick установлено на 75, что эквивалентно качеству Photoshop 45. Изображения, сжимаются ImageMagick с тем же визуальным качеством, что и в Photoshop, но больше в размере на 50%!!!</p> <p>ЗАМЕЧАНИЕ: работает ТОЛЬКО ЕСЛИ объект IMAGE - НЕ GIFBUILDER.</p>	
image_effects	int /stdWrap	Добавление параметров к командам масштабирования. Эта функция не работает при "image_compression" равном 1!!! 1 => "-rotate 90", 2 => "-rotate 270", 3 => "-rotate 180", 10 => "-colorspace GRAY", 11 => "-sharpen 70", 20 => "-normalize",	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>23 => "-contrast", 25 => "-gamma 1.3", 26 => "-gamma 0.8"</p> <p>ЗАМЕЧАНИЕ: работает ТОЛЬКО ЕСЛИ объект IMAGE - НЕ GIFBUILDER.</p>	
image_frames	Array + .key /stdWrap	<p>Фреймы: .key указывает на используемый фрейм.</p> <p>".image_frames.x" — это свойства маска imgResource ("._m"), переназначающее свойства [imgResource].m для imageObjects. Используется для маскирования изображения во фрейме. Смотрите, как это сделать в настройке по умолчанию и IMGTEXT в таблице static_template.</p> <p>Пример:</p> <pre>1 { mask = media/uploads/darkroom1_mask.jpg bgImg = GIFBUILDER bgImg { XY = 100,100 backColor = {\$bgCol} } bottomImg = GIFBUILDER bottomImg { XY = 100,100 backColor = black } bottomImg_mask = media/uploads/darkroom1_bottom.jpg }</pre> <p>ЗАМЕЧАНИЕ: параметры jpg-качества, посылаемые как обычные ".params" в imgResource отменяются. Кроме того, в результате этой операции всегда получаются файлы jpg или gif!</p> <p>ЗАМЕЧАНИЕ: работает ТОЛЬКО ЕСЛИ объект IMAGE - НЕ GIFBUILDER.</p>	
editIcons	string	(Смотрите stdWrap.editIcons).	
noStretchAndMarginCells	boolean /stdWrap	Если установлено (1), ячейки, используемые для добавления левых и правых отступов не будут добавлены. Вы потеряете возможность установки отступов для объектов, если выберите "in text". Поэтому это не рекомендуется, но может быть полезно по разным причинам.	
stdWrap	->stdWrap		

[tsref:(cObject).IMGTEXT]

Пример:

```
tt_content.textpic.default {
  5 = IMGTEXT
  5 {
    text < tt_content.text.default
    imgList.field = image
    textPos.field = imageorient
    imgPath = uploads/pics/
    imgObjNum = 1
    1 {
      file.import.current = 1
```

```
file.width.field = imagewidth
imageLinkWrap = 1
imageLinkWrap {
    bodyTag = <BODY bgColor=black>
    wrap = <A href="javascript:close();"> | </A>
    width = 800m
    height = 600m
    JSwindow = 1
    JSwindow.newWindow = 1
    JSwindow.expand = 17,20
}
}
maxW = 450
maxWInText = 300
cols.field = imagecols
border.field = imageborder
caption {
    1 = TEXT
    1.field = imagecaption
    1.wrap = <font size="1"> |</font>
    1.wrap2 = {$cBodyTextWrap}
}
borderThick = 2
colSpace = 10
rowSpace = 10
textMargin = 10
}
30 = HTML
30.value = <br>
}
```

CASE

Это очень гибкий объект, результат которого может сильно разниться, в зависимости от заданного ключа. Принцип, сродни конструкции “switch” в PHP.

Свойство “key” должно соответствовать одному из значений, находящемуся в массиве “Array”. Если его там нет, будет использовано свойство “default”. В качестве значений в “Array” можно использовать любые строки, кроме названий других свойств. Поэтому запрещены: “setCurrent”, “key”, “stdWrap” и “if”. И, конечно, “default” имеет специальное значение.

Свойство:	Тип данных:	Описание:	По умолчанию:
setCurrent	string /stdWrap	Установка "текущего" значения.	
key	string /stdWrap	Используется для установки значений, которые сравниваются со значениями в “Array”. В общем это может быть не только простая строка, но динамическое значение, получаемое посредством свойств stdWrap из определенных источников, обычно из поля текущей записи (смотрите приведенный пример).	default
default	cObject	Выводимое по умолчанию для всех значений “key”, не соответствующим никакому значению в “Array”.	
Array...	cObject	Определение выводимого для разных значений.	
stdWrap	->stdWrap	stdWrap для любого объекта, независимо от предоставляемого в значении “key”.	
if	->if	Если “if” возвращает ложь, то результат не возвращается.	

[tsref:(cObject).CASE]

Пример:

В этом примере выбирается формирование двух вариантов содержимого, в зависимости от содержимого поля "layout" — "1" или нет ("default"). Результат в любом случае заключается в "|
". Если поле "header" не установлено ("false") всегда возвращается пустая строка.

```
stuff = CASE
stuff.key.field = layout
stuff.if.isTrue.field = header
stuff.stdWrap.wrap = |<br />

stuff.default = TEXT
stuff.default {
    ...
}
stuff.1 = TEXT
stuff.1 {
    ...
}
```

LOAD_REGISTER

Возможность загрузить значения в массив `$GLOBALS["TSFE"]->register[]`. Ничего не возвращается! Удобно, что некоторые предопределенные настройки (вроде содержимого страницы) могут использоваться в различных местах, но с разными значениями, определенными как значения регистра, которые возможно изменить при формировании страницы.

Свойство:	Тип данных:	Описание:	По умолчанию:
Array... [field name]	string /stdWrap	<p>Пример: (установка "contentWidth", "label" и "head")</p> <pre>page.27 = LOAD_REGISTER page.27 { contentWidth = 500 label.field = header head = some text head.wrap = }</pre>	

[tsref:(cObject).LOAD_REGISTER]

RESTORE_REGISTER

Сброс последних изменений в массиве регистра, установленных через LOAD_REGISTER.

Работает наподобие стека, в котором сохраняется оригинальный регистр при вызове LOAD_REGISTER. После чего вызов RESTORE_REGISTER сObject восстанавливает значения из стека (оригинальные).

RESTORE_REGISTER не имеет свойств.

FORM

Возможность создания форм.

```
textarea: Label | [* = required][field name =] textarea[,cols,rows,"wrap= [eg. "OFF"]"] | [defaultdata] | Special evaluation configuration (see note below)
input: Label | [* = required][field name =] input[,size,max] | [defaultdata] | Special evaluation configuration (see note below)
password: Label | [* = required][field name =] input[,size,max] | [defaultdata]
```

```

file:      Label | [* = required][field name (*1)=] file[,size]
check:     Label | [* = required][field name =]check | [checked=1]
select:    Label | [* = required][field name =]select[,size (int/"auto"),
"m"=multiple] | label [=value] , ...
radio:     Label | [* = required][field name =]radio | label [=value] , ...
hidden:   [[field name =]hidden | value
submit:   Label |[field name =]submit | Caption
reset:    Label |[field name =]reset | Caption
label:    Label | label | Label value
property: [Internal, see below]
-----
```

Предопределенные элементы типа "select" и "radio":

Это пример, где "Brown" — предопределенный элемент из блока выбора:

```
Haircolor: | *haircolor=select| Blue=blue , Red=red , *Brown=brown
```

Можно ввести несколько предопределенных элементов, вводя звездочку перед каждым из них.

Переназначение свойств:

Это может быть сделано со следующими свойствами из приведенной ниже таблицы:

type, locationData, goodMess, badMess, emailMess

синтаксис:

```
| [property] =property | value
```

(*1) (название поля для файлов)

Для файлов, прикрепленных к письмам, нужно использовать следующие названия полей:

```
attachment, attachment1, ... , attachment10
```

Отображение формы:

Необходимо установить свойство "layout". Если не сделать этого, форма не будет сформирована! За информацией обратитесь к примеру и приведенной ниже таблице.

Пример:

```

temp.mailform = FORM
temp.mailform {

  dataArray {
    10.label = Name:
    10.type = name=input

    20.label = Nachricht:
    20.type = nachricht=textarea,40,10

    100.type = submit=submit
    100.value = Absenden
  }
  recipient = info@example.com
  layout = <div class="some-class">###LABEL### ###FIELD###</div>
}
```

Правильный ответный email:

Для писем, закрепленных за адресами email людей, подписанных на получение почты, используйте название поля "email", например:

```
Email: | *email=input |
```

Специальные условия

Префиксом “*” перед названием большинства типа полей можно определить обязательные поля. Проверка осуществляется в JavaScript; при этом форма будет отправлена при заполненном поле.

Альтернативно, можно проверять значение поля регулярным выражением или email адресом для некоторых типов (textarea, password, input).

Это делается через определение “Настройку специальных условий” для типов, вроде части 4 в строке настроек (смотрите приведенный ниже пример).

Типы специальных условий разделяются двоеточием (“:”).

Первая часть определяет **ключевое слово** для условия. Текущий параметр — “EREG” (для регулярного выражения) и “EMAIL” (для условия с email адресами).

Если определен ключ “EREG”, вторая и третья части — сообщение об ошибке и регулярное выражение, соответственно.

Примеры:

```
Ваш адрес: | address=textarea,40,10 | | EREG : Можно вводить лишь символы от A до Z: ^[a-zA-Z]*$  
Ваш email: | *email=input | | EMAIL
```

Свойство:	Тип данных:	Описание:	По умолчанию:
data	string /stdWrap	Это данные, установленные для формы. Смотрите ниже. Вместо перехода строки, можно использовать " ".	
dataArray	[массив элементов формы]	<p>Альтернативный способ установки полей формы. Вместо синтаксиса с разделителями в виде вертикальных черт, поддерживаемых свойством .data, возможно определять элементы в стиле обычных массивов TypoScript.</p> <p>.dataArray добавляется к уже введенным значениям .data. Каждая запись в dataArray — числовая и имеет три главных свойства: <i>label</i>, <i>type</i>, <i>value</i>, а также свойство <i>required</i>. 'label' и 'value' имеют свойство stdWrap. Имеется свойство .valueArray, являющееся альтернативой свойству .value. Это такой же массив, в том же стиле, с нумерованными значениями, имеющими свойства <i>label</i>, <i>value</i> и <i>selected</i>. 'label' имеет свойства stdWrap.</p> <p>Пример:</p> <pre>dataArray { 10.label = Имя: 10.type = name=input 10.value = [Введите имя] 10.required = 1 20.label = Цвет глаз: 20.type = eyecolor=select 20.valueArray { 10.label = голубой 10.value = 1 20.label = красный 20.value = 2 20.selected = 1 } 40.type = submit=submit 40.value = Отправить }</pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>}</p> <p>Это соответствует строке в свойстве .data:</p> <pre>Имя: *name=input [Введите имя] Цвет глаз: eyecolor=select голубой=1, *красный=2 submit=submit Отправить</pre> <p>Зачем нужен этот способ? Хороший вопрос, но использование этого способа имеет огромное преимущество в том, что метки отделены от кода. Кроме того, значительно легче удалять или добавлять новые элементы в форму.</p> <p>Вставка поля email после поля имени делается так:</p> <pre>dataArray { 15.label = Email: 15.type = input 15.value = your@email.com 15.specialEval = EMAIL }</pre> <p>Или перевод формы на датский (настройка config.language на 'dk'):</p> <pre>dataArray { 10.label.lang.dk = Navn: 10.value.lang.dk = [Indtast dit navn] 20.label.lang.dk = Øjenfarve 20.valueArray { 10.label.lang.dk = Blå 20.label.lang.dk = Rød } 40.value.lang.dk = Send }</pre>	
radioWrap	->stdWrap	Обертка меток для радиокнопок.	
radioWrap.accessibilityWrap	wrap /stdWrap	Определяет обертку радио кнопок при включении режима accessibility (см. далее свойство "accessibility").	<fieldset###RADIO_FIELD_ID###><legend>###RADIO_GROUP_LABEL###</legend> </fieldset>
radioInputWrap	->stdWrap	Обертки элемента ввода и метки радиокнопки.	
type	integer, string	<p>Тип (action="" of the form):</p> <p>Integer: расценивается, как страница в TYPO3.</p> <p>String: расценивается как обычный URL (например "formmail.php" или "fe_tce_db.php").</p> <p>Empty: выбирается текущая страница.</p> <p>ЗАМЕЧАНИЕ: если тип – integer/empty форма будет отправлена на странице в TYPO3 и, если эта страница имеет значение для target/no_cache, то оно будет использовано вместо target/no_cache по умолчанию ниже.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>ЗАМЕЧАНИЕ: если установлено значение перенаправления, то цель для перенаправления переназначает цель, установленную в action-url.</p> <p>ЗАМЕЧАНИЕ: может быть переназначена функцией переназначения свойств данных формы (смотрите выше).</p>	
target	target /stdWrap	Цель для формы по умолчанию.	
method	form-method /stdWrap	<p>Пример: GET</p>	POST
no_cache	string /stdWrap	Параметр no_cache по умолчанию.	
noValueInsert	boolean /stdWrap	<p>Значения по умолчанию, которые передаются той же странице (и той же форме, например форме поиска), заново вставляются в форму вместо любых данных по умолчанию, которые могут быть установлены. Это применяется ТОЛЬКО если установлено "no_cache=1"! (страница, взятая из кеша, конечно, может не включать определенные пользователем значения в форме..)</p> <p>Если этот параметр установлен, содержимое всегда будет содержимым по умолчанию.</p>	
compensateFieldWidth	double /stdWrap	Параметр переназначения для значения настройки с тем же названием. Смотрите "CONFIG" выше.	
locationData	boolean / string /stdWrap	<p>Если это значение истина, к форме добавляется скрытое поле с названием "locationData". В это поле загружается значение вроде этого: [id страницы]:[таблица текущей записи]:[id текущей записи]</p> <p>Например, если поле формы вставлено на страницу с uid = "100", как элемент содержимого страницы из таблицы "tt_content" с id "120", то значение должно быть "100:tt_content:120".</p> <p>Значение использует, например cObject SEARCHRESULT. Если здесь обнаружено значение \$GLOBALS["HTTP_POST_VARS"]["locationData"], поиск осуществляется, как будто он выполнялся на этой странице! Это очень полезно, если нужно применить функционал поиска для страницы с полем "stype" установленным на "L1", это значит, что поиск осуществляется на первом уровне корневой линии. Предположим, что имеется специализированная страница поиска. Этой странице известно, из locationData, что поиск выполнялся с другого места сайта.</p> <p>Если "locationData" не только истина, но также установлено на "HTTP_POST_VARS", то значение будет содержать содержимое \$GLOBALS["HTTP_POST_VARS"]["locationData"] вместо истинных данных о местоположении страницы. Это нужно делать для полей поиска, так как здесь содержится начальная точка</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>поиска.</p> <p>ЗАМЕЧАНИЕ: может быть переназначена функцией переназначения свойств данных формы (смотрите выше).</p>	
redirect	string /stdWrap	<p>URL для перенаправления (генерирует скрытое поле "redirect")</p> <p>Integer: расценивается, как страница в TYPO3. String: расценивается как обычный URL. Empty: выбирается текущая страница.</p> <p>ЗАМЕЧАНИЕ: если установлено это значение, то его цель переназначает цель свойства "type".</p>	
recipient	(список строк /stdWrap	Email получателя содержимого письма из формы (генерирует скрытое поле "recipient")	No email
goodMess	string /stdWrap	<p>Сообщение для функции событий формы, при правильно заполненной форме.</p> <p>ЗАМЕЧАНИЕ: может быть переназначена функцией переназначения свойств данных формы (смотрите выше)</p>	No message
badMess	string /stdWrap	<p>Сообщение при отсутствующем обязательном поле. Это сообщение появляется над списком полей.</p> <p>ЗАМЕЧАНИЕ: может быть переназначена функцией переназначения свойств данных формы (смотрите выше).</p>	No message
emailMess	string /stdWrap	<p>Сообщение при неверном значении поля email адреса.</p> <p>ЗАМЕЧАНИЕ: может быть переназначена функцией переназначения свойств данных формы (смотрите выше).</p>	
image	->IMAGE (cObject)	<p>Если это изображение правильное, то кнопка отправки формы формируется в виде этого изображения!!!</p> <p>ЗАМЕЧАНИЕ: CurrentValue устанавливается для метки заголовка перед формированием изображения.</p>	
layout	string	<p>Определяется, как будут расположены друг относительно друга метка и поле.</p> <p>Свойство обязательное, необходимо его устанавливать! Иначе форма не будет сформирована.</p> <p>Пример: Вместо "###FIELD###" подставляются данные поля, а вместо "###LABEL###" - метка.</p> <pre>layout = <tr><td>###FIELD###</td><td>###LABEL###</td></tr></pre> <p>Также можно использовать маркер ###COMMENT###, который ТАКЖЕ является значением метки, но</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		вставляется в .commentWrap свойства stdWrap (см. далее).	
fieldWrap	->stdWrap	Поле: обертки для поля.	
labelWrap	->stdWrap	Метки: обертки для метки.	
commentWrap	->stdWrap	Комментарии: обертки для комментариев, ЕСЛИ используются ###COMMENT###.	
REQ	boolean /stdWrap	Определяет, должно ли требуемые поля быть проверены и отмечены.	
REQ.fieldWrap	->stdWrap	Поле: обертки для требуемых полей	свойство "fieldWrap"
REQ.labelWrap	->stdWrap	Метки: обертки для меток требуемых полей	свойство "labelWrap"
REQ.layout	string /stdWrap	То же, что и "layout" выше, но для требуемых полей	свойство "layout"
COMMENT.layout	string /stdWrap	Альтернативный шаблон для комментариев.	свойство "layout"
CHECK.layout	string /stdWrap	Альтернативный шаблон для флажков.	свойство "layout"
RADIO.layout	string /stdWrap	Альтернативный шаблон для радиокнопок.	свойство "layout"
LABEL.layout	string /stdWrap	Альтернативный шаблон для типов меток.	свойство "layout"
stdWrap	->stdWrap	Обертка для формы (перед добавлением тега form).	
hiddenFields	[массив cObject]	<p>Используется для установки hiddenFields из TS.</p> <p>Пример:</p> <pre>hiddenFields.pid = TEXT hiddenFields.pid.value = 2</pre> <p>Формирование скрытых полей с названием "pid" и значением "2".</p> <p>Доступно дополнительное свойство: stdWrap, see ->stdWrap.</p>	
params	параметры тега элемента формы /stdWrap	<p>Дополнительные параметры для элементов формы.</p> <p>Пример:</p> <pre>params = style="width:200px;" params.textarea = style="width:300px;" params.check =</pre> <p>Устанавливается ширина в 200 пикселей по умолчанию, за исключением блока выбора и текстовых областей, шириной в 300 пикселей.</p> <p>stdWrap доступны для вторичных свойств, например params.tagname.</p>	
wrapFieldName	wrap /stdWrap	Это обертка названий полей, перед тем, как они будут заключены в теги поля формы.	

Свойство:	Тип данных:	Описание:	По умолчанию:
		Пример: Если установлено значение <code>tx_myextension[input][]</code> , то поле с названием "email" будет заключено в это: <code>tx_myextension[input][email]</code> .	
noWrapAttr	boolean /stdWrap	Если это значение истинно, то все атрибуты обертки элементов текстовой области удаляются. Это необходимо для совместимости с XHTML. Атрибуты обертки могут быть отменены для конкретных полей, использованием специального ключевого слова "disabled" в качестве значения атрибута wrap.	
arrayReturnMode	boolean /stdWrap	Если установлено, теги <code><form></code> и содержимое формы будут возвращены как отдельные элементы, включая другие практические значения. Этот режим предназначен для использования в расширениях, где значение, возвращаемое как массив, может быть полезнее.	
accessibility	boolean /stdWrap	Если установлено, то форма будет совместима с рекомендациями доступности (XHTML совместимость). Это включает: <ul style="list-style-type: none"> строки меток будут заключены в <code><label for="formname fieldname-hash"> ... </label></code> Все элементы формы будут иметь атрибут id, в зависимости от названия формы, с добавлением md5-хешированного названия поля. Замечание: в TYPO3 4.0 и позже, CSS Styled Content настроен на формирование доступных форм по умолчанию.	
formName	string /stdWrap	Альтернативное название для данной формы. По умолчанию должно быть уникальным (случайным) хешем. <code><form name=" . . ."></code>	
fieldPrefix	string /stdWrap	Альтернативный префикс названия полей в данной форме. Другими словами, ко всем полям приписывается название формы (либо уникальный хеш, либо название, установленное в свойстве "formName"). Если установлен "0", префикс не используется вообще.	
dontMd5FieldNames	boolean /stdWrap	ID для всех элементов формы используют md5 хеши из названий полей. Установка этого свойства истинным, отменяет это. При этом используются просто названия полей, с префиксом в виде названия формы в качестве ID. Это может быть полезно для стилизации специально названных полей через CSS.	

[tsref:(cObject).FORM]

Пример: авторизация

В случае создания формы регистрации, поддерживаются следующие поля:

- "username" = имя пользователя;
- "userident" = пароль;

- "login_status" = "logout" при выходе, "login" для авторизации.

Если вставить "<!--###USERNAME###-->" где-нибудь в документе, то это будет заменено на имя пользователя, если пользователь зарегистрирован!

Если нужно, чтобы форма регистрации преобразовывалась в форму выхода, нужно использовать условия. Смотрите этот пример TS (из static_template "styles.content (default)":

```
# форма авторизации
styles.content.loginform {
    data = Пользователь:|*username=input || Пароль:|*userident=password
}
[usergroup = *]
styles.content.loginform.data = Пользователь: <!--###USERNAME###--> || |submit=submit| Выход
[global]
```

Пример: почтовая форма

Создается простая почтовая форма (это не TypoScript, а установочный код, который следует вставить непосредственно в поле "основной текст" записи содержимого страницы типа "FORMMAIL":

```
Name: | *replyto_name= input | Enter your name here
Имя: | *replyto_name= input | Введите свое имя
Email: | *replyto_email=input |
Like TV: | tv=check |
|formtype_mail = submit | Отослать!

|html_enabled=hidden | 1
|subject=hidden| Это тема
|recipient_copy=hidden | copy@email.com
|auto_respond_msg=hidden| Здравствуйте / Это – автоматическое уведомление. //Мы получили Ваше
сообщение.
| from_name=hidden | Website XY
| from_email=hidden | noreply@website.com
| organisation=hidden | организация XY
| redirect=hidden | 16
| priority=hidden | 5
| tv=hidden | 0
| tv=hidden | 0
```

- "replyto_name": если поле называется вроде этого, то его значение используется в почтовой программе в качестве имени адресата и не будет показано в содержимом письма. Выберете другое название для поля, вроде the_name, для использования его значения в качестве обычного поля. Помните, что звездочка (*) обозначает обязательное поле. Установлено значение по умолчанию ("Введите свое имя")
- "replyto_email": если поле называется вроде этого, то его значение используется в качестве email адреса для ответа и не будет показано в содержимом письма. Для получения значения в качестве адреса отправителя для почтовой программы, используйте в качестве названия поля "email".
- "Like TV" переключатель. По умолчанию "выключен".
- "formtype_mail" – название кнопки подтверждения. И она **должна** присутствовать, если нужна встроенная почтовая форма TYPO3. Эта переменная заставляет TYPO3 среагировать на ввод значений и расценивать их как содержимое почтовой формы!
- "html_enabled" — формирует почтовое сообщение в виде HTML.
- "use_base64" — вместо обычного текста, сообщение посыпается в зашифрованном виде base64.

- "subject": введите тему своего письма.
- "recipient_copy" : копия письма отсылается по этому адресу. Можно ввести несколько адресов через запятую (,). Письмо, отсылаемое recipient_copy то же самое, но посыпается отдельным письмом, кроме того, письмо-копия посыпается только при посылке письма для "recipient".
- "auto_respond_msg": автоматическое уведомление. Отсылается, если известен email адрес "submitter" (поле: "email"). Его значение — сообщение, разбиваемое на строки слешем "/". Каждый слеш — новая строка в письме. Первая строка используется в качестве темы письма.
- "from_name": в этом параметре можно установить название в заголовке письма from, которое будет видно в почтовой программе.
- "from_email": в этом параметре можно установить email в заголовке письма from, который будет виден в почтовой программе.
- "organisation": в этом параметре можно установить параметры для организации в почтовом заголовке, они не будут показаны в письме, а только в его заголовке.
- "redirect": через этот параметр, можно определить страницу TYPO3 (id страницы) или внешний URL (www.example.com), на которую будет перенаправлен пользователь после подтверждения формы. Если этот параметр не установлен, он будет возвращен снова на форму.
- "priority": через этот параметр, можно установить приоритет письма, от 1 (не важно) до 5 (очень важно). По умолчанию – 3.
- "tv" (снова, но скрыто). Повторение поля — разумный ход, так как значение "tv" обычно НЕ представляется значением "ложь", если оно не выбрано. Добавление этой строки сохраняет значение по умолчанию для "tv".

SEARCHRESULT

Объект может использоваться для отображения результатов поиска.

Искомые слова загружаются в регистр формы, связанной со страницами:

Пример:

```
register:SWORD_PARAMS = '&sword_list[]=word1&sword_list[]=word2 ....'
```

За деталями обратитесь к информации по typolink!

SEARCHRESULT возвращает результаты лишь со страниц типа "Обычные" (1), "Дополнительные" (2) и "Не в меню" (5)

Свойство:	Тип данных:	Описание:	По умолчанию:
allowedCols	string	Список (разделяемый ":") позволенных столбцов таблицы. Пример: <code>pages.title:tt_content.bodytext</code>	

Свойство:	Тип данных:	Описание:	По умолчанию:
layout	string	Определяет, как отображается содержимое поиска. Пример: Заменяет следующие поля: ###RANGELOW###: низший диапазон результаов, напр. "1"; ###RANGEHIGH###: высший диапазон результатов, напр. "10"; ###TOTAL###: всего результатов; ###RESULT###: сам результат; ###NEXT###: кнопка "следующий"; ###PREV###: кнопка "предыдущий".	
next	cObject	Этот cObject будет помещен в ссылку на следующий результат поиска. Этот код замещает метку "###NEXT###".	
prev	cObject	Этот cObject будет помещен в ссылку на предыдущий результат поиска. Этот код замещает метку "###PREV###".	
target	target /stdWrap	Цель для ссылок следующий/предыдущий!	
range	integer /stdWrap	Количество одновременно выводимых результатов!	20
renderObj	cObject	cObject для предоставления результатов поиска. \$cObj->массив данных установлен на запись результатов поиска. Помните, что все поля именуются [tablename]_[fieldnam]. Так, название страницы — это поле "pages_title". Кроме того, также присутствует следующее поле из таблицы страниц: uid	
renderWrap	wrap /stdWrap		
resultObj	cObject	cObject предваряющий строки в результате поиска.	
noResultObj	cObject	cObject, используемый, если в результате поиска нет строк.	
noOrderBy	boolean /stdWrap	Если установлено, результат НЕ сортируется после lastUpdated, tstamp для таблицы страниц.	
wrap	wrap /stdWrap	Контейнер для всего содержимого...	
stdWrap	->stdWrap	Контейнер для всего содержимого...	
addExtUrlsAndShortCuts	boolean	Если установлено, то типы страниц 3 и 4 (внешняя ссылка и ссылка) добавляются к типам документов для поиска. Но, на данный момент страницы не будут выбраны, если на них нет хотя бы одной записи tt_content! Это происходит из-за того, что страницы и таблица tt_content (или другая) объединены. Поэтому, должна быть хотя бы одна запись в tt_content для страниц внешней ссылки / ссылки, чтобы они были отображены.	

Свойство:	Тип данных:	Описание:	По умолчанию:
languageField. [2nd table]	string	Установка названия поля для включения фильтрации по языку. Работает подобно настройке "languageField" в ->select. Пример: <code>languageField.tt_content = sys_language_uid</code>	

[tsref:(cObject).SEARCHRESULT]

ЗАМЕЧАНИЕ: "sword" и "scols" ДОЛЖНЫ быть установлены для участия в поиске.

переменная "sword" = поиск слов(а);

переменная "scols" = поиск в столбцах, разделитель ":". Например:
pages.title:pages.keywords:tt_content.bodytext

переменная "stype" = начальная точка поиска: false = текущая страница, L-2 = страница перед текущей, L-1 = текущая страница, L0 = корневой уровень, L1 = с первого уровня, L2 = со второго уровня;

переменная \$GLOBALS["HTTP_POST_VARS"]["locationData"]: если установлено, поиск производится как бы от другой страницы сайта, заданной здесь в значении "locationData". Смотрите описание cObject "FORMS".

Поиск примет это во внимание, только если locationData указывает во вне фактической корневой линии сайта.

внутреннее:

переменная "scount": если установлено, то используется как searchCount – общее количество столбцов поиска. Этим способом не нужно восстанавливать число!

переменная "spointer": указатель на начальную запись для поиска.

LATER:

переменная "alldomains" : boolean: если установлено, поиск будет осуществляться в других доменах;

переменная "allsites" : boolean: если установлено, поиск будет осуществляться в других сайтах (определяется установкой "root" активного шаблона);

переменная "depth": глубина.

Синтаксис при поиске

При поиске можно использовать три типа операторов:

- AND: "+", "and" (UK), "og" (DK)
- OR: "or" (UK), "eller" (DK)
- NOT: "-", "not" (UK), "uden" (DK)

По умолчанию используется оператор AND. Если заключить фразу в кавычки "", то будет осуществляться поиск целой строки. Поиск не чувствителен к регистру, части слова также ищутся.

Примеры:

1. *menu backend* – поиск страниц со словами и "menu", и "backend".
2. *"menu backend"* – поиск целой фразы "menu backend".
3. *menu or backend* – поиск страниц, либо со словом "menu", либо с "backend".
4. *menu or backend not content* - поиск страниц, либо со словом "menu", либо с "backend", но не с "content".

Примеры запросов

В этом случае "pagecontent" выбран в качестве полей для поиска. Он включает *tt_content.header*, *tt_content.bodytext* и *tt_content.imagecaption*.

Это предваряет такие запросы:

```
SELECT pages.title AS pages_title, pages.subtitle AS pages_subtitle, pages.keywords AS pages_keywords, pages.description AS pages_description, pages.uid, tt_content.header AS tt_content_header, tt_content.bodytext AS tt_content_bodytext, tt_content.imagecaption AS tt_content_imagecaption
FROM pages, tt_content
WHERE(tt_content.pid=pages.uid) AND (pages.uid IN (2,5,6,20,21,22,29,30,31,3,4,8,9,16,1) AND
pages.doktype in (1,2,5) AND pages.no_search=0 AND NOT tt_content.deleted AND NOT
tt_content.hidden AND (tt_contentstarttime<=985792797) AND (tt_contentendtime=0 OR
tt_contentendtime>985792797) AND tt_content.fe_group IN (0,-1) AND NOT pages.deleted AND
NOT pages.hidden AND (pages.starttime<=985792797) AND (pages.endtime=0 OR
pages.endtime>985792797) AND pages.fe_group IN (0,-1)) ...
```

Часть "... pages.uid IN (2,5,6,20,21,22,29,30,31,3,4,8,9,16,1)..." – это список uid страниц, по которым осуществляется поиск. Этот список берется из ветви дерева страниц и ограничивает поиск этой ветвью, вместо всей таблицы страниц.

```
1. ... AND ((tt_content.header LIKE '%menu%' OR tt_content.bodytext LIKE '%menu%' OR
tt_content.imagecaption LIKE '%menu%') AND (tt_content.header LIKE '%backend%' OR
tt_content.bodytext LIKE '%backend%' OR tt_content.imagecaption LIKE '%backend%')) GROUP
BY pages.uid
2. ... AND ((tt_content.header LIKE '%menu backend%' OR tt_content.bodytext LIKE '%menu
backend%' OR tt_content.imagecaption LIKE '%menu backend%')) GROUP BY pages.uid
3. ... AND ((tt_content.header LIKE '%menu%' OR tt_content.bodytext LIKE '%menu%' OR
tt_content.imagecaption LIKE '%menu%') OR (tt_content.header LIKE '%backend%' OR
tt_content.bodytext LIKE '%backend%' OR tt_content.imagecaption LIKE '%backend%')) GROUP
BY pages.uid
4. ... AND ((tt_content.header LIKE '%menu%' OR tt_content.bodytext LIKE '%menu%' OR
tt_content.imagecaption LIKE '%menu%') OR (tt_content.header LIKE '%backend%' OR
tt_content.bodytext LIKE '%backend%' OR tt_content.imagecaption LIKE '%backend%')) AND NOT
(tt_content.header LIKE '%content%' OR tt_content.bodytext LIKE '%content%' OR
tt_content.imagecaption LIKE '%content%')) GROUP BY pages.uid
```

Помните, что регистр здесь не важен. Так по запросу "menu" будет найдено "menu", "menus", "menuitems" и т.д.

USER и USER_INT

Вызывается либо функция PHP, либо метод в классе. Полезно для включения собственной обработки данных или содержимого.

В основном, это определяемый пользователем cObject, так как это просто вызов функции или метода, которым Вы управляете!

Важно знать, что если вызывается метод в классе (который, конечно, является объектом), внутренняя переменная "cObj" этого класса устанавливается с *ссылкой* на родительский cObj. Смотрите файл typo3/sysext/cms/tslib/media/scripts/example_callfunction.php, он может

быть полезен в качестве примера. В основном это API для функций, более или менее понятны для вас. Просмотрите раздел “Включение PHP сценариев” в конце этого документа.

Если этот объект создается как USER_INT, он будет выполняться без кеширования, вне обработки основной страницы.

Свойство:	Тип данных:	Описание:	По умолчанию:
userFunc	название функции	<p>Название функции. Если определить название через ">", это будет расценено как вызов метода в классе. Посыпаются два параметра: переменная содержимого (в этом случае пустая, но не при использовании из функций stdWrap .postUserFunc и .preUserFunc) и второй параметр — массив со свойствами этого cObject, если они есть.</p> <p>Пример: Этот TypoScript отображает все заголовки элементов содержимого на странице в обратном порядке. Обратите внимание на typo3/sysext/cms/tslib/media/scripts/example_callfunction.php!</p> <pre>page = PAGE page.typeNum=0 includeLibs.something = typo3/sysext/cms/tslib/media/scripts/example_callfunction.php page.30 = USER page.30 { userFunc = user_various->listContentRecordsOnPage reverseOrder = 1 }</pre> <p>ЗАМЕЧАНИЕ: при использовании функции, ее название должно начинаться с “user_”. При использовании класса, его название должно начинаться с “user_” (при этом нет ограничений на название методов).</p>	
includeLibs	список ресурсов /stdWrap	<p>Это свойство применимо только если объект создан как USER_INT.</p> <p>Это список ресурсов через запятую, включаемых как сценарии PHP (посредством функции include_once()), при включении этого сценария.</p> <p>Это возможно сделать, так как все включаемые файлы должны быть определены до включения сценариев.</p>	

[tsref:(cObject).USER/(cObject).USER_INT]

TEMPLATE

При помощи этого cObject можно определить шаблон (например, файл HTML), используемый в качестве основы всего сайта. Внутри файла шаблона можно определить маркеры, которые впоследствии будут заменены динамическим содержимым TYPO3.

Свойство:	Тип данных:	Описание:	По умолчанию:
template	cObject	<p>Должен загружаться с кодом шаблона. Если это не так, объект ничего не возвращает.</p> <p>Пример:</p> <pre>page.10 { template = FILE template.file = fileadmin/template.html }</pre> <p>в качестве шаблона для сайта используется файл fileadmin/template.html.</p>	
subparts	Массив... cObject	<p>Массив меток подразделов (чувствителен к регистру). Подчасть в шаблоне определяется двумя маркерами. Метки должны быть заключены в "###" с обоих сторон. Метки нужно вставлять внутрь тегов-комментариев!!!</p> <p>Пример:</p> <p>Подраздел "HELLO" в шаблоне:</p> <pre><!-- start of subpart ###HELLO### --> This is the HTML-code, that will be loaded in the register and will be replaced with the result... <!-- end ###HELLO### --></pre> <p>Теперь следующий код TypoScript заменяет подраздел "HELLO" текстом, заданным в "value":</p> <pre>page.10.subparts { HELLO = TEXT HELLO.value = En subpart er blevet erstattet!! }</pre> <p>ЗАМЕЧАНИЕ:</p> <p>Перед формированием объекта содержимого, все подразделы извлекаются из массива и загружаются в регистр, поэтому возможно позже загрузить из оттуда. Ключ регистра для кода каждой подразделы "SUBPART_[theSubpartkey]".</p> <p>Кроме того, текущее значение загружается с содержимым каждой подразделы как раз перед анализом cObject для подразделы. Поэтому очень просто становиться загрузить подраздел из cObject (например: ".current=1")</p> <p>Например, подраздел выше имеет ключ регистра "SUBPART_HELLO".</p> <p><i>Это справедливо только ЕСЛИ не установлено свойство .nonCachedSubst! (см. далее).</i></p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
relPathPrefix	<i>string / properties</i>	<p>Поиск относительных ссылок (например к изображениям или таблицам стилей) и установка префикса к ним.</p> <p>При определении свойства (в верхнем регистре), оно будет соответствовать тегам HTML и определять для них альтернативные пути. Смотрите пример ниже.</p> <p>Если свойство называется "style", оно устанавливает альтернативный путь для оболочки "url()", которая может быть в разделах <style>.</p> <p>Пример:</p> <pre>page.10 = TEMPLATE page.10 { template = FILE template.file = fileadmin/template.html relPathPrefix = fileadmin/ relPathPrefix.IMG = fileadmin/img/ }</pre> <p>В этом примере ко всем найденным альтернативным путям приписывается "fileadmin/", если же это атрибут тега img, приписывается "fileadmin/img/".</p>	
marks	<i>Массив... cObject</i>	<p>Это массив маркеров-меток (чувствителен к регистру). Метки определяются одним маркером в шаблоне. Маркер заключается в "####" с обоих сторон. В противоположность подразделам, можно НЕ вставлять маркеры подразделов в теги-комментарии HTML! (Они не будут удаляться).</p> <p>Пример:</p> <p>В шаблоне:</p> <pre><div id="copyright"> &copy; ####DATE##### </div></pre> <p>Следующий код TypoScript теперь динамически подставляет вместо маркера "DATE" текущий год:</p> <pre>page.10.marks { DATE = TEXT DATE { data = date : U strftime = %Y } }</pre> <p>Маркеры заменяются функцией str_replace. Подчасти загружаются в регистр и также доступны в cObject маркеров (только если не установлено .nonCachedSubst!).</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
wraps	<i>Массив...</i> cObject	<p>Это массив маркеров-оберток (чувствителен к регистру).</p> <p>Лучше продемонстрировать на примере:</p> <p>Пример:</p> <p>В шаблоне имеется подраздел "MYLINK":</p> <pre>This is <!---###MYLINK###-->a link to my<!-- ###MYLINK###--> page!</pre> <p>При помощи следующего кода TypoScript подраздел будут заменен оберткой, в которую вставляется содержимое, возвращаемое MYLINK cObject.</p> <pre>page.10.wraps { MYLINK = TEXT MYLINK.value = }</pre>	
workOnSubpart	string /stdWrap	Это не обязательное определение подраздела, над которым производится работа. Другими словами, если это значение определено, этот подраздел извлекается из шаблона и становится основой целого объекта шаблона.	
markerWrap	wrap /stdWrap	Это обертка, в которую помещаются маркеры. Значение по умолчанию ### ### , в результате чего маркеры представляются как ###[ключ_маркера]### . Любые пробелы убираются перед установкой вокруг ключ_маркера .	### ###
substMarksSeparately	boolean /stdWrap	<p>Если установлено, то маркеры подставляются в содержимое ПОСЛЕ подстановки подразделов и оберток. Обычно маркеры не подставляются внутрь подразделов и заключаются в обертки при использовании обычного режима кеширования в TEMPLATE cObject. Но возникает проблема при использовании маркеров внутри подразделов!</p> <p>Установка этого режима не помещает в кеш подстановочные маркеры, процесс выполняется последовательно.</p> <p>Другое решение — отключение кеширования, см. далее.</p>	
nonCachedSubs	boolean /stdWrap	<p>Если включено, то это в корне меняет режим подстановки этого cObject. Обычно, необработанный шаблон читается и разделяется на разделы, определяемые маркерами, подразделы и ключи-обертки. Плюс — хорошая скорость, так как "предварительно проанализированный" шаблон помещается в кеш. Минус — то, что шаблоны, зависящие от порядка подстановки, не очень хорошо работают.</p> <p>В этом режиме, сначала в шаблоне подставляются маркеры через str_replace, один за одним. Затем, одна за одной, подставляются подразделы. И наконец — последовательно подставляются обертки.</p> <p>Очевидно, что теряется возможность ссылаться на другие части шаблона через ключи регистра, как описано выше.</p>	
stdWrap	->stdWrap		

Пример:

```
page.10 = TEMPLATE
page.10 {
    template = FILE
    template.file = fileadmin/test tmpl
    subparts {
        HELLO = TEXT
        HELLO.value = This is the replaced subpart-code.
    }
    marks {
        Testmark = TEXT
        Testmark.value = This is replacing a simple marker in the HTML-code.
    }
    workOnSubpart = DOCUMENT
}
```

В этом примере загружается и используется шаблон под названием test.tml.

FLUIDTEMPLATE

Объект TypoScript FLUIDTEMPLATE работает схожим с обычным объектом на основе "маркеров" TEMPLATE образом. Но здесь не используются маркеры или подразделы, а разрешены переменные в стиле Fluid с фигурными скобками.



Замечание

для работы всего этого необходима установка расширений "fluid" и "extbase".

Свойство:	Тип данных:	Описание:	По умолчанию:
file	string /stdWrap	Файл шаблона fluid.	
layoutRootPath	filepath /stdWrap	Установка специфичных путей шаблона. Обычно это подлежащие файлы шаблона в Layouts/.	
partialRootPath	filepath /stdWrap	Специфичный путь частей. Обычно это подлежащие файлы шаблона в Partials/.	
format	keyword /stdWrap	Формат текущего запроса.	html
extbase.pluginName	string /stdWrap	Переменные для инициализации extbase.	
extbase.controllerExtensionName	string /stdWrap	Название расширения диспетчера (controller).	
extbase.controllerName	string /stdWrap	Название диспетчера (controller).	
extbase.controllerActionName	string /stdWrap	Название действия.	
variables	Массив... cObjects	Набор переменных, которые должны быть доступны в шаблоне fluid. Ключами являются названия переменных во Fluid. Зарезервированными переменными являются "data" и "current", они заполняются автоматически из текущего набора данных.	
stdWrap	->stdWrap		

[tsref:(cObject).FLUIDTEMPLATE]

Пример:

Шаблон Fluid (в fileadmin/templates/MyTemplate.html) может быть таким:

```
<h1>{data.title}<f:if condition="{data.subtitle}">, {data.subtitle}</f:if></h1>
<h3>{mylabel}</h3>
<f:format.html>{data.bodytext}</f:format.html>
```

Можно использовать его совместно с таким кодом TypoScript:

```
page = PAGE
page.10 = FLUIDTEMPLATE
page.10 {
    file = fileadmin/templates/MyTemplate.html
    partialRootPath = fileadmin/templates/partial/
    variables {
        mylabel = TEXT
        mylabel.value = Label coming from TypoScript!
    }
}
```

В результате заголовок страницы и метка из TypoScript будут вставлены в названии.

MULTIMEDIA

Этот элемент вставляет файл мультимедиа. Текстовые файлы выводятся непосредственно. Java, Flash, аудио и видео файлы используют тег embed.

Свойство	Тип данных:	Описание:	По умолчанию:
file	resource /stdWrap	<p>Файл мультимедиа. Возможные типы: txt, html, htm: вставляются непосредственно, можно использовать лишь свойства ".stdWrap". class: Java-приложение. swf: анимация Flash. swa, dcr: анимация ShockWave. au, wav, mp3: звук. avi, mov, asf, mpg, wmv: клипы (AVI, QuickTime, MPEG4).</p>	
params	string /stdWrap	<p>Параметры для объектов мультимедиа. Используются для определения autostart, type, width, height и т.п. Каждый тип файлов имеет свой набор параметров. Неполный список приведен за текущей таблицей.</p> <p>Пример:</p> <pre>params (type = application/x-shockwave-flash width = 200 height = 300)</pre> <p>Сформирует подобный тег</p> <pre><embed type="application/x-shockwave-flash" width="200" height="300"></pre> <p>Для параметров устанавливаемых по умолчанию (таблица ниже) пустая строка удалит параметры из тега embed.</p> <p>Пример:</p> <pre>params (height =)</pre> <p>Замечание: если здесь установить width или height, то width или height, установленные через ".width" и ".height", будут переназначены.</p>	
width	integer /stdWrap	Атрибут width тега embed. Не используется для файлов txt, html, htm и звуковых.	200
height	integer /stdWrap	Атрибут height тега embed. Не используется для файлов txt, html, htm и звуковых.	200
stdWrap	->stdWrap		

[tsref:(cObject).MULTIMEDIA]

Значимые параметры для .params

Для разных типов файлов могут быть установлено множество разных параметров. Здесь приведен неполный список некоторых из этих параметров:

au, wav, mp3:

Параметр:	Описание:	По умолчанию:
width	Width of the controls. If not set, the browser defaults to 200.	

Параметр:	Описание:	По умолчанию:
height	Высота элемента. Если не установлено, по умолчанию в браузерах используется 16.	
loop	Повторение звука по окончании воспроизведения. true или false.	
autoplay	Автоматическое начало воспроизведения звука. true или false.	

avi, mov, asf, mpg, wmv:

Parameter:	Описание:	По умолчанию:
width	Ширина клипа.	200
height	Высота клипа.	200
autoplay	Автоматическое начало проигрывания видео. true или false. Замечание: не для "mov", здесь параметр называется "autoplay". Смотрите пример ниже.	

swf, swa, dcr:

Parameter:	Описание:	По умолчанию:
width	Ширина элемента. Если не установлено, по умолчанию в браузерах используется примерно 200.	200
height	Высота элемента. Если не установлено, по умолчанию в браузерах используется примерно 200.	200
quality	Качество видео.	high

class:

Parameter:	Описание:	По умолчанию:
width	Ширина объекта.	200
height	Высота объекта.	200

Пример для QuickTime (mov):

```
params (
    width = 256
    height = 208
    autoplay = true
    controller = true
    loop = false
    pluginspage = http://www.apple.com/quicktime/
)
```

SVG

Так можно вставить SVG. Можно воспользоваться непосредственно данными XML или

ссылкой на файл. Flash зарезервирован для использования в браузерах, не имеющих встроенной поддержки SVG, так что это будет работать и в, например, IE 6/7/8.

Свойство:	Тип данных:	Описание:	По умолчанию:
width	integer /stdWrap	Ширина SVG.	600
height	integer /stdWrap	Высота SVG.	400
src	file resource /stdWrap	Файл ресурса SVG. Пример: src = fileadmin/svg/tiger.svg	
value	XML /stdWrap	Непосредственные данные XML для SVG. Игнорируются, если определен "src".	
noscript	string /stdWrap	Выводится, если невозможно вывести SVG.	
stdWrap	->stdWrap		

[tsref:(cObject).SVG]

Пример:

```
10 = SVG
10 {
    width = 600
    height = 600
    value (
        <rect x="100" y="100" width="500" height="200" fill="white" stroke="black" stroke-width="5px"/>
        <line x1="0" y1="200" x2="700" y2="200" stroke="red" stroke-width="20px"/>
        <polygon points="185 0 125 25 185 100" transform="rotate(135 125 25)" />
        <circle cx="190" cy="150" r="40" stroke="black" stroke-width="2" fill="yellow"/>
    )
    noscript.cObject = TEXT
    noscript.cObject.value = No SVG rendering possible, please use a browser.
}
```

Этот пример покажет несколько геометрических фигур.

EDITPANEL

Это объект содержимого вставляется только если пользователь внутреннего интерфейса авторизован на сайте, а также установлено и загружено расширение редактирования во внешнем интерфейсе. Приведенная ниже справка относится к системному расширению "feedit". При этом EDITPANEL также требует отображения Панели администрирования (config.admPanel = 1) и включения параметра "Отобразить значки редактирования". При вставке панели редактирования, кеширование страницы отключается.

Панель редактирования вставляет значки для перемещения, удаления, скрытия и создания записей.

В связке с css_styled_content, EDITPANEL выводится для каждого элемента содержимого на странице. Также можно вставить EDITPANEL в шаблон, используя TypoScript.

Пример

```
page = PAGE
page.10 = EDITPANEL
page.10 {
    ...
}
```

При этом нечего будет редактировать во внешнем интерфейсе, но можно воспользоваться панелью для, например, создания новых записей.



Замечание

для работы требуется установка расширения "feedit".

Свойство:	Тип данных:	Описание:	По умолчанию:
label	string /stdWrap	Заголовок панели. Возможно вставить заголовок записи посредством %s. Пример: <code>label = Раздел %s</code>	
allow	string	Определение доступных функций. Кроме того, этот список может уменьшиться, если BE_USER не имеет прав на совершение каких-либо действий. Значения перечисляются через запятую. Здесь можно выбирать из следующих параметров: toolbar,edit,new,delete,move,hide (панель инструментов — общий список значков на странице, поэтому используйте ее лишь для записей страниц).	
newRecordFromTable	string	Отображение панели для создания новых элементов (вверху списка) на странице из этой таблицы.	
newRecordInPid	int	Определение ID страницы, на которой создаются новые записи (за исключением новых страниц).	
line	boolean / int	Если установлено, то после панели будет помещена черная линия. Это значение определяет расстояние от черной линии до панели.	
edit.displayRecord	boolean	Если установлено, то редактируемая запись отображается над формой редактирования.	
onlyCurrentPid	boolean	Если установлено, то с панелью будут отображены лишь записи с pid, соответствующие текущему id (TSFE->id).	
innerWrap	wrap /stdWrap	Обертки панели редактирования.	
outerWrap	wrap /stdWrap	Обертка всей панели редактирования, включая черную линию (если установлена).	
printBeforeContent	boolean	Обычно панель редактирования отображается под соответствующим ей элементом. Если установлен этот параметр, панель выводиться перед редактируемым элементом. Пример: <code>tt_content.stdWrap.editPanel. printBeforeContent = 1</code> так панель редактирования выводится перед редактируемыми элементами при использовании <code>css_styled_content</code> .	0
previewBorder	boolean / int	Если установлено, то элементы скрыты, со временем начала/окончания, будут предварительно видны с рамкой. Целое значение определяет толщину рамки.	

Свойство:	Тип данных:	Описание:	По умолчанию:
previewBorder.innerWrap previewBorder.outerWrap previewBorder.color	wrap / HTML color	innerWrap обертки элементов содержимого (включая значки) внутри рамки предварительного просмотра (таблица HTML). outerWrap обертки всего элемента содержимого, включая рамку. color обозначает цвет рамки.	
stdWrap	->stdWrap		

[tsref:(cObject).EDITPANEL]

GIFBUILDER

GIFBUILDER

GIFBUILDER это объект, использующийся во многих ситуациях для создания файлов gif. Везде, где подразумевается ->GIFBUILDER объект, имеются ввиду применяемые свойства.

Используя TypoScript можно определить "числовой массив" "ОБЪЕКТОВ GIFBUILDER" (вроде "TEXT", "IMAGE" и т.п.), которые будут преобразованы в изображения друг за другом.

Название "GIFBUILDER" пришло из тех времен, когда поддерживался только лишь формат файлов GIF. PNG и JPG тогда только создавались (настраивается через \$TYPO3_CONF_VARS['GFX']).

ЗАМЕЧАНИЕ (+calc)

Всегда, когда функция "+calc" добавляется к значению в типе данных для приведенных ниже свойств, можно использовать размерности объектов TEXT и IMAGE из массива GifBuilderObj. Это осуществляется посредством тега вроде: "[10.w]" или "[10.h]", где "10" номер в массиве GifBuilderObj, а "w"/"h" обозначает ширину или высоту объекта.

Специальное свойство "lineHeight" (напр. "[10.lineHeight]") использует высоту одной строки текста.

Используя специальную функцию max(), можно определить максимальное из нескольких значений. Пример:

```
XY: [10.w]+[20.w], max([10.h], [20.h])
```

Вот полный пример (взят из "styles.content (default)"):

```
styles.header.gfx1 = IMAGE
styles.header.gfx1 {
    wrap = {$styles.header.gfx1.wrap}
    file = GIFBUILDER
    file {
        XY = [10.w]+10 ,{$styles.header.gfx1.itemH}
        backColor = {$styles.header.gfx1.bgCol}
        reduceColors = {$styles.header.gfx1.reduceColors}
        10 = TEXT
        10 {
            text.current = 1
            text.crop = {$styles.header.gfx1.maxChars}
            fontSize = {$styles.header.gfx1.fontSize}
            fontFile = {$styles.header.gfx1.file.fontFile}
            fontColor = {$styles.header.gfx1.fontColor}
            смещение = {$styles.header.gfx1.fontOffset}
        }
    }
}
```

Как видите, изображение gif имеет ширину, определяемую как ширина выводимого на нем текста + 10 пикселей. Высота — фиксированное значение константы {\$styles.header.gfx1.itemH}.

Основной объект (Top Level Object) "_GIFBUILDER"

Для GIFBUILDER возможно определить глобальные настройки через объект основного уровня "_GIFBUILDER". Одно из доступных свойств глобальной настройки GIFBUILDER —

это "charRangeMap".

.charRangeMap

Через это свойство можно глобально определить шрифтовые файлы для определенных диапазонов символов. Например, GIFBUILDER для формирования gif файлов для латинских символов может понадобиться определенный шрифт, а из другого шрифта true type могут быть взяты японские глифы. При этом нужно определить использование шрифтовых файлов для различных диапазонов Unicode кодов символов.

В объекте GIFBUILDER это можно сделать через параметр "splitRendering", но если на сайте имеются сотни объектов GIFBUILDER, то не очень удобно добавлять по 5-10 строк настроек для формирования каждого текста. Вместо этого возможно использование этой глобальной настройки для определения соответствия символов и базовых шрифтов.

Свойство:	Тип данных:	Описание:	По умолчанию:
[array]	string	<p>Название основного шрифта, соответствующего данной конфигурации. Помните, что используется лишь <i>название файла шрифта</i> — путь удаляется. Это сделано для облегчения настройки и решения проблем при перемещении файла шрифта в другое место внутри расширения.</p> <p>Например, при использовании файлов шрифтов "EXT:mytext/fonts/arial.ttf" или "t3lib/fonts/arial.ttf", оба они будут соответствовать одной и той же конфигурации.</p> <p>Ключ: Значение ключа массива используется при вызове настройки в конфигурации "splitRendering" индивидуальных объектов GIFBUILDER. В приведенном примере ключ – "123". Примечание: если ключ уже находится в локальной настройке GIFBUILDER, то ее содержание сохраняется и не заменяется. Поэтому возможно создать локальную настройку, замещающую глобальную.</p> <p>Пример:</p> <pre>_GIFBUILDER.charRangeMap { 123 = arial.ttf ... }</pre>	
[array].charMapConfig	TEXT / splitRendering .[array] configuration	<p>Установка настройки splitRendering. За деталями обратитесь к объекту GIFBUILDER TEXT.</p> <p>Пример:</p> <pre>_GIFBUILDER.charRangeMap { 123 = arial.ttf 123 { charMapConfig { fontFile = t3lib/fonts/vera.ttf value = -65 fontSize = 45 } fontSizeMultipliator = 2.3 }</pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<pre> } }</pre> <p>В этом примере объект GIFBUILDER TEXT со шрифтом "arial.ttf" имеет splitConfiguration, использующую "t3lib/fonts/vera.ttf" для всех символов в кодировке Unicode ниже/равных 65 в value.</p>	
[array].fontSizeMultiplier	double	Если установлено, будет взят размер шрифта объекта TEXT GIFBUILDER и умножен на это значение, что переопределит свойство "fontSize" внутри "charMapConfig".	
[array].pixelSpaceFontSizeRef	double	<p>Если установлено, то это служит множителем для четырех свойств [x/y]Space[Before/After] разбиения для формирования, для связи размера шрифта с этим значением.</p> <p>Другими словами, так как размер пробела в пикселях может изменяться в зависимости от используемого размера шрифта, можно просто определить это через данное значение: какому размеру шрифта оптимально подходят размеры пробелов, а значения для остальных размеров будут выставлены автоматически.</p> <p>Пример:</p> <pre>_GIFBUILDER.charRangeMap { 123 = arial.ttf 123 { charMapConfig { fontFile = t3lib/fonts/vera.ttf value = 48-57 color = green xSpaceBefore = 3 xSpaceAfter = 3 } pixelSpaceFontSizeRef = 24 } }</pre> <p>Здесь xSpaceBefore и xSpaceAfter будут "3", при размере шрифта 24. Если эту настройку использовать в объекте GIFBUILDER TEXT, с размером шрифта 16, то значение пробела будет "16/24", соответственно равно "2".</p>	

[tsref:_GIFBUILDER.charRangeMap]

Названия объектов в этом разделе

Как только в этом разделе встречается что-либо с названием "объект", подразумевается объект "GifBuilderObj", а не на "cObjects" из предыдущего раздела. Иначе может получиться путаница, так как объекты "IMAGE" и "TEXT" встречаются в обоих разделах.

Свойство:	Тип данных:	Описание:	По умолчанию:
1,2,3,4...	GifBuilderObj + .if (->if)	.if (->if) свойство всех объектов gifbuilder. Если свойство присутствует, но НЕ установлено, объект НЕ формируется! Это соответствует функциональности ".if" функции stdWrap.	

Свойство:	Тип данных:	Описание:	По умолчанию:
XY	x,y +calc /stdWrap	Размер файла gif.	100,20
format	"gif" / "jpg"	Выводимый тип. "jpg"/"jpeg" = изображение jpg.	gif
reduceColors	posint (1-255) /stdWrap	Уменьшение количества цветов (для файла gif).	
transparentBackground	boolean /stdWrap	Установка этого значения делает фон прозрачным. TYPO3 делает прозрачным цвет, находящийся в позиции 0,0 в изображении (верхний левый угол). При обработке текста, нужно ВЫКЛЮЧИТЬ параметр niceText, так как результат без этого параметра будет более приемлемым.	
transparentColor	HTMLColor /stdWrap	<p>Определение прозрачного цвета.</p> <p>Пример значений: #ffffcc red 255,255,127</p> <p>Параметры: transparentColor.closest = 1 Это включает близкие цвета в этот параметр. Это нужно, если изображение плохо "очищено" или "замылено".</p> <p>ЗАМЕЧАНИЕ: это может не работать, при использовании параметров reduceColors или niceText.</p>	
quality	posint (10-100)	Качество JPG (если ".format" = jpg/jpeg)	
backColor	GraphicColor /stdWrap	Фоновый цвет для gif.	white
смещение	x,y +calc /stdWrap	Смещение всех объектов в gif.	0,0
workArea	x,y,w,h +calc /stdWrap	Определение рабочей области в файле gif. Все GifBuilderObj рассматривают это как размерности файла gif в зависимости от размещения, перекрывающих изображений и т.п. Лишь текстовые TEXT, превышающие границы рабочей области будут выводиться вне этой области.	
maxWidth	pixels /stdWrap	Максимальная ширина файла gif.	
maxHeight	pixels /stdWrap	Максимальная высота файла gif.	

[tsref:->GIFBUILDER]

TEXT

Свойство:	Тип данных:	Описание:	По умолчанию:
text	->stdWrap	Это текстовая строка для файла gif. Элемент формируется, только если строка не пустая. Массив cObj->data загружается вместе с записью страницы, если, например, объект GIFBUILDER используется в GMENU или IMGMENU	

Свойство:	Тип данных:	Описание:	По умолчанию:
breakWidth	integer /stdWrap	Определяет максимальную ширину для объекта, наложение на элементы приведет к автоматическому переносу строки.	
breakSpace	float	Определяет фактор кратный высоте строки текущего элемента.	1.0
textMaxLength	int	Максимальная длина текста. Это просто обычный разрыв, исключающий неприятности при обработке очень длинных текстов!	100
maxWidth	pixels /stdWrap	Установка максимальной ширины в пикселях для текста. Уменьшите fontSize, если текст не помещается в эту ширину. Не поддерживается установка альтернативных fontSize в параметрах splitRendering. (Автор Rene Fritz <r.fritz@colorcube.de>)	
doNotStripHTML	boolean	Если установлено, теги HTML во вставленной строке НЕ удаляются. Во всех других случаях, код HTML по умолчанию удаляется!	0
fontSize	posint	Размер шрифта.	12
fontColor	GraphicColor /stdWrap	Цвет шрифта.	black
fontFile	resource	Шрифт (вы можете загрузить шрифт truetype!).	Nimbus (Arial-clone)
angle	degree	Угол вращения текста. Замечание: не доступно, если установлено spacing/wordSpacing.	0 Диапазон: -90 til 90
align	align	Выравнивание текста.	left
смещение	x,y +calc /stdWrap	Смещение текста.	0,0
antiAlias	boolean	Сглаживание FreeType. Помните, что по умолчанию "включено"! Замечание: параметр не доступен, если включен .niceText.	1
iterations	posint	Сколько раз текст должен быть "распечатан" сам в себя. Появляется эффект жирного текста. Замечание: параметр не доступен, если включен .niceText.	1
spacing	posint	Если истина, текст НЕ печатается. Эта возможность используется, если необходим объект-тень от текста, но сам текст не нужен.	0
wordSpacing	posint	Если установлен этот параметр, текст не обрабатывается. Тень и объем также, так что не применяйте их!!! Но это возможность формирования imageMap на основе размеров текста без его обработки.	= ".spacing" * 2
hide	boolean	Если истина, текст НЕ печатается.	0

Свойство:	Тип данных:	Описание:	По умолчанию:
		Эта возможность используется, если необходим объект-тень от текста, но сам текст не нужен.	
hideButCreateMa p	boolean	Если установлен этот параметр, текст не обрабатывается. Тень и объем также, так что не применяйте их!!! Но это дает возможность формирования imageMap на основе размеров текста без его обработки.	
emboss	GifBuilderObj ->EMBOSS		
shadow	GifBuilderObj ->SHADOW		
outline	GifBuilderObj ->OUTLINE		
imgMap	->IMGMAP ->stdWrap properties for "altText" and "titleText" in this case		
niceText	boolean	<p>Это очень востребованная возможность, помогающая формировать маленькие символы гораздо лучше, чем обычно это делает библиотека freetype. Но это сильно загружает систему!</p> <p>Принцип — создание черно-белого файла gif размера в два и более раз большего, чем требуемый файл, затем увеличенный текст распечатывается в этот файл. После чего ImageMagick (IM) уменьшает маску и маскирует текст для оригинального файла gif посредством временной маски.</p> <p>Фактически формируется шрифт двойного размера и уменьшается, добавляя более плавную форму для символов. Некоторые шрифты лучше подходят для этого, чем другие. Если качество не требуется, не используйте эту функцию.</p> <p>Некоторые свойства:</p> <ul style="list-style-type: none"> .before = параметры IM перед масштабированием; .after = параметры IM после масштабирования; .sharpen = значение резкости для маски (после масштабирования), целое 0-99 (дает возможность сделать размытый текст более четким!!!); .scaleFactor = фактор масштабирования, целое 2-5. 	
splitRendering.co mpX splitRendering.co mpY splitRendering. [array]		Split the rendering of a string into separate processes with individual configurations. By this method a certain range Разбивает формирование строки на отдельные процессы с индивидуальными настройками. Посредством этого, некоторые наборы символов могут формироваться другим шрифтом или размером шрифта. Полезно, если нужно использовать разные шрифты на одной строке, где могут быть латинские символы вместе, например, с японскими и используются разные файлы шрифтов.	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>Так можно сформировать ключевые слова другим шрифтом/размером/цветом.</p> <p>Свойства:</p> <p><code>splitRendering.compX</code> = дополнительное расстояние в пикселях между частями в направлении x; <code>splitRendering.compY</code> = дополнительное расстояние в пикселях между частями в направлении y; <code>splitRendering.[array]</code> = ключевые слова [charRange, highlightWord] <code>splitRendering.[array] {</code> <code> fontFile = альтернативный файл шрифта для формирования этих символов;</code> <code> fontSize = альтернативный размер шрифта для формирования этих символов;</code> <code> color = альтернативный цвет для формирования этих символов, работает ТОЛЬКО без "niceText";</code> <code> xSpaceBefore = x-расстояние перед этой частью;</code> <code> xSpaceAfter = x-расстояние после этой части;</code> <code> ySpaceBefore = y-расстояние перед этой частью;</code> <code> ySpaceAfter = y-расстояние после этой части;</code> <code>}</code></p> <p>Ключ: charRange <code>splitRendering.[array].value</code> = список диапазонов символов через запятую (напр. "100-200") в виде Unicode кодов. Список воспринимает начальную и конечную точки, например "- 200" или " 200 -" и единичные значения, например "65, 66, 67"</p> <p>Ключ: highlightWord <code>splitRendering.[array].value</code> = выделяемые слова, для этого проводится регистра-зависимый поиск.</p> <p>Ограничения:</p> <ul style="list-style-type: none"> • Значения компенсации в пикселях не корректируются для фактора масштабирования, используемого в <code>niceText</code>. В основном это означает, что при использовании <code>niceText</code>, эти значения будут вдвое уменьшены. • При использовании изменения расстояния между словами, не работает режим "highlightWord". • Переопределение цвета не работает вместе с "niceText". <p>Пример:</p> <pre>10.splitRendering.compX = 2 10.splitRendering.compY = -2 10.splitRendering.10 = charRange 10.splitRendering.10 { value = 200-380 , 65, 66 fontSize = 50 fontFile = t3lib/fonts/nimbus.ttf xSpaceBefore = 30 } 10.splitRendering.20 = highlightWord 10.splitRendering.20 { value = TheWord color = red }</pre>	

SHADOW

Свойство:	Тип данных:	Описание:	По умолчанию:
textObjNum	pos-int	Должно указывать на объект TEXT, если это свойство shadow не является непосредственным свойством объекта TEXT ("отдельная тень"). При этом shadow должна знать, тенью какого объекта TEXT она должна быть! С другой стороны, если тень является свойством объекта text, это свойство не нужно.	
смещение	x,y	Смещение тени.	
color	GraphicColor	Цвет тени.	
blur	posint (1-99)	Размытие тени. Значащими являются значения выше 40, вроде 40,50,60,70,80,90. Замечание: к сожалению, возможности размытия ImageMagick не слишком сильны в версии 4.2.9. Это относится к версиям выше 5.2.0, в которые добавлена функция размытия по Гауссу. Но на деле последняя разработка ImageMagick еще плохо сочетается с TYPO3.	
opacity	posint (1-100)	Непрозрачность (прозрачность^-1) 100% непрозрачность = 0% прозрачности. Применяется лишь для размытия.	
intensity	posint(0-100)	"Размер" тени. Это значение, если оно большое, комбинируется с размытием тени — создается что-то вроде размытых контуров.	

[tsref:->GIFBUILDER.(GObj).SHADOW]

EMBOSS

Объем — фактически две тени, смещенные в разных направлениях и с разными цветами, создающие эффект блика на объемном тексте.

Свойство:	Тип данных:	Описание:	По умолчанию:
textObjNum	pos-int	Должно указывать на объект TEXT, если свойство shadow не является свойством самого объекта TEXT ("отдельная тень"). При этом shadow нужно указать, тенью какого объекта TEXT она должна быть! С другой стороны, если тень является свойством объекта text, это свойство не нужно.	
смещение	x,y	Смещение объема.	
highColor	GraphicColor	Цвет верхней кромки.	
lowColor	GraphicColor	Цвет нижней кромки.	
blur	posint (1-99)	Размытие тени. Значащими являются значения выше 40, вроде 40,50,60,70,80,90.	
opacity	posint (1-100)	Непрозрачность (прозрачность -1) 100% непрозрачность = 0% прозрачности. Применяется лишь для размытия.	
intensity	posint(0-100)	"Размер" объема. Большое значение комбинируется с размытием — создаются размытые контуры.	

[tsref:->GIFBUILDER.(GObj).EMBOSS]

OUTLINE

Контуры обычно плохо выглядят, так как они делаются через наложение текста 4-8 раз. Лучше попытайтесь использовать тень с высокой интенсивностью. Это лучше работает!

Свойство:	Тип данных:	Описание:	По умолчанию:
textObjNum	pos-int	Должно указывать на объект TEXT, если это свойство shadow не является непосредственным свойством объекта TEXT ("отдельная тень"). При этом shadow должна знать, тенью какого объекта TEXT она должна быть! С другой стороны, если тень является свойством объекта text, это свойство не нужно.	
thickness	x,y	Толщина по каждому из измерений, диапазон 1-2	
color	GraphicColor	Цвет контура	

[tsref:->GIFBUILDER.(GBObj).OUTLINE]

BOX

Свойство:	Тип данных:	Описание:	По умолчанию:
dimensions	x,y,w,h +calc /stdWrap	Размерность блока. x,y смещение. w,h размерность. При размерности 1, получим линию, шириной в 1 пиксель!	
color	GraphicColor	Цвет заливки.	black
opacity	pos-int (1-100)	Непрозрачность (прозрачность -1) 100% непрозрачность = 0% прозрачности.	100
align	VHalign	<p>Пара значений, определяющих горизонтальное и вертикальное выравнивания.</p> <p>Значения: Горизонтальное выравнивание: r/c/l соответственно по правому краю, по центру, по левому краю. Вертикальное выравнивание: t/c/b соответственно по верху, по центру, по низу.</p> <p>Пример: Центрирование по горизонтали, и по низу по вертикали: align = c, b</p>	l, t

[tsref:->GIFBUILDER.(GBObj).BOX]

ELLIPSE

Свойство:	Тип данных:	Описание:	По умолчанию:
dimensions	x,y,w,h +calc /stdWrap	Размерности заполненного эллипса. x,y - смещение. w,h - размерности. Размерности 1 соответствует ширине в 1 пиксель!	

Свойство:	Тип данных:	Описание:	По умолчанию:
color	GraphicColor	Цвет заливки. Пример: <pre>file = GIFBUILDER file { XY = 200,200 format = jpg quality = 100 10 = ELLIPSE 10.dimensions = 100,100,50,50 10.color = red }</pre>	black

[tsref:->GIFBUILDER.(GObj).ELLIPSE]

IMAGE

Свойство:	Тип данных:	Описание:	По умолчанию:
file	imgResource	Файл изображения.	
смещение	x,y +calc /stdWrap	Смещение.	0,0
tile	x,y	Мозаичное x,y изображение. Максимум - 20 в каждом направлении. Если нужно больше — используйте большее изображение.	
align	VHalign	<i>Смотрите в "Справочнике по типам данных" в начале этого документа или в таблице "BOX".</i>	
mask	imgResource	Не обязательный файл-маска для изображения.	

[tsref:->GIFBUILDER.(GObj).IMAGE]

EFFECT

Синтаксис:

.value = [Property] = [value] | [Property] = [value]

Пример:

```
lib.image = IMAGE
lib.image {
    file = GIFBUILDER
    file {
        XY = 1024,768
        format = jpg
        10 = IMAGE
        10.file = fileadmin/image.jpg

        20 = EFFECT
        20.value = gamma=1.3 | flip | rotate=180
    }
}
```

Свойство:	Тип данных:	Описание:	По умолчанию:
gamma	0.5 - 3.0	Значение гаммы.	1.0
blur	1-99	Размытие краев внутрь изображения.	0
sharpen	1-99	Резкость края внутрь изображения.	0
solarize	0-99	Уменьшение количества цветов.	
swirl	0-100	Изображение закручивается или раскручивается от центра.	0

Свойство:	Тип данных:	Описание:	По умолчанию:
wave	амплитуда, длина	Все горизонтальные края преобразуются в волну заданной амплитуды и длины. Максимальное значение амплитуды и длины равно 100. Пример: 20 = EFFECT 20.value = wave=1,20	
charcoal	0-100	Эффект рисунка углем и его интенсивность.	
gray	-	Преобразование изображения оттенки серого. Пример: Придание изображению легкой волнистости и преобразование его в оттенки серого. 20 = EFFECT 20.value = wave=1,20 gray	
edge	0-99	Скругление углов.	
emboss	-	Эффект рельефа: подсветка или затенение, заменяющие светлые и темные границы изображения.	
flip	-	Переворот по вертикали.	
flop	-	Переворот по горизонтали.	
rotate	0-360	Угол вращения по часовой стрелки в градусах. Размерность изображения будет при необходимости увеличена, так что ничего не обрежется в оригинале.	0
colors	2-255	Количество разных цветов, используемых в изображении.	
shear	-90 - 90	Сдвиг по горизонтали.	
invert	-	Инвертирование цветов.	

[tsref:->GIFBUILDER.(GBObj).EFFECT]

WORKAREA

Установки другой рабочей области.

Свойство:	Тип данных:	Описание:	По умолчанию:
set	x,y,w,h + calc /stdWrap	Настройка размеров рабочей области. x,y — смещение. w,h — размеры. При использовании "calc" ознакомьтесь с соответствующими замечаниями по разделу "GIFBUILDER".	
clear	(isset)	Установка текущих значений по умолчанию. Проверка для использования isset().	

[tsref:->GIFBUILDER.(GBObj).WORKAREA]

CROP

Замечание: этот объект устанавливает размеры workArea соответственно новым размерам изображения!

Свойство:	Тип данных:	Описание:	По умолчанию:
backColor	GraphicColor	Смотрите "Справочник по типам данных".	backColor оригинала
align	VHalign	Горизонтальное и вертикальное выравнивание рамки обрезки. Смотрите "Справочник по типам данных".	l, t
crop	x,y,w,h + calc /stdWrap	x,y – смещение рамки обрезки от позиции, определяемой в "align". w,h — размеры рамки. При использовании "calc" ознакомьтесь с соответствующими замечаниями по разделу "GIFBUILDER".	

[tsref:->GIFBUILDER.(GBObj).CROP]

SCALE

Замечание: этот объект устанавливает размеры workArea соответственно новым размерам изображения!

Свойство:	Тип данных:	Описание:	По умолчанию:
width	pixels + calc /stdWrap	Ширина масштабируемого изображения.	
height	pixels + calc /stdWrap	Высота масштабируемого изображения.	
params	ImageMagickP arams	Параметры для обработки.	

[tsref:->GIFBUILDER.(GBObj).SCALE]

ADJUST

Позволяет выставлять тональный диапазон, наподобие диалога "уровни" в Photoshop. Можно установить входные и выходные уровни, таким образом, переопределить тональный диапазон изображения. Если нужно скорректировать гамму, взгляните на объект EFFECT.

Пример:

```
20 = ADJUST
20.value = inputLevels = 13, 230
```

Свойство:	Тип данных:	Описание:	По умолчанию:
inputLevels	low, high	<p>Этот параметр позволяет переопределить тоновый диапазон, сделать тени темнее, светлая светлее и увеличить контраст.</p> <p>Диапазон значений для "low" и "high" — целые числа от 0 до 255, где "high" должно быть больше "low".</p> <p>Значение "low" при этом будет переназначено в тон 0, а значение "high" — в 255.</p> <p>Пример:</p> <p>В этом примере тоновый диапазон изображения начнется с 50 для оригинала (что станет 0 в итоговом изображении) и закончится на 190 для оригинала (что станет 255 в новом изображении).</p> <pre>20 = ADJUST 20.value = inputLevels = 50, 190</pre>	

Свойство:	Тип данных:	Описание:	По умолчанию:
outputLevels	low, high	<p>Этот параметр позволяет переопределить тоновый диапазон, сделать тени светлее, света темнее и уменьшить контраст.</p> <p>Диапазон значений для "low" и "high" — целые числа от 0 до 255, где "high" должно быть больше "low".</p> <p>Начало тонового диапазона, то есть 0, будет переназначено в "low", а конечное значение, то есть 255, — в "high".</p> <p>Пример: В этом примере тоновый диапазон изображения станет таким, что исчезнут пиксели с тоном ниже 50 и выше 190 для оригинала (что станет 0 и 255, соответственно, в новом изображении).</p> <pre>20 = ADJUST 20.value = outputLevels = 50, 190</pre>	
autoLevels	-	Автоматическая установка уровней.	

[tsref:->GIFBUILDER.(GBObj).ADJUST]

Не GifBuilderObj

IMGMAP

Используется в GifBuilderObj "TEXT" для создания карты изображений для файла gif. Особенно востребовано с объекте меню IMGMENU..

Свойство:	Тип данных:	Описание:	По умолчанию:
url	url	url для ссылки.	Автоматически предоставляется для объектов меню IMGMENU
target	target	Цель для ссылки.	Автоматически предоставляется для объектов меню IMGMENU
noBlur	Boolean	Обычно, при использовании браузера MSIE, графические ссылки "размываются". При этом убирается некрасивая окантовка вокруг ссылок. Если установлено это свойство, ссылки НЕ размываются при "onFocus".	Автоматически предоставляется для объектов меню IMGMENU
explode	x,y	Так можно "расширить" и "сжать" карту изображений. Полезно для увеличения области реакции на ссылку, чем символы текста.	
altText	string	<p>Значение атрибута alt.</p> <p>(Используется из объектов TEXT Gifbuilding, имеющих свойства stdWrap. В других случаях таких свойств нет).</p>	
titleText	string	<p>Значение атрибута title.</p> <p>(Используется из объектов TEXT Gifbuilding, имеющих свойства stdWrap. В других случаях таких свойств нет).</p>	

[tsref:->IMGMAP]

Объекты MENU

Общие свойства

Здесь приведены общие для *всех* объектов меню свойства, если не установлены другие!

Свойство:	Тип данных:	Описание:	По умолчанию:
sectionIndex		(см. далее).	
alternativeSortingField		<p>Обычно элементы меню упорядочиваются по полям "sorting" таблиц pages и tt_content. Здесь возможно ввести список полей, которые будут использоваться в операторе SQL "ORDER BY" вместо этого.</p> <p>Примеры (для таблицы "pages"): alternativeSortingField = title desc (формирование меню в обратном алфавитном порядке)</p> <p>ОГРАНИЧЕНИЯ: Это свойство работает с обычными меню, меню sectionsIndex и специальными меню, типа "directory".</p>	
minItems	int	<p>Минимальное количество элементов в меню. Если количество страниц меньше этого числа, вставляются фиктивные меню с заголовком "..." и uid=[currentpage_id].</p> <p>Имеет приоритет над HMENU.minItems</p>	
maxItems	int	<p>Максимальное количество элементов в меню. Элементы сверх этого игнорируются.</p> <p>Имеет приоритет над HMENU.maxItems</p>	
begin	int +calc	<p>Первый элемент меню.</p> <p>Пример: формирование меню, в котором пропускаются первые два элемента, меню начинается с элемента 3: <code>begin = 3</code></p> <p>Имеет приоритет над HMENU.begin</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
JSWindow	boolean	<p>Если установлено, ссылки элементов меню будут открываться через JavaScript во всплывающем окне.</p> <p>.newWindow булева переменная, позволяющая каждому элементу меню открываться в собственном всплывающем окне, вместо открытия их в одном и том же окне.</p> <p>.params – список параметров, посылаемых в функцию открытия окна JavaScript, например: width=200,height=300,status=0,menubar=0</p> <p>ПРИМЕЧАНИЕ: не работает с JSMENU.</p>	
imgNamePrefix	string	Перфикс для названий изображений. Этот префикс добавляется с uid страницы.	"img"
imgNameNotRandom	boolean	<p>Если установлено, названия изображений не назначаются случайным образом. Полезно при работе с этими изображениями во внешнем JavaScript.</p> <p>ПРИМЕЧАНИЕ: не устанавливайте, если работаете с меню вместе с sectionIndex! При этом требуются уникальные названия элементов, на основе чего-либо еще, кроме uid родительской страницы!</p>	
debugItemConf		<p>Вывод (через функцию debug()) массив настроек для каждого элемента меню. Полезно для отладки optionSplit и т.д....</p> <p>Применимо к GMENU, TMENU, IMGMENU</p>	
overrideld	integer (id страницы)	<p>Если установлено, все ссылки в меню будут указывать на страницу с этим id. Вместо фактического uid страницы, отсылается параметр "&real_uid=[uid]".</p> <p>Это интеллектуальная функция, применяющаяся при вставке меню из другого места, возможно общего меню, так, чтобы элементы меню вели на одни и те же страницы, формирующиеся должным образом по real_uid.</p> <p>Применимо к GMENU, TMENU, IMGMENU</p>	
addParams	string	<p>Дополнительные параметры для ссылок меню.</p> <p>Пример: &some_var=some%20value Должно быть не обработанным url. Применимо к GMENU, TMENU, IMGMENU</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
showAccessRestrictedPages	integer (id страницы) / keyword "NONE"	<p>Если установлено, в меню будут включены страницы, разрешенные для доступа группе внешних пользователей. Конечно же страницы не будут доступны, поэтому ссылка будет вести на страницу с установленным значением ID. На этой странице можно разместить форму регистрации или какое-либо сообщение.</p> <p>Если используется значение "NONE", ссылка не будет изменена, и при щелчке сайт сформирует событие "страница не найдена" (что, разумеется, может быть использовано для захвата события и совершения соответствующего действия).</p> <p>Свойства:</p> <p>.addParam = дополнительный параметр для URL, который может содержать два маркера; ###RETURN_URL###, замещаемый ссылкой, как если бы страница доступна, и ###PAGE_ID###, замещаемый id исходящей страницы (может использоваться для поиска fe_group, требуемой для доступа).</p> <p>Пример:</p> <pre>showAccessRestrictedPages = 22 showAccessRestrictedPages.addParams = &return_url=###RETURN_URL###&pageId=###PAGE_ID###</pre> <p>В примере, ссылки из меню на запрещенные для доступа страницы ведут на страницу с id 22, на которой указаны возвращаемый URL, через переменную GET "return_url", и id страницы в переменной GET "pageId".</p>	
itemArrayProcFunc	название функции	<p>Первая переменная, передаваемая в эту функцию — это массив "menuArr" с элементами меню, собранными на основе типа меню.</p> <p>Этим массивом можно свободно управлять или дополнять его. Только не забудьте снова возвратить его!</p> <p>Замечание:</p> <p>свойство .parentObj <u>жестко связано</u> с ссылкой на вызов объекта tslib_menu. Здесь, например ->id – uid элемента меню, генерирующего подменю и т.п.</p> <p>Предварительная установка состояния элемента</p> <p>Состояния элементов, вроде SPC, IFSUB, ACT, CUR и USR, можно переназначить через установку ключа ITEM_STATE в записи страницы. За дополнительной информацией обратитесь к cObject HMENU/special=userdefined.</p>	
submenuObjSuffixes	string +optionSplit	Определение суффикса для альтернативных объектов подменю. Полезно для создания специальных подменю, в зависимости от родительских элементов меню. смотрите пример ниже.	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>Пример:</p> <p>В этом примере формируется меню, в котором объекты меню второго уровня зависят от номера элемента меню первого уровня, для которого формируются подменю. Для второго уровня используются "2" (по умолчанию), "2a" и "2b" (альтернативы). Что из этого используется, определяется через "1.submenuObjSuffixes", с настройками "a * * b". Это значит, что первый элемент меню использует настройку "2a", последний – "2b", а промежуточные – "2" (суффикс не устанавливается).</p> <pre> page.200 = HMENU page.200 { 1 = TMENU 1.wrap = <div style="width:200px; border: 1px solid;"> </div> 1.expAll = 1 1.submenuObjSuffixes = a * * b 1.N0.allWrap =
</pre> <pre> 2 = TMENU 2.N0.allWrap = <div style="background:red;"> </div></pre> <pre> 2a = TMENU 2a.N0.allWrap = <div style="background:yellow;"> </div></pre> <pre> 2b = TMENU 2b.N0.allWrap = <div style="background:green;"> </div> }</pre> <p>Результат приведен на следующем рисунке (применимо к тестовому сайту):</p>  <p>Применимо к GMENU, TMENU, GMENU_LAYERS, TMENU_LAYERS и GMENU_FOLDOUT при ≥ 2го уровня меню.</p>	

Общие состояния элементов для TMENU, GMENU и IMGMENU:

Это общие свойства для TMENU, GMENU и IMGMENU. То есть, они не используются в, например, JSMENU.

Свойство:	Тип данных:	Описание:	По умолчанию:
NO	Boolean / (config)	<p>По умолчанию "обычное" состояние формирования элементов меню. Необходимо для всех меню. Определяя свойства для "NO", не нужно устанавливать для него "1". То есть, без установки свойства "NO=1", меню в любом случае будет сформировано (для TMENU это имеет смысл).</p> <p>Простейшее меню в TYPO3 можно создать так:</p> <pre>page.20 = HMENU page.20.1 = TMENU page.20.1.NO = 1</pre> <p>Это простые теги <a>, в которые помещены заголовки страниц.</p>	1
IFSUB IFSUBRO	Boolean / (config)	Включение/настройка для элементов меню с подстраницами.	0
ACT ACTRO	Boolean / (config)	Включение/настройка для элементов меню из корневой линии.	0
ACTIFSUB ACTIFSUBRO	Boolean / (config)	Включение/настройка для элементов меню из корневой линии, имеющих подстраницы.	0
CUR CURRO	Boolean / (config)	Включение/настройка для элемента меню, представляющего текущую страницу.	0
CURIFSUB CURIFSUBRO	Boolean / (config)	Включение/настройка для элемента меню, представляющего текущую страницу и имеющего подстраницы.	0
USR USRRO	Boolean / (config)	Включение/настройка для элементов меню с ограниченным доступом, к которым имеется доступ у данного пользователя.	0
SPC	Boolean / (config)	<p>Включение/настройка для страниц "распорок". Распорки – это страницы типа "распорка". Это не просматриваемые страницы, но скорее "контейнеры" для разделения меню.</p> <p>Замечание: при использовании GMENU, роловеры не работают!</p>	0
USERDEF1 USERDEF1RO	Boolean / (config)	<p>Определенные пользователем, за детальной информацией по использованию обратитесь к .itemArrayProcFunc.</p> <p>Можно назначить значения USERDEF1 и USERDEF2 (+...RO) для ITEM_STATE посредством сценария/пользовательской функции для обработки массива элементов меню. Смотрите HMENU/special=userdefined или свойство .itemArrayProcFunc объектов меню.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
USERDEF2 USERDEF2RO	Boolean / (config)	(смотрите выше)	

[tsref:(cObject).HMENU.(mObj_itemStates)]

Приоритет: USERDEF2, USERDEF1, SPC, USR, CURIFSUB, CUR, ACTIFSUB, ACT, IFSUB

Все состояния *RO требуют установки настроек "RO" по умолчанию.

[menuObj].sectionIndex

Это разделяемое для всех menuObj свойство. Если оно установлено, то меню не будет состоять из ссылок на страницы "следующего уровня", а будет связано с родительскими страницами, но дополнены "#"-ссылками на cObject, формирующиеся на страницах. То есть элементы меню будут связаны с элементами содержимого (на colPos=0!) на странице. Индекс раздела.

.sectionIndex = [boolean]

Если установлено, то выбираются все элементы содержимого (из таблицы tt_content) из "Column" = "обычная" и включенным флагом "Включить в меню". Это соответствует элементу "Карта сайта" типа "Меню секций".

.sectionIndex.type = "all" / "header"

Если дополнительное свойство установлено на "all", то включение флагка "Включить в меню" не оказывает влияния, и выбираются все элементы из colPos=0.

Если это свойство установлено в "header", то выбираются лишь элементы с видимым заголовком (и заполненным полем "заголовок"!). То есть, если установлен "Скрытый" заголовок элемента, страница не появится в меню.

Записи данных /за сценой

При выборе записей меню происходит следующее: запись родительской страницы используется в качестве базовой для записи меню. Это значит, что для всего меню используются любые свойства "no_cache" или "target" родительской страницы.

Но, конечно, передаются некоторые поля из записей tt_content: вот их соответствия:

```
$temp[$row['uid']]=$basePageRow;
$temp[$row['uid']]['title']=$row['header'];
$temp[$row['uid']]['subtitle']=$row['subheader'];
$temp[$row['uid']]['starttime']=$row['starttime'];
$temp[$row['uid']]['endtime']=$row['endtime'];
$temp[$row['uid']]['fe_group']=$row['fe_group'];
$temp[$row['uid']]['media']=$row['media'];
$temp[$row['uid']]['header_layout']=$row['header_layout'];
$temp[$row['uid']]['bodytext']=$row['bodytext'];
$temp[$row['uid']]['image']=$row['image'];
$temp[$row['uid']]['sectionIndex_uid']=$row['uid'];
```

Отсюда видно, что

- поля "header" и "subheader" из tt_content отображаются как "title" и "subtitle" в записях страниц. Поэтому не нужно изменять стандартные объекты меню в соответствии с ними...
- поля "starttime", "endtime", "fe_group", "media" из tt_content отображаются на те же поля в записях страниц.
- поля "header_layout", "bodytext" и "image" отображаются на несуществующие поля в

записях страниц.

- новое поле "sectionIndex_uid" вводиться в запись страницы, обнаруженную функцией `t3lib_tstemplate->linkData()`. Если это поле присутствует в записи страницы, функция `linkData()` вставляет знак диез (#) и номер поля.

Замечание:

Невозможно создать подменю для меню с `sectionIndex`. Это не имеет смысла, поскольку эти элементы не являются страницами, а, значит, не имеют "дочерних" страниц.

GMENU

GMENU работает как объект под cObject "HMENU" и формирует графическую навигацию, в которой каждая ссылка является отдельным файлом gif.

Свойство:	Тип данных:	Описание:	По умолчанию:
RO	Boolean	Включение / отключение настройки RollOver. Если это истинно, RO становится объектом GIFBUILDER, определяющим шаблон элемента меню, когда указатель мышки находится над ним.	0
expAll	Boolean	Если это истинно, то всегда будет показано меню на уровне под элементом меню. Это соответствует разворачиванию следующего уровня меню при щелчке на элементе. Можно включить это, как поведение по умолчанию для всех элементов меню.	
collapse	Boolean	Если установлено, "активные" элементы меню, с развернутым следующим уровнем элементов, будут снова сворачиваться.	
accessKey	Boolean	Если включено, кнопки доступа будут установлены для ссылки в меню.	
noBlur	Boolean	Обычно, при использовании браузера MSIE, графические ссылки "размываются". При этом убирается некрасивая окантовка вокруг ссылок. Если установлено это свойство, ссылки НЕ размываются при "onFocus".	
target	target	Цель для ссылок из меню.	self
forceTypeValue	int	Если установлено, параметр &type ссылки принудительно устанавливается в это значение независимо от цели. Переназначается глобальный эквивалент "config", если установлен.	
stdWrap	->stdWrap	<p>Обертка всего элемента посредством stdWrap.</p> <p>Пример:</p> <pre>2 = TMENU 2 { stdWrap.dataWrap = <ul class="{register : parentProperty}"> NO { ... } }</pre>	
wrap	wrap	Обертка только если в меню имеются элементы!	

Свойство:	Тип данных:	Описание:	По умолчанию:
applyTotalH	objNumsList (смещение)	<p>Полная высота ранее сформированного элемента меню добавляется к смещению GifBuilderObj в этом списке.</p> <p>Пример: полезно при создание меню с индивидуальными элементами, но общим фоновым изображением, растянутым на все поле, занимаемое меню. При этом нужно установить фоновое изображение для каждого объекта GIFBUILDER и включить номер объекта в этот список. Взгляните на реализацию в static_template "styles.gmenu.bug".</p>	
applyTotalW	objNumsList (смещение)	Полная ширина ранее сформированного элемента меню добавляется к смещению GifBuilderObj в этом списке.	
min	x,y (calcInt)	Устанавливает минимальный размер для всего меню.	
max	x,y (calcInt)	Устанавливает максимальный размер для всего меню.	
useLargestItem X	boolean	Если установлено, то ширина всех элементов меню будет равна большему из них.	
useLargestItem Y	boolean	Если установлено, то высота всех элементов меню будет равна большему из них.	
distributeX	int+	Если установлено, то общая ширина всех элементов меню будет соответствовать этому числу пикселей, путем добавления/удаления одинакового количества пикселей от ширины каждого элемента. Переназначает любые установки для ".useLargestItemX"	
distributeY	int+	Если установлено, то общая высота всех элементов меню будет соответствовать этому числу пикселей, путем добавления/удаления одинакового количества пикселей от высоты каждого элемента. Переназначает любые установки для ".useLargestItemY"	
removeObjects OfDummy	objNumsList	Если для меню принудительно установлено использование определенного минимального размера, то это — список объектов в объекте gifbuilder, которые удаляются из этого последнего элемента. Это важно, если среди элементов меню есть такие, которые применимы лишь в качестве фактических элементов меню!!!	
disableAltText	boolean	Если включено, то параметр alt не устанавливается для изображений. Его можно назначить вручную через "imgParams" (см. далее)	

Свойство:	Тип данных:	Описание:	По умолчанию:
IProcFunc	название функции	В функцию передается внутренний массив "I", ожидается, что он же будет возвращен. Последующий за вызовом этой функции элемент меню формируется посредством интеграции implode() массива \$I[parts] в переданный массив. Поэтому можно изменить его в собственных целях. Смотрите пример в typo3/sysext/cms/tslib/media/scripts/example_itemArray ProcFunc.php	
[Common Item States, see above] + rollover version for all, except SPC	->GIFBUILDER + дополнительные свойства! Смотрите таблицу ниже.	<p>Это параметры GIFBUILDER для каждой категории элементов меню которые могут быть сформированы.</p> <p>Замечание: для набора GMENU также возможно определить настройку RollOver для состояний элементов. То есть для "Активного" состояния объект GIFBUILDER определяется через ACT, а объекта RollOver GIFBUILDER – через ACTRO. Этот шаблон применяется для BCEX состояний, за исключением SPC.</p> <p>ОСОБЕННОСТИ: Функция ->OptionSplit обрабатывает всю настройку GIFBUILDER перед формированием элементов.</p>	

[tsref:(cObject).HMENU.(mObj).GMENU

Дополнительные свойства для состояний элементов меню

Для состояний элемента GMENU доступны дополнительные свойства, несмотря на то, что основной объект объявлен как GIFBUILDER.

Конечно, представление этих свойств на одном уровне с объектом GIFBUILDER довольно грязное решение, но в данной ситуации применимо. Так как это необратимо, то в таком состоянии и находится уже долгое время.

Свойство:	Тип данных:	Описание:	По умолчанию:
noLink	boolean	Если установлено, элемент НЕ имеет ссылки!	
imgParams	params	Параметры для тега 	
altTarget	string	Альтернативная цель, переназначающая цель для GMENU	
altImgResource	imgResource	Определение использования альтернативного изображения. Если сюда возвращено изображение, оно переназначает все настройки GIFBUILDER.	
ATagParams	string /stdWrap	Дополнительные параметры.	
ATagTitle	string /stdWrap	Определение атрибута title тега a (также смотрите TMENUITEM).	
additionalParams	string /stdWrap	Определение параметров, добавляемых к концу URL. Это должен быть подготовленный код для вставки в качестве последнего параметра. Детально смотрите typolink->additionalParams	

Свойство:	Тип данных:	Описание:	По умолчанию:
wrap	wrap	Обертка элемента меню.	
allWrap	wrap /stdWrap	Обертка всего элемента.	
wrapItemAndSub	wrap /stdWrap	Если установлено, вместе с uid элемента подставляется "{elementUid}".	
subst_elementUid	boolean	stdWrap для всего элемента.	
allStdWrap	->stdWrap	Если установлено, элемент НЕ имеет ссылки!	

[tsref:(cObject).HMENU.(mObj).GMENU.(itemState)]

GMENU_LAYERS / TMENU_LAYERS

GMENU_LAYERS / TMENU_LAYERS работают как расширения к GMENU/TMENU, это значит, что свойства, представленные ниже являются дополнительными к представленным выше.

xMENU_LAYERS используются для создания двух- (и более!) уровневых меню, в которых уровни 2 и выше отображаются на слоях DHTML. Большинство функций работает в современных браузерах, включая Netscape, Microsoft Internet Explorer, Mozilla, Konqueror и Opera. Можно делать всевозможные каскадные меню.

Замечание: необходимо включить библиотеку

"typo3/sysext/cms/tslib/media/scripts/gmenu_layers.php" (для GMENU_LAYERS) и/или "typo3/sysext/cms/tslib/media/scripts/tmenu_layers.php" (для TMENU_LAYERS) , также нужно развернуть xMENU_LAYERS, чтобы в этом был смысл (используйте флаг expAll).

Совместимость: MSIE 4+, Netscape 4+ и 6+, Opera 5+, Konqueror.

Замечания:

- Netscape 4 не поддерживает перекрытие слоев указателем мышки.
- В Opera, по всей видимости, имеются проблемы с событием mouseout, при перемещении с элемента на слой. Поэтому событие может не сформироваться перед выводом слоя. Это случается только если слой расположен очень близко расположен к переключающему элементу. При этом могут появиться проблемы, когда rollover состояние элементов не сбрасывается.
- Возможные ошибки; было замечено, что Опера с каскадными уровнями может внезапно отказаться взаимодействовать со страницей, даже с обычными ссылками. Это может быть ошибкой JavaScript, но в этом нет уверенности, поскольку не работают даже обычные ссылки. С одноранговыми меню проблем не наблюдалось.

Свойство:	Тип данных:	Описание:	По умолчанию:
layerStyle	<DIV>-tag params	Параметры для тегов слоев <DIV> документа HTML. Возможно не понадобиться их изменять. Пример: position: absolute; VISIBILITY: hidden;	position: absolute; visibility: hidden;
lockPosition	"x" / "y" / ""	Если установлены "x" или "y", меню блокируется на слое и не следует за курсором мыши (что происходит без этой установки). "x" или "y", соответственно определяют	

Свойство:	Тип данных:	Описание:	По умолчанию:
		общую ширину (x) или высоту (y), добавляемую к смещению x или y для меню. То есть, нужно установить значение "x" для горизонтального GMENU_LAYERS, а "y" – для вертикального.	
dontFollowMouse	boolean	<p>Если установлено, и lockPosition пустое (то есть слои меню следуют за мышью), меню НЕ будет следовать за курсором мыши, но появиться в месте, где курсор мыши щелкнет по переключающему элементу. Полезно, если точные позиции элементов неизвестны.</p> <p>Внимание: нельзя устанавливать displayActiveOnLoad для элементов меню, у которых включен этот параметр (так как неизвестна абсолютная позиция слоя).</p>	
lockPosition_adjust	int	Число, добавляемое к ширине/высоте элементов меню, для компенсации, например вертикальный отступ или другие расстояния между изображениями в GMENU_LAYERS	
lockPosition_addSelf	boolean	<p>Обычно ширина и высота элементов (+lockPosition_adjust) прибавляется после формирования элемента. Это хорошо, когда слои меню формируются слева направо или вниз.</p> <p>Но при использовании directionLeft/directionUp, может понадобиться добавить ширину элементов перед ними.</p> <p>Если это так, установите этот параметр.</p>	
xPosOffset	int	Смещение меню от точки "активации" (если не установлено lockPosition) / от верхнего левого угла страницы (если установлено lockPosition)	
yPosOffset	int	То же, что и выше, но для направления y.	
topOffset	int	<p>Смещение элементов меню сверху браузера. Должно отличаться от значения свойства .layerStyle. Должно быть установлено при использовании directionUp.</p> <p>Используется вместе с определением lockPosition=x или xPosOffset.</p>	
leftOffset	int	<p>Смещение элементов меню от левой границы браузера. Должно отличаться от значения свойства .layerStyle. Должно быть установлено при использовании directionLeft.</p> <p>Используется вместе с определением lockPosition=x или xPosOffset.</p>	
blankStrEqFalse	boolean	Если установлено, свойства topOffset, leftOffset, xPosOffset, yPosOffset должны работать с "blank", если это действительно пустые строки, а не с просто "zero". Нужно включить, если нужно работать с нулевыми смещениями. Это типичная установка при выборе относительного позиционирования.	

Свойство:	Тип данных:	Описание:	По умолчанию:
directionLeft	boolean	Установите для право-ориентированных элементов (выпывающих влево). Пока не работает с Opera, так как неизвестно, как Opera читает ширину каждого слова. Это может работать при установке ширины слоев меню в .layerStyles непонятно почему.	
directionUp	boolean	Установите для по низу ориентированных меню (выпывающих снизу вверх).	
setFixedWidth	int	Ширины и высота элементов GMENU_LAYERS обычно берется из графических элементов. Для TMENU_LAYERS этот способ не подходит. Поэтому вы можете использовать .setFixedWidth и .setFixedHeight для установки желаемой величины. Конечно это может быть обработано браузерами не так, как хотелось бы. Альтернативно, можно использовать свойство "relativeToTriggerItem", позиционирующее слои меню относительно элемента, над которым располагается указатель мыши. Но и здесь есть недостатки. Золотая середина — использовать меню с пустым lockPosition и включить dontFollowMouse. При этом всего лишь нужно определить координаты следования x или y, и меню появиться в месте щелчка по элементу. Замечание: активно, если значение НЕ пустая строка. Установка этого значения в ноль, означает, что для элементов в GMENU_LAYERS ширина вычисляться не будет.	
setFixedHeight	int	Смотрите "setFixedWidth". То же, но для высоты.	
bordersWithin	I,t,r,b,l,t	Оставляет края слоя в этих пределах пикселей. Ноль — "не установлено". (Синтаксис: список целых чисел, по часовой стрелке: лево, верх, право, низ, лево, верх)	
displayActiveOnLoad	boolean	Если установлено, слой подменю открывается при загрузке страницы. Если для элементов основного меню еще установлено .freezeMouseover и определено RO, также отображаются элементы, принадлежащие элементам подменю. Свойства: .onlyOnLoad (boolean) If set, then the display of the active item will happen only when the page is loaded. The display will not be restored on mouseout of other items. Внимание: если объект GMENU_LAYER каскадный, убедитесь, что у всех элементов перед ним (у которого установлен этот атрибут) этот атрибут также установлен!	

Свойство:	Тип данных:	Описание:	По умолчанию:
freezeMouseover	boolean	<p>Если установлено, то любые эффекты перемещения мыши для основного меню удаляются, не уходе с меню, а при переходе на другой элемент (или при восстановлении скрытого или слоя по умолчанию).</p> <p>Свойства: .alwaysKeep (boolean) если установлено, скрытые элементы останутся, даже если подменю закрыто.</p>	
hideMenuWhenNotOver	int+	Если установлено (> 1), меню будет скрываться, как только курсор мыши покинет меню. Значение этого параметра определяет ширину (в пикселях) зоны вокруг элемента, при нахождении курсора на которой, слой не будет скрыт.	
hideMenuTimer	int+	Количество миллисекунд, через которые скроется меню, при установке hideMenuWhenNotOver.	
dontHideOnMouseUp	boolean	Если установлено, слои в меню не скрываются после щелчка по нему мышкой. Полезно, когда элементы меню загружают страницу в другой фрейм.	
layer_menu_id	string	<p>Если нужно определить название меню на странице. Обычно это не нужно!</p> <p>Внимание: не используйте подчеркивания и специальные символы в этой строке. Придерживайтесь alpha-numeric.</p>	[случайная 6 символьная хеш-строка]
relativeToTriggerItem	boolean	<p>Появляется возможность позиционирования слоев меню относительно переключающего их элемента. При этом нужно учитывать:</p> <ul style="list-style-type: none"> • Не работает с Netscape 4 – позиция переключающего слоя вычисляется нулевой, поэтому смещение для всех меню будет 0,0 + ваше значение. • Эта функция заключает элемент меню в некие <code><div></code> теги, прямо перед тем, как все меню обертывается посредством кода <code>.wrap</code> (для GMENU_LAYERS) или <code>.allWrap</code> (для TMENU_LAYERS). Оборотная сторона этого в том, что 1) для горизонтального меню, всегда помещайте свои элементы меню в таблицу, так как переходы строки не сработают из-за <code><div></code>-тегов и 2) убедитесь, что ячейки таблицы обернуты посредством свойств <code>.wrap/.allWrap</code> соответственно. • Эффективно работают лишь на первом xMENU_LAYER в каскаде. Для последующих элементов xMENU_LAYERS используйте "relativeToParentLayer". <p><i>Если установлено, свойства <code>xPosOffset</code>, <code>yPosOffset</code> и <code>lockPosition*</code> не функционируют (свойства <code>directionLeft</code>, <code>directionUp</code>, <code>topOffset</code> и <code>leftOffset</code></i></p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p><i>продолжают работать)</i></p> <p>Дополнительные свойства:</p> <p>.addWidth = добавление ширины элемента-переключателя;</p> <p>.addHeight = добавление высоты элемента-переключателя.</p>	
relativeToParentLayer	boolean	<p>Если установлено, слой позиционируется относительно предыдущего слоя (родительского) в серии каскадных xMENU_LAYERS. В основном к смещению текущего меню просто добавляется относительная позиция родительского слоя.</p> <p>Внимание: это свойство имеет смысл, только если имеется фактический предыдущий слой GMENU_LAYER, позицию которого можно получить! Поэтому это должно быть каскадное меню!</p> <p>Дополнительные свойства:</p> <p>.addWidth = добавляется ширина родительского слоя;</p> <p>.addHeight = добавляется высота родительского слоя.</p>	

[tsref:(cObject).HMENU.(mObj).GMENU_LAYERS, (cObject).HMENU.(mObj).TMENU_LAYERS]

Пример:

```
page.includeLibs.gmenu_layers = media/scripts/gmenu_layers.php
page.10 = HMENU
page.10.1 = GMENU_LAYERS
page.10.1 {
    layerStyle = position: absolute; VISIBILITY: hidden;
    xPosOffset = -30
    lockPosition = x
    expAll=1
    leftOffset = 15
    topOffset = 30
}
page.10.1.N0 {
    backColor = #cccccc
    XY = [10.w]+10, 14
    10 = TEXT
    10.text.field = title
    10.смещение = 5,10
}
page.10.2 = GMENU
page.10.2.wrap = <nobr>|</nobr>
page.10.2.N0 {
    backColor = #99cccc
    XY = [10.w]+10, 14
    10 = TEXT
    10.text.field = title
    10.смещение = 5,10
}
```

GMENU_FOLDOUT

GMENU_FOLDOUT работает как расширение GMENU, это значит, что свойства, представленные ниже являются дополнительными к представленным выше.

GMENU_FOLDOUT используется для создания динамически сворачивающихся

двухуровневых меню.

Работает в Netscape, Mozilla, Microsoft internet Explorer и Opera. Меню первого уровня – GMENU, так как за это отвечает GMENU_FOLDOUT, следующий уровень (о котором говорится, как о втором уровне) может быть и TMENU, и GMENU.

ЗАМЕЧАНИЕ: необходимо включить библиотеку "typo3/sysext/cms/tslib/media/scripts/gmenu_foldout.php".

Сценарий реализации взят из http://www9.ewebsite.com/skripts/foldoutmenu_move.htm

Совместимость: MSIE 4+, Netscape 4+ и 6+, Opera 5+

Свойство:	Тип данных:	Описание:	По умолчанию:
dontLinkIf_submenu	boolean	Если установлено, элементы с подменю не имеют связей. Элементы без подменю всегда связаны.	
foldTimer	int	Время анимации, миллисекунды.	40
foldSpeed	int, range 1-100	Количество шагов анимации. Для выключения анимации установите 1.	1
stayFolded	boolean	Не закрываться при щелчке по новому элементу (уровня 1).	
bottomHeight	int, pixels	Высота нижнего слоя. Важно, когда на нижнем слое находится содержимое или фоновый цвет, иначе слой обрежется.	100
menuWidth	int, pixels	Ширина основного слоя целого меню. Важная настройка, особенно для нижнего слоя, который может подрезаться. Всегда старайтесь устанавливать ширину меню в пикселях.	170
menuHeight	int	Высота всего слоя меню. Не столь важна.	400
subMenuOffset	x,y	Смещение подменю для каждого элемента меню. Важно, так как если не установить это значение, элементы появятся поверху "родителя".	
menuOffset	x,y	Смещение основного слоя меню на странице. От левого верхнего угла.	
menuBackColor	HTML-цвет	Фон меню. Если не установлен, прозрачный (что не очень хорошо работает в случае установки .foldSpeed отличной от 1. По усмотрению).	
dontWrapInTable	boolean	По умолчанию, каждый элемент меню первого уровня заключается в таблицу: <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0><TR><TD> [здесь HTML для элемента меню...] </TD></TR></TABLE> При этом уровни одинаково формируются во всех браузерах. Но можно отключить это, установив этот параметр. Замечание: использование <TBODY> ограничено для Netscape 4+.	0
bottomContent	cObject	Содержимое нижнего слоя, находящегося внизу меню.	

Свойство:	Тип данных:	Описание:	По умолчанию:
adjustItemsH	int	Настройка вычисления высоты слоев меню первого уровня (называемого верхним). Пример: -10 Будет вычтено 10 пикселей от высоты слоя при вычислениях.	
adjustSubItemsH	int	Настройка вычисления высоты слоев меню второго уровня (подэлементов, называемых подменю). Смотрите выше.	
arrowNO arrowACT	imgResource	При определении arrowNO и arrowACT, и правильном imgResources, эти изображения используются как "традиционный указатель", показывающие развернут ли элемент (активен) или нет. NO — обычный, ACT — активный. Изображение вставляется сразу перед элементом меню. Для изменения позиции, вставьте маркер ###ARROW_IMAGE### в обертку для элемента, вместо него будет вставлено изображение.	
arrowImgParams	 params	Параметры изображения-стрелки. Пример: hspace=5 vspace=7	
displayActiveOnLoad	boolean	Если установлено, то активные пункты меню свернутся при "onLoad".	

[tsref:(cObject).HMENU.(mObj).GMENU_FOLDOUT]

Пример:

```
## GMENU_FOLDOUT
includeLibs.gmenu_foldout = typo3/sysext/cms/tslib/media/scripts/gmenu_foldout.php

temp.foldoutMenu = HMENU
temp.foldoutMenu.1 = GMENU_FOLDOUT
temp.foldoutMenu.1.expAll = 1
temp.foldoutMenu.1.NO {
    wrap = | <br>
    XY = 150,20
    backColor = silver

    10 = TEXT
    10.text.field = title
    10.fontSize = 12
    10.fontColor = Blue
    10.смещение = 2,10
}
temp.foldoutMenu.1.R0 < temp.foldoutMenu.1.NO
temp.foldoutMenu.1.R0 = 1
temp.foldoutMenu.1.R0 {
    10.fontColor = red
}
temp.foldoutMenu.2 = TMENU
temp.foldoutMenu.2.NO {
    LinkWrap = <nobr><font face=verdana size=1 color=black><b>|</b></font></nobr><br>
    stdWrap.case = upper
}
temp.foldoutMenu.1 {
    dontLinkIfSubmenu = 1
    stayFolded=1
```

```

foldSpeed = 6
subMenuOffset = 10,18
menuOffset = 100,20
menuBackColor = silver
bottomBackColor = silver
menuWidth = 170

arrowNO = typo3/sysext/cms/tslib/media/bullets/arrow_no.gif
arrowACT = typo3/sysext/cms/tslib/media/bullets/arrow_act.gif
arrowImgParams = hspace=4 align=top

bottomContent = TEXT
bottomContent.value = Hello World! Here is some content!
}

```



Создается меню, вроде приведенного выше. Одна важная строка:

```
temp.foldoutMenu.1.expAll = 1
```

Если не установить ее (также, как GMENU_LAYERS) то второй уровень не сформируется!

TMENU

Свойство:	Тип данных:	Описание:	По умолчанию:
expAll	Boolean /stdWrap	Если это истинно, то всегда будет показано меню на уровне под элементом меню. Это соответствует ситуации, когда пользователь щелкает по элементу меню и оно открывается на следующий уровень. Это дает возможность сделать такое поведение, поведением по умолчанию для всех элементов меню.	
collapse	boolean	Если установлено, то "активные" элементы меню, развернутые на следующий уровень, будут снова сворачиваться.	
accessKey	boolean	Если установлено, на ссылки меню устанавливаются кнопки доступа.	

Свойство:	Тип данных:	Описание:	По умолчанию:
noBlur	boolean	Обычно, при использовании браузера MSIE, графические ссылки "размываются". При этом убирается некрасивая окантовка вокруг ссылок. Если установлено это свойство, ссылки НЕ размываются при "onFocus".	
target	target	Цель для ссылок из меню.	self
forceTypeValue	int	Если установлено, параметр &type ссылки принудительно устанавливается в это значение независимо от цели.	
stdWrap	->stdWrap	Обертка всего элемента посредством stdWrap. Пример: смотрите GMENU.stdWrap.	
wrap	wrap	Обертка, если в меню имеются элементы!	
IProcFunc	название функции	В функцию передается внутренний массив "l", ожидается, что он же будет возвращен. Последующий за вызовом этой функции элемент меню формируется посредством интеграции implode() массива \$l[parts] в переданный массив. Поэтому можно изменить его в собственных целях. Смотрите пример в typo3/sysext/cms/tslib/media/scripts/example_itemArray ProcFunc.php	
[Common Item States, see above]	->TMENUITEM	Параметры TMENUITEM для каждой категории элементов меню, которые могут быть сформированы. ОСОБЕННОСТИ: функция ->OptionSplit обрабатывает все настройку GIBUILDER перед формированием элементов меню.	

[tsref:(cObject).HMENU.(mObj).TMENU]

TMENUITEM

Это запись для страницы элемента меню — то же самое, что было в GMENU/gifbuilder. При этом, если нужно получить данные из текущей записи страницы, используйте stdWrap.data = page : [field name]

Свойство:	Тип данных:	Описание:	По умолчанию:
allWrap	wrap /stdWrap	Обертка всего элемента.	
wrapItemAndSub	wrap /stdWrap	Обертка для всего элемента и содержащихся в нем подэлементов.	
subst_elementUid	boolean	Если установлено, все вхождения строки "{elementUid}" в общем html коде элемента (после заключения в .allWrap} заменяются на uid элемента меню. Полезно при вставке идентификационного кода в HTML для управления свойствами через JavaScript.	
RO_chBgColor	string	При установленном свойстве RO (см. далее),	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>можно установить это свойство на определенный набор параметров, позволяющих изменять фоновый цвет, например ячейки таблицы, когда указатель мыши будет находиться над текстовой ссылкой.</p> <p>Синтаксис: <code>[over-color] [out-color] [id-prefix]</code></p> <p>Пример: <code>page = PAGE page.typeNum = 0 page.10 = HMENU page.10.wrap = <table border=1> </table> page.10.1 = TMENU page.10.1.NO { allWrap = <tr><td valign=top id="1tmenu{elementUid}" style="background:#eeeeee;"> </td></tr> subst_elementUid = 1 R0_chBgColor = #cccccc #eeeeee 1tmenu R0 = 1 }</code></p> <p>В этом примере задается цвет ячеек таблицы #eeeeee и он меняется на #cccccc (и обратно), когда указатель мыши находится над ней. Стока "1tmenu" - уникальный id для элементов меню. Он может не понадобиться (если одни и те же пункты меню не повторяются на странице), но важно то, что id ячейки таблицы имеет тот же префикс перед {elementUid} в метке (помечено красным). Другая важная деталь — УСТАНОВИТЬ цвет фона для ячейки по умолчанию через атрибут style (помечено синим). Если это не сделать, браузеры Mozilla будут вести себя странно, не захватывая событие mouseout при первом переключении.</p>	
before	HTML /stdWrap		
beforeImg	imgResource		
beforeImgTagParams	- параметры		
beforeImgLink	boolean	Если установлено, то это изображение будет связано с тем же тегом <A>, что и текст.	
beforeROImg	imgResource	Если установлено, ".beforeImg" и ".beforeROImg" должны создать пару для rollOver.	
beforeWrap	wrap	обертка для кода ".before".	
linkWrap	wrap		
stdWrap	->stdWrap	stdWrap для текста ссылки!	
ATagBeforeWrap	boolean		
ATagParams	параметры <A> /stdWrap	<p>Дополнительные параметры.</p> <p>Пример: <code>class="board"</code></p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
ATagTitle	string /stdWrap	Позволяют определить атрибут "title" для тега <a> пункта меню. Пример: <code>ATagTitle.field = abstract // description</code> При этом будут использоваться поля abstract или description для атрибута .	
additionalParams	string /stdWrap	Параметры, добавляемые в конец URL. Это должен быть готовый код, для вставки после последнего параметра. Детали смотрите в typolink->additionalParams	
doNotLinkIt	boolean /stdWrap	Если установлено, текст ссылки вообще не связан!	
doNotShowLink	boolean /stdWrap	Если установлено, текст совсем не отображается (учитывая распорки)	
stdWrap2	wrap /stdWrap	StdWrap для всего текста ссылки и тега A. (Помните, что по умолчанию в функцию stdWrap передается простое значение " ").	
RO	boolean	Если установлено, rollOver включен для ссылок.	
after...	[mixed]	Набор свойств "before..." дублируется свойствами "after...". Разница в том, что сформированное свойством .after... помещается после ссылки, а не до нее.	
altTarget	target	Если установлено, альтернативная цель переназначает свойство target для TMENU.	
allStdWrap	->stdWrap	stdWrap для всего элемента.	

[tsref:(cObject).HMENU.(mObj).TMENUITEM]

IMGMENU

Изображения для меню создаются из одного большого объекта GIFBUILDER, на основе объекта GIFBUILDER ".main" с добавлением свойств объектов GIFBUILDER для каждого элемента (NO, ACT, SPC... и т.д.).

Свойство:	Тип данных:	Описание:	По умолчанию:
target	target	Цель для ссылок из меню.	self
forceTypeValue	int	Если установлено, параметр &type ссылки принудительно устанавливается в это значение независимо от цели.	
noBlur	Boolean	Обычно, при использовании браузера MSIE, графические ссылки "размываются". При этом убирается некрасивая окантовка вокруг ссылок. Если установлено это свойство, ссылки НЕ размываются при "onFocus".	
wrap	wrap		
params	-параметры		

Свойство:	Тип данных:	Описание:	По умолчанию:
main	->GIFBUILDER	Базовая настройка карты изображений! Определяет "основу"!	
dWorkArea	смещение + calc	Базовое смещение элементов GIFBUILDER (так называемое "распределение").	
[Common Item States, see above]	->IMGMENUITEM M + .distrib	<p>Это параметры TMENUITEM для каждой категории элементов меню, которые могут быть сформированы.</p> <p>ОСОБЕННОСТИ: функция ->OptionSplit обрабатывает все настройку GIFBUILDER перед формированием элементов меню.</p> <p>.distrib — это (x,y,v,h +calc) распределение элементов меню. Это возможность отделения элементов друг от друга. Для размерностей ширины (X) и высоты (Y) каждой из ссылок можно использовать коды "textX" и "textY". Это работает путем добавления WORKAREA-GifBuilderObj между каждым IMGMENUITEM ("подстановка" объекта GIFBUILDER), а рабочая область определяет размещение текста. Так "x,y" определяет смещение, которое будет иметь следующий элемент (в основном — это ширина предыдущего элемента!), а "v,h" определяет размеры текущего элемента. Согласно этому примеру, взятому из static_template "template: MM": NO.distrib = textX+10, 0, textX+10, textY+5 В будущем TypoScript сможет обеспечить лучшие возможности для позиционирования объектов GIFBUILDER на карте изображений!</p> <p>ImgMap автоматически используется в ссылках! (это свойство ".imgMap" текстовых объектов в объектах GIFBUILDER, автоматически устанавливается, независимо от текущих настроек).</p>	
imgMapExtras	теги <area...>	Дополнительные теги <area...> для карты изображений.	
debugRenumberObject	boolean	Если установлено, итоговая конфигурация объекта GIFBUILDER выводится для отладки сделанных настроек.	

[tsref:(cObject).HMENU.(mObj).IMGMENU]

IMGMENUITEM

Свойство:	Тип данных:	Описание:	По умолчанию:
1,2,3,4...	->GifBuilderObj	<p>ЗАМЕЧАНИЕ: Это возможность создания карты изображений; все IMGMENUITEMS включаются в один большой Gifbuilderobj (и перенумеруются!). Из-за этого Gifbuilderobjects следующего уровня не могут получить данные для каждого элемента меню. Та же использование [##.w] и [##.h] с +calc в данный момент не поддерживается в IMGMENUITEM.</p> <p>При этом все объекты IMAGE первого уровня проверяются; если "file" или "mask" для любых объектов IMAGE установлены на "GIFBUILDER", объект Gifbuilder анализируется на присутствие каких-либо объектов TEXT, и если они обнаружены, то "отмечаются", это значит, что функция stdWrap вызывается во время установки \$cObj->data-array на элемент меню.</p> <p>В данном примере, предоставляется текст каждого элемента меню, формируя заголовок в маске, а не непосредственно в изображении. Заметьте, что объект "NO.10" представлен для формирования координат карты изображений!!!</p> <pre> NO.6 = IMAGE NO.6.file = masked_pencolor*.gif NO.6.mask = GIFBUILDER NO.6.mask { XY = 500, 200 backColor = black 10 = TEXT 10 { text.field = title fontFile = fileadmin/fonts/caflisch.ttf fontSize = 34 fontColor = white angle = 15 смещение = 48,110 } 20 = EFFECT 20.value = blur=80 } NO.10 = TEXT NO.10 { text.field = title fontFile = fileadmin/fonts/caflisch.ttf fontSize = 34 angle = 15 смещение = 48,110 hideButCreateMap = 1 } </pre>	

[tsref:(cObject).HMENU.(mObj).IMGMENUITEM]

JSMENU

Свойство:	Тип данных:	Описание:	По умолчанию:
levels	int, 1-5	Количество уровней	1
menuName	string	Название меню JavaScript. Если на странице имеется более одного JSMENU, нужно установить это значение для каждого.	

Свойство:	Тип данных:	Описание:	По умолчанию:
target	target	Цель для элементов меню.	
forceTypeValue	int	Если установлено, параметр &type ссылки принудительно устанавливается в это значение независимо от цели.	
1,2,3,4...	JSMENUITEM	Настройки для уровней.	
wrap	wrap	Обертка для полей выбора.	
wrapAfterTags	wrap	Обертка вокруг полей выбора с их обертками и тегов form, например в коде JS.	
firstLabelGeneral	string	Базовая первая метка. Может быть переназначена установками в каждом из JSMENUITEM	
SPC	boolean	Если установлено, распорки будут присутствовать в меню, иначе нет будут.	

[tsref:(cObject).HMENU.(mObj).JSMENU]

JSMENUITEM

Свойство:	Тип данных:	Описание:	По умолчанию:
noLink	boolean	Обычно, при выборе пункта меню в блоке выбора, обновляется блок выбора следующего уровня (если он есть), и если в нем нет элементов (из-за отсутствия подстраниц), ссылки ведут на текущую страницу. Если установлен этот параметр, то элементы меню в блоке выбора ни с чем не будут связаны. Лишь будет обновляться содержимое блока выбора для следующего уровня.	
alwaysLink	boolean	Если установлено, элементы в блоке выбора всегда будут иметь ссылки. Имеет приоритет над "noLink".	
showFirst	boolean	Если установлено, будет показана первая ссылка при обновлении меню.	
showActive	boolean	Если установлено, будет выбран активный уровень, если он присутствует.	
wrap	wrap	Обертка блока выбора.	
width	int+	Начальная ширина блоков устанавливается количеством _ (подчеркивание).	14
elements	int+	Начальный номер элементов в меню. Конечно переопределяется фактическим текстом из элементов меню.	5
additionalParams	string	Дополнительные параметры блока <select>. Например, можно установить ширину через параметр style, примерно так: style="width: 200px;"	
firstLabel	string	Первая метка в меню (по умолчанию – пустая).	

[tsref:(cObject).HMENU.(mObj).JSMENUITEM]

Пример:

```
# Меню:
temp.jsmenu = HMENU
temp.jsmenu.1 = JSMENU
temp.jsmenu.1 {
```

```
levels = 2
1.wrap = |<br>
2.wrap = |<hr>
}

# Вставка на страницу.
page = PAGE
page.typeNum =0
page.5 = TEXT
page.5.field = title
page.10 < temp.jsmenu
```

Отобразиться меню с двумя блоками выбора.

Приложение А – дополнения media/scripts/ media/scripts/ в общем

В директории typo3/sysext/cms/tslib/media/scripts (в старых версиях просто media/scripts) по большей части содержатся сценарии php, называемые "внешние модули", в пакетах функциям, включенным в библиотеки typo3/sysext/cms/tslib/. Несмотря на то, что они распространяются вместе с TYPO3, как и tslib/, это основа для разработки функциональности для внешнего интерфейса. Большинство из этих сценариев могут рассматриваться, как примеры для вашего собственного кода. Обратите внимание на слово "большинство", так как некоторые из них написаны давно и не актуальны на текущий момент.

Сведения об 'example templates'

Для каждого сценария расширения даются один или несколько шаблонов-примеров. Эти шаблоны являются частью документации по возможностям расширений, так как они описывают возможности маркеров и подразделов, и являются учебным примером. Поэтому шаблоны примеров могут меняться при появлении новых возможностей или по другим причинам.

Поэтому, используя шаблоны, *нельзя* полагаться на значения по умолчанию, так как в будущем они могут измениться! Сделайте копию, измените ее в соответствие с собственными целями, и создайте настройку TypoScript для расширения в которой прописано использование собственного файла шаблона!

fe_adminLib.inc

Файлы:

Файл:	Описание:
fe_adminLib.inc	Основной класс, использующийся для отображения форм администрирования внешнего интерфейса. Вызывайте из из USER_INT cObject с "userFunc = user_feAdmin->init". За примерами обратитесь к static_templates. Замечание: использование USER_INT cObject позволяет сценарию работать независимо от кеша страниц, что необходимо!!!
fe_admin_dmailsubscrip.tpl	Пример файла шаблона для подписки на информационные бюллетени пользователей в таблице tt_address. Этот шаблон используется в static_template "plugin.feadmin.dmailsubscription".
fe_admin_fe_users.tpl	Пример файла шаблона для создания новых внешних пользователей (fe_users). Этот шаблон используется в static_template "plugin.feadmin.fe_users".

Описание

Этот класс используется для создания форм для управления базой данных из внешнего интерфейса, *независимо от внутреннего (TBE)*. Если внешним пользователям нужно редактировать содержимое базы данных, можно использовать этот класс.

Аутентификация производиться через регистрацию fe_user, при этом можно отметить записи с fe_user_uid, чтобы запись принадлежала определенному fe_user. Другая

возможность аутентификации — по электронной почте. В этом случае доступ к записи открывается, если адрес email имеется в определенном поле. При аутентификации через fe_user, можно получить меню элементов для редактирования. При аутентификации по email, запрос отправляется на почтовый адрес. В письме содержится список доступных записей.

Все это создано на основе файлов HTML шаблонов, которые нужно разработать самостоятельно. С другой стороны, вы получаете полную свободу для разработки собственных форм.

Пример:

За примерами обратитесь к static_templates 'plugin.feadmin.*'. Опробуйте их на тестовом сайте TYPO3.

Статический шаблон

plugin.feadmin.*

Привнесенные GET или POST переменные:

Название:	Описание:
cmd	Команда.
preview	Флаг предварительного просмотра.
backURL	Обратный URL.
rU	UID записи.
aC	Код аутентификации.
fD	Установленные данные (массив полей).
FE	Массив данных для внешнего редактирования, синтаксис: FE[<i>таблица</i>][<i>поле</i>] = значение.

fe_adminLib.inc properties

Свойство:	Тип данных:	Описание:	По умолчанию:
templateFile	resource	Файл шаблона, примеры в typo3/sysext/cms/tslib/media/scripts/fe_user_admin tmpl	
templateContent	string	Также можно установить эти свойства, как значения, непосредственно в шаблоне.	
table	tablename	Таблица для редактирования. Замечание: полный список полей, доступных для редактирования для каждой рассматриваемой таблицы, определен в TCA с ключом ["feInterface"]["fe_admin_fieldList"]. Как отличный пример,смотрите определение таблицы для fe_users.	
defaultCmd	string	Определение действия по умолчанию (если &cmd= не установлено при вызове страницы).	
clearCacheOfPages	[список целых чисел]	Список id страниц, для которых очищается кеш при каждой удачной операции РЕДАКТИРОВАНИЯ, СОЗДАНИЯ или УДАЛЕНИЯ.	

Свойство:	Тип данных:	Описание:	По умолчанию:
debug	boolean	Если установлено, будет выводиться служебная информация из fe_adminLib, по которой можно отслеживать ошибки.	
Действия:			
edit	boolean /actionObject	<p>Если установлено, редактирование в основном разрешено.</p> <p>Но нужно определить:</p> <p>.fields (список полей) доступные для редактирования. Каждое из полей должно находиться в["feInterface"]["fe_admin_fieldList"] из массива TCA, определяющего полный список полей, доступных для редактирования для fe_adminLib.</p> <p>.overrideValues.[fieldname] (строковое значение) определяет значения для определенных полей, переопределяющие ЛЮБЫЕ введенные значения из формы. Переопределение происходит после анализа введенных значений свойством .parseValues из fe_adminLib, но перед оценкой через .required и .evalValues ниже. Например, это может быть полезно, если нужно скрыть редактируемую запись, так как сначала их нужно просмотреть.</p> <p>.required (список полей из .fields), определяет, какие поля обязательные. Правильное введенное здесь поле будет подчастью ###SUB_REQUIRED_FIELD_[fieldname]###, удаляемой из шаблонов, если они истинны, и, поэтому, в порядке. Ниже смотрите информацию об этой подчасти.</p> <p>.evalValues.[fieldname] (список кодов оценки) определяют оценки форм по конкретным полям. Смотрите ниже.</p> <p>.preview (булево) включает предварительный просмотр отправляемой формы. Требуется шаблон для предварительного просмотра в файле шаблона. Смотрите ниже о названиях маркеров подчастей.</p> <p>.menuLockPid (булево, принудительно блокирует меню редактируемых элементов на .pid (только редактирование)</p> <p>.userFunc_afterSave (название функции) вызывается после сохранения записи. Отдаваемое содержимое — это массив с текущей (и предыдущей) записью.</p>	
create	boolean /actionObject	<p>То же, что и .edit выше, за исключением другого состояния.</p> <p>Плюс три дополнительных свойства:</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>.noSpecialLoginForm (булево) – при установке, fe_adminLib НЕ ищет маркер TEMPLATE_CREATE_LOGIN, а всегда ищет TEMPLATE_CREATE</p> <p>.defaultValues.[fieldname] (строковое значение); то же .overrideValues, но устанавливает значения по умолчанию при начальном отображении формы.</p>	
delete	boolean	Могут или нет удаляться записи. По прежнему требуется аутентификация (владелец или по email). Установка переменной "preview" позволяет предварительный просмотр перед фактическим удалением записи.	
infomail	boolean	<p>Информационные бюллетени — это обычные текстовые письма, создаваемые на основе шаблонов из файла шаблона. Они могут использоваться, например, для пересылки забытого пароля для пользователей, но полное содержание бюллетеней определяет содержимое шаблона.</p> <p>Обычно используется лишь бюллетени по умолчанию (infomail.default), например для пересылки паролей. Конечно, можно использовать и другие ключи. Смотрите ниже.</p>	
infomail.[key]	(настройка свойств для infomail)	<p>Для того, чтобы fe_adminLib отсыпал информационные бюллетени, нужно определить эти переменные в переменных GET или формах HTML.</p> <p>Fetch – если целое число, то ищется uid, как значение "fetch". Если нет, ищется поле email (определенное в свойстве fe_adminLib, см. далее).</p> <p>key – указатель на используемую настройку infomail.[key].</p> <p>Свойства:</p> <p>.dontLockPid (булева) — выборка записей, только из .pid для fe_adminLib.</p> <p>.label (строка) – суффикс маркеров,смотрите "Маркеры email" ниже.</p>	
setfixed	boolean /properties	<p>Позволен установленный фиксированный ввод, возможно из ссылки в информационном бюллетене, или письме-уведомлении.</p> <p>Синтаксис:</p> <p>[fixkey].[fieldname] = fieldvalue – используется для настройки в информационном бюллетене ссылки на фиксированный ввод в таблице через маркеры SYS_SETFIXED_*. Смотрите ниже (свойство setfixed из fe_adminLib).</p> <p>Специальный ключ "DELETE" — просто булева переменная.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>.userFunc_afterSave (название функции) вызывается после сохранения записи. Отдаваемое содержимое — это массив с текущей (и предыдущей) записью.</p> <p>Понятие: Понятие "setfixed" лучше всего объясняется типичным сценарием — фактически, наиболее частой ситуацией ее использования: Представьте, что имеются несколько пользователей, работающих с информацией на сайте. Но перед вводом информации в базу данных, ее нужно проверить — просто просмотреть и, либо удалить, либо утвердить. В настройке "create" для fe_adminLib, устанавливается поле "hidden" (скрыть) для записи, перезаписываемое на 1. Таким образом запись по умолчанию скрыта. Затем настраивается setfixed-fixkey, устанавливающее поле "hidden" (скрыть) на 0. Эта установка формирует список параметров для URL, которые, наконец, вставляются вместо соответствующих маркеров в шаблон письма. Ссылки содержат все необходимые для аутентификации и изменения значений полей данные, так, чтобы изменения вносились одним щелчком по ссылке. Таким образом, администратор, щелкнув по ссылке в информационном письме, сможет изменить запись! Или, конечно же, щелкнув по другой ссылке с cmd=delete link, сможет удалить ее... Имеется специальный ключ "fieldname", который используется для указания определенного '_FIELDLIST', определяя список полей в записи, содержащих код аутентификации. Если ничего не определено, используется md5-хеширование для всей записи, то есть, любые изменения ссылки setfixed, приведут к невозможности ее использования. Если же, с другой стороны, установить _FIELDLIST = uid,pid, запись будет доступна для редактирования, пока актуальны значения uid и pid.</p> <p>Пример: Это общая настройка свойств email, с простой настройкой setfixed:</p> <pre>email.from = kasper@typo3.com email.fromName = Kasper Skårhøj email.admin = kasper@typo3.com setfixed.approve { hidden = 0 _FIELDLIST = uid,pid } setfixed.DELETE = 1 setfixed.DELETE._FIELDLIST = uid</pre> <p>Теперь, если вставить следующий маркер в шаблон email:</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>###SYS_SETFIXED_approve###</p> <p>будут подставлены примерно следующие параметры:</p> <p>&cmd=setfixed&rU=9&fD[hidden]=0&aC=5c403d90</p> <p>Все, что нужно для ссылки — правильный url (для вызова fe_adminLib!), например:</p> <p>###THIS_URL#####FORM_URL#####SYS_SETFIXED_approve###</p> <p>и для удаления:</p> <p>...###SYS_SETFIXED_DELETE###</p>	
Другие			
authcodeFields	<i>[список полей]</i>	<p>Список полей через запятую, на основе которого формируется authCode. В основном, этот список должен включать “uid” только в некоторых случаях. Если список включает несколько полей, нужно удостовериться, что authCode меняется, при изменении этих полей. При этом должно быть заново выслано письмо с новым кодом аутентификации.</p> <p>.addKey (строка) строка, добавляемая к md5-хешу для authCode. Введите любую комбинацию символов. Имеется ввиду то, что сторонние люди, не зная этого кода, не смогут восстановить md5-хеш, исключительно по uid.</p> <p>.addDate (настройка даты) используется для создания временного кода. Скажем, если ввести здесь значение “d-m-Y”, код будет работать до полуночи, а затем должен использоваться другой код.</p> <p>.codeLength (целое) определяет длину кода аутентификации. По умолчанию — 8 символов. В любом случае, предшествует TYPO3_CONF_VARS[SYS][encryptionKey].</p> <p>Совет: Если нужно формировать authCodes по стандартам authCodes (используется в direct mailer из t3lib_div::stdAuthCode()), установите TYPO3_CONF_VARS[SYS][encryptionKey] на уникальное значение (это необходимо в любом случае) и добавьте ТОЛЬКО “uid” как authcodeField. Это достаточно безопасно.</p>	
email		<p>.from (строка, email) email адрес отправителя отсылаемых писем.</p> <p>.fromName (строка) имя отправителя. Если</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>установлено, будет использоваться в форме NAME <EMAIL></p> <p>.admin email адрес администратора, уведомляемого об изменениях.</p> <p>.field (строка/целое) поле записи, в котором находится адрес email для отправления писем. Если содержимое поля — целое число, подразумевается uid fe_user, владельца записи и email адрес этого пользователя.</p>	
pid	int+	Pid в котором хранятся/получаются записи.	Текущая страница
fe_userOwnSelf	boolean	Если установлено, для fe_user, созданные этим модулем, поле fe_cruser_id устанавливается на их собственный uid, что означает, что они "владеют" своей записью и, поэтому, могут редактировать данные в ней. Все другие таблицы, имеющие поле fe_cruser_id, в разделе "ctrl" настроек \$TCA, автоматически устанавливают его на id текущего fe_user.	
fe_userEditSelf	boolean	Если установлено, fe_users, независимо, владеют ли они собственной записью или нет, могут редактировать свою запись.	
allowedGroups	[список целых чисел]	Список uid fe_group, которым позволено редактировать записи через эту форму. Обычно это делать позволено лишь fe_user владельцу.	
evalFunc	название функции	Функция, посредством которой можно манипулировать dataArray перед его сохранением. dataArray передается в функцию как \$content и ДОЛЖЕН быть возвращен из этой функции. Свойство "parentObj" – жесткая ссылка на объект fe_adminLib.	
formurl	->typolink	Содержит свойства typolink для URL (action tag) формы.	
parseValues.[field]	[список parseCodes]	<p>ParseCodes:</p> <p>int – возвращает целое значение из введенного;</p> <p>lower – возвращает версию введенного в нижнем регистре;</p> <p>upper – возвращает версию введенного в верхнем регистре;</p> <p>nospace – удаляет все пробелы;</p> <p>alpha, num, alphanum, alphanum_x – возвращает из введенного лишь символы алфавита (a-z) и/или цифры, alphanum_x также разрешает _ и -;</p> <p>trim – удаляет пробелы в конце строки;</p> <p>setEmptyIfAbsent – проверяет, пустое ли поле, если значение не передано. Нужно для сброса поля с флагками.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>random[x] – случайное значение от 0 до x;</p> <p>files[semicolon-list(!) of extensions, none=all] – поле, содержащее файлы. Детали смотрите ниже!</p> <p>multiple – установите, если вводится из блока с множественным выбором (не забудьте добавить ...[] к названию поля, так как его значение будем массивом!);</p> <p>checkArray – установите, если нужно, чтобы несколько флагков делали установку бит в одно поле. В этом случае, каждый флагок должен начинаться с [x], где x — это номер бита, начиная с нуля. Начальные значения флагков на форме должны быть ложью.</p> <p>uniqueHashInt[semicolon-list(!) of other fields] – формируется уникальный хеш (32 битовое целое) содержимого определенных полей. Значения этих полей сначала преобразуются в нижний регистр, а затем сохраняются лишь символы алфавита и цифры.</p>	
userFunc_updateArray	название функции	Указатель на пользовательскую функцию, в которую передается массив значений, перед тем, как он будет использован для обновления формы через JavaScript.	
evalErrors.[field]. [evalCode]		<p>Определение сообщений об ошибках, вставляемых в маркеры</p> <p>###EVAL_ERROR_FIELD_[fieldname]### при ошибке сравнения.</p> <p>Смотрите ниже описание сравнения.</p>	
cObjects. [marker_name]	cObject	<p>Это – cObject, которые можно вставить на место маркеров в шаблон.</p> <p>Пример: Скажем, cObject определен так:</p> <pre>cObject.myHeader = TEXT cObject.myHeader.value = This is my header</pre> <p>при этом можно включить этот cObject в большинство шаблонов через маркер с названием ###CE_myHeader### или ###PCE_myHeader### (см. далее о разнице).</p>	
wrap1	->stdWrap	Глобальная обертка 1. Будет разбито на маркеры ###GW1B### и ###GW1E###. Не изменяет введенное значение через настройки, а просто помещает его во что-нибудь.	
wrap2	->stdWrap	Глобальная обертка 2 (смотрите выше).	
color1	string /stdWrap	Значение для ###GC1### маркера (Global color 1)	
color2	string /stdWrap	Значение для ###GC2### маркера (Global color 2)	

Свойство:	Тип данных:	Описание:	По умолчанию:
color3	string /stdWrap	Значение для ###GC3### маркера (Global color 3)	

[tsref:(script).fe_adminLib]

Основные подразделы

Существует определенная система в наименовании основных подразделов в файле шаблона. Приведенные ниже маркеры используются в результате действия "сохранение". Код *[action]* может быть DELETE, EDIT или CREATE, в зависимости от значения cmd.

Маркер подраздела:	Описание:
###TEMPLATE_[action]_SAVED###	Используется для вывода HTML.
###TEMPLATE_SETFIXED_OK### (general) ###TEMPLATE_SETFIXED_OK_[fixkey]###	Используется для успешной ссылки setfixed.
###TEMPLATE_SETFIXED_FAILED###	Используется для не успешной ссылки setfixed. Помните, что если дважды щелкнуть по setfixed ссылке, второй переход будет ошибочным. Это происходит потому, что ссылка setfixed связана с оригинальной записью, и, при ее изменении, код аутентификации будет неверным!
###EMAIL_TEMPLATE_[action]_SAVED###	Используется для сообщения email, отсылаемого пользователю сайта.
###EMAIL_TEMPLATE_[action]_SAVED-ADMIN###	Используется для сообщения email, отсылаемого администратору.
###EMAIL_TEMPLATE_SETFIXED_[fixkey]###	Используется для уведомлений по email об удачных setfixed операциях.
###EMAIL_TEMPLATE_SETFIXED_[fixkey]-ADMIN###	Также для email администратора.

Аналогичная система применяется для маркеров подразделов при действиях EDIT и CREATE при инициализации и отображении форм:

###TEMPLATE_[action]### или, при авторизованном fe_user (только CREATE):
###TEMPLATE_[action]_LOGIN###

... и, если флаг &preview отсылается (включая DELETE)

###TEMPLATE_[action]_PREVIEW###

Подразделы, которые должны существовать:

Эти подразделы должны быть определены в любом шаблоне.

Subpart marker:	Описание:
###TEMPLATE_AUTH###	Отображается, если авторизация, либо fe_user, либо по email коду, прошла неудачно. Нужно разработать корректное отображение, соответственно проблеме!

Subpart marker:	Описание:
###TEMPLATE_NO_PERMISSIONS###	Это сообщение об ошибке, возникшей при успешной авторизации, но не позволяющей редактировать или удалять запись по другим причинам (неверный владелец fe_user/group).

Подразделы 'infomail' Email

Все подразделы email могут отсылаться как HTML. Это происходит, если первое и последнее слова в шаблоне, соответственно, <html> и </html>. Кроме того, должен быть загружен класс t3lib_htmlmail.

Subpart:	Описание:
###EMAIL_TEMPLATE_NORECORD###	
###EMAIL_TEMPLATE_[infomail_key]###	
###SUB_RECORD###	

'infomail' Email markers

Marker:	Описание:
###SYS_AUTHCODE###	
###SYS_SETFIXED_[fixkey]###	

Соглашения относительно FORM

Название формы для fe_adminLib должны находиться после названий таблиц, которые предполагается редактировать. Например, если нужно отредактировать записи в таблице "fe_users", нужно использовать тег FORM, вроде этого:

```
<FORM name="fe_users_form" method="POST" action="....">
```

Поля, изменяющие данные записей имеют следующий синтаксис, FE[таблица][поле]. То есть, если нужно редактировать поле "city" в записи tt_address, нужно использовать элемент, вроде этого:

```
<INPUT name="FE[tt_address][city]">
```

Кнопка сохранения может иметь любое название, за исключением "doNotSave", что предотвратит сохранение. Если необходима кнопка "Отмена", используйте JavaScript в событии onClick, для изменения document.location.

Общие маркеры

Marker:	Описание:
###GW1B### / ###GW1E###	Global wrap 1, начало и конец (заголовки).
###GW2B### / ###GW2E###	Global wrap 2, начало и конец (основной текст).
###GC1### / ###GC2### / ###GC3###	Global color от 1 до 3.

Marker:	Описание:
###FORM_URL###	URL, используемый в формах: index.php?id= <i>page-id</i> &type= <i>page-type</i>
###FORM_URL_ENC###	То же, что и выше, но rawurlencode.
###BACK_URL###	значение backUrl. Установка на значение переменной приходящего "backURL".
###BACK_URL_ENC###	То же, что и выше, но rawurlencode.
###REC_UID###	UID редактируемой записи. Установка на значение приходящей переменной "rU".
###AUTH_CODE###	Приходящая переменная "aC".
###THE_PID###	Значение "thePid" - место сохранения записи.
###THIS_ID###	id текущей страницы.
###THIS_URL###	URL текущего сценария, как он получен из t3lib_div::getThisUrl().
###HIDDENFIELDS###	Набор скрытых полей, требуемых для вставки в формы. По умолчанию они включают "cmd", "aC" и "backURL".

В большинстве случаев, можно использовать маркеры, вроде такого:

###FIELD_[field name]###

где [field name] — это название поля из записи. Используются все поля в записи.

И наконец, можно вставить cObject, определенные в TypoScript, посредством этого набора маркеров (смотрите свойство .cObject в приведенной ниже таблице):

###CE_[cObjectName]###
###PCE_[cObjectName]###

###PCE_* отличается от ###CE_* cObject, тем, что они формируются со вновь созданным cObj (в отличие от родительского cObj для fe_adminLib), когда массив данных загружается значениями из ->dataArr, массива отданного в сценарий. Поэтому эти маркеры полезны для вывода данных предварительного просмотра. И наконец, как PCE_, так и CE_ типы cObject маркеров, могут использоваться с любым элементом в меню редактирования (список доступных записей), если использовать префикс маркера "ITEM_", например ###ITEM_PCE_[cObjectName]###.

Проверка полей формы

Вывод сообщения об ошибке для REQUIRED (требуемых) полей

Когда шаблон формы отображает все подразделы, маркеры

###SUB_REQUIRED_FIELDS_WARNING###

и

###SUB_REQUIRED_FIELD_[field name]###

удаляются. При обычной ошибке "required" (поле не заполнено), поле SUB_REQUIRED_FIELDS_WARNING остается, и отображается содержащееся в нем сообщение об ошибке.

Для определенности, скажем, что поле "email" не заполнено. Тогда можно вставить в подраздел маркер

```
###SUB_REQUIRED_FIELD_email###
```

Обычно он удаляется, но он остается, если поле email содержит ошибку, при этом нужно дать специальное предупреждение, относящееся к этому полю.

Вывод других сообщений об ошибках

Можно использовать и другие формы оценки, кроме простой проверки “required”. Это относится к режимам “create” и “edit”, через свойства “.evalValues.[поле] = [список кодов]”. Для того, чтобы сказать пользователям, что конкретно неправильно, можно определить сообщения об ошибках посредством свойства .evalErrors, значения которого будут замещать маркер **###EVAL_ERROR_FIELD_[field name]###**.

Например, среди списка кодов оценки поля email определен код "uniqueLocal". Это сделано для того, чтобы убедиться, что адрес email не используется повторно. Это можно определить так:

```
create.evalValues {
    email = uniqueLocal, email
}
```

И устанавливается сообщение об ошибке, вроде этого:

```
evalErrors.email {
    uniqueLocal = Возможно Вы уже регистрировались с этим адресом email!
    email = Неверный адрес email!
}
```

Если адрес email уже имеется в базе данных, происходит ошибка, и поле **###EVAL_ERROR_FIELD_email###** замещается сообщением об ошибке “Возможно Вы уже регистрировались с этим адресом email!”.

Заполнение формы значениями по умолчанию

Значения по умолчанию в форму можно передать, используя тот же синтаксис, что и в форме. Например, так назначаются значения по умолчанию для name и email:

```
...?FE[tt_address][name]=Mike%20Tyson&FE[tt_address]
[email]=mike@trex.us&doNotSave=1&noWarnings=1
```

Обратите внимание, голубым цветом отмечены значения полей (должны быть rawurlencode). В javascript эта функция вызывается через `escape()`, а красным — необходимые значения, если НЕ нужно сохранять запись при этом действии и НЕ нужно выводить сообщение об ошибке, если в некоторые требуемые поля не введены значения.

Список кодов-проверок

Проверочные коды:	Описание:
uniqueGlobal	Требуется, чтобы значение поля было уникальным в глобальном масштабе, что это значение не должно встречаться в тех же полях для любых других записей из этой таблицы.
uniqueLocal	Подобно uniqueGlobal, но требуется, чтобы значение было уникальным <i>лишь для</i> записей с текущим PID. Так, если записи имеют разные значения pid, у них может быть одно и то же значение этого поля.

Проверочные коды:	Описание:
twice	Требуется, чтобы значение этого поля соответствовало значению вторичного поля [fieldname]_again посылаемого в форму. Полезно для ввода паролей. Например, если название поля для пароля – "user_pass", то просто добавьте второе поле, с названием "user_pass_again" и установите код-оценку "twice".
email	Требуется, чтобы значением поля был адрес email, хотя бы в форме [name]@[*][domain].[tld]
required	Просто требуемое поле (обрезаемое значение). 0 (ноль) не соответствует требованиям!
atLeast atMost	Определение минимум / максимум символов, введенных в поле. Пример , требуется хотя бы 5 символов: atleast [5]
inBranch	inBranch требует, чтобы значение (обычно поле pid) входили в список id страниц (pid), определенных в параметрах inBranch. Параметры задаются следующим образом [root_pid; depth; beginAt] Пример , список pid, глубиной в один уровень, начиная со страницы 4 (включительно): inBranch [4;1]
unsetEmpty	Никакой ошибки не выводиться. Просто сбрасывает поля, если значения в поле нет. То есть текущие значения не перезаписываются, если в поле нет значения.

[tsref:(script).fe_adminLib.evalErrors.(field).(evalCode)]

Загрузка файлов

fe_adminLib может получать файлы через формы. Но на данный момент имеются существенные ограничения. В идеале, процесс должен обрабатываться классом t3lib_tcemain, использующимся во внутреннем интерфейсе. Фактически это возможно и будет так, но не на данном этапе. Замечательно то, что tcemain.php отлично справляется с копированием/удалением файлов в определенном поле, и даже обрабатывает это независимо от метода сохранения списка файлов или использования отношений ММ к записям (смотрите раздел tables.php в "Inside TYPO3").

Так файлы обрабатываются в fe_adminLib и текущие ограничения:

- Загружать файлы можно ТОЛЬКО через режим "создания" записи. В любом случае, нельзя отредактировать прикрепленный файл (это может измениться в будущем). В любом случае, можно использовать режим "удаления".
- Можно использовать режим ПРЕДВАРИТЕЛЬНОГО ПРОСМОТРА вместе с "созданием". Работает примерно так: если используется режим предварительного просмотра, временно загруженный файл копируется с уникальным именем (с приставкой названия таблицы) в папку typo3temp/. Затем значение поля устанавливается на название файлов в списке. При одобрении предварительно просматриваемых записей пользователем, временные файлы копируются в папку uploads/*(или другую, определенную в ТСА). Ограничения в том, что временные файлы в typo3temp/ НЕ удаляются при копировании в папку для загрузок (может быть исправлено) и конечно не удаляется при ошибках (не может быть исправлено, так как пользователь может вернуться). Если предварительный просмотр не утвержден по причине изменения каких-либо значений, файлы нужно загружать снова (может быть исправлено).
- Расширения ТСА позволяют игнорировать поля! Но можно определить список расширений, которые могут загружать файлы, через свойство .parseValues из

fe_adminLib

- Ограничения на размер файлов из ТСА игнорируются! Но можно определить максимальный размер файла в килобайтах посредством свойства .parseValues из fe_adminLib
- Работает лишь с полями, настроенными на список названий файлов через запятую (не MM,смотрите документ “Inside TYPO3”, относительно MM отношений для файлов).

Рекомендуется использовать отдельную папку для файлов, получаемых через fe_adminLib. В TYPO3 testsite используется папка uploads/photomarathon/ для изображений. Это упрощает очистку при ошибках в файлах и неверных ссылках в записях.

Названия полей для файлов

Представим, что имеется поле “picture” в таблице “user_cars”, элемент формы выглядит примерно так:

```
<input type="file" name="FE[user_cars][picture][]">
```

Если нужно загрузить несколько файлов в это поле, элементы формы будут выглядеть примерно так:

```
<input type="file" name="FE[user_cars][picture][]">
<input type="file" name="FE[user_cars][picture][]">
<input type="file" name="FE[user_cars][picture][]">
```

Для полей с названиями файлов используйте тип blob и зарезервируйте для них по крайней мере 32 символа.

Замечание: убедитесь, что всегда в конце названия поля добавляете квадратные скобки ('...[]')! Иначе это не сработает!

tipafriendLib.inc

Tip a Friend!

Tip a friend:

You're now sending this link to an email recipient:

http://192.168.1.4/typo3/32/testsite-32b3/index.php?1278&type=1&backPID=58&tt_news=1

Your Name: *	<input type="text"/>
Your Email: *	<input type="text"/>
Recipient email: *	<input type="text"/> (Separate many recipients by comma)
Message:	<input type="text"/>
HTML message:	<input type="checkbox"/>
(You <u>must</u> fill in fields with * correctly!)	
Send	

Файлы:

Файл:	Описание:
tipafriendLib.inc	Основной класс для отображения формы Tip-a-Friend Вызовите его из объекта USER cObject через 'userFunc = user_tipafriend->main_tipafriend'
tipafriend_template.tml	Пример файла шаблона.

Пример:

Рабочую настройку смотрите в static_template 'plugin.tipafriend'.

Статический шаблон

plugin.tipafriend

tipafriendLib.inc properties

Свойство:	Тип данных:	Описание:	По умолчанию:
templateFile	resource	Файл шаблона. Смотрите пример в 'media/scripts/tipafriend_template.tml'	
code	string /stdWrap	Код для определения того, что делает сценарий. Чувствительно к регистру.	
defaultCode	string	Код по умолчанию (смотрите выше) при пустом значении. По умолчанию ничего не установлено и появляется поПощь.	

Свойство:	Тип данных:	Описание:	По умолчанию:
wrap1	->stdWrap	Глобальная обертка 1. Разбивается в маркерах ###GW1B### и ###GW1E### . Не изменяет введенное значение, а только лишь помещает его во что-то. Пример: wrap1.wrap = 	
wrap2	->stdWrap	Глобальная обертка 2 (смотрите выше).	
color1	string /stdWrap	Значение для маркера ###GC1### (Глобальный цвет 1)	
color2	string /stdWrap	Значение для маркера ###GC2### (Глобальный цвет 2)	
color3	string /stdWrap	Значение для маркера ###GC3### (Глобальный цвет 3)	
typolink	->typolink	Настройка TypoLink для TIPLINK на страницу TIPFORM. .additionalParams добавляет параметр "&tipUrl="	
htmlmail	boolean	Если установлено, создается и отсылается в виде HTML (обычная текстовая версия также отсылается).	

[tsref:(script).tipafriend]

plaintextLib.inc

Файлы:

Файл:	Описание:
plaintextLib.inc	Основной класс, используемый для отображения содержимого с простым текстом. Вызывается из USER cObject через "userFunc = user_plaintext->main_plaintext".
plaintext_content.tml	Пример файла-шаблона.

Пример:

Рабочую настройку смотрите в 'plugin.alt.plaintext'.

Статический шаблон

plugin.alt.plaintext

plaintextLib.inc properties

Свойство:	Тип данных:	Описание:	По умолчанию:
siteUrl	url	Url сайта.	
defaultOutput	untrimmed string	Вывод по умолчанию, если CType не сформирован.	
uploads.header	untrimmed string	Заголовок для загрузок	
images.header	untrimmed string	Заголовок для изображений	
images.captionHeader	untrimmed string	Заголовок для графических заголовков	
images.linkPrefix	untrimmed string	Префикс для ссылок на изображения	
.header			
defaultType	int	Тип, используемый по умолчанию	

Свойство:	Тип данных:	Описание:	По умолчанию:
date	date-config	Для даты заголовка	
datePrefix	untrimmed string	Префикс для даты заголовка	
linkPrefix	untrimmed string	Префикс для ссылок заголовка	
[1-5].preLineLen	int	Длина строки перед заголовком	
[1-5].postLineLen	int	Длина строки после заголовка	
[1-5].preBlanks	int	Количество пустых строк перед заголовком	
[1-5].postBlanks	int	Количество пустых строк после заголовка	
[1-5].stdWrap	->stdWrap	Для текста заголовка	
[1-5].preLineChar	string	Символ в pre-line	
[1-5].postLineChar	string	Символ в post-line	
[1-5].preLineBlanks	int	Количество пустых строк между заголовком и pre-line	
[1-5].postLineBlanks	int	Количество пустых строк между заголовком и post-line	
[1-5].autonumber	boolean	Если установлено, номер, предваряющий каждый заголовок. Номер относиться к количеству элементов содержимого в выборке.	
[1-5].prefix	untrimmed string	Префикс строки заголовка	
bulletlist.[0-3].bullet	untrimmed string	Маркер для маркированного списка, шаблон [0-3]	
bulletlist.[0-3].secondRow	untrimmed string	Если установлено, используется для строк во второго уровня маркированных списков.	
menu	cObject	cObject для формирования меню. Выводимое освобождается от тегов и ссылок. Далее, все преобразуются в символ chr(10)	
shortcut	cObject	cObject для формирования других элементов. Смотрите настройки ниже, просто использующие этот объект для формирования других tt_content элементов в виде простого текста.	
bodytext.stdWrap	->stdWrap	stdWrap для основного текста. Смотрите примерную настройку ниже.	
userProc	название функции	Позволяет обработать каждый элемент содержимого перед выводом. Свойство "parentObj" массива настроек, содержит ссылки на объекты plainText, вызывая функцию.	

[tsref:(script).plaintextLib]

Тип данных "untrimmed string" значит, что можно вводить строку как обычно, но при вводе значения в двух вертикальных линиях, оно НЕ будет очищаться. Обычные значения очищаются от пустых знаков в начале и конце строки.

Пример:

```
lib.renderObj = USER
lib.renderObj.userFunc = user_plaintext->main_plaintext
lib.renderObj {
    header.defaultType = 1
    header.date = D-m-Y
    header.datePrefix = |Date: |
    header.linkPrefix = | - Headerlink: |
    header.1.preLineLen = 76
    header.1.postLineLen = 76
    header.1.preBlanks = 1
    header.1.stdWrap.case = upper

    header.2 < .header.1
    header.2.preLineChar = *
    header.2.postLineChar = *

    header.3.preBlanks = 2
    header.3.postBlanks = 1
    header.3.stdWrap.case = upper

    header.4 < .header.1
    header.4.preLineChar= =
    header.4.postLineChar= =
    header.4.preLineBlanks= 1
    header.4.postLineBlanks = 1

    header.5.preBlanks = 1
    header.5.autonumber = 1
    header.5.prefix = |: >> |

    siteUrl = {$plugin.alt.plaintext.siteUrl}
    defaultOutput (
|
[Unrendered Content Element; ### CType### ]
|
)
uploads.header = |DOWNLOADS:|
images.header = |IMAGES:|
images.linkPrefix = | - Imagelink: |
images.captionHeader = |CAPTION:|
bulletlist.0.bullet = |* |
bulletlist.1.bullet = |# |
bulletlist.2.bullet = | - |
bulletlist.3.bullet = |> |
bulletlist.3.secondRow = |. |
bulletlist.3.blanks = 1

menu = <tt_content.menu.20
shortcut = <tt_content.shortcut.20
shortcut.0.conf.tt_content = <lib.renderObj
shortcut.0.tables = tt_content

bodytext.stdWrap.parseFunc.tags {
    link < styles.content.parseFunc.tags.link
    typolist = USER
    typolist.userFunc = user_plaintext->typolist
    typolist.siteUrl = {$plugin.alt.plaintext.siteUrl}
    typolist.bulletlist < temp.renderObj.bulletlist
    typohead = USER
    typohead.userFunc = user_plaintext->typohead
    typohead.siteUrl = {$plugin.alt.plaintext.siteUrl}
    typohead.header < temp.renderObj.header
    typocode = USER
    typocode.userFunc = user_plaintext->typocode
    typocode.siteUrl = {$plugin.alt.plaintext.siteUrl}
}
}
```

Приложение В – стандартные шаблоны

static_template

В этом разделе справочника по TypoScript представлены стандартные шаблоны, входящие в статические таблицы TYPO3, "static_template". Недопустимо самостоятельно изменять эти таблицы, вместо этого отправляете предложения по исправлению ошибок или добавлению шаблонов, либо других разделов TypoScript, через сайт www.typo3.com.

"static_template" публикуются в новых версиях. Старые записи в static_template НЕ изменяются от версии к версии (до финального выпуска), если они не находятся в разработке и явно не отмечены в примечании, что они еще не доделаны! Изменения могут вноситься до тех пор, пока проект TYPO3 не будет окончательно выпущен!

Media

Стандартные шаблоны используют стандартные медиа файлы, вроде gif изображений и шрифтов. Они расположены в папке "typo3/sysext/cms/tslib/media/" (в старый версиях в папке "media/") относительно корня сайта TYPO3.

Приложение С – включение сценариев PHP

Введение

Несмотря на то, что многое можно сделать через сам TypoScript, иногда лучшим решением оказывается включение на сайт своих сценариев PHP. Но следует понимать и учитывать некоторые обстоятельства. Например, система буферизации: страница, воспроизведенная через TYPO3 обычно кешируется в базе данных. Это делается из соображений производительности при отображении той же страницы в следующий раз. Но это также означает, что свой код можно только лишь включить, если он выводит что-либо по правилам шаблона, в который он включен! Невозможно вывести в TypoScript код для конкретного браузера, если шаблон не может различать браузеры. Если сделать это, то страница будет помещена в кеш с кодом HTML, ориентированным на специфический браузер, и в следующий раз, в другом браузере, эта страница будет возвращена из кеша с неправильным кодом HTML. Если в настройках установлены корректные условия для "другого браузера", то при запросе страницы будет сформирован другой код (который также будет помещен в кеш, но для использования в другом браузере!), при этом оба браузера получат одну и ту же страницу из кеша, но для нее будет использоваться разный код HTML.

Настройки TypoScript

Следующие объекты относятся к непосредственному включению кода PHP в шаблоны.

PHP_SCRIPT

Так включается сценарий PHP. Не нужно давать сценарию расширение ".php", лучше ".inc", так как он предназначен для включения, а не для непосредственного выполнения.

ЗАМЕЧАНИЕ: этот параметр игнорируется, если в localconf.php установлен параметр \$TYPO3_CONF_VARS['FE']['noPHPscriptInclude']=1;

Свойство:	Тип данных:	Описание:	По умолчанию:
file	resource /stdWrap	<p>Включаемый файл. В файле должен быть настоящий код PHP! Он включается посредством "include()";</p> <p>Указания:</p> <p>1) Все содержимое должно быть в переменной \$content. Вывода через echo быть не должно!</p> <p>2) Вызовите <code>\$GLOBALS['TSFE']->set_no_cache()</code>, если нужно отключить кеширование страницы. Установите во время разработки! А также, если выводимое не должно помещаться в кеш.</p> <p>ЗАМЕЧАНИЕ: если во включаемом сценарии имеется ошибка анализа, функция <code>\$GLOBALS['TSFE']->set_no_cache()</code> НЕ выполняется, поэтому кеширование не отключается. При ошибках анализа нужно очищать кеш страниц вручную после исправления ошибки!</p> <p>3) массив <code>\$conf</code> содержит настройки для <code>PHP_SCRIPT cObject</code>. Сделайте <code>debug(\$conf)</code> для просмотра его содержимого при отладке!</p> <p><i>Смотрите дальнейшие приложения в этом документе для указаний по написанию собственных включаемых сценариев PHP.</i></p>	
stdWrap	<code>->stdWrap</code>		

[tsref:(cObject).PHP_SCRIPT]

PHP_SCRIPT_INT

(смотрите PHP_SCRIPT)

Свойство:	Тип данных:	Описание:	По умолчанию:
file	resource /stdWrap	<p>Включаемый файл. В файле должен быть настоящий код PHP! Он включается посредством "include()";</p> <p>Предназначение:</p> <p>В основе работает также, как PHP_SCRIPT. Но главная разница в том, что вставка PHP_SCRIPT_INT (internal — внутренне, в противоположность external — внешне, смотрите ниже) просто вставляет строку-разделитель в код, а затем сериализует текущий cObj, затем помещает его в <code>\$GLOBALS['TSFE']->config['INTincScript']-array</code>. Этот массив сохраняется с кешированным содержимым страницы.</p> <p>Но включение сценария подобным образом позволяет избежать отключения кеширования страницы. Причина в том, что кешированное содержимое страницы содержит строку-разделитель, и при получении "статической" страницы из кеша, в строку разделитель вставляется объект динамического содержимого.</p> <p>Это компромисс из всех трех PHP_SCRIPT-cObjects, так как данные страницы, так как все данные страницы помещаются в кеш, но по прежнему включается сценарий pagegen.php, инициализирующий все классы, объекты и т.п. При этом обеспечивается окружение для сценария почти такое же, как и для PHP_SCRIPT, так как он вызывается изнутри объекта класса <code>tslib_cObj</code>.</p>	

Свойство:	Тип данных:	Описание:	По умолчанию:
		<p>Возможно работать со всеми функциями этого класса. Но все "статическое" содержимое страницы формируется единожды, кешируется, а динамически формируется только лишь включаемый сценарий.</p> <p>Правила:</p> <ul style="list-style-type: none">- вызовы <code>\$GLOBALS['TSFE']->set_no_cache()</code> и <code>\$GLOBALS['TSFE']->set_cache_timeout_default()</code> в этой ситуации не имеют никакого значения.- Разбор ошибок не затрагивается кешированием.- Имейте ввиду, что определенные глобальные переменные не могут быть установлены как обычно, а, значит, и не доступны обычным образом при работе в этом режиме. Большинство сценариев не потребуют внесения изменений.- Хрупкие зависимости и использование <code>LOAD_REGISTER</code>, так как <code>PHP_SCRIPT_INT</code> формируется <i>после</i> вывода кешированного содержимого, поэтому изменяется порядок событий, используемый <code>LOAD_REGISTER</code> может не работать.- Нельзя использовать вложения <code>PHP_SCRIPT_INT</code> и <code>PHP_SCRIPT_EXT</code> в <code>PHP_SCRIPT_INT</code>. Возможно только вложение <code>PHP_SCRIPT cObjects</code>.	
includeLibs	<i>список ресурсов</i>	<p>Список ресурсов через запятую, включаемых как сценарии PHP (через функцию <code>include_once()</code>) при включении этого сценария.</p> <p>Это возможно, так как любые включаемые файлы будут известны перед включением сценариев. Это невозможно для обычного PHP_SCRIPT cObject.</p>	
stdWrap	<code>->stdWrap</code>		

[tsref:(cObject).PHP_SCRIPT_INT]

PHP_SCRIPT_EXT

(смотрите PHP_SCRIPT)

Свойство:	Тип данных:	Описание:	По умолчанию:
file	resource /stdWrap	<p>Включаемый файл. В файле должен быть настоящий код PHP! Он включается посредством "include()";</p> <p>Предназначение: В основе работает также, как PHP_SCRIPT_INT, так как для этого типа включаемого сценария по прежнему вставляется строка-разделитель. Но разница в том, что содержимое разрывается перед самым выводом в браузер. В основном это означает, что PHP_SCRIPT_EXT (external -внешний, так как включается в глобальную область в файле index_ts.php!!!) может напрямую выводить данные через операторы echo! Это слишком "сырой" вариант PHP_SCRIPT, так как он не включается изнутри объекта, и возможно использовать лишь самые стандартные функции при вызове из TYPO3. Это самый быстрый из трех возможных PHP_SCRIPT-cObjects, так как все данные страницы кешируются, а динамическое содержимое формируется напрямую сценарием php.</p> <p>Правила:</p> <ul style="list-style-type: none">- Все содержимое может либо 1) непосредственно выводиться через echo, либо 2) возвращаться в \$content.- вызовы \$GLOBALS['TSFE']->set_no_cache() и \$GLOBALS['TSFE']->set_cache_timeout_default() в этой ситуации не имеют никакого значения.- Разбор ошибок не затрагивается кешированием.- В глобальной области имен, массив \$REC содержит текущую запись при "вставке" файла на страницу, а массив \$CONF содержит настройки для сценария.- Не путайте с глобальными переменными под названиями \$EXTiS_* .	
includeLibs	список ресурсов	<p>Список ресурсов через запятую, включаемых как сценарии PHP (через функцию include_once()) при включении этого сценария.</p> <p>Это возможно, так как любые включаемые файлы будут известны перед включением сценариев. Это невозможно для обычного PHP_SCRIPT cObject.</p>	
stdWrap	->stdWrap		

[tsref:(cObject).PHP_SCRIPT_EXT]

Включение собственного сценария

Сценарий включается через функцию PHP_SCRIPT внутри класса "tslib_cObj" сценария "tslib_content.php". Таким образом файл является частью этого объекта (tslib_cObj) и функции. Вот почему необходимо возвращать все содержимое в переменной "\$content", а любая настройка TypoScript доступна из массива "\$conf" (он может быть не установлена совсем, поэтому проверяйте ее через is_array()!).

\$conf

Массив \$conf содержит настройки для PHP_SCRIPT cObject. Воспользуйтесь `debug($conf)` для просмотра его содержимого в целях отладки!

\$content

Все содержимое возвращается в эту переменную.

Помните, ничего не выводите непосредственно в своем сценарии (кроме отладки кода)!

Пробелы

Так как ничего не посыпается в браузер перед тем, как все не будет сформировано и возвращено в `index_ts.php`, изначально отвечающий за процесс формирования страницы, необходима убедиться, что нет никаких пробелов до и после тегов `<?...?>` во включаемых или библиотечных сценариях!

\$GLOBALS['TSFE']->set_no_cache()

Если нужно отключить кеширование страницы, вызовите функцию `$GLOBALS['TSFE']->set_no_cache()`. Вызывайте ее в процессе разработки! А также, если создаваемое содержимое не должно помещаться в кеш.

Замечание: если в сценарии PHP допущена синтаксическая ошибка, препятствующая его выполнению, то функция `$GLOBALS['TSFE']->set_no_cache()` не будет выполнена, а страница будет помещена в кеш! При этом исправьте ошибку, очистите кеш страниц вручную и попытайтесь вывести страницу снова. Это справедливо лишь для PHP_SCRIPT, а не для PHP_SCRIPT_INT и PHP_SCRIPT_EXT, которые выполняются *после* помещения страницы в кеш!

Пример:

```
$GLOBALS['TSFE']->set_no_cache();
```

\$this->cObjGetSingle(value , properties)

Получение объекта-содержимого из массива \$conf (обратитесь к разделу "Пример" ниже за примером использования!).

Пример:

```
$content = $this->cObjGetSingle($conf['image'], $conf['image.']);
```

Так будет возвращен любой IMAGE cObject в свойстве "image" массива conf для включаемого сценария!

\$this->stdWrap(value, properties)

stdWrap для "value" (значения) содержимого для настроек массива "properties".

Пример:

```
$content = $this->stdWrap($content, $conf['stdWrap.']);
```

Это stdWrap для содержимого со свойствами ".stdWrap" массива \$conf!

Внутренние переменные в основном объекте внешнего содержимого, TSFE (TypoScript Front End)

Некоторые переменные в глобальном объекте TSFE, о которых нужно знать. Они все ТОЛЬКО ДЛЯ ЧТЕНИЯ (Читай: не меняйте их!!!!). Смотрите полное описание в классе `tslib_fe`.

Если, например, понадобиться доступ к переменной "id", можно написать:
`$GLOBALS['TSFE']->id`

Переменная:	PHP-тип:	Описание:	По умолчанию:
id	int	id страницы.	
type	int	Тип.	
page	array	Запись страницы.	
fe_user	object	Текущий внешний пользователь. Запись пользователя в <code>\$GLOBALS['TSFE']->fe_user->user</code> , если он авторизован.	
loginUser	boolean	Флаг, говорящий что пользователь авторизован.	0
rootLine	array	rootLine (полный путь к корню дерева страниц, не только для текущего сайта!). Корневая линия для текущего сайта находится в <code>\$GLOBALS['TSFE']->t3lib->rootLine</code> .	
sys_page	object	Объект с функциями страницы (pagefunctions object). Смотрите <code>t3lib/page.php</code> .	
gr_list	string (list)	Список групп, упорядоченный по номерам. Группа -1 = нет авторизации.	
beUserLogin	boolean	Флаг, говорящий об авторизации внутреннего пользователя!	0

Глобальные переменные

Переменная:	PHP-тип:	Описание:	По умолчанию:
BE_USER	object	Объект внутреннего пользователя (если есть).	не установлено
TYPO3_CONF_VARS	array	Настройки TYPO3.	
TSFE	object	Основной объект внешнего интерфейса.	

Пример

Пример использования включения сценариев.

Здесь мы хотели бы использовать сторонние библиотеки, не являющиеся частью TYPO3. Поэтому мы используем возможность включения библиотеки с самого начала анализа страницы.

Сначала поместим такую строку TypoScript в поле "Настройка" для шаблона:

```
config.includeLibrary = fileadmin/scripts/include.inc
```

Теперь подключен файл **include.inc** (в `typo3/sysext/cms/tslib/pagegen.php`). Здесь он выглядит так:

файл: fileadmin/scripts/include.inc

```
<?
...
    include('fileadmin/scripts/hello_world.inc');
    include('fileadmin/scripts/other_library.inc');
...
?>
```

Как видите, этот файл включает нашу библиотеку "hello_world" и остальные!

Файл **hello_world.inc** выглядит так:

файл: fileadmin/scripts/hello_world.inc

```
<?
class hello_world {
    function theMessage() {
        return "Hello World";
    }
?>
```

До сих пор ничего не происходит, за исключением подключения наших библиотек, готовых к использованию.

Теперь нужно использовать выводимое в классе hello_world где-нибудь на странице. В коде настроек TypoScript, создадим объект содержимого, включающий третий сценарий:

```
page.100 = PHP_SCRIPT
page.100.file = fileadmin/scripts/surprise.inc
```

surprise.inc выглядит так:

файл: fileadmin/scripts/surprise.inc

```
<?
    $hello_world_object = new hello_world; // Создается новый экземпляр
    $contentBefore = $this->cObjGetSingle($conf['cObj'], $conf['cObj.']);
    $content = $contentBefore . $hello_world_object->theMessage();
    $content = $this->stdWrap($content, $conf['stdWrap.']);
?>
```

Строка 1: создается объект PHP \$hello_world_object.

Строка 2: получение содержимого определенного нами cObject, "cObj".

Строка 3: результат строки 2 объединяется с результатом функции "theMessage" объекта \$hello_world_object.

Строка 4: окончательное содержимое использует свойство функции stdWrap ".stdWrap" массива \$conf.

Выводится:

с такой настройкой

```
page.100 = PHP_SCRIPT
page.100.file = fileadmin/scripts/surprise.inc
```

выводится следующее:

Hello World

с такой настройкой

```
page.100 = PHP_SCRIPT
page.100 {
    file = fileadmin/scripts/surprise.inc
    cObj = TEXT
    cObj.value = Joe says:&nbsp;
}
```

следующее:

```
Joe says: Hello World
```

с такой настройкой

```
page.100 = PHP_SCRIPT
page.100 {
    file = fileadmin/scripts/surprise.inc
    cObj = TEXT
    cObj.value = Joe says:&nbsp;
    stdWrap.wrap = <font color="red"> | </font>
    stdWrap.case = upper
}
```

следующее:

```
JOE SAYS: HELLO WORLD
```

Конец урока.

Сохранение данных пользователя или сессии

Это довольно просто в TYPO3.

Данные пользователя — это данные, сопровождающие авторизацию пользователей. Как только пользователь вышел с сайта, данные становятся недоступны и не могут быть изменены.

Данные сессии — данные о просмотре сайта пользователем. Пользователь может быть авторизованным пользователем, но данные о его сессии связаны с "сесном просмотра", а не с его id пользователя. Это значит, что человек все еще будет предоставлять данные о своей сессии, несмотря на то, что он вышел с сайта. Как только он закроет страницу в браузере, данные о нем исчезнут.

Также нужно знать, что данные сессии по умолчанию доступны в течение 24 часов.

Получение и сохранение данных пользователя / сессии производится через следующие функции:

`$GLOBALS['TSFE']->fe_user->getKey(type, key)`

"type" либо "user", либо "ses", что определяет область данных — пользователь или сессия .

"key" — это "название" под которым сохраняются данные. Это могут быть массивы и обычные скаляры.

Помните, что ключ "recs" зарезервирован для встроенной "покупательской корзины". Как и

"sys" (стандартные модули и код для TYPO3).

Пример:

```
if ($GLOBALS['TSFE']->loginUser) {  
    $myData = $GLOBALS['TSFE']->fe_user->getKey('user', 'myData');  
} else {  
    $myData = $GLOBALS['TSFE']->fe_user->getKey('ses', 'myData');  
}
```

Так из данных пользователя получаются сохраненные данные с ключом "myData", но, если нет авторизованных пользователей, они получаются из данных сессии.

\$GLOBALS['TSFE']->fe_user->setKey(type, key, data)

"type" либо "user", либо "ses", это определяет область данных — данные пользователя или сессии.

"key" — это "название", под которым сохраняются данные.

Помните, что ключ "recs" зарезервирован для встроенной "покупательской корзины". Как и "sys" (стандартные модули и код для TYPO3).

"data" — это переменная, которую нужно сохранить. Это могут быть массивы и обычные скаляры.

Пример:

```
$myConfig['name'] = 'paul';  
$myConfig['address'] = 'Main street';  
$GLOBALS['TSFE']->fe_user->setKey('ses', 'myData', $myConfig);
```

Так сохраняется массив \$myConfig с ключом "myData" в данных сессии. Они хранятся до тех пор, пока "paul" находится на сайте!

Использование встроенной "покупательской корзины"

TYPO3 имеет функционал покупательской корзины для данных сессии.

При утверждении данных формы (метод post/get, или через строку запроса), они сохраняются в массиве "recs" как данные сессии с ключом recs

Синтаксис следующий:

```
recs[table_name][uid_of_record]
```

Пример:

Этот элемент формы меняет значение регистра формы с uid=345 из таблицы "tt_products" в typo3. Обратите внимание, что сама по себе запись НЕ меняется ни в коем случае, меняется лишь "счетчик" в данных сессии, указывающий на "количество элементов", которые изменились в таблице.

```
<input name="recs[tt_products][345]">
```

Замечание о флажках:

При создании формы с переключателями, значения переключателей посылаются в MSIE/Netscape ТОЛЬКО при его включении! Если нужно отсыпать значения отключенного переключателя, создайте скрытые поля формы с теми же названиями перед переключателем!

Пример:

```
<INPUT type="hidden" name="recs[tt_content][345]" value="0">
<INPUT type="checkbox" name="recs[tt_content][345]" value="1">
```

Очистка "корзины"

Так можно очистить корзину:

```
<INPUT type="hidden" name="recs[clear_all]" value="1">
```

Приложение D – index.php

Введение

index.php — это основной сценарий, служащий для отображения страниц в TYPO3 / TypoScript. Эта страница отображает некоторую информацию о сценарии и его использовании.

Обычно страница запрашивается по значениям ее "id" и, возможно, "type".

"id" указывает на страницу. Это – целое число. Если предоставляется строка, то она рассматривается как псевдоним, и ищется соответствующая страница.

"type" определяет какого типа эта страница. Всегда целое число (0-255). Если "type" не определен, считается, что он равен нулю. "type" используется для набора фреймов. Набор фреймов может иметь "type=0" (или ничего), а страницы в разных фреймах должны иметь "type=1", "type=2" и "type=3". В TypoScript объект PAGE определяется для каждого типа, поэтому TYPO3 создает разные страницы, в зависимости от их типа. Обычно, объект PAGE отображает содержимое с названием "page" и с "type=1".

Передача данных в index.php

Может понадобиться передать данные в index.php по различным причинам. Имеется стандартная процедура передачи их в сценарий:

Авторизация/выход:

Определяется классом "t3lib_userauth", просматривающим переменную "logintype". Если она установлена, авторизация пройдена.

Вводиться может как через GET, так и через POST методы.

Авторизация:

```
logintype = "login";
pass = пароль;
user = имя пользователя;
```

pid = id страницы с записями пользователей. Это значение не нужно, если установлена переменная \$TYPO3_CONF_VARS['FE']['checkFeUserPid'].

(redirect = не используется)

Выход:

```
logintype = "logout".
```

За дополнительной информацией обратитесь к cObject FORMS.

Поиск

Определяется по cObject SEARCHRESULT, производящим поиск при установленных "sword" && "scols". Поиск ДОЛЖЕН предоставляться на страницу с объектом содержимого для поиска!

Вводиться может как через GET, так и через POST методы.

Поиск:

sword = искомое слово;

stype = тип поиска;

scols = таблица/столбец для поиска;

locationData = ссылка на запись с формой. Используется для поиска начальной точки поиска (ТОЛЬКО в методе POST);

(redirect = не используется);

scount = используется в результатах поиска, для отображения количества результатов.

spointer = используется в результатах поиска, для отображения начальной точки для следующих результатов.

За дополнительной информацией обратитесь к cObject SEARCHRESULT.

Почтовые формы

Определяется основным сценарием "index.php", ищущим установленную переменную "formtype_mail" (может быть кнопкой отправки).

ДОЛЖНО вводиться через метод POST. Также должны быть соответствующие REFERER и HTTP_HOST. К тому же, должна отправляться переменная locationData и хотя бы указатель на uid читаемой страницы.

Передача данных в базу данных

Определяется основным сценарием "index.php", ищущим установленную переменную "formtype_db" (может быть кнопкой отправки).

ДОЛЖНО вводиться через метод POST. Также должны быть соответствующие REFERER и HTTP_HOST. Для установки сценария обработки вводимых данных должна быть ссылка на объект FE_DATA.

Смотрите примеры в папке typo3/sysext/cms/tslib/media/scripts/, например "guest_submit.inc".