

# Командно-строчные утилиты Communiware версия 0.912

В.Б. Вагнер

11 мая 2004 г.

## Содержание

<b>1</b>	<b>Настройка среды исполнения</b>	<b>4</b>
<b>2</b>	<b>cmwactl</b>	<b>4</b>
<b>3</b>	<b>dump_item</b>	<b>6</b>
3.1	Способы задания набора итемов . . . . .	7
3.1.1	Явное задание списка . . . . .	7
3.1.2	Дамп поддеревьев . . . . .	7
3.1.3	Дамп сайта . . . . .	8
3.1.4	Дамп по SQL-запросу . . . . .	8
3.2	Управление сохраняемой информацией . . . . .	9
3.3	Прочие ключи . . . . .	9
<b>4</b>	<b>load_item</b>	<b>9</b>
4.1	Управление проверкой синтаксиса . . . . .	10
4.2	Пакетные режимы и логгинг . . . . .	10
4.3	Отмена ограничений на изменение итемов . . . . .	11
4.4	Переименование итемов при загрузке . . . . .	11
<b>5</b>	<b>load_item.www</b>	<b>12</b>
<b>6</b>	<b>putitem</b>	<b>13</b>

<b>7</b>	<b>viitem</b>	<b>14</b>
7.1	Краткая справка по командам <code>vim</code> . . . . .	14
<b>8</b>	<b>dump_meta</b>	<b>14</b>
8.1	Задание явного списка метаобъектов . . . . .	15
8.2	Дамп онтологии сайта . . . . .	16
<b>9</b>	<b>load_meta</b>	<b>17</b>
<b>10</b>	<b>createserver</b>	<b>17</b>
10.1	Копирование сайтов . . . . .	18
<b>11</b>	<b>greptempl</b>	<b>19</b>
<b>12</b>	<b>clearpictures</b>	<b>19</b>

## Введение

Помимо web-интерфейсов Communiware располагает рядом интерфейсов командной строки, доступных из консоли той же машины, на которой работает Communiware-сервер. В состав этих утилит входят:

**cmwactl** управляет запуском, и остановкой Communiware-сервера

**dump\_item** сохраняет заданные итемы в текстовый файл, напоминающий по структуре электронный почтовый ящик.

**load\_item** загружает итемы, сохраненные утилитой *dump\_item* в базу данных Communiware.

**load\_item.www** загружает файл созданный *dump\_item* в удаленный Communiware-сервер, пользуясь стандартным интерфейсом веб-мастера.

**viitem** позволяет отредактировать указанные итем.

**dump\_meta** сохраняет коллекцию метасущностей (типов итем, типов связей и так далее) в XML-файле.

**load\_meta** загружает в базу данных Communiware XML-файл с описанием онтологии.

**putitem** Создает итем, получая информацию об атрибутах и связях из командной строки, а текст из файлов на локальном диске.

**createserver** Создает Коммунивер-сайт на текущем сервере.

**greptempl** ищет заданный текст во всех шаблонах данного сервера.

**clearpictures** является скорее служебным скриптом, предназначенным для запуска из `cron`, чем утилитой командной строки. Однако в некоторых случаях имеет смысл запускать эту утилиту вручную. Она находит все пустые каталоги и каталоги, которым не соответствует итем в базе данных, и удаляет их.

Все эти утилиты обладают набором общих черт. Они (за исключением *putitem*) используют стандартные соглашения командной строки Unix<sup>1</sup>.

В качестве имени пользователя `Communiware` утилиты используют имя пользователя Unix, от которого они работают, не требуя дополнительной авторизации. В случае, если имя Unix-пользователя не совпадает с именем пользователя `Communiware`, администратор `Communiware`-сервера должен поместить пару `unix-имя`-`Communiware-имя` в файл `$(CMWHOME)/conf/aliases`.

В ограниченных пределах эти утилиты позволяют работать с `Communiware`-базой, находящейся на машине, отличной от той, на которой они запускаются. Если база данных другой машины доступна по сети с использованием родного протокола для данного SQL-сервера, то утилиты будут работать с удаленной базой. При этом невозможно доступ к каталогу иллюстраций и бинарных файлов, если только он не смонтирован по NFS или аналогичному протоколу.

Для работы с удаленной базой данных требуется создать в `$(CMWHOME)/conf` файл с именем *имя.config*, имеющий тот же формат, что и `Communiware.config` и содержащий другие параметры соединения с базой данных.

При вызове утилиты ей указывается параметр `-с имя`, вызывающий использование указанного конфигурационного файла вместо конфигурационного файла по умолчанию.

---

<sup>1</sup>разбор параметров реализован через модуль `GetOpt::Std` языка Perl

## 1 Настройка среды исполнения

Все утилиты командной строки Communiware требуют наличия определенных переменных среды (environment)

Во-первых, это переменная CMWHOME, задающая корень иерархии директорий, в которых установлен Communiware. Изменяя значение этой переменной, можно выбирать между несколькими экземплярами Communiware, установленными на одной машине.

Во-вторых, требуются переменные окружения, задающие параметры клиентских библиотек выбранного SQL-сервера. Для Oracle это ORACLE\_HOME, задающая местоположение служебных файлов Oracle и NLS\_LANG, задающая параметры локализации клиента (которые могут не совпадать с параметрами локализации как сервера баз данных, так и web-сервера.

Для PostgreSQL аналогичную роль играют переменные PG\_LIB и PGCLIENTENCODING.

Следует отметить, что различные русские кодировки, поддерживаемые Communiware, не на 100% совпадают по набору используемых символов. Например, в кодировке windows-1251 присутствуют некоторые знаки препинания, отсутствующие в koï8-r. Поэтому, кодировка используемой утилитами load\_item и dump\_item при переносе сайта, должен учитывать возможность потери информации при двойной перекодировке.

С другой стороны, при работе с утилитой viitem кодировка должна соответствовать кодировке экранных шрифтов и клавиатуры.

Утилиты dump\_meta и load\_meta кодировко-независимы, так как используемый ими XML-формат всегда хранится в кодировке UTF-8, но из-за особенностей реализации перекодировки в PostgreSQL-версии, при работе с этими утилитами также рекомендуется выставлять кодировку, соответствующую кодировке хранения.

## 2 cmwactl

cmwactl представляет собой модифицированную версию скрипта arachectl, входящего в комплект Apache версии 1.3 и выше. Модифицированная версия, входящая в комплект Communiware позволяет работать сразу с обоими апачами — фронтэндом и бэкэндом. cmwactl написан на Perl, в отличие от оригинального arachectl, поэтому на нем

можно установить `suid`-бит. В стандартной установке `Communiware` он устанавливается именно так, и сам производит проверку, обладает ли запустивший его пользователь достаточными правами.

Синтаксис:

```
cmwactl [-f|-b] команда
```

Опции **-f** и **-b** задают экземпляр `Apache`, которому передается команда. `-f` означает, что мы управляем *только* фронтэндом, `-b` — только бэкэндом. По умолчанию `cmwactl` выполняет команду сразу над обоими апачами.

В отличие от всех остальных утилит `cmwactl` не поддерживает опцию `-s`, поскольку управлять удаленным `web`-сервером он все равно не может.

`cmwactl` поддерживает следующие команды:

**start** Запускает `web`-сервер. Эта команда обычно используется в стартовых скриптах системы. В некоторых случаях ее осмыслено выполнять в ручную.

**stop** Останавливает `web`-сервер.

**restart** Посылает `Web`-серверу сигнал (`SIGUSR1`), заставляющий его перечитать конфигурационные файлы и перезагрузить все модули `Communiware`, а также обновить закешированную информацию из базы данных. Все `http` запросы, которые обрабатываются в текущий момент будут обработаны до конца, прежде чем изменения в конфигурации вступят в силу. Следует учесть, что объем потребляемой `web`-сервером памяти при выполнении команды `restart` несколько увеличивается.

В случае, если в момент выполнения команды `restart` `web`-сервер не был запущен, он запускается, с выдачей соответствующего сообщения на консоль.

**graceful** Синоним `restart`. Для совместимости с оригинальным `apachectl`.

**hard** Жесткий `restart`. Посылает `web`-серверу сигнал `SIGHUP`, заставляющий его прервать обработку всех `http`-запросов и переинициализироваться. Заметим, что оригинальный `apachectl` именно так поступает по команде `restart`, а для мягкого рестарта использует команду `graceful`.

**hard-restart** Синоним hard.

**complete-restart** Эквивалент последовательности stop/start.

**conftest** Проверяет синтаксис файлов конфигурации и выдает сообщения об ошибках. Эту команду рекомендуется применять перед выполнением команды restart, поскольку в случае ошибок в конфигурационных файлах web-сервер аварийно завершается.

**configtest** Синоним conftest.

**reconfig** Эта команда приводит к регенерации автоматически генерируемого фрагмента конфигурационного файла frontend-a, и рестарту обеих апачей. Эту команду *необходимо* выполнить после создания нового Коммунивер-сайта (см раздел 10), а также при изменении расширенных атрибутов итема сайта, таких как HOSTNAME и URLTOP.

**status** Выдает отчет о статусе web-сервера — сколько процессов работает, какие действия они выполняют и т.д. Для успешного выполнения этой команды требуется наличие mod\_status

**fullstatus** Более подробное сообщение о статусе.

**dump** Выдает на stdout директивы конфигурационного файла после разбора их web-сервером.

**vhosts** Выдает список виртуальных хостов. Список виртуальных хостов обычно включает не все Коммунивер-сайты, а только те из них которые имеют собственное уникальное значение доменного имени.

**modules** Выдает список модулей, вкомпилированных в соответствующий apache.

### 3 dump\_item

Утилита `dump_item` выдает всю информацию об указанных итемах (атрибуты, связи, текст, иллюстрации) на стандартный вывод. Основное ее применение — перенос информации с одного коммунивер-сервера на другой, хотя возможны и другие, например, пакетное редактирование.

Формат, выдаваемый `dump_item` понимается утилитами `load_item` (с. 9) и `load_item.www` (с. 12).

При дампе нескольких итемов `dump_item` старается расположить их так, чтобы связи между ними были разрешимы, т.е. активный итем был записан первым, а пассивные, ссылающиеся на него, позднее. Если это не удастся, то выдается сообщение «обнаружена взаимная рекурсия» («mutual recursion encountered») и утилита аварийно завершается.

Связи, ведущие к итемам, не вошедшим в состав дампа, к вредным эффектам не приводят. Считается что пользователь знает, что делает, и на том сервере, куда дамп будет загружаться, необходимые итемы уже созданы.

## 3.1 Способы задания набора итемов

### 3.1.1 Явное задание списка

Простейший способ задания набора итемов — явное перечисление их идентификаторов в командной строке:

```
dump_item итем1 итем2 ...
```

Количество итемов, которые можно сдать таким образом ограничено только системными ограничениями на длину командной строки.

Список итемов может быть как набран вручную, так и получен автоматически с помощью утилиты `grepempl` (см. раздел 11) или путем обработки результатов `sql`-запроса скриптом. В общем, вся мощь текстовых утилит Unix в вашем распоряжении.

### 3.1.2 Дамп поддеревьев

Достаточно часто возникает необходимость дампа не только самого итема, но и его потомков по определенной связи (как правило, одной из входящих в метасвязь BELONGS).

Для этой цели применяется ключ `-d`.

```
dump_item -d итем1 итем2 ...
```

приводит к дампу не только самих перечисленных итемов, но и их потомков.

В случае если необходимо задать тип связи, отличный от BELONGS, используйте ключ `-D`.

Например:

```
dump_item -D PART book
```

Приведет к дампу итема книги `book` вместе со всеми частями и главами, но без дискуссий, которые привязаны к ним связью `PARENT`, также входящей в метасвязь `BELONGS`.

### 3.1.3 Дамп сайта

При переносе сайтов с сервера разработки на эксплуатационный сервер типичной операцией является дамп определенных типов итемов принадлежащих указанному сайту (например, шаблонов и рубрик) и игнорирование всех остальных, предназначенных исключительно для отладки.

Если указать `dump_item` ключ `-s`, то он воспримет все свои аргументы не как имена итемов, а как имена типов, и сдампит все итемы указанных типов, которые принадлежат сайту, указанному в параметре ключа `-s`.

Например, для переноса типичного новостного сайта можно использовать следующую команду:

```
dump_item -s news TEMPLATE TOPIC NEWS_SOURCE\  
EXPIRATION_RULE USER_GROUP
```

### 3.1.4 Дамп по SQL-запросу

Наиболее гибким способом дампа итемов является задание произвольного SQL-запроса, отбирающего их. Для этого предназначена опция `-q`. Параметром этой опции является SQL-запрос, который должен возвращать одну колонку, содержащую идентификаторы нужных итемов. Эту опцию нельзя сочетать с `-d2`, `-D` и `-s`, а также с явным указанием итемов в командной строке.

Последнее, впрочем несущественно, так как файлы дампов можно объединять с помощью команды `cat` или просто дописывать один дамп в конец другого с помощью символа переназначения `>>`.

Например, сдампить все шаблоны используемые шаблоном `replic_t`, можно с помощью команды

```
dump_item -q "select active from item_rel  
where passive='replic_t' and linktype_id='USES'"
```

---

<sup>2</sup>Кстати, а почему?



## 3.2 Управление сохраняемой информацией

У `dump_item` есть несколько ключей, позволяющих управлять сохраняемой информацией.

**-p** позволяет включить в дамп зашифрованные пароли итемов типа «Персона». По умолчанию пароли в дампе не включаются, поэтому для переноса пользователей с сервера на сервер эту опцию указывать обязательно.

**-A** (анонимный дамп). Позволяет исключить из дампа информацию об авторах, переводчиках и т.д. Данную опцию рекомендуется использовать при переносе итемов на сайт, на котором не предполагается создавать пользователей, работавших над данными итемами на текущем сайте, поскольку даже если вы не прописываете авторов шаблонов, ядро CompuWare автоматически прописывает связь PUBLISHER с персонай, создавшей данный итем.

**-i** Исключает иллюстрации и прочие двоичные данные. Может оказаться полезной при работе с удаленной базой данных, каталог иллюстраций которой недоступен.

## 3.3 Прочие ключи

**-c** Задаёт альтернативный конфигурационный файл CompuWare.

**-r** Отменяет проверку на взаимозависимости. Итемы будут помещаться в дампы в том порядке, в котором они указаны в командной строке или возвращены SQL-запросом.

**-n тип итем** — вместо дампа существующего итема создаёт „заготовку” итема указанного типа — набор заголовков, соответствующих атрибутам и связям данного типа итем. Вместе с этой опцией может быть указан идентификатор, который будет прописан в поле Item-Id. Предназначена в основном для использования утилитой `viitem` (с. 14)

## 4 `load_item`

Утилита `load_item` предназначена для загрузки в базу данных CompuWare итемов, сохранённых с помощью `dump_item`. Наиболее распространённым способом её применения является просто указание одного или нескольких файлов дампа в командной строке:

```
load_item file.dmp file2.dmp ...
```

Если в командной строке не указано имени файла, то утилита читает дамп со стандартного ввода. Поэтому ее можно использовать в составе конвейера для переноса итемов с одного сайта на другой, например

```
dump_item -c alternate_config -s site TEMPLATE \  
ARTICLE | load_item
```

или

```
dump_item -d book | \ ssh otherhost load_item
```

## 4.1 Управление проверкой синтаксиса

По умолчанию `load_item` выполняет все те же проверки на корректность синтаксиса, что и `web-интерфейсы` `Communiware`, и прерывает обработку дампа при обнаружении первой же ошибки. В выдаваемом при этом сообщении указывается номер ошибочной строки относительно начала дампа.

Ключ `-n` позволяет запретить проверку синтаксиса и выполнять загрузку итемов с `Content-Type HTML` в режиме эквивалентном `AUTOFIX=1`.

Ключ `-d` позволяет отключить стандартное поведение `Communiware` по удалению „опасных” тэгов (`iframe`, `script`) из текста итем. Данный режим позволяет разработчику сайта использовать итемы с `Content-Type=HTML` для служебных целей, требующих использования `client-side` скриптинга. Аналогичного режима, доступного через `Web-интерфейс` не существует.

Ключ `-i` заставляет `load_item` не загружать содержащиеся в дампе файлы иллюстраций.

Ключ `-f` задает стиль форматирования для текстов, которые содержатся в дампе в формате `RICHTEXT`, а не `HTML`. Такой файл дампа не может быть создан `dump_item` который всегда сбрасывает итем в дамп в том виде, в каком он содержится в базе данных, но может быть создан, например, при помощи `viitem` (см. раздел 7).

## 4.2 Пакетные режимы и логгинг

Ключ `-b` заставляет `load_item` не прекращать обработку дампа по первой же ошибке. Вместо этого ошибочный итем игнорируется, и продолжается обработка следующего. Такой режим может привести к неразрешенности некоторых ссылок, если последующие итемы ссылаются на

пропущенный, а также к тому, что часть сообщений об ошибках будет пропущена. Поэтому пакетный режим рекомендуется комбинировать с логгингом.

Ключ **-I** *имя-файла* приводит к дублированию всех сообщений об ошибках в указанный файл.

Ключ **-L** *имя-файла* приводит к записи в файл SQL-операторов, которые восстанавливают связи между итемами, которые не удалось создать при загрузке. При указании этого ключа невозможность установки связи по причине отсутствия итема на целевом сервере не является фатальной ошибкой. Итем все равно создается, но с неполным набором связей.

### 4.3 Отмена ограничений на изменение итемов

По умолчанию, `load_item` отказывается загружать итемы, которые уже существуют в базе, но имеют другой тип. Цель этого поведения — избежать ошибочного перезаписывания итема в случае если при загрузке возник конфликт имен. Ключ **-t** позволяет `load_item` изменять типы существующих итемов, а ключ **-T** позволяет вносить изменения в итем, который в базе данных имеет один тип, а в дампе — другой, не изменяя самого типа итема.

### 4.4 Переименование итемов при загрузке

Во многих случаях при переносе итемов с сервера на сервер требуется избежать конфликта имен. Особенно часто это касается итемов идентификаторы которых сгенерированы Comptuneware автоматически.

Ключ **-r** (`renumber`) приводит к тому, что при загрузке таких итемов `load_item` будет генерировать для них новые идентификаторы. При этом ведется таблица соответствия идентификаторов в дампе и в базе, и связи у итемов, ссылающихся на ранее переименованный итем, устанавливаются правильно.

Поскольку загрузка большого дампа в штатном режиме (т.е. без **-p** и **-L**) редко производится за один раз, то в случае ошибочного завершения `load_item` сохраняет эту таблицу в файле `renumber.dat` в текущей директории, а при обнаружении такого файла при старте считывает его и использует для инициализации таблицы переименования.

В случае невозможности создания файла `renumber.dat` в текущей директории, `load_item` пытается создать его сначала в домашней

директории текущего пользователя, а затем в `/tmp`, о чем выдает соответствующее предупреждающее сообщение. Считывается этот файл только из текущей директории.

Данный механизм позволяет загружать один и тот же дамп несколько раз (после ручного исправления ошибок в файле дампа) с использованием одной и той же таблицы переименования.

В случае необходимости переименования нечисловых (т.е. содержательных) идентификаторов, можно воспользоваться ключами `-x prefix`, который добавляет указанный префикс ко всем нечисловым именам, и `-X prefix`, который удаляет указанный префикс из всех нечисловых имен, содержащих такой префикс.

В случае, если сайт на который загружается дамп, не совпадает по названию с сайтом, откуда был сделан дамп, можно воспользоваться ключом `-s имя сайта`, который изменит имя сайта на указанное во всех загружаемых итемах.

## 5 *load\_item.www*

Утилита *load\_item.www* предназначена для загрузки дампа на коммунивер-сервер, доступ к которому есть только через web-интерфейс.

Она загружает итемы с помощью HTTP-запросов к интерфейсу веб-мастера.

Утилите необходимо указать имя сервера (hostname) с помощью ключа `-h`. Можно также указать префикс URL скриптов (`-u`), если он отличается от стандартного `/scripts`. Имя пользователя, если оно отличается от имени пользователя Unix на локальной машине, можно задать с помощью ключа `-U`. Пароль на удаленном коммунивер-сервере запрашивается с терминала, или может быть указан с помощью ключа `-P`. Этой возможностью пользоваться не рекомендуется, так как пароль, указанный в командной строке принципиально доступен любому пользователю, работающему на той же машине, что и Вы.

*load\_item.www* поддерживает опции `-n`, `-r -l` и `-f`, имеющие тот же смысл, что и в *load\_item*.

## 6 putitem

В отличие от `load_item`, основное назначение которой — перенос итемов с сервера на сервер, `putitem` предназначена для создания новых итемов из локальных файлов, например, для импорта в CompuWare унаследованного статического контента.

Она позволяет создавать итемы с `Content-Type=HTML` не только из HTML-файлов, но и из RTF, а итемы с `Content-Type=Binary` из любых файлов.

Вся информация об атрибутах и связях задается с помощью ключей командной строки. В связи с необходимостью использовать десятки различных ключей, `putitem` использует не классические односимвольные ключи, а многосимвольные, например `-item`, `-topic`.

Если `load_item` обрабатывает заданные идентификаторы по алгоритму «если существует, обновить, иначе создать», то для `putitem` указание идентификатора с помощью ключа `-item` означает обновление существующего итема, и ошибку если такого итема не существует. Для создания нового итема с содержательным идентификатором используйте ключ `-nick`.

Получить список всех ключей, соответствующих атрибутам и связям можно, запустив `putitem -help`.

Опции, задающие альтернативный конфигурационный файл и файл для записи ошибок также отличаются от стандартных: `-config` и `-log` соответственно.

Если при загрузке HTML-файла не указана опция `-title`, `putitem` пытается использовать в качестве атрибута `TITLE` содержимое заголовка HTML (тега `<title>`).

С помощью `putitem` принципиально возможно создать несколько итемов за один вызов. При этом требуется не задавать содержательных имен, т.к. ключ `-nick` можно указать только один раз, и все атрибуты и связи должны совпадать. Так что как максимум можно создать несколько статей в одной рубрике или несколько глав в книге.

`putitem` позволяет загружать HTML-файлы с иллюстрациями при условии, что файлы иллюстраций расположены в той же директории, что и HTML-файл и в HTML используются относительные ссылки.

```

```

## 7 viitem

Утилита `viitem` предназначена для редактирования итемов из командной строки сервера. Ее синтаксис:

`viitem` *идентификатор-и́тема*

Фактически при запуске `viitem` выполняется `dump_item` указанного итема, потом на созданный временный файл запускается текстовый редактор, а при сохранении файла запускается `load_item`.

`viitem` ориентирован на использование редактора `vim`. При использовании этого редактора загрузка итема в базу производится не при выходе из редактора, а при выполнении команды сохранения, а в случае обнаружения ошибок синтаксиса курсор позиционируется на строку в которой произошла ошибка (к сожалению, это не распространяется на ошибки в атрибутах итема и неправильное задание связей).

Кроме этого, `vim` поддерживает синтаксическую подсветку HTML, достаточно удобную и для шаблонов Communiware.

Поэтому рекомендуется при запуске `viitem` устанавливать переменную среды `EDITOR` в `vim`, если `vim` установлен на сервере.

Для создания нового итема с помощью `viitem` нужно указать ключ **-n** *тип-и́тем*.

При этом будет создана заготовка заголовка содержащая по строке на каждый атрибут и каждую допустимую связь для данного типа `item`.

При создании нового шаблона необходимо не забыть заполнить поле `Template-Type`, а при создании любого итема — поле `Server`.

### 7.1 Краткая справка по командам `vim`

## 8 dump\_meta

Утилита `dump_meta` выводит на стандартный вывод метаинформацию Communiware — описание типов итемов, типов связи и так далее.

Ее вывод представляет собой XML-файл соответствующий DTD Ontology.

В отличие от утилит, работающих с итемами, `dump_meta` всегда выводит информацию в кодировке UTF-8.

Существует два способа задания списка метаобъектов, описания которых надо вывести — явное задание списка объектов и указание о выводе всех метаобъектов, относящихся к конкретному сайту.

В атрибуты тэга `Ontology`, создаваемого утилитой `dump_meta` помещается имя текущего пользователя Unix в качестве автора онтологии и номер установленной версии `Communiware` в качестве минимально необходимой для использования данной онтологии версии `Communiware`.

Название онтологии можно задать в командной строке с помощью ключа **-n** (по умолчанию записывается строка `Unknown Ontology`).

Номер версии самой онтологии, если это необходимо, можно указать в с помощью ключа **-V**.

Если в состав онтологии входят фильтры, то проверяется наличие в них конструкций, специфичных для текущего сервера базы данных, и если они обнаружены, то тэг `<query>` записывается в выходной XML-файл с атрибутом `sql-dialect`, что предотвращает его использование в несовместимых с текущим диалектом серверах баз данных.

## 8.1 Задание явного списка метаобъектов

Список метаобъектов каждого типа указывается в качестве параметра соответствующего ключа, т.е. должен быть передан утилите как один параметр.

Разделителями внутри списка могут быть как запятые, так и пробелы.

Так, например, если необходимо сдампить типы `TOPIC` и `ARTICLE`, то следующие два вызова `dump_meta` будут правильными:

```
dump_meta -t ARTICLE, TOPIC
dump_meta -t "ARTICLE TOPIC"
```

а вот этот — неправильным:

```
dump_meta -t ARTICLE TOPIC
```

Для указания метаобъектов можно использовать следующие ключи:

- t** задает список типов итемов.
- l** задает список типов связи.
- a** задает список вычисляемых и групповых атрибутов.
- v** задает список типов виртуальных итем.
- f** задает список фильтров.

**-F** задает список шаблонов, все фильтры, используемые в которых, необходимо включить в онтологию.

## 8.2 Дамп онтологии сайта

Для дампа онтологии сайта необходимо использовать команду

```
dump_meta -s имя сайта
```

При указании сайта, `dump_meta` использует ряд эвристик для определения того, какие метаобъекты относятся к данному сайту.

Типы итемов отбираются, если на сайте присутствует хотя бы один итем такого типа или если для сайта или какой-то его рубрики назначен шаблон по умолчанию для такого типа. Тип связи отбирается, если в этом типе связи может участвовать хотя бы один из типов итемов, используемых на сайте. Фильтр отбирается, если он упоминается хотя бы в одном шаблоне сайта.

Из построенного таким образом списка выбрасываются те метаобъекты, которые входят в стандартную онтологию `Compuware`.

Запретить это можно посредством указания ключа **-S**.

Вышеописанная эвристика срабатывает далеко не всегда, и может оказаться что какие-то существенно важные для функционирования сайта объекты в дампе онтологии не попали. В этом случае рекомендуется повторить дамп, скомбинировав оба способа задания параметров.

В онтологию сайта кроме используемых в нем типов итемов и связей, и фильтров, используемых в его шаблонах, включаются:

- текущие значения виртуальных страниц;
- текущие значения шаблонов по умолчанию;
- минимальный фрагмент дерева рубрик, достаточный для того, чтобы в онтологию попали все точки привязки шаблонов по умолчанию;

Если в файле онтологии встречается упоминание итема (например, шаблона), то в него помещается минимальное описание данного итема, позволяющее при загрузке онтологии создать итем с таким именем, и разрешить ссылку на него. Предполагается, что реальное содержание итема будет загружено позднее с помощью `load_item`.



## 9 `load_meta`

Утилита `load_meta` загружает в Communiware-сервер онтологию в формате XML, созданную с помощью `dump_meta`. В качестве параметров командной строки ей указывается имя файла онтологии. Если имя не указано, то онтология читается со стандартного ввода.

Эта утилита не имеет никаких ключей, кроме стандартного `-c`, задающего альтернативный конфигурационный файл Communiware. Онтология должна быть в кодировке UTF-8.

## 10 `createserver`

Утилита `createserver` создает новый коммунивер-сайт. Для того, чтобы этот сайт стал виден по WWW, после создания сайта требуется выполнить команду `cmwactl reconfig`.

Минимально необходимым набором параметров для `createserver` является указание только идентификатора итема сайта. Все остальные значения имеют разумные умолчания, и могут быть изменены впоследствии с помощью `viitem` или web-интерфейсов.

Тем не менее, `createserver` позволяет задать значительную часть параметров непосредственно в момент создания.

Вторым параметром после имени сайта указывается префикс URL, который будет появляться перед именами итемов сайта в браузере. По умолчанию это */имя-сайта*.

Остальные параметры задаются с помощью ключей,

**-t** Заголовок (title) итема сайта.

**-a** Уровень авторизации. Уровень авторизации 0 (по умолчанию) соответствует отсутствию проверок прав доступа на чтение, уровень 1 — возможность ограничить доступ на чтение к определенным итемам с помощью связи `ALLOWEDUSERS`, 2 — вся авторизация (при редактировании) производится только по HTTPS, 3 — сочетание HTTPS и защиты от несанкционированного чтения.

Изменение уровня авторизации в дальнейшем будет требовать выполнения `cmwactl reconfig`, так что рекомендуется сразу задавать правильный уровень авторизации.

- h Задает hostname нового сайта. Используется, если hostname не совпадает с заданным по умолчанию. Использование отдельного имени хоста возможно только в том случае, если он resolves в DNS в один из IP-адресов данного сервера. Изменение hostname виртуального сайта требует запуска `cmwactl reconfig`.
- m Задает идентификатор пользователя, который будет иметь права модератора на данном сайте. Если при создании сайта не указать имя модератора, то возможность изменить что-либо на данном сайте будут иметь только пользователи с правами суперпользователя `Communityware`.
- T Задает имя сайта-прототипа. По умолчанию `default`. Новому сайту прописываются те же шаблоны по умолчанию и виртуальные страницы, что и у сайта-прототипа. Если указана опция **-C**, то итемы будут копироваться именно с сайта-прототипа.
- C Задает список типов итем, которые должны быть скопированы с сайта-прототипа на новый сайт. По умолчанию не копируются никакие итемы. Если указана опция **-C**, то должна быть указана и **-x**.
- x Задает префикс, который добавляется к именам копируемых итемов.
- X задает префикс, который отбрасывается из имен копируемых итемов, если он там присутствует.

## 10.1 Копирование сайтов

При создании сайта можно указать, какие типы итемов нужно скопировать с сайта-прототипа. Обычно в этот список входят шаблоны, часто — служебные итемы, такие как `EXPIRATION_RULE` или `NEWS_RESOURCE`, иногда — рубрики.

Если указан набор копируемых типов, то все шаблоны по умолчанию, виртуальные страницы и другие ссылки на итемы копируемых типов будут ссылаться на копии итемов, принадлежащие данному сайту. Это касается в том числе и ссылок в тексте шаблонов. При копировании указывается префикс, который требуется дописать к именам итем нового сайта (и, возможно, префикс, который надо удалить из имен итем старого сайта). Выбирая префикс, следует помнить что общая длина идентификатора итем ограничена 20 символами.

## 11 greptempl

Утилита `greptempl` выполняет поиск заданного регулярного выражения по всем шаблонам текущего сервера.

В отличие от системных утилит `grep` и `egrep` она использует синтаксис регулярных выражений, принятый в `perl`.

Утилита поддерживает следующие ключи командной строки:

- s **site** Ограничить поиск только шаблонами указанного сайта
- t **type** Ограничить поиск только шаблонами указанного типа (PAGE, ELEMENT, MAIL).
- n Как и в стандартном `grep` — выводить рядом с найденными строками их номера.
- i Как и в стандартном `grep` — регистронезависимый поиск.
- l Как и в стандартном `grep` — выводить не строки, в которых найден образец, а только имена объектов, в которых он найден. Имена выводятся в формате `site:item` и после некоторой обработки их можно, например, передать в командную строку `dump_item`

Например:

```
dump_item `greptempl -l образец|cut -f 2 -d:`
```

## 12 clearpictures

Обычно при удалении итемов `Communiware`, принадлежащие им иллюстрации также удаляются. В некоторых случаях, например при удалении с помощью вручную выполненного SQL-запроса, этого может не происходить. Кроме того, если из итема удалить все картинки, сохранив сам итем, директория этого итема в каталоге иллюстраций останется. Для борьбы с такими оставшимися картинками, бесполезно занимающими место на диске, предназначена утилита `clearpictures`. Она сканирует область иллюстраций `Communiware`, удаляет оттуда все пустые каталоги, и все файлы, относящиеся к несуществующим итемам. Единственный ключ, который она воспринимает — `-c` позволяет задать альтернативный конфигурационный файл `Communiware`. Следует учесть, что для этой утилиты строго необходимо, чтобы каталог картинок был доступен по файловой системе.