

Newstrack – программный продукт (далее система), предназначенный для автоматического наполнения блогов без участия человека.

Система автоматизирует процесс сбора информации с множества RSS и ATOM потоков (фидов), обработку полученного контента и автоматическую публикацию в другие блоги через протокол XML-RPC. Модуль управления Newstrack, выполнен в виде плагина для популярной платформы ведения блогов Wordpress (для легкости встраивания и унификации привычных интерфейсов).

### **Возможности системы**

- Структурированная организация хранения фидов (в виде разделов)
- Робот автоматического сбора обновлений фидов (поддержка Unix и Windows платформ)
- Публикация контента в удаленные блоги. Поддерживаются XML-RPC протоколы metaweblog (Wordpress) и Blogger (с google авторизацией).
- Отдельные парсеры для обработки контента и предварительной обработки перед публикацией. Возможность многократного использования парсеров.
- Парсеры позволяют обрабатывать и видоизменять контент в любых пределах
- Возможность перелинковки содержимого при экспорте (можно связать множество блогов и автоматически публиковать контент с ссылками на другие)
- Множество примеров парсеров в поставке
- Возможность экспорта нескольких фидов в множество блогов (или в один блог)
- Поддержка перелинковки фидов (внедрение в посты прямых ссылок из других фидов)
- Возможность расширения системы.

## Установка системы

- **Внимание!** Newstrack работает только на PHP5, так же для работы системы обязательно наличие следующих модулей PHP5: **mbstring**, **iconv**, **curl** и **DOM**. Также необходимы расширения **PEAR: MDB2, MDB2#mysql**. Если они отсутствуют в системе и нет возможности установки глобального PEAR, то эти модули могут поставляться отдельно по запросу.
- Предварительная настройка опций PHP для правильной работы плагина:
  - Создайте файл `.htaccess` (или же откройте существующий)
  - Добавьте в него следующие опции конфигурации PHP
    - **`php_value magic_quotes_gpc 1`**
    - **`php_value mbstring.internal_encoding utf-8`**
- Распакуйте дистрибуционный архив.
- Скопируйте папку `nst` в корневой каталог Вашей инсталляции `wordpress`
- Настройка системы
  - В каталоге `nst` зайдите в подкаталог `conf` и отредактируйте файл `config.php`
    - Скопируйте в него настройки доступа к базе данных из Вашего файла `wp-config.php` :
    - `define('DB_NAME', 'nst'); // The name of the database`
    - `define('DB_USER', 'user'); // Your MySQL username`
    - `define('DB_PASSWORD', 'password'); // ...and password`
    - `define('DB_HOST', 'localhost'); // 99% chance you won't need to change this value`
  - убедитесь что каталог `cache` (и входящие в него подкаталоги) доступны на запись всем (`chmod 777`)
- Установка плагина
  - Скопируйте файлы из дистрибуционного каталога `wordpress` в каталог `wp-content/plugins`
  - В админке Wordpress зайдите в раздел `Plugins` и активируйте плагин `Newstrack`
  - После Активации появится модуль `Newstrack` в главном меню админки, зайдите в него
  - При первом запуске, Вам будет предложено создать базу данных – нажмите `Install Database` .
  - Плагин установлен.
- Установка под Windows
  - В процессе работы система активно использует утилиту `curl`, если под Unix она обычно всегда присутствует в системе то под Windows –нет.
  - Для работы `Newstrack` необходимо ее скачать и скопировать в каталог `c:\windows` (именно здесь ее будет искать робот)
  - Скачать утилиту под Win32 можно здесь:
    - <http://fileforum.betanews.com/search?search=cURL+for+Windows>

## Настройка автозапуска роботов сбора и экспорта

Робот сборщик фидов:

- `_feed_p.php` – работает как под Windows так и unix
- `_wp_robot.php` – работает в многопоточном режиме под unix (до 20 одновременных закачек)

Настройте cron или task scheduler на периодический запуск данного скрипта (одного из)

Робот, Выполняющий экспорт

- `_queue_p.php`

Лучше всего вызывать данный скрипт сразу за роботом сборщиком фидов

Рбот, чистящий базу данных

- `_purge_data.php`

Осуществляет очистку данных старше, чем 60 дней (настраиваемо).

Пример Unix скриптов для автоматического запуска можно найти в подпапке `nst/shell`

Подкорректируйте пути в систему в файлах `*.sh`, дайте им права на выполнение и отредактируйте cron файл `crt`, в котором, прописаны оптимальные значения автозапуска.

Во время установки, очень важно понимать, где и какие пути вы прописываете, ибо ни у каждого свои.

## Описание модулей системы

После установки плагина в Wordpress появится отдельный модуль Newstrack, доступный из главного меню.



- Dashboard – вывод информации о последних обновленных фидах, лицензии, дате запуска роботов и др.
- Feeds – настройка фидов и параметров экспорта
- External Blogs – настройка внешних блогов (блогов куда будет публиковаться контент)
- Content Parsers – настройки шаблонов обработки (парсеров)
- Processing Templates – настройка шаблонов экспорта (обработки)
- Proxies – работа с прокси для экспорта
- Tools – Настройки User Agent, прокси, опции отладки SQL
- Log – просмотр логов работы роботов сбора и экспорта.

## Модуль Feeds

Модуль Feeds предназначен для управления RSS/ATOM потоками и их группами, а в частности – добавления, редактирования, удаления . Также в этом модуле настраиваются параметры экспорта фидов во внешние блоги.

Check	Title	Added	Exports	Atcls	Action
<input type="checkbox"/>	Computers				
<input type="checkbox"/>	APC Magazine	2007-12-06 17:10:38	0	134	Exports Edit Articles
<input type="checkbox"/>	BetaNews.Com	2007-12-06 17:10:09	1	278	Exports Edit Articles
<input type="checkbox"/>	CNET News.com - Business Tech	2007-12-06 17:10:12	1	403	Exports Edit Articles

Модуль отображает в табличном виде следующую информацию

- Check – поставьте отметку для выполнения операции удаления
- Title – название группы или фида
- Added – дата добавления фида в систему
- Exports – количество привязанных внешних блогов
- Atcls - количество статей на данный момент
- Action – действие
  - [Exports] - настройка экспорта
  - [Edit] – редактировать параметры фида
  - [Articles] – просмотр статей фида


### Фильтр Root Group

Если фидов очень много (древовидная иерархия организации фидов и групп не позволила реализовать удобный метод многостраничного отображения ) – Вы можете показать только детей определенной группы – Выберите родительскую группу из выпадающего списка Root Group. При этом будут показаны только фиды и группы, которые являются ее «детьми».

### Операции

- Для добавления нового фида нажмите [Add Feed]
- Для импорта OPML файла нажмите [Import OPML]
- Для удаления Выбранных фидов (и/или групп) нажмите [Delete Selected]
- Для редактирования атрибутов фида или группы нажмите Edit в выбранной строке

### Добавление нового фида (редактирование существующего)

	<input type="button" value="Save"/>	<input type="button" value="Reset"/>	<input type="button" value="Cancel &amp; Return"/>	<input type="button" value="Detect"/>	<input type="button" value="Preview"/>	<input type="button" value="Exports"/>
						
<b>Feed Category:</b>	IT Russian <input type="button" value="v"/> / <input type="text"/>					
<b>Feed URL:</b>	<input type="text" value="http://feeds.feedburner.com/internetno"/>					
<b>Authentication:</b>	No auth <input type="button" value="v"/> Login: <input type="text"/> Password: <input type="text"/>					
<b>Feed Title:</b>	<input type="text" value="Интернетные штучки"/>					
<b>Feed Title URL:</b>	<input type="text" value="http://internetno.net"/>					
<b>Feed Description:</b>	<input type="text" value="веб2.0, социальные сети, ajax, обзоры интересных сайтов"/>					
<b>Tags:</b>	<input type="text"/>					
<b>Feed Image URL:</b>	<input type="text" value="http://internetno.ru/logo_64.gif"/>					
<b>Added:</b>	[2007-06-28 14:01:39]					
<b>Last Poll:</b>	[2007-12-06 17:10:35 - 2007-12-06 17:10:35]					
<b>Status Code:</b>	0					

Здесь Вы задаете Категорию/Группу фида (**Feed Category**) , URL фида (**Feed URL**), параметры аутентификации (если есть) (**Authentication**) , а так же описательные атрибуты – название фида (**Feed Title**), адрес сайта(**Feed Title URL**), описание (**Feed Description**) и теги (**Tags**). **Feed Image URL** – адрес логотипа фида.

Остальные строчки являются информационными – когда был добавлен фид, дата последнего обновления и код статуса (если не ноль – произошла ошибка).

Для быстрого добавления достаточно ввести адрес сайта и нажать [Detect] если сайт содержит meta теги с ссылками на его фиды – будет выбран первый фид и автоматически просканирован на

атрибуты. Вы можете также ввести только адрес фида, после [Detect] фид будет обработан и его заголовки будут автоматически подставлены в соответствующие поля.

Кнопка [Preview] служит для предварительного просмотра RSS фида (экспериментально).

[Exports] ведет непосредственно в раздел настройки экспорта данного фида. Кнопка доступна только при редактировании уже добавленного фида.

При добавлении фида желательно указывать группу назначения т к фидами рассортированными на группы, легче управлять. Вы можете создать нужную группу/подгруппу сразу, во время добавления фида.

**Замечание** – имейте ввиду, что лучше добавлять прямые источники информации . Старайтесь избегать ссылки на feedburner , т к такие фиды могут содержать нежеланный контент в виде рекламы, кнопок feedflare, специфичных URL (для учета статистики) , и других нежелательных элементов. Если Вы в дальнейшем будете ставить ссылки на feeburner items (как бы соблюдая правила приличия и оставляя ссылку на источник), то они будут выглядеть, мягко говоря, несуразно.

Когда Вы ввели/изменили все необходимые данные фида – нажмите [Save],

[Cancel & Return] вернет вас назад к списку фидов.

[Reset] сбросит форму в начальное состояние

*В других модулях кнопки [Save], [Reset] и [Cancel & Return] выполняю, т е же смысловые функции.*

## Удаление группы Фидов

Check	Title
<input type="checkbox"/>	PC News
<input type="checkbox"/>	Интернетные штучки
<input type="checkbox"/>	Новости Компьюлента
<input checked="" type="checkbox"/>	Хабрахабр: интернет

Поставьте отметки напротив тех фидов, которые Вы хотите удалить и нажмите [Delete Selected], Вы попадете на страницу подтверждения удаления, где будете должны поставить отметку напротив Check here to confirm your action для подтверждения удаления. Нажмите [Delete] для финального удаления выделенных объектов.

**Are you sure you want to delete 1 feeds and 0 Feed Groups ?**

**Feeds:** Хабрахабр: интернет

Check here to confirm your action

## Создание Группы (Контейнера) Фидов

**Create Feed Group**

**Parent Feed Category:**

**Feed Category Name:**

- Выберите родительскую группу (Parent Feed Category)
- Введите название Группы (Feed Category Name)

## Exports – настройка экспорта

Для перехода на страницу добавления блогов для экспорта нажмите [Exports] в соответствующей строке в списке фидов.

Данная страница предполагает то, что Вы уже настроили парсер обработки содержимого фида (Content Parser), Шаблон экспорта (Processing Template) и сам блог для экспорта (External Blog).

**Feed Category:** IT Russian

**Feed URL:** <http://www.compulenta.ru/rss.xml>

**Feed Title:** [Новости Компьюлента](#)

---

Return to Feeds    Edit Feed    Create New Export

Active	Title	Blog	Template	Accessed	Added	Action
<input checked="" type="checkbox"/>	Компьюлента В newstrack	Newstrack новости рунета	Компьюлента технологии	--	2007-04-30 05:22:49	<a href="#">Edit</a>
<input checked="" type="checkbox"/>	Компьюлента технологии	Topnews	Компьюлента технологии	--	2007-04-29 01:44:30	<a href="#">Edit</a>

После перехода на страницу настройки экспортов, Вы увидите список блогов, куда экспортируются статьи выбранного фида. Блогов может быть сколько угодно. Те Вы можете экспортировать одну и ту же статью несколько раз. Естественно предполагать, что для нового фида – список экспортов будет пустым. Для добавления экспортного Блога нажмите [Create New Export] – после чего Вы попадете на страницу выбора атрибутов экспорта:

**Feed Category:** IT Russian

**Feed URL:** <http://www.compulenta.ru/rss.xml>

**Feed Title:** [Новости Компьюлента](#)

**Export Name:**

**Target Blog:**  [Blog Settings](#)

**Processing Template:**  [Edit Template](#)

**Parser Template:**  [Edit Parser](#)

**Export Time Offset:**

**Active:**

**Added:** [2007-04-30 05:22:49]

[Save](#)    [Reset](#)    [Cancel & Return](#)    [Test](#)

**Test Parameters :** Articles to preview :  Match:  against

- **Export Name** – название для списка (исключительно для описания)
- **Target Blog** – целевой блог – т.е. куда будут экспортироваться статьи – список из уже введенных и сконфигурированных внешних блогов (см модуль External Blogs ) – [Blog Settings] открыть отдельно страницу настройки выбранного блога (для удобства)
- **Processing Template** – шаблон обработки/экспорта / [Edit Template] – открыть страницу редактирования (используется для быстрого доступа ) выбранного шаблона
- **Parser Template** – шаблон обработки содержимого. [Edit Parser] – редактировать выбранный парсер (используется для быстрого доступа )
- **Export Time Offset** – смещение времени в целевом блоге (на данный момент не реализовано)
- **Active** – экспорт активен (иногда полезно отключить экспорт, не удаляя его)
- **Added** – когда данный экспорт был добавлен (информация только при редактировании существующего экспорта)

[Test] служит для тестирования параметров экспорта с выбранными настройками.

### Тестирование параметров экспорта с выбранными настройками

Тестирование производится с использованием находящихся (скачанных ранее) статей,

Параметры тестирования (**Test Parameters**)

**Articles to Preview** – количество статей (отсортированных в обратном порядке попадания в систему – самая свежая – первая и т. п.) - по умолчанию 5.

Опционально параметры фильтрации : **Match** (Category|Title|Subtitle|Body) - фильтровать где выбранное поле попадает под условия регулярного выражения (**against**)

При тестировании выбирается **Articles to Preview** статей из базы (которые совпадают с фильтром – если задан), затем к каждой их статей применяется выбранный Content Parser, а затем шаблон экспорта (Processing Template). Результат обработки выводится в новое окно, где зеленым цветом выбраны статьи из фида, которые готовы для экспорта в целевой блог.

Замечание: статья внутри системы описывается определенными переменными (полями базы данных), названия которых используются при создании шаблонов экспорта и парсеров контента. Бо большому счету они являются аналогами полей в RSS/ATOM фиде. Наиболее важными являются

- **Title** – заголовок статьи
- **Subtitle** – описание (анонс)
- **Body** – полный текст статьи – зачастую отсутствует или же статья целиком попадает в subtitle
- **Category** – список тегов – категорий статьи

Полный список полей приведен в разделе, где описан язык обработки контента и шаблонов экспорта.

## Модуль External Blogs

Модуль External Blogs предназначен для управления внешними целевыми блогами. Целевой блог в контексте системы это тот сайт, который принимает публикуемые новости по протоколу XML-RPC. Для настройки целевого блога достаточно знать адрес его xml-rpc точки доступа, тип блога и параметры авторизации. В настоящее время система поддерживает только блоги с MetaWeblog API (Wordpress, Movable Type) и Blogger (ATOM с Google авторизацией).

Главная страница модуля отображает список сконфигурированных блогов в табличном виде.

Check	Title	Description	Last Access	Action
<input type="checkbox"/>	Newstrack новости рунета		2007-06-03 07:46:23	<a href="#">Edit</a>
<input type="checkbox"/>	Topnews		2007-06-03 08:46:46	<a href="#">Edit</a>
<input type="checkbox"/>	BlogForward : Money		2007-06-01 16:31:17	<a href="#">Edit</a>
<input type="checkbox"/>	Midget Cars		2007-05-13 19:01:23	<a href="#">Edit</a>
<input type="checkbox"/>	Extra Technology News		2007-06-01 12:16:24	<a href="#">Edit</a>

Для редактирования настроек блога нажмите [Edit] в соответствующей строке. Для создания нового – нажмите [New Blog] после чего Вы попадете на страницу создания/редактирования параметров внешнего блога.

**Blog Title:**

**Blog URL:**

**API Type:**

**Blog Id:**

**Authentication:** Login:  Password:

**XML-RPC:**

**Description:**

**Last Access:** [2007-06-03 07:46:23]

Описание атрибутов:

- Blog Title – название блога в системе
- Blog URL – адрес блога
- API Type – тип API для доступа
- Blog Id – зарезервировано
- Authentication – параметры авторизации (для Blogger это Ваш google account)
- XML-RPC – адрес точки доступа.

### Замечание по настройке XML-RPC адреса

Обычно у Wordpress блогов, XML-RPC адрес формируется из адреса самого блога с добавленным в конце файлом xmlrpc.php (пример <http://www.newstrack.ru/xmlrpc.php> )

У Blogger все по другому. Для определения адреса XML- RPC Blogger блога нужно зайти браузером на сам блог и посмотреть его исходный текст (View Source). В разделе head Вы найдете link тэг который содержит этот адрес, он называется service.post и выглядит приблизительно так :

```
<link rel="service.post" type="application/atom+xml" title="Extra Technology News - Atom" href="http://www.blogger.com/feeds/6416281110785812402/posts/default" />
```

Адрес <http://www.blogger.com/feeds/6416281110785812402/posts/default> и будет искомым XML-RPC коннектором для нашего Blogger блога.

Кнопка [Detect] попытается определить атрибуты блога самостоятельно , однако она не всегда корректно находит адрес XML-RPC так что этот параметр лучше задавать вручную.

## Модуль Content Parsers

Данный модуль, равно как и модуль Processing Templates позволяют создавать и редактировать скрипты обработки содержимого фидов, перед тем как они попадут на целевой блог. Модуль Content Parsers отображает уже сконфигурированные парсеры контента, позволяет редактировать их и добавлять новые.

[New Parser](#)

Check	Title	Description	Action
<input type="checkbox"/>	Korrespondent.net	Parse website Korrespondent.net website	<a href="#">Edit</a>
<input type="checkbox"/>	Podrobnosti.ua	Podrobnosti.ua parser	<a href="#">Edit</a>
<input type="checkbox"/>	Obkom.net.ua		<a href="#">Edit</a>
<input type="checkbox"/>	лента ру		<a href="#">Edit</a>
<input type="checkbox"/>	Компьюлента		<a href="#">Edit</a>
<input type="checkbox"/>	habrahabr		<a href="#">Edit</a>
<input type="checkbox"/>	Venturebeat		<a href="#">Edit</a>
<input type="checkbox"/>	cnet		<a href="#">Edit</a>

Для редактирования настроек парсера нажмите [Edit] в соответствующей строке. Для создания нового – нажмите [New Parser] после чего Вы попадете на страницу создания/редактирования параметров парсера.

Описание атрибутов/параметров:

- **Parser ID** – уникальный идентификатор парсера в системе
- **Parser Name** - имя парсера в системе
- **Description** – описание
- **Processing Rules** – код обработки на XML языке, который описан далее.

**Parser ID:**

**Parser Name:**

**Description:**

**Processing Rules:**

```
<template>
<ifempty var="body">
<fetch/>
<iconv/>
<ifmatch>
<regex><![CDATA[!<div\s+id="article">(.*?)<p><a.+?Обсудить!msi]]
></regex>
<assign var="body" val="$1" />
</ifmatch>
<replace to="" var="body">
<regex><![CDATA[!<div\s+class="info">.+?</div>!msi]]></regex>
</replace>
<replace to="" var="body">
<regex><![CDATA[!(<div[^>]*>|</div>!msi)]></regex>
</replace>
<update var="body" />
```

## Content Parser Test

**Feed to test:**

**Article to test:**

### Тестирование кода парсера (Content Parser Test)

Выберите фид, для которого Вы пишете парсер (**Feed to test**). После выбора фида в выпадающий список **Article to test** будут загружены 10 последних статей данного фида. Выберите статью с которой вы хотите протестировать фид и нажмите [Test]. Результат тестирования откроется в новом окне. (включая возможные ошибки XML ). Результатом теста является массив полей после обработки парсером.

## Язык написания контент парсеров

Собственно языком обработки является набор вложенных XML тегов, которые выполняются последовательно. Ключевым понятием при написании парсера, является применение инструкций совпадения с использованием регулярных выражений. Операции, которые соответствуют тегам, выполняются последовательно одна за другой. Язык не поддерживает циклов, однако поддерживает условия, а соответственно вложения.

### Структура шаблона

```
<template>
```

... Набор тегов обработки ...

```
</template>
```

Тег обработки обычно оперирует рядом атрибутов параметров, которые дополняют операцию. Параметры задаются как в виде атрибутов тега (например `<iconv from="enc" to="to_encoding" />`) или же полноценные текстовые теги : `<regex><![CDATA[ ]]></regex>`.

### Как происходит обработка.

При запуске парсера, подразумевается что он будет оперировать рядом predetermined информационных переменных соответствующих названию полей записи из базы данных, в которые загружена обрабатываемая статья. Результатом обработки является тот же набор переменных.

Названия переменных:

title	Заголовок из RSS/ATOM файла (содержимое тега <title>)
subtitle	Подзаголовок (содержимое тега <description>)
body	Тело статьи (<content:encoded>) часто пустое до обработки
url	Адрес статьи (содержимое тега <link>)
img	Адрес enclosure (содержимое тега < enclosure >) – практически не используется
d_date	Дата новости (в формате MYSQL DATETIME ("Y-m-d H:i:s")) (<pubDate>, <dc:date>)
feed_id	Внутренний код фида
digest	Внутренний уникальный код статьи (32 символа)
link	Не используется
author	Автор статьи (содержимое тега <author> или < dc:creator>)
category	Категории или теги статьи (содержимое тега <category>)
comments	Адрес (URL) для комментариев (содержимое тега <comments>)
comments_rss	Адрес фида (URL) для комментариев (содержимое тега < wfw:commentRSS>)
feed_title	Название фида
feed_title_url	Адрес сайта фида (главной страницы)
rssurl	Адрес фида
feed_notes	Описание фида
feed_image_url	Адрес логотипа сайта из фида
timezone	Смещение временной зоны (задана из настроек блога)

В процессе работы парсера можно создавать другие произвольные переменные, которые могут служить вспомогательными.

Наиболее естественной задачей парсера является трансформация переменных статьи их одного вида в другой, готовый для экспорта на другой сайт.

### Список управляющих тегов и их атрибутов

Тэг	Атрибуты/под тэги	Описание
fetch		Считать вебстраницу по адресу переменной url (скачать исходный текст статьи) . Результат помещается во внутренний буфер обработки buf.
skip		Инструктирует систему остановить обработку статьи и игнорировать ее. Удобно если в результате анализа выяснилось что статья не соответствует тематике.
stop		Полностью остановить обработку фида
iconv	<b>from</b> - из какой кодировки <b>fail</b> – какую кодировку выбрать если не удалось опередить автоматически (на основе http заголовков и мета тегов) <b>to</b> – в какую кодировку (utf-8) по умолчанию	Конвертирование кодировки внутреннего буфера обработки (buf). Применяется для нормализации кодировки и перевода в нужную (практически всегда utf-8) для дальнейшей обработки.  обычно задается только атрибут fail – если вы уверены что сайт выдает неверную кодировку в заголовках.  <b>Пример:</b> <iconv fail="cp1251" /> - если не удастся опередить кодировку страницы, будет использована cp1251 (Windows) и буфер будет переведен в кодировку utf-8 (по умолчанию)
update	<b>var</b> - переменная или список переменных, разделенных запятой.	Производит модификацию полей базы данных из переменных(ой) заданных в поле <b>var</b> . Обычно является одним из завершающих тегов в процессе обработки. Завершает модификацию обрабатываемой записи.  <b>Пример:</b> <update var="body" />
ifmatch	<b>rexp</b> - регулярное выражение <b>text</b> - текст для сравнения <b>not</b> - флаг инверсии	Выполняет сравнение текущего буфера с регулярным выражением, заданном в rexp (может задаваться в виде подтега) или же с текстом заданном в text . (можно использовать только один из параметров) Если результат сравнения положительный (или отрицательный – если задан атрибут <b>not="1"</b> ) то выполняется набор подтегов втнури ifmatch.  Данный тег является ключевым в процессе выделения

		<p>содержимого из буфера. Т к здесь Вы задаете регулярные выражения выделяющие текст, а затем с помощью тега &lt;assign&gt; назначаете результат в рабочие переменные.</p> <p>Параметр &lt;regex&gt; задается в виде preg_match совместимого регулярного выражения т е с соблюдением разделителей и опций . Данный параметр следует задавать в виде подтега , задавая регулярное выражение в CDATA. После отработки регулярного выражения – его результаты могут быть в дальнейшем использованы в атрибутах тега <b>assign</b> в виде макросов: \$0, \$1, \$2 .. \$9.</p> <p><b>Пример:</b>  <pre>&lt;ifmatch&gt; &lt;regex&gt; &lt;![CDATA[!&lt;p\s+class="maintext"&gt;(.+?)&lt;table.+?javascript:print()!msi]]&gt; &lt;/regex&gt; &lt;assign var="body" val="\$1" /&gt; &lt;update var="body" /&gt; &lt;/ifmatch&gt;</pre></p> <p>Производится сравнение буфера с регулярным выражением, заданным в regex.</p> <p>Если результат позитивен, то содержимое первого совпадения будет помещено в переменную body (тег &lt;assign&gt;)</p> <p>Затем данное поле будет записано в запись базы данных для дальнейшего использования. Фактически данная операция выделяет содержимое статьи из буфера и помещает его в базу данных.</p>
assign	<p><b>var</b> - целевая переменная</p> <p><b>val</b> - текст или выражение (может быть как атрибутом так и подтегом )</p> <p><b>where</b> - метод добавления</p>	<p>Копирует текстовое выражение заданное параметром <b>val</b> в переменную <b>var</b>. Если <b>var</b> опущен, то подразумевается внутренний буфер обработки.</p> <p>Атрибут <b>where</b> задает как текст будет помещен в целевую переменную.</p> <p>Если атрибут он опущен, то текст заменит содержимое полностью, если он равен <i>pre</i> , то текст будет добавлен вначале содержимого целевой переменной, и если атрибут равен <i>append</i> то текст будет добавлен к концу содержимого целевой переменной.</p> <p>Следует заметить, что val может задаться в виде подтега, а не атрибута.</p> <p>В val можно записывать макросы, которые будут заменены результатами работы регулярного выражения последнего использованного тега &lt;ifmatch&gt;. Макросы выглядят, так же, как и на языках Perl/PHP: \$0, \$1, \$2 и т п.</p>

replace	<p><b>var</b> - обрабатываемая переменная</p> <p><b>regex</b> - регулярное выражение для замены</p> <p><b>to</b> - текст замены</p>	<p>Выполняет замену текста определенного регулярным выражением <b>regex</b> на содержимое параметра <b>to</b> в переменной, заданной атрибутом в переменной <b>var</b>.</p> <p>Фактически выполняется операция:</p> <pre>var = preg_replace(regex, to, var)</pre> <p>если <b>var</b> опустить, то будет производиться обработка рабочего буфера.</p> <p><b>Пример:</b>  <pre>&lt;replace to="" var="subtitle"&gt; &lt;regex&gt;&lt;![CDATA[!&lt;a[^\&lt;^&gt;]+class="habracut".+?&lt;/a&gt;!msi]]&gt;&lt;/regex&gt; &lt;/replace&gt;</pre> </p> <p>- очистка всех ссылок по шаблону в переменной subtitle</p>
ifempty	<p><b>var</b> - тестируемая переменная</p>	<p>Тег проверки содержимого переменной заданной атрибутом var, если она пуста, то выполняется набор вложенных тегов – инструкций.</p> <p>необходим для того что бы ограничить использование парсера если он уже однажды обработал статью при повторном использовании.</p> <p>Часто является основным условием перед обработкой контента.</p> <p><b>Пример:</b>  <pre>&lt;template&gt; &lt;ifempty var="body"&gt; &lt;fetch/&gt; &lt;iconv/&gt; ... &lt;update var="body" /&gt; &lt;/ifmatch&gt; &lt;/ifempty&gt; &lt;/template&gt;</pre> </p> <p>Производится проверка переменной <b>body</b>, если она пуста (<b>&lt;ifempty&gt;</b>), происходит загрузка содержимого статьи (<b>&lt;fetch /&gt;</b>), ее конвертация в кодировку utf-8. Обработка ... и запись в базу данных. В дальнейшем, при повторном вызове парсера для данной статьи, обработка не будет выполняться, т.к. мы записали результат обработки в базу (<b>&lt;update var="body"/&gt;</b>) и <b>&lt;ifempty&gt;</b> не сработает.</p>

## Пример сложного парсера контента

- <template>
- <replace to="" var="title"> <!--замена текста в виде "что угодно:" в поле title -->
  - o <regex><![CDATA[!^.+?;!msi]]></regex>
- </replace>
- <update var="title" /> <!--сохранение поля title в базе -->
- <replace to="" var="subtitle"> <!--очистка ссылок в поле subtitle (to="" !!!)-->
  - o <regex><![CDATA[!<a[^<>]+class="habracut".+?</a>!msi]]></regex>
- </replace>
- <update var="subtitle" /> <!--сохранение поля subtitle в базе -->
- <ifempty var="body"> <!--далее – выполнять вложенные инструкции если body пустой -->
  - o <fetch/> <!--скачать статью -->
  - o <iconv/> <!--перевести ее в utf-8 -->
  - o <ifmatch> <!--выделить текст согласно рег выражению regex -->
    - <regex><![CDATA[!<div\s+class="groups\_topic\_text">(.\*?)(<br\s+clear="left">|<div\s+class="smallcom">)]!msi]]></regex>
    - <assign var="body" val="\$1" /> <!--сохранить результат в переменной body -->
  - o </ifmatch>
  - o <update var="body" /> <!--сохранение поля body в базе -->
- </ifempty> <!-- конец ifempty -->
- </template> <!-- конец шаблона -->

В поставку входит несколько парсеров контента, готовых к использованию. Вы легко сможете самостоятельно разобраться с языком шаблонов обработки, изучив эти примеры.

## Модуль Processing Templates

Данный модуль позволяет создавать и редактировать скрипты пост обработки статей перед экспортом на целевой блог. Модуль Processing Templates отображает уже сконфигурированные обработки, позволяет редактировать их и добавлять новые.

Данный модуль практически повторяет функциональность модуля Content Parsers.

Check	Title	Description	Action
<input type="checkbox"/>	Украина keywords		<a href="#">Edit</a>
<input type="checkbox"/>	Украина		<a href="#">Edit</a>
<input type="checkbox"/>	Лента ру Технологии		<a href="#">Edit</a>
<input type="checkbox"/>	Лента ру Украина		<a href="#">Edit</a>
<input type="checkbox"/>	лента ру спорт		<a href="#">Edit</a>
<input type="checkbox"/>	Компьюлента технологии		<a href="#">Edit</a>
<input type="checkbox"/>	экспорт темы интернет		<a href="#">Edit</a>

Для редактирования настроек шаблона нажмите [Edit] в соответствующей строке. Для создания нового – нажмите [New Template] после чего Вы попадете на страницу создания/редактирования параметров шаблона.

Причина разделения парсеров содержимого и экспорта проста – Вы можете использовать один парсер контента для отдельного фида, однако иметь возможность по-разному форматировать экспорт для разных блогов, применяя разные шаблоны обработки. (например обрабатывать один фид, а затем с помощью шаблонов обработки разделять статьи по тематике)

Описание атрибутов/параметров:

- **Template Name** - имя шаблона в системе
- **Description** – описание
- **Processing Rules** – код обработки на XML языке, который описан далее.

**Template Name:**

**Description:**

**Processing Rules:**

```
<template>
<actions>
<act match="buf">
<replace to="%category%" />
<expand/>
<striptags />
</act>
<act match="buf">
<ifmatch not="1">
<regexp><![CDATA[!(Сделки|игры|Ноутбуки|Пираты|видеокамеры|
<skip/>
</ifmatch>
</act>
<act match="category">
<replace to="Технологии" />
</act>
```

## Язык написания шаблонов обработки

Язык структурно близок к языку обработки контента, однако содержит гораздо больше управляющих и условных тегов. Как и в шаблоне обработки контента применяется набор вложенных XML тегов, которые выполняются последовательно. Ключевым понятием при написании шаблона, является применение инструкций совпадения с использованием регулярных выражений. Операции, которые соответствуют тегам, выполняются последовательно одна за другой. Язык не поддерживает циклов, однако поддерживает условия, а соответственно вложения. Язык оперирует тем же набором переменных, что и язык обработки контента. Те на вход получает набор переменных, результат обработки уже непосредственно экспортируется на целевой блог.

## Список управляющих тегов и их атрибутов

Тэг	Атрибуты/под тэги	Описание
<b>actions</b>		Начало группы обработки (наподобие begin .. end в паскале или фигурных скобок c/php)
<b>skip</b>		Инструктирует систему остановить обработку статьи и игнорировать ее. Удобно если в результате анализа выяснилось что статья не соответствует тематике.
<b>stop</b>		Полностью остановить обработку фида
<b>act</b>	<b>match</b> - список переменных для обработки (через запятую) <b>present</b> - флаг условие наличия поля <b>empty</b> - флаг – если поле пустое	<p>Выполнить группу действий над переменными заданными списком <b>match</b> (title, subtitle, body) и т.п. по умолчанию (если match не задан) подразумевается subtitle и body</p> <p>Внутри тега задается набор тегов обработки, которые описаны далее.</p> <p>Если задан атрибут <b>present</b> то набор операция будет выполняться если переменная непустая (т.е. содержит что то)</p> <p><b>Empty</b> является антонимом атрибута <b>present</b> – т.е. выполнять если пустой.</p> <p>В дальнейшем для всех вложенных в act операций переменная из match является переменной по умолчанию (над которой производится операция). Переменная может быть задана непосредственно с использованием атрибута <b>var</b></p>
<b>notags</b>	<b>var</b> - имя переменной	<p>Удалить теги в текущей переменной (заданной родительским тегом act или же в переменной заданной атрибутом var).</p> <p>Аналог PHP функции strip_tags</p>
<b>regex, rexp</b>	<b>var</b> - имя переменной <b>from</b> - регулярное выражение <b>to</b> - текст замены	<p>Выполняет var = preg_replace(from,to, var) т.е. замену согласно регулярному выражению from.</p> <p>В <b>to</b> могут быть использованы макросы: \$0, \$1, \$2 .. \$9. Соответствующие скобкам в регулярном выражении <b>from</b></p>
<b>assign</b>	<b>var</b> - имя переменной <b>val</b> - имя переменной	<p>Выполняет копирование содержимого переменной <b>val</b> в <b>var</b> (или в текущую, по контексту) или попросту присваивание <b>var = val</b></p>

<b>replace</b>	<b>var</b> - имя переменной <b>from</b> - текст <b>to</b> - текст замены	Выполняет замену текста <b>from</b> на <b>to</b> в переменной <b>var</b> , или в текущей переменной по контексту.  Аналог <code>var = str_replace(from, to, var)</code> , если <b>from</b> не задать, то текст из <b>to</b> , заменит содержимое переменной полностью (операция присваивания)
<b>append</b>	<b>var</b> - имя переменной	Добавить в текущую переменную содержимое тега ( операция добавить в конец)
<b>prepend</b>	<b>var</b> - имя переменной	Добавить в текущую переменную содержимое тега ( операция вставить в начало текста)
<b>clean</b>	<b>var</b> - имя переменной	Очистка переменной
<b>fix</b>	<b>var</b> - имя переменной	Производит коррекцию всех href и src атрибутов в тексте, те заменяет относительные адреса ы на полные (относительно адреса переменной url)
<b>block</b>	<b>var</b> - имя переменной <b>tag</b> - название тега	Блокировать теги (img) в которых <b>src</b> совпадает со списком слов из подтегов данного тега. Удобно для фильтрации рекламы и т п.  Пример: <pre>&lt;block&gt; &lt;pheedo&gt;pheedo\.com&lt;/pheedo&gt; &lt;pheedo&gt;feedflare&lt;/pheedo&gt; &lt;/block&gt;</pre> Фильтрует все картинки, которые содержат <code>pheedo.com</code> или <code>feedflare</code>
<b>ifmatch</b>	<b>var</b> - имя переменной <b>regex</b> - регулярное выражение <b>text</b> - текст <b>not</b> - флаг отрицания	тег условного выполнения Выполняет сравнение переменной с регулярным выражением, заданном в <code>regex</code> (может задаваться в виде подтега) или же с текстом заданном в <code>text</code> . (можно использовать только один из параметров) Если результат сравнения положительный (или отрицательный – если задан атрибут <code>not="1"</code> ) то выполняется набор подтегов внутри <code>ifmatch</code> .  Параметр <code>&lt;regex&gt;</code> задается в виде <code>preg_match</code> совместимого регулярного выражения т е с соблюдением разделителей и опций . Данный параметр следует задавать в виде подтега , задавая регулярное выражение в CDATA.

<b>expand</b>	<b>var</b> - имя переменной	<p>Операция автоподстановки переменных в тексте. Происходит замена фрагментов текста вида %abcd% на значения соответствующих переменных.</p> <p><b>Пример:</b> Текст текущей переменной: Source: &lt;a href="% url%"&gt;%title%&lt;/a&gt;</p> <p>url = <a href="http://mysite.com/link1.html">http://mysite.com/link1.html</a> title= Время собирать камни</p> <p>Будет преобразован в текст Source: &lt;a href="<a href="http://mysite.com/link1.html">http://mysite.com/link1.html</a>"&gt;Время собирать камни &lt;/a&gt;</p> <p>Данный тег используется совместно с тегami append/prepend/replace в конце трансформации т к позволяет манипулировать текстом, внедряя содержимое результатов других преобразований</p>
<b>getlinks</b>	<b>feed</b> - ID фидов из которых берется контент <b>limit</b> - сколько записей брать <b>random</b> - выбрать случайных random статей из limit	<p>Служит для вставки текстов и значение из других фидов, зарегистрированных в системе.</p> <p>выбирает из заданных фидов limit последних по дате записей , если задан random то выбирает из limit random случайных. Все записи попадают в макросы, доступные как %IN_FIELD% т е %lO_title%, %lO_url% , %l1_title% и т п, которые можно использовать в теге append. После добавления макросов в текст обязательно выполняем <b>expand</b>.</p> <p>Расширенный пример использование можно найти здесь: <a href="http://www.lordtime.com/forum/viewtopic.php?f=2&amp;t=3">http://www.lordtime.com/forum/viewtopic.php?f=2&amp;t=3</a></p>
<b>nolinks</b>	<b>var</b> - имя переменной	Операция удаления всех ссылок в тексте. Подссылочные тексты остаются (тексты, теги, изображения). Тем самым получаем текст без ссылок,
<b>nofollow</b>	<b>var</b> - имя переменной	Добавление всем ссылкам атрибута rel="nofollow", делаем внешние ссылки запрещенными для индексирования (дабы избежать чрезмерного цитирования с оригинальных сайтов)
...		В системе возможно функционирование других тегов, которые появляются в результате использования плагинов. См далее.

Заметим, что можно использовать не только predetermined переменные, которые соответствуют полям базы данных, но и произвольные, которые будут работать как временные. Вы

можете вырезать части текста для обработки и сохранять их во временных переменных для дальнейшей трансформации при необходимости.

## Пример

- <template>
  - o <actions>
    - <act match="buf"> <!-- работаем с временной переменной buf -->
      - <replace to="%category%" /> <!-- помещаем туда текст %category% -->
      - <expand/> <!-- производим автозамену переменных в buf - там будет текст переменной -->
      - <striptags /> <!-- чистим html теги -->
    - </act>
    - <act match="buf"> <!-- снова работаем с временной переменной buf -->
      - <ifmatch not="1"> <!-- условие с отрицанием - далее рег выражение для сравнения-->
        - o <rexp><![CDATA[!(Технологии|Интернет|Игры)!umsi]]></rexp>
        - o <skip/> <!-- если buf не совпало с регулярным выражением - пропускаем статью -->
      - </ifmatch>
    - </act>
    - <act match="category"> <!-- теперь - работаем с переменной category-->
      - <replace to="Технологии" /> <!-- назначаем ей текст Технологии -->
    - </act>
    - <act match="subtitle" present="1"> <!-- работаем с subtitle если он заполнен -->
      - <clean/> <!-- чистим от HTML мусора -->
      - <expand/> <!-- растягиваем все макросы (на всякий случай)-->
      - <fix/> <!-- чиним ссылки-->
    - </act>
    - <act match="body" present="1"> <!-- работаем с body если он заполнен -->
      - <clean/> <!-- чистим от HTML мусора -->
      - <append><![CDATA[<p>Источник: <a href="%url%" target="\_blank">%feed\_title%</a></p>]]></append> <!-- добавили ссылку на источник прямо в текст статьи -->
      - <expand/> <!-- растягиваем все макросы (url, feed\_title) -->
      - <fix/> <!-- чиним ссылки-->
    - </act>
    - o </actions>
  - </template>

## Поддержка плагинов

Newstrack поддерживает расширение функциональности за счет использования внешних плагинов, написанных на языке PHP

Примеры плагинов можно найти в папке `nst/plugins`

### Структура папки `nst/plugins`

Файлы плагинов вида `Filename.N.php`

Где,

- `Filename` - имя плагина (уникально)
- `N` порядковый номер (могут совпадать, инициализируются в последовательности от меньшего до большего)

Плагин представляет собой файл класса, который написан на PHP. При запуске newstrack, создается экземпляр данного класса. Во время работы вызываются методы с predetermined именами. На данный момент поддерживаются два метода – один вызывается когда встречен незнакомый тег в шаблоне парсера, а второй аналогично в шаблоне экспорта.

Для шаблона парсера метод описан следующим образом:

- `function custom_parser_tag($data)`

Для шаблона экспорта:

- `function custom_export_tag($data)`

параметры рассмотрим на прмере:

Допустим мы хотим создать тег который добавит в заданное поле некий текст

Структуру плагина рассмотрим на создании дополнительного тега для `Export Template`

Generic код:

```
class plugin_export {  
  
    function plugin_export(&$nst, $priority = 0 ) {  
  
        $this->nst = $nst;  
        $this->pty = $priority;  
        $this->nst->log("Export plugin initialized");  
  
    }  
}
```

```
function custom_export_tag($data) {  
  
    global $db;  
    $export = &$data[0];  
    $node = &$data[1];  
    $arr = &$data[2];  
    $rv = &$data[3];  
  
    if($node->nodeName == 'custom' )    {  
        $what = $export->get_param($node, 'var');  
        $what = $what ? $what : 'title';  
        $arr[$what] .= ' [parsed]';  
    }  
  
}  
}
```

Данный плагин добавляет дополнительную функциональность в виде тега custom который поддерживает 1 атрибут **var** – название переменной в массиве текущей обрабатываемой статьи.

В качестве параметров в метод плагина передается массив \$data содержащий ссылки на 4 объекта-массива:

- \$export – объект выполняющий обработку (нужен для вызова некоторых его полезных функций)
- \$node – DOM объект нашего тега
- \$arr массив с переменными статьи (отсюда берем данные и туда же их возвращаем)
- \$rv = переменная для статуса выполнения – запишите туда -1 для остановки работы шаблона

Если Вы пишете свой плагин – обязательно включайте в него объявления внешних переменных, иначе Вы запутаетесь как возвращать обработанный текст.

В вышеприведенном примере мы проверяем что имя тега = custom, затем считываем значение атрибута var, и добавляем слово `[parsed]` в конец текста, указанного в переменной var.

Теперь если в шаблоне экспорта написать `<custom var="title" />` то в заголовке каждой статьи которая пройдет через шаблон добавится слово [parsed]

Само собой очевидно, что логику обработки можно значительно расширить, передавая параметры через атрибуты тегов.

Передача параметров удобна, но что если мы хотим задавать параметры сразу для каждого блога, не создавая каждый раз новый плагин с одним измененным параметром?

Ответ прост – при создании элемента экспорта (там где Вы выбираете блог и шаблоны – есть поля parser params и export params.

В них и имеет смысл вписывать параметры для парсера в виде paramname= value (1 на строчоку). Данные праметры будут доступны в том же массиве arr в виде \$arr[“param\_paramname”] (те добавлен префикс , дабы не смешивать переменные)

Теперь Вы можете легко подключать, напрмер, свой внешний обработчик текстов, и легко передавать ему параметры – просто написав обработчик в виде тега , а парметры задавать каждый раз для индивидуального блога.

Дл более детального ознакомления с системой плагинов – рекомендуем ознакомиться с плагином keywords, который берет список ключевых слов и выполняет замену в тексте на ссылки. Данный плагин демонстрирует гибкость подключения и простоту расширения newstrack.

## Полезные Ссылки

- Регулярные выражения: <http://ru.php.net/manual/ru/reference.pcre.pattern.syntax.php>
- Wordpress: <http://wordpress.org/>
- Русский Wordpress: <http://mywordpress.ru/>, <http://maxsite.org/>, <http://ru.wordpress.org>
- Форум поддержки: <http://www.lordtime.com/forum/viewforum.php?f=1>
- Новости проекта: <http://www.linkads.ru> и <http://www.newstrack.ru/category/newstrack/>
- Регулярные выражения (с чего начать): [http://phpclub.ru/detail/article/regexp\\_1](http://phpclub.ru/detail/article/regexp_1)
- Лучший фид ридер (был бы если б не бедность? и жестокие инвест. законы нашего гондураса): <http://www.newsalloy.com>

## Координаты автора

- Email: [lt@lordtime.com](mailto:lt@lordtime.com)
- Skype: v\_danylyuk
- ICQ: 44726644

Некоторые сайты, работающие на Newstrack:

- <http://www.newstrack.ru>
- <http://www.gamegeek.ru>
- <http://auto.blogforward.com>
- <http://extratech.blogspot.com>
- <http://windowsup.blogspot.com>
- <http://realty.blogforward.com>
- и много много других ...