

Битрикс: Управление сайтом 5.0

Руководство по интеграции



Содержание

Введение	3
Новая технология интеграции	4
Шаблоны и сайты	4
Управление шаблонами. Структура шаблона	5
Включаемые области и компоненты	7
Интеграция в деталях	9
Структура файлов	9
Типовая страница	9
Свойства страниц и папок	10
Создание шаблона сайта	10
Управление метаданными	10
Установка кодировки страницы	11
Включение таблиц стилей	12
Настройка цепочки навигации	15
Установка заголовка документа	18
Подключение административной панели	19
Размещение рекламных областей	21
Настройка меню	23
Настройка дополнительных редактируемых областей	27
Работа с компонентами	29
Модуль подписки	32
Модуль поиска	33
Главный модуль	34
Модуль статистики	34
Модуль информационных блоков	35
Модуль Веб-форм	36
Модуль Опросов	37
Использование файлов языковых сообщений	37
Настройка продукта на техподдержку со стороны партнеров	38
Формат файла описания шаблонов компонентов	39
Приложение 1. Рекомендации по подготовке HTML шаблона	41
Приложение 2. Настройка внешнего вида дополнительных элементов	42

Введение

Данное руководство предназначено для разработчиков веб-сайтов на основе программного продукта «Битрикс: Управление сайтом 5.0». В документе подробно рассматривается процедура интеграции продукта в новый или уже существующий сайт.

В данном документе подразумевается, что разработчик владеет базовыми технологиями для разработки веб-сайта, такими как HTML, CSS, PHP.

Новая технология интеграции

Версия 5.0 продукта «Битрикс: Управление сайтом» включает в себя целый ряд новшеств, направленных на ускорение процесса разработки сайтов.

Сохраняя традиционную гибкость системы, *шаблоны дизайна* позволяют в течение нескольких часов провести интеграцию дизайна с программным продуктом «Битрикс: Управление сайтом» и приступить к построению структуры проекта и размещению *программных компонентов*.

Шаблон сайта может быть загружен в систему одним файлом с использованием встроенного веб-интерфейса продукта и применен к одному или нескольким сайтам. Используемые в системе шаблоны могут быть выгружены (экспортированы) в виде комплекта файлов и использованы в других проектах.

Шаблоны и сайты

Продукт «Битрикс: Управление сайтом 5.0» поддерживает многосайтовость, что позволяет создать несколько *сайтов* на одной копии продукта. Каждый сайт может характеризоваться собственным доменным именем, внешним видом, языком интерфейса и информационным наполнением.

Внешний вид сайта определяется *шаблоном дизайна*. Для каждого сайта может быть назначено неограниченное количество шаблонов. Использование шаблонов открывает широкие возможности для настройки внешнего вида сайта и изменения его дизайна в зависимости от различных условий.

Продукт позволяет гибко настроить дизайн для различных сайтов, разделов сайта, использовать специальный праздничный дизайн в течение определенного времени, задать отдельные шаблоны дизайна для различных групп посетителей сайта, в зависимости от определенного параметра в адресе и т.п. (рис. 1.1).

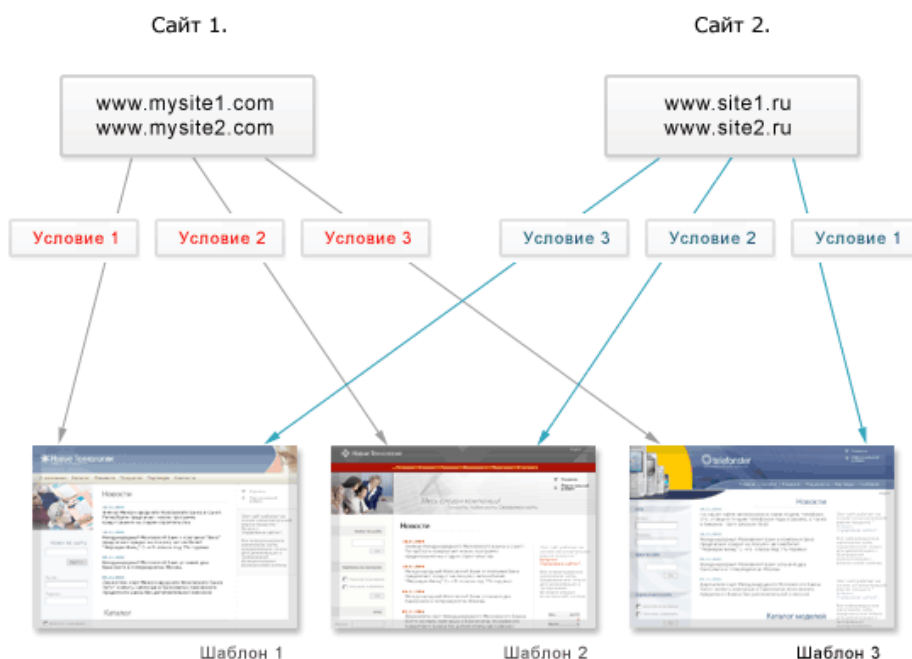


Рис. 1.1. Шаблоны и сайты.

Выбор и назначение шаблонов для каждого сайта выполняется в административном разделе: Настройки – Настройки продукта – Сайты – Список сайтов.

Для каждого шаблона дизайна может быть задано условие, при котором данный шаблон будет применен к страницам сайта (рис. 1.2). Если условие не задано, то шаблон применяется по умолчанию как основной.


*Шаблон	Сорт.	Тип условия	Условие
Шаблон демо-версии	1	[без условия]	<без условия>
Версия для печати	2	Параметр в URL	print = Y
Мобильные телефоны	3	Для папки или файла	/catalog/phone
Корпоративный шаблон 1	4	Для групп пользователей	Зарегистрированные пользователи Партнеры Подписчики Редакторы сайта Администраторы техподде

Рис. 1.2. Настройка шаблонов сайта.

В соответствии с условиями, представленными на рисунке 1.2, шаблоны дизайна будут применяться к страницам сайта следующим образом:

1. **Шаблон демо-версии** будет использоваться в качестве основного шаблона сайта, так как для него не задано условие показа.
2. Шаблон **Версия для печати** будет использован, если URL адрес страницы содержит параметр **print=Y**. Например, ссылка на странице сайта <http://www.site2.com/?print=Y> позволит посетителю получить версию документа для печати.
3. Шаблон **Мобильные телефоны** будет использован для страниц раздела сайта </catalog/phone/>.
4. **Корпоративный шаблон 1** будет использован, если посетитель авторизовался на сайте как пользователь группы Партнеры.

! На заметку: условие может содержать любой программный код на языке PHP, в том числе вызовы функций API продукта. Для получения дополнительной информации смотрите документацию к продукту.

Вы можете убедиться, что шаблон настроен правильно, нажав на иконку  рядом с выпадающим списком шаблонов. Вы увидите сайт с выбранным шаблоном в режиме *предварительного просмотра*, еще до применения изменений.

При создании шаблона сайта возможно использование различных программных условий, влияющих на отображение тех или иных элементов шаблона для различных разделов сайта. Для этого для раздела сайта нужно определить некоторое свойство, значение которого будет проверяться в шаблоне сайта:

```
<? if ($APPLICATION->GetProperty("SECT_PROP ") == "Y"):?>
```

Подобные условия можно вводить для любых элементов шаблона. Например, можно отключать показ включаемых областей или управлять показом навигационной цепочки, и т.п.

К одному сайту может быть применено неограниченное число шаблонов. Один и тот же шаблон может быть использован многократно для любых сайтов при различных условиях.

Управление шаблонами. Структура шаблона

Дизайн сайта обычно состоит из трех основных частей (рис. 2.1):

- § Верхняя часть дизайна (**header**).
- § Рабочая область страницы, в которой размещаются информационные материалы сайта, компоненты или любой другой программный код.
- § Нижняя часть дизайна (**footer**).

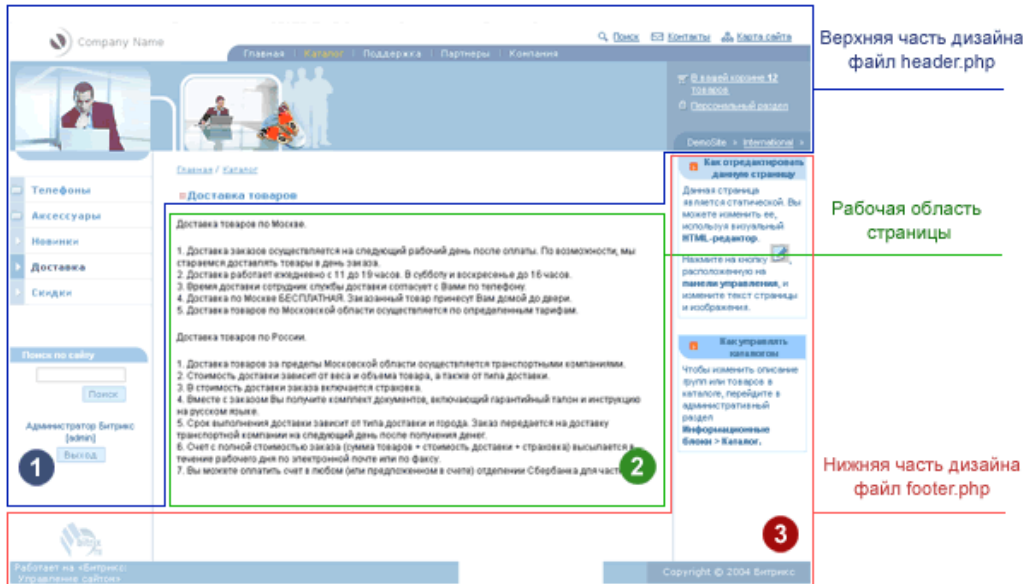


Рис. 2.1. Основные части дизайна.

Рассмотрим, как в шаблоне настраиваются части дизайна на примере шаблона **demo**, который включен в поставку продукта.

Управление шаблонами дизайна осуществляется в административном разделе: Настройки – Настройки продукта – Сайты – Шаблоны сайтов. В указанном разделе можно просмотреть существующие шаблоны, модифицировать их или же добавить свои собственные шаблоны сайта. Например, зайдём в режим редактирования шаблона **demo**, выбрав действие "Изменить".

Код шаблона представлен в поле «Внешний вид шаблона сайта».

⚠ Обратите внимание на наличие в коде шаблона разделителя `#WORK_AREA#`, который используется для указания границы между верхней и нижней частью дизайна. Именно в этом месте выполняется подключение рабочей области страницы.

Шаблон в этом поле представляет собой **объединенные верхнюю и нижнюю часть дизайна** сайта. В HTML код вставляются компоненты и функции на языке программирования PHP, которые обеспечивают показ различного рода информации: метаданных, заголовка страницы, таблицы каскадных стилей, административной панели и последующее управление этой информацией с использованием визуальных инструментов.

Каждая часть шаблона храниться в отдельном файле:

- § верхняя часть шаблона дизайна храниться в файле `.../<идентификатор шаблона>/header.php`;
- § нижняя часть шаблона дизайна храниться в файле `.../<идентификатор шаблона>/footer.php`.

⚠ Никаких ограничений в шаблонах и внешнем виде сайта.

Использование шаблона с программным кодом обеспечивает необходимую гибкость при реализации интернет-проекта, позволяет реализовать как простые шаблоны, так и шаблоны с произвольной программной логикой и индивидуальным внешним видом. Если вы не очень хорошо знаете PHP, включение программных модулей можно выполнить простым переносом соответствующего кода из других шаблонов.

Каскадные стили, используемые в шаблоне, представлены на закладке «Файл стилей». Таблица стилей хранится в файле `<идентификатор шаблона>/styles.css`.

Все шаблоны хранятся в директории `/bitrix/templates/`. Набор файлов каждого шаблона находится в поддиректории, название которой соответствует идентификатору шаблона. В рассматриваемом нами примере шаблон размещается в каталоге: `/bitrix/templates/demo/`.

⚠ Примечание: При создании нового шаблона через интерфейс задается его идентификатор, название, описание для показа в списке, код шаблона внешнего вида сайта, таблица стилей и набор используемых включаемых компонентов и картинок. При сохранении шаблона автоматически создается поддиректория `/bitrix/templates/<идентификатор шаблона>`.

Каждый шаблон имеет схожую структуру файлов и поддиректорий. Простейший шаблон может состоять всего из нескольких основных файлов: `header.php`, `footer.php`, `styles.css`, а также нескольких файлов для шаблонов меню.

Все графические элементы, используемые в шаблоне, размещаются в директории `/bitrix/templates/<идентификатор шаблона>/images/`.

В каталоге шаблона могут быть также размещены другие файлы и компоненты. Посмотреть состав шаблона в Менеджере файлов можно, перейдя по ссылке, показанной рядом с идентификатором шаблона.

В виде отдельных файлов могут быть представлены любые области дизайна, которые предполагается использовать в шаблоне сайта, и к редактированию которых необходимо иметь быстрый доступ. Примерами таких областей могут служить: область с указанием авторских прав (copyright), область с контактной информацией и т.п.

Включаемые области и компоненты

Для динамического управления современным сайтом с использованием визуальных инструментов некоторые элементы дизайна в шаблоне представлены в виде программных компонентов.

Вернемся к шаблону **demo** и познакомимся с основными программными компонентами и включаемыми областями. На рисунке 3.1 выделены компоненты и включаемые области в данном шаблоне.

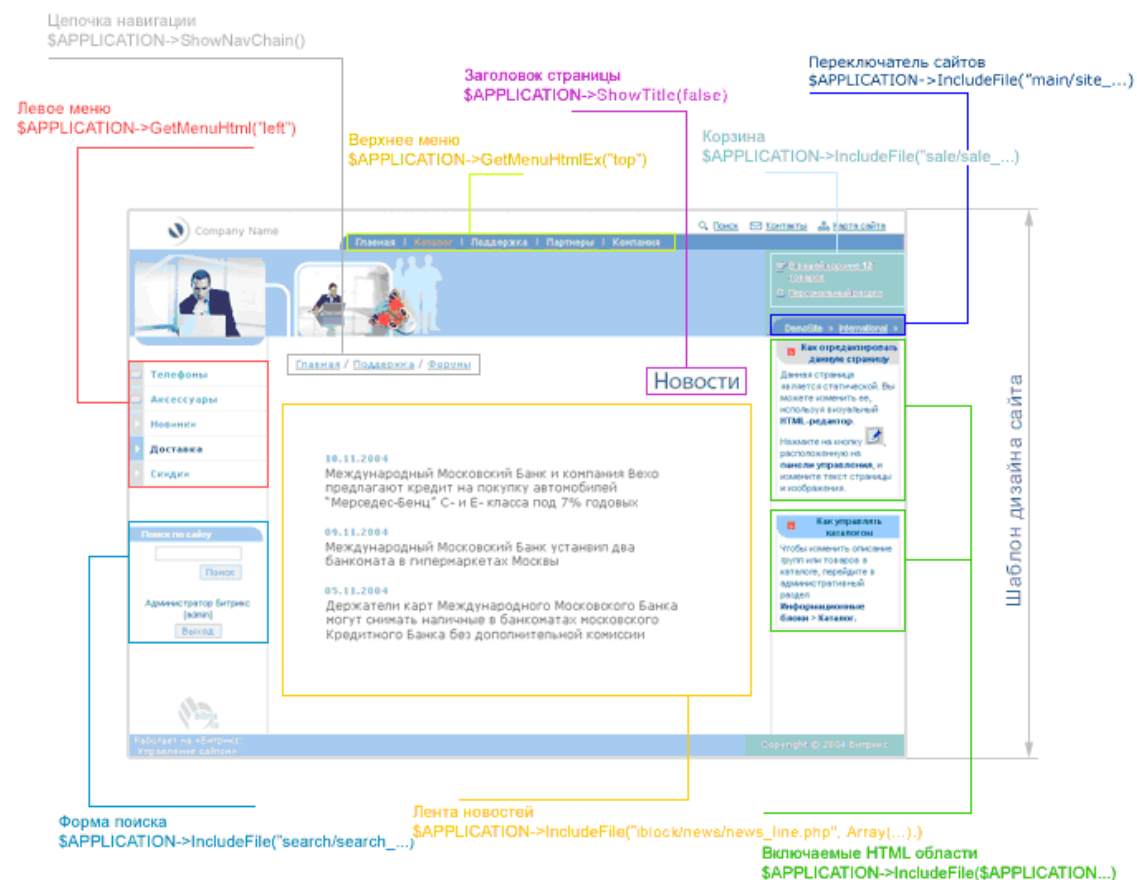



Рис. 3.1. Включаемые области в шаблоне.

В HTML-коде прототипа проводятся замены выделенных областей на включения соответствующих компонентов (рис. 3.1).

В код вставляются функции, которые обеспечивают показ различного рода информации: метаданных, заголовка страницы, подключение таблицы каскадных стилей, вывод административной панели.

Отдельно создаются шаблоны элементов навигации: навигационная цепочка, меню сайта.

После создания HTML-прототипа и проведения требуемых подстановок функций и вызовов для программных компонентов, в вашем распоряжении будет PHP-шаблон дизайна сайта.

Все размещенные в шаблоне сайта области и компоненты доступны для быстрого редактирования из публичного раздела при работе в режиме показа включаемых областей (кнопка  на административной панели, рис. 3.2).

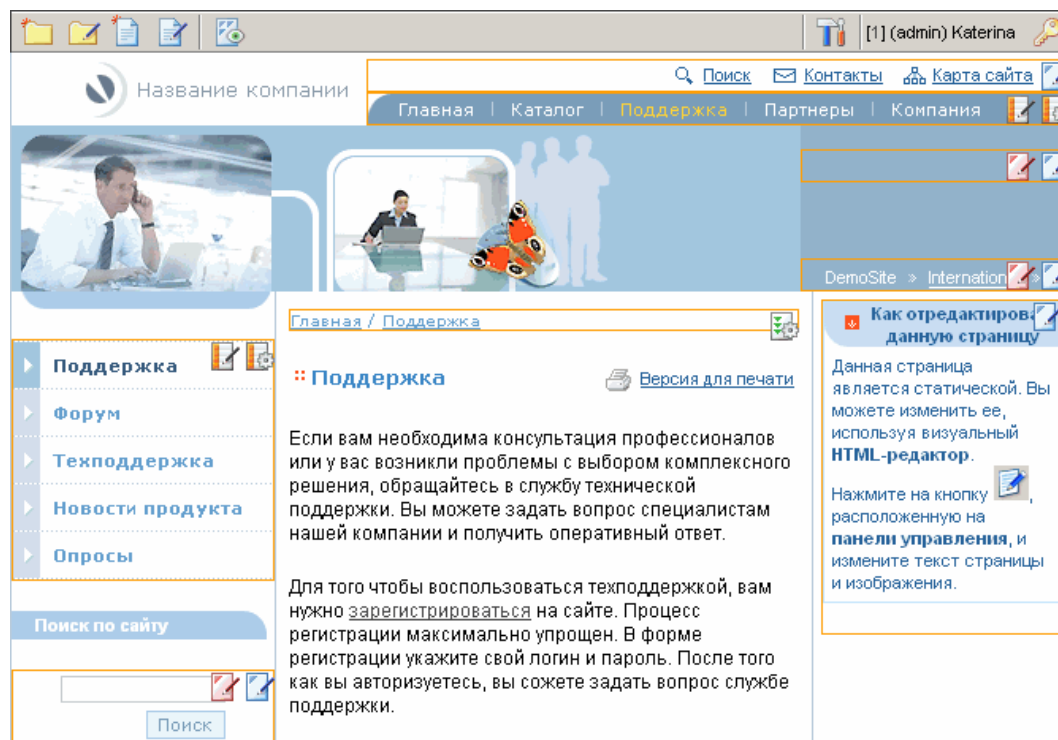


Рис. 3.2. Режим показа включаемых областей и компонентов.

Включаемые компоненты могут иметь различные элементы управления, представленные различными иконками.

Так, например, с помощью элементов управления рекламного баннера можно перейти к редактированию текущего баннера или же перейти к списку баннеров в административной части, отфильтрованному по выбранному типу баннера.

С помощью элементов управления, доступных для меню, можно перейти к редактированию пунктов данного меню для текущего подраздела, либо перейти к редактированию шаблона меню.

На сайте могут быть использованы различные включаемые HTML-области. Включаемые области оформляются отдельными файлами и могут быть показаны, например, только для текущего раздела либо только для текущей страницы и т.п. Элемент управления каждой включаемой области позволяет перейти непосредственно к ее редактированию.

Интеграция в деталях

Структура файлов

Программный продукт «Битрикс: Управление сайтом» разработан таким образом, что внешнее представление (или публичная часть) отделено от программного ядра. Основная программная часть инсталлированного продукта размещается в папке `/bitrix/` в корневом каталоге сайта.

Подкаталоги этой папки содержат следующие файлы:

<code>/bitrix/templates/</code>	шаблоны внешнего вида (дизайна сайта) и используемые компоненты. Основная работа по интеграции существующего дизайна и продукта производится с файлами, находящимися в этой директории;
<code>/bitrix/admin/</code>	интерфейс административной части. Вызов различных форм управления и редактирования для всех основных модулей системы;
<code>/bitrix/cache/</code>	кэш-файлы, созданные в процессе кэширования динамической информации;
<code>/bitrix/php_interface/</code>	дополнительные системные файлы (информация для подключения к базе данных, некоторые другие файлы);
<code>/bitrix/modules/</code>	библиотеки классов и функций для основных модулей системы;
<code>/bitrix/images/</code>	картинки для модулей, установленных в системе
<code>/bitrix/tools/</code>	служебные файлы;
<code>/bitrix/updates/</code>	используется для загрузки обновлений;
<code>/bitrix/</code>	дополнительные служебные файлы.

Типовая страница

Сборка типовых страниц производится подключением *пролога* и *эпилога* для каждой страницы сайта. В общем случае структура документов, которые размещаются на сайте, выглядит так:

```
<?
require($_SERVER["DOCUMENT_ROOT"]."/bitrix/header.php");
$APPLICATION->SetTitle("Битрикс: Управление сайтом 4.0");
?>

// Тело документа. Содержательная часть.

<?
require($_SERVER["DOCUMENT_ROOT"]."/bitrix/footer.php");
?>
```

Свойства страниц и папок

Предлагаемая в продукте возможность задавать свойства страниц и разделов позволяет организовать гибкое управление показом информации на сайте. Задавая определенные свойства на страницах сайта, можно устанавливать значения метаданных для различных страниц, организовать смену графических элементов дизайна в зависимости от раздела сайта или отдельной страницы и т.д.

Значения свойств страниц и разделов могут быть заданы через соответствующий административный интерфейс либо непосредственно из программного скрипта.

В продукте зарезервировано несколько названий свойств, значения которых проверяются и используются некоторыми функциями.

Примеры зарезервированных свойств:

- § **title**, свойство используется для установки дополнительного заголовка страницы (см. Установка заголовка документа);
- § **adv_desired_target_keywords**, свойство используется для установки желательных (desired) ключевых слов для показа рекламы на странице (см. Размещение рекламных областей);
- § **not_show_nav_chain**, свойство используется для выключения показа навигационной цепочки на странице или в разделе сайта (см. Настройка цепочки навигации).

Создание шаблона сайта

В процессе перевода шаблона от HTML-варианта к полноценному PHP-шаблону производится замена HTML-кода на вызов соответствующих PHP-функций, методов и программных компонентов.

Обратите внимание, что в версии 5.0 продукта используется набор функций, которые позволяют управлять вставкой в шаблон метаданных, заголовка страницы, навигационной цепочки, включаемых областей и т.п.

Использование функций [ShowMeta\(\)](#), [ShowTitle\(\)](#), [ShowCSS\(\)](#) и т.д. позволяет производить инициализацию отдельных элементов непосредственно из скрипта на странице или из компонента. Например, заголовок страницы может быть добавлен уже после вывода результатов работы скрипта. Таким образом, если ранее требовалось производить инициализацию заголовка страницы до подключения основного дизайна, то теперь появилась возможность производить установку заголовка страницы непосредственно из кода в рабочей области страницы. Ниже приводится детальное описание используемых в типовом шаблоне функций и программных компонентов.

Управление метаданными




Как это выглядит


Примером управления метаданными в продукте может служить механизм задания ключевых слов и описаний для страниц и разделов сайта. По умолчанию в поставляемой версии продукта настроено управление именно этими двумя типами метаданных, но по аналогичной схеме список возможных вариантов может быть дополнен и другими.



Как это работает

Работа с метаданными осуществляется через управление свойствами страниц и разделов сайта. Изменение свойств страницы может быть выполнено при редактировании текущего документа встроенным визуальным HTML-редактором в режиме "Редактировать как HTML" (кнопка 

верхней панели инструментов). В режиме визуального редактирования перейти на закладку «Свойства» и ввести значения соответствующих свойств страницы.

Также свойства страницы могут быть заданы в режиме редактирования страницы как «Текст». Значения свойств для всех страниц раздела сайта задаются в отдельном диалоге, доступном по кнопке  административной панели.



Как это реализуется

Для вывода соответствующих метаданных в коде страницы необходимо воспользоваться следующей функцией:

```
<head>
...
<?APPLICATION->ShowMeta("keywords")?>
<?APPLICATION->ShowMeta("description")?>
...
</head>
```

В результате работы функции в код страницы будет подставлен следующий HTML-код (пример):

```
<meta name="keywords" content="Мобильные телефоны, аксессуары, Siemens,
Motorolla">
<meta name="description" content="Магазин мобильных телефонов">
```

Свойства страницы могут быть установлены динамически из скрипта. Например, для страниц показа информации каталога или новостей (информационных блоков) свойства страницы могут быть установлены в соответствии с определенными свойствами элементов информационных блоков. Таким образом, можно создавать свойства keywords и description для элементов каталога и динамически подставлять их в код страницы.

Установка кодировки страницы



Как это выглядит

Некоторые языки не могут быть представлены одновременно в одной кодировке, например, русский и немецкий языки. Поэтому, если в системе используется подобное сочетание языков, для каждого языка нужно определить кодовую таблицу, с использованием которой будут отображаться текстовые сообщения.



Как это работает

Управление языками публичной части системы осуществляется из административного раздела отдельно для каждого сайта: Настройки – Настройки продукта – Сайты – Список сайтов. Для каждого сайта может быть указана индивидуальная кодировка, формат вывода даты и времени. Настройка формата показа даты и времени позволяет правильно отображать эти данные в публичном разделе сайта.

Параметры:	
*Язык:	[ru] Russian ▾
*Формат даты: (например: DD.MM.YYYY)	DD.MM.YYYY
*Формат даты и времени: (например: DD.MM.YYYY HH:MI:SS)	DD.MM.YYYY HH:MI:SS
*Кодировка:	windows-1251

Рис. 4.1. Установка параметров языка для публичного раздела сайта



Как это реализуется

Кодировка страниц определяется с помощью константы PHP, которая получает значение из настроек языка текущего сайта:

```
<head>
...
<meta http-equiv="Content-Type" content="text/html; charset=<?echo LANG_CHARSET?>">
...
</head>
```

Включение таблиц стилей



Как это выглядит

Для форматирования различных элементов на сайте используются несколько таблиц стилей.

Таблицы стилей настраиваются для каждого шаблона сайта, используемого в системе, и размещаются в соответствующих папках наряду с другими файлами настроек интерфейса для конкретного шаблона.

Важным элементом при формировании таблицы стилей является создание названий стилей. Названия следует создавать для тех стилей, которые планируется использовать при редактировании страниц в режиме HTML-редактора. Стили будут доступны в редакторе из выпадающего списка под именами, заданными в файле `/<идентификатор шаблона>/styles.php` (файл с именами стилей).

Пример таблицы стилей для шаблона сайта:

Стили для верхнего меню:

```
.topmenu {font-family: Arial, Helvetica, sans-serif; font-size:12px; font-weight:
bold; color: #FFFFFFF; text-decoration: none;}

.topmenuact {font-family: Arial, Helvetica, sans-serif; font-size:12px; font-
weight: bold; color: #FAC535; text-decoration: none;}
```

Стили для таблиц:

```
.tableborder {background-color:#9C9A9C;}
```

```
.thead {background-color:#D8D9DA;}

tbody {background-color:#F8F8F8;}

tbodytext, .theadtext {font-family: Arial, Helvetica, sans-serif; font-size:12px;}

tbodytext {color:#000000;}

theadtext {color:#000066;}
```

Стили для оформления текста:

```
.text {font-family: Arial, Helvetica, sans-serif; font-size:12px; font-weight:normal;} /*Обычный текст */

.titletext {font-family: Arial, Helvetica, sans-serif; color:#585858; font-size:15px; font-weight: bold; line-height: 18px;} /*Текст заголовка*/

.subtitletext {font-family: Arial, Helvetica, sans-serif; color:#000000; font-size:11pt; font-weight: bold;} /*Текст подзаголовка*/
```

Пример файла с названиями для стилей шаблона сайта:

```
<?
$arStyles = Array(

    "text" => "Нормальный",

    "smalltext" => "Мелкий",

    "smalltextwhite" => "Мелкий серый",

    "smalltextblack" => "Мелкий черный",

    "newstext" => "Текст новостей",

    "titletext" => "Заголовок",

    "subtitletext" => "Подзаголовок",

    "tableborder" => "Таблица - рамка",

    "thead" => "Таблица - шапка",

    "tbody" => "Таблица - тело",

    "tableborders" => "Вкл. область - рамка",

    "tableheads" => "Вкл. область - шапка",

    "tablebodys" => "Вкл. область - тело",

    "inputbutton" => "Кнопка",
```

```
);
return $arStyles;
?>
```



Как это работает

Для настройки таблиц стилей служит закладка «Файл стилей» на странице редактирования шаблона дизайна.

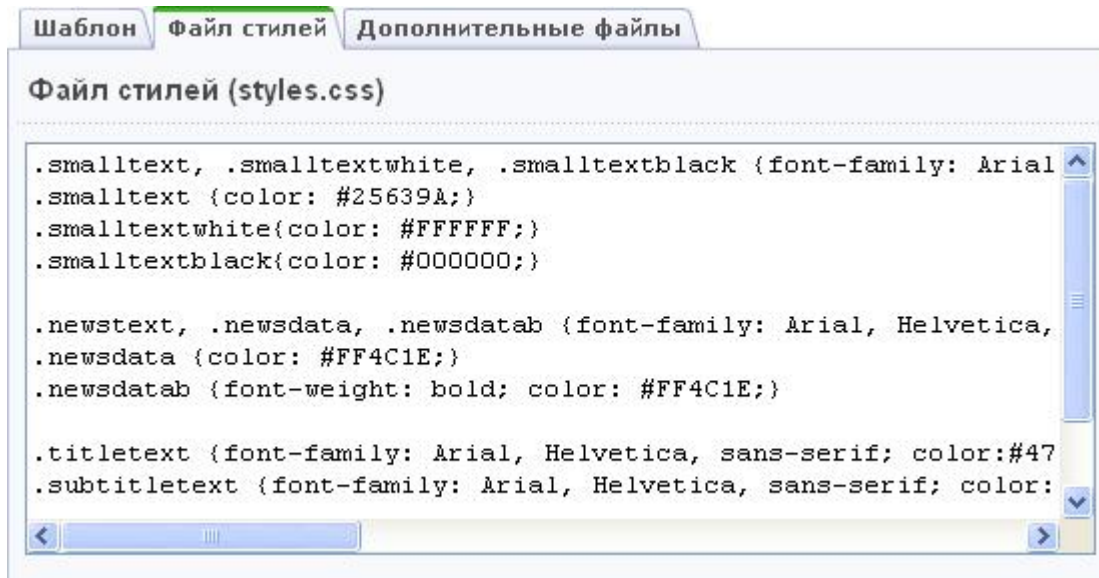


Рис. 4.2. Управление таблицей стилей в настройках шаблона сайта



Как это реализуется

Таблицы стилей подключаются к шаблону сайта в области <head> пролога с помощью функции [ShowCSS\(\)](#).

```
<?
$APPLICATION->ShowCSS();
?>
```

Функция [ShowCSS\(\)](#) производит подключение файла стилей из текущего шаблона сайта, а также всех дополнительных стилей определенных для данной страницы функцией [SetAdditionalCSS\(\)](#).

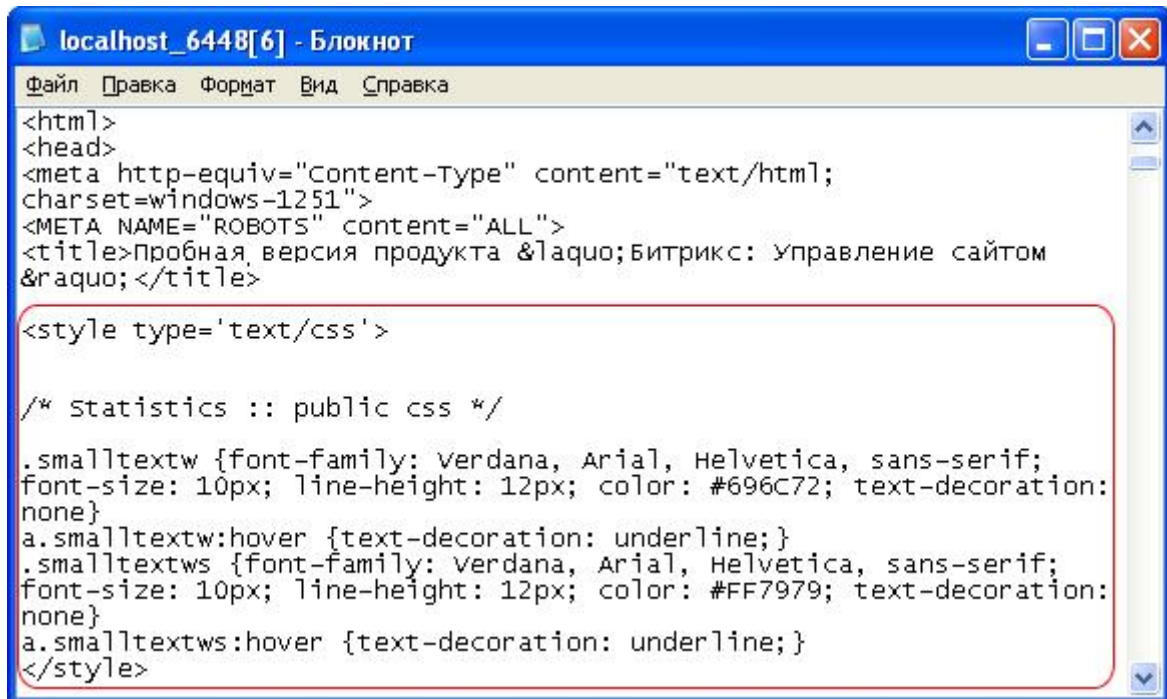
```
<?
$APPLICATION-> SetAdditionalCSS("/bitrix/templates/demo/additional.css");
?>
```

Дополнительные стили могут использоваться, например, для оформления форума, веб-форм, таблиц, некоторых типов меню и т.д.

⚠️ Обратите внимание! При использовании функции `ShowCSS()` без параметров подключение стилей будет выполнено в виде ссылки на CSS файл:

```
<LINK href="/bitrix/templates/demo/styles.css" type="text/css" rel="STYLESHEET">
```

При этом стили, подключаемые с использованием `SetAdditionalCSS()`, будут включены в код страницы с использованием PHP функции `require()`, т.е. будут полностью включены итоговый код страницы (рис. 4.3).



```
localhost_6448[6] - Блокнот
Файл Правка Формат Вид Справка
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
<META NAME="ROBOTS" content="ALL">
<title>Пробная версия продукта &laquo;Битрикс: Управление сайтом
&raquo;</title>
<style type='text/css'>

/* statistics :: public css */

.smalltextw {font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px; line-height: 12px; color: #696C72; text-decoration:
none}
a.smalltextw:hover {text-decoration: underline;}
.smalltextws {font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px; line-height: 12px; color: #FF7979; text-decoration:
none}
a.smalltextws:hover {text-decoration: underline;}
</style>
```

Рис. 4.3. Включение CSS стилей в код страницы

В случае использования функции `ShowCSS()` с параметром `false` файл стилей для текущего дизайна будет также включен в код страницы с использованием `require()`:

```
<?
$APPLICATION->ShowCSS(false);
?>
```

Настройка цепочки навигации



Как это выглядит

Навигационная цепочка позволяет визуально представить глубину вложенности пунктов меню, начиная с главной страницы и до текущего документа. Значения, которые подставляются в навигационную цепочку, могут быть заданы как для каждого раздела, так и для каждого документа.

Для представления навигационной цепочки на странице сайта создается отдельный PHP-шаблон.

Рис. 4.3. Цепочка навигации


В случае необходимости, значение пункта навигационной цепочки для данной страницы, определяется непосредственно в документе. Для этого используется функция `AddChainItem()`.

Обратите внимание, что некоторые скрипты и компоненты, поставляемые в составе дистрибутива продукта, добавляют определенные элементы в навигационную цепочку. Так, например, компонент каталога последовательно добавляет в навигационную цепочку названия каталогов и групп товаров по мере погружения вглубь по уровням каталога. Аналогичным образом в навигационную цепочку добавляются названия форумов и тем форумов.



Как это работает

По умолчанию в качестве ссылки на раздел в цепочке навигации используется заголовок раздела. Значение заголовка сохраняется в файле `.section.php` соответствующего раздела.

Изменение заголовка раздела может быть выполнено в форме настройки свойств папки соответствующего раздела (кнопка  на административной панели), либо непосредственным редактированием файла `.section.php`.

Пример файла `.section.php`:

```
<?
$SSectionName = "Главная страница";
?>
```



Как это реализуется

Шаблон показа навигационной цепочки определяется в файле `/bitrix/templates/<название шаблона>/chain_template.php` индивидуально для каждого шаблона сайта. Структура шаблона показа навигационной цепочки аналогична структуре шаблона показа меню. Шаблон включает область пролога, основную область (тело), область эпилога. Основная область задает шаблон для показа одного элемента. Построение всей цепочки осуществляется в цикле.

Пример шаблона навигационной цепочки:

```
<?
//---Область пролога цепочки
$SChainProlog="";

//---Область тела шаблона
$SChainBody = "";

//---Переменная $ITEM_INDEX содержит текущий номер элемента.
if($ITEM_INDEX > 0)


$SChainBody = " &nbsp;&raquo;&nbsp; ";

$SChainBody .= "<a href=\"\".$LINK.\"\"
class=\"smalltext\">\".htmlspecialchars($TITLE).\"</a>\";
```

```
//---Область эпилога

$sChainEpilog="";

?>
```

Быстро перейти к редактированию шаблона навигационной цепочки из публичного раздела вы можете, если в режиме показа редактируемых областей воспользуетесь иконкой .

Шаблон навигационной цепочки может быть задан дополнительно для отдельного раздела сайта. Для этого непосредственно в файле `.section.php` определяется переменная `$sChainTemplate`, где задается полный путь к шаблону показа навигационной цепочки.

Шаблон навигационной цепочки может быть также задан непосредственно при вызове функции `ShowNavChain()` как один из параметров функции.

В пробной версии продукта шаблон навигационной цепочки размещен в папке шаблона сайта по умолчанию: `/bitrix/templates/.default/`.

Код для вызова навигационной цепочки в общем шаблоне показа:

```
<?APPLICATION->ShowNavChain();?>
```

Навигационная цепочка может быть использована на различных страницах сайта, например, вывод навигационной цепочки используется в компоненте поиска, в параметрах компонента задается путь к шаблону навигационной цепочки.

В пробной версии продукта дополнительный шаблон для использования с компонентом поиска находится в папке шаблона сайта по умолчанию `/bitrix/templates/.default/chain_template_search.php`.

Ниже приведены пример добавления элементов в навигационную цепочку с использованием функции `AddChainItem()`:

```
<?

//---Первым параметром функции AddChainItem() задается название,

//---которое будет показано в навигационной цепочке, вторым параметром задается

//---ссылка для перехода.

//---Значения параметров могут быть как статическими, так и динамическими.

//---в приведенном примере название раздела задано статической величиной, а ссылка
формируется динамически.

$APPLICATION->AddChainItem("Детальная информация о товаре",
"catalog.php?BID=".$arIBlock["ID"]."&ID=".$arSection["ID"]);

//---В приведенном ниже примере обе величины формируются динамически.

//---В качестве названия подставляется текущее значение раздела каталога.

$APPLICATION->AddChainItem($arSection["NAME"],
"catalog.php?BID=".$arIBlock["ID"]."&ID=".$arSection["ID"]);

?>
```

Для того чтобы выводить в навигационную цепочку название текущей страницы, необходимо вставить вызов функции `AddChainItem()` в файле `footer.php`, т.е. после вывода содержимого рабочей области.

```
<?APPLICATION->AddChainItem($APPLICATION->GetTitle());?>
```

В шаблоне показа навигационной цепочки можно настроить показ элементов, для которых не задана ссылка, в виде обычного текста (например, выводить название текущей страницы без ссылки):

```
if (strlen($LINK)>0)
    $sChainBody .= "<a href=\"\".\".\" class='\".\".\">\".\".\"</a>";
else
    $sChainBody .= "<font class='\".\".\">\".\".\"</font>";
```

Показ навигационной цепочки может быть отключен на определенных страницах или в определенном разделе сайта. Для этого для страницы или папки необходимо задать свойство с кодом: `not_show_nav_chain`.

На выбранных страницах или в заданных разделах сайта следует установить значение данного свойства равным `Y`.

Установка заголовка документа



Как это выглядит

Заголовок документа выводится в верхней части браузера и в центральной части страницы документа.

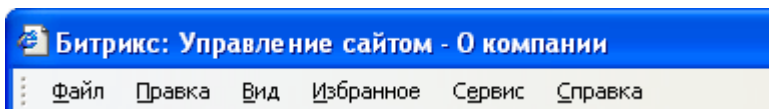


Рис. 4.4. Вывод заголовка документа в верхней части браузера

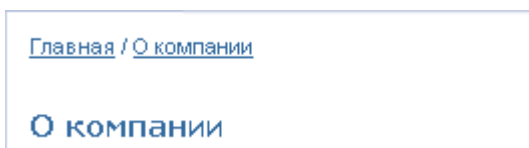



Рис. 4.5. Вывод заголовка в содержательной части документа



Как это работает

Изменение заголовка документа может быть выполнено при редактировании документа встроенным визуальным HTML-редактором в режиме "Редактировать как HTML" (кнопка  административной панели). Заголовок документа задается в поле «Заголовок страницы».

Заголовок страницы также может быть изменен в режиме редактирования документа как «Текст» или при непосредственном редактировании HTML-кода документа.

Заголовок, выводимый в верхней части окна браузера, может отличаться от основного заголовка страницы. По умолчанию для его задания на странице определено свойство **title**.



Как это реализуется

Задание заголовка может выполняться в любом месте страницы. Стандартно, заголовок страницы вставляется до вызова основного пролога документа следующим образом:

```
<?
//---Установка статического заголовка
$APPLICATION->SetTitle("О компании");
//---Установка динамического заголовка. В качестве заголовка страницы
//---используется имя информационного блока для заданного $ID.
$arIBlock = GetIBlock($ID, "news")
$APPLICATION->SetTitle($arIBlock["NAME"]);
... ?>
```

Заголовок, выводимый в верхней строке браузера, может быть установлен с использованием различных механизмов. По умолчанию заголовок устанавливается в отдельно заведенном свойстве страницы **title**. Если указанное свойство не заполнено, то заголовок окна браузера будет установлен равным текущему заголовку страницы.

Вывод заголовка производится заданием следующего кода в шаблоне дизайна:

```
<?$APPLICATION->ShowTitle()?>
```

Приведенный код размещается в области **<head>** основного пролога:

```
<head><title><?$APPLICATION->ShowTitle()?></title></head>
```

и в области **<body>** на месте вывода заголовка документа.

```
<?$APPLICATION->ShowTitle(false)?>
```

Параметр **false** в последнем случае указывает, что для установки заголовка не нужно проверять значение свойства **title**, т.е. используется стандартный заголовок, установленный с использованием функции `SetTitle()`.

Подключение административной панели



Как это выглядит

После авторизации на сайте для администратора становится доступна панель инструментов в верхней части страницы. С ее помощью можно управлять параметрами текущего раздела, перейти к редактированию текущей страницы и дополнительных редактируемых областей, добавить и изменить меню текущего раздела, а также быстро перейти в административный раздел сайта или завершить сеанс работы с сайтом.



Рис. 4.6. Административная панель



Как это работает

С каждой кнопкой панели инструментов связан определенный раздел административной части системы.

Вот не полный перечень функций доступных из административной панели (дополнительно доступны кнопки для перехода к редактированию документов через модуль «Документооборота» и для управления содержимым информационных блоков):



Создание новой папки. Открывает диалог для создания новой директории в структуре сайта. Задается как имя папки, так и название раздела, которое будет использоваться при построении навигационной цепочки. Предлагает возможность автоматически создать индексную страницу раздела по заданному шаблону и сразу же перейти к ее редактированию.



Свойства текущей папки. Переводит в диалог для управления свойствами папки текущего раздела сайта. Можно задать/изменить заголовок раздела, используемый для построения навигационной цепочки.



Создание нового документа. Переводит в режим создания нового документа (страницы сайта) в текущем разделе сайта.



Редактирование текущей страницы. Позволяет модифицировать текущую страницу сайта



Переключение в режим отображения включаемых областей. Позволяет отобразить все включаемые области на текущей странице сайта. С помощью иконок, расположенных в верхнем правом углу включаемой области, можно перейти к управлению выбранным элементом непосредственно из публичного раздела.



Переход в административный раздел.



Завершение сеанса.



Как это реализуется

Код для включения административной панели в верхней части сайта задается сразу после тега `<body>` перед началом первой таблицы.

```
<?
$APPLICATION->ShowPanel ( ) ;
?>
```

Отдельные модули могут добавлять в панель свои кнопки (например, модуль «Документооборота» и модуль «Информационных блоков»).

Также пользователь может самостоятельно добавить кнопки в панель.

! **Примечание:** Для модуля «Информационные блоки» управление кнопками административной панели вынесено в настройки визуальных компонентов.

Добавление может быть выполнено несколькими способами:

- 1) изменением файла `"/bitrix/php_interface/include/add_top_panel.php"`. Функция `$APPLICATION->ShowPanel()`; производит проверку наличия указанного файла и выводит в панель описанные в нем кнопки.
- 2) использованием в скрипте показа функции `Ciblock::ShowPanel()`, в случае вывода дополнительных кнопок для модуля «Информационных блоков». Количество кнопок может варьироваться, например, в зависимости от того выполняется ли вывод списка элементов информационного блока или детальный показ информации по одному элементу.

Размещение рекламных областей



Как это выглядит

На сайте может быть представлено несколько различных типов рекламы и выделено несколько рекламных областей. Это может быть как стандартная баннерная реклама, так и текстовые рекламные области. Реклама может быть постоянной или показываться с определенной вероятностью, задаваемой администратором. Показ рекламы может быть регламентирован по определенным разделам сайта и т.д. Более подробная информация о рекламе представлена в соответствующем разделе «Помощи» по продукту.

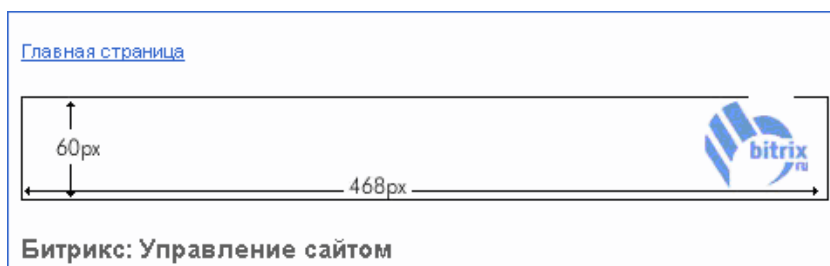


Рис. 4.7. Пример рекламной области



Как это работает



Заведение типов рекламы и добавление конкретных рекламных элементов производится через административный интерфейс в разделе Сервисы – Реклама. Название типа рекламного блока задается администратором произвольно, и будет использоваться для выбора конкретной рекламы. Например, для верхней рекламы может быть установлен тип TOP, TOP1 и т.п.

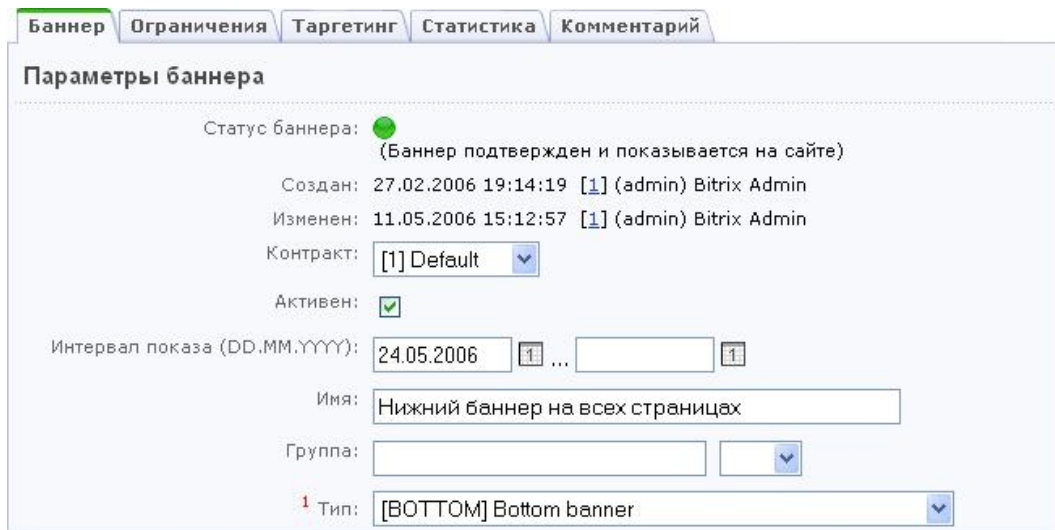
|| Добавить тип || Excel

⏪ ⏩ 1 ⏪ ⏩ | На странице: 20 | Типы баннеров 1 - 13 из 13

<input type="checkbox"/>		ID	Изменено	Акт.	Сорт.	Имя	Описание	Баннеров
<input type="checkbox"/>		TOP	10.05.2006 15:21:16	Да	10	Top banner		0
<input type="checkbox"/>		LEFT	10.05.2006 15:21:16	Да	20	Left banner		3

Рис. 4.8. Управление типами рекламы на сайте

Быстрый доступ к управлению соответствующим рекламным баннером (или списком баннеров выбранной рекламной области) можно выполнить по нажатию на иконку  «Редактировать данный баннер» (или  «Отфильтрованный список баннеров»), отображаемую непосредственно рядом с баннером в режиме показа редактируемых областей.



Параметры баннера

Статус баннера: ●
(Баннер подтвержден и показывается на сайте)

Создан: 27.02.2006 19:14:19 [1] (admin) Bitrix Admin
Изменен: 11.05.2006 15:12:57 [1] (admin) Bitrix Admin

Контракт: [1] Default

Активен:

Интервал показа (DD.MM.YYYY): 24.05.2006 ...

Имя: Нижний баннер на всех страницах

Группа:

1 Тип: [BOTTOM] Bottom banner

Рис. 4.9. Форма добавления рекламы определенного типа



Как это реализуется

Размещение рекламных областей производится в заранее определенных при разработке шаблона местах.

```
<?
//---Пример размещения рекламной области в левой части сайта.
//---Аналогично производится выбор любого другого типа:
//---TOP, BOTTOM, COUNTER,... (первый параметр функции)
//---Могут быть использованы как предустановленные, так и задаваемые пользователем
типы.
//---В качестве двух дополнительных параметров может указываться HTML-код
обрамляющий рекламную область сверху и снизу.
$APPLICATION->ShowBanner("LEFT", '<div align="center">', '<br></div><br>');
?>
```

Показ рекламы на странице осуществляется в зависимости от набора ключевых слов баннера и страницы сайта.

Ключевые слова баннера могут быть заданы как для отдельного баннера, так и для рекламного контракта в целом.

Ключевые слова страницы сайта разделяются на обязательные (**required**) и желательные (**desired**). Более подробная информация представлена в руководстве «Управление рекламой с помощью ключевых слов» и документации продукта.

Обратите внимание, что по умолчанию, если в коде страницы не установлено каких-либо ключевых слов для рекламы, считается что функция **CAdvBanner::SetDesiredKeywords**

использует в качестве параметра значение свойства страницы **adv_desired_target_keywords**. Если значение данного свойства не задано, то в качестве параметра функции используется значение свойства **keywords**.

Функция **CAdvBanner::SetDesiredKeywords** вызывается автоматически в момент сборки страницы с указанными выше параметрами, ее не нужно дополнительно вызывать в файле [header.php](#), если нет необходимости переопределить ключевые слова для показа рекламы.

Настройка меню



Как это выглядит

Любое меню на сайте строится на основе двух составляющих:

- § массива данных, редактируемых в административном разделе (в поставку входят демонстрационные меню);
- § шаблона внешнего представления меню.

Массив данных определяет состав меню, задает названия и ссылки для всех пунктов меню. Массив данных меню хранится в файле [.<тип меню>.menu.php](#) в папке соответствующего раздела сайта. При отображении в Менеджере файлов данному файлу автоматически присваивается имя *Меню типа «<типменю>»*.

Шаблон меню – это код на PHP, определяющий внешний вид меню. Шаблон меню обрабатывает массив данных, давая на выходе HTML-код.

Для настройки шаблона в HTML-коде пролога необходимо выделить повторяющиеся части меню (рис. 4.10). Например, для горизонтального меню – это ячейки таблицы, а для вертикального – строки. Затем повторяющаяся часть выносится в шаблон меню, а само меню строится в прологе или эпилоге с помощью вызова PHP-функции построения меню.

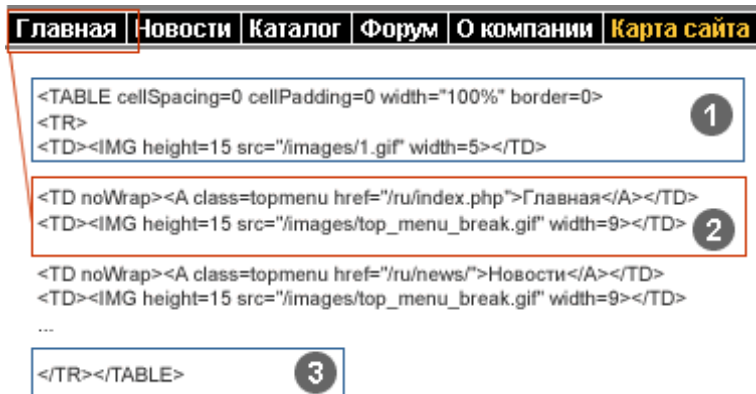


Рис. 4.10. Выделение меню.

Задача формирования меню включает:

- § выделение HTML элементов для построения меню,
- § создание шаблонов меню,
- § включение (вызов) функции показа меню в общем шаблоне ("прологе" и "эпилоге"),
- § заполнение меню в соответствии со структурой сайта.

При выделении области верхнего меню нужно обратить внимание на повторяющиеся элементы: пункты меню и разделители. В примере, приведенном на рисунке 4.10, повторяющимися элементами являются ячейки таблицы (2). При выделении области левого меню повторяющимися элементами будут строки таблицы.

При создании шаблона меню потребуется также создать дополнительные стили в таблице стилей. Например, для текстового меню: цвет пункта меню и цвет текущего (активного) пункта меню. Отдельного представления в шаблоне могут потребовать заголовки разделов. Например,

возможно использование графических и текстовых обозначений того, что данный пункт связан не с конечным документом, а ведет в новый раздел (директорию) и т.п.

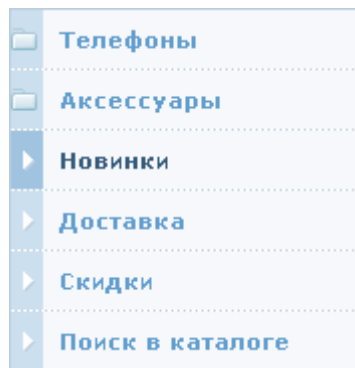


Рис. 4.11 Элементы меню.



Как это работает

Управление меню каждого раздела производится через административный интерфейс. Для редактирования меню раздела можно найти соответствующий файл в Менеджере файлов либо использовать кнопку "Редактировать пункты меню" рядом с соответствующим меню в режиме показа включаемых областей ().

Одному разделу сайта могут соответствовать несколько меню различных типов: верхнее, левое, нижнее и т.п. В самом общем случае на сайте существует одно "верхнее меню", которое соответствует самому верхнему уровню и подключается во всех разделах. Также в системе обычно используется – "левое" (или "правое") меню второго уровня, которое определяет пункты для документов и подразделов данного раздела (директории).

Типы меню задаются из административного раздела в секции Настройки – Настройки модулей – Управление структурой. Например, пусть в системе заданы два типа меню:

- § Левое меню – тип "left";
- § Верхнее меню – тип "top".

Настройки
Доступ

Настройка параметров модуля

Редактор страниц по умолчанию: Визуальный HTML редактор ▼

Использовать предыдущую версию визуального HTML редактора:

Разрешить в визуальном HTML редакторе выводить стили без названий:

Типы меню:

Тип	Название
left	Левое меню
top	Верхнее меню

Рис. 4.12. Задание типов меню.

Тип, заданный в настройках, будет использован как префикс файла шаблона меню. Указанный префикс используется также для идентификации файлов с пунктами меню для конкретного раздела.

Для хранения данных для каждого типа меню создается отдельный файл. Например, для хранения данных левого меню текущего раздела - `.left.menu.php`, или `.top.menu.php` для верхнего меню.

Существует два режима редактирования меню: упрощенный и расширенный. В упрощенном режиме задаются только название пункта меню, ссылка для перехода и значение сортировки.

Название	Ссылка	Сортировка	Удалить
Главная	/index.php	10	<input type="checkbox"/>
Каталог	/catalog/	20	<input type="checkbox"/>
Поддержка	/support/	30	<input type="checkbox"/>

Рис. 4.13. Упрощенный режим редактирования меню.

В расширенном режиме для управления доступны следующие данные:

- § название пункта меню;
- § ссылка для перехода;
- § набор дополнительных ссылок, которые соответствуют этому же пункту меню. В данном поле задается набор ссылок на страницы, при переходе на которые будет также подсвечиваться данный пункт меню. Например, для того, чтобы при просмотре любой страницы раздела «Новости компании» подсвечивался пункт меню **Новости компании**, в данном поле нужно указать ссылку на папку, содержащую все страницы раздела (или перечислить необходимые страницы): `/about/news/`.
- § сортировка - определяет положение пункта меню в общем списке, чем меньше значение, тем выше будет показан элемент;
- § условия показа. Например, позволяет внести ограничения на показ данного пункта меню пользователям с определенными правами доступа;
- § дополнительные параметры – набор произвольных параметров, которые могут быть обработаны в шаблоне показа меню и представлены соответствующим образом. Например, если пункт меню является заголовком секции, это может быть указано в параметрах пункта так: название параметра - SEPARATOR, значение - "Y". При разработке шаблона можно проверять значение этого параметра и при показе выделять данный пункт меню разделителем.

Пункты меню:

Вставить пункт >>

Название:

Ссылка:

Сортировка:

Удалить

Доп. ссылки для подсветки:

Тип условия:

Условие:

Параметры:

Название	Значение
SEPARATOR	= Y

Вставить пункт >>

Рис. 4.14 Расширенный режим редактирования меню.

Все данные, которые заносятся через административный интерфейс, будут сохранены в папке соответствующего раздела сайта в файле `.left.menu.php` для левого меню и `.top.menu.php` для верхнего меню соответственно. Файл содержит массив значений для всех элементов меню.

Файлы размещаются в тех разделах (директориях), где требуется их показ. Если для данного раздела не задан соответствующий файл меню, система производит поиск файла на уровень выше.


Построение меню для показа происходит следующим образом. В общем шаблоне показа меню содержится вызов функции вывода меню на экран. При загрузке страницы данная функция проверяет наличие в текущем разделе файла содержащего массив значений для меню, вызывает шаблон построения для данного типа меню и выводит HTML меню на экран.



Как это реализуется

Создание шаблонов меню начинается с выделения соответствующих HTML областей в шаблоне сайта. После этого выделяются неизменные верхняя и нижняя часть шаблона, а также повторяющиеся элементы.

Шаблоны для верхнего и левого меню хранятся в папке `/bitrix/templates/<идентификатор шаблона сайта>/` в файлах `top.menu_template.php` и `left.menu_template.php` соответственно.

Быстрый доступ к редактированию шаблона каждого типа меню можно осуществить по нажатию на иконку  «Редактировать шаблон меню», расположенную рядом с соответствующим меню в режиме показа включаемых областей.

Все шаблоны меню имеют одинаковую структуру:

- § область пролога шаблона меню;
- § область с описанием замен для различных условий обработки шаблона;
- § область тела шаблона меню;
- § область эпилога шаблона меню.

⚠ Примечание: Пример построения шаблона левого меню приводится в электронной версии документации, поставляемой на диске.

Вызов меню из основного шаблона показа производится включение соответствующего кода в областях, определенных при создании шаблона.

```
<?

//---Код включения для верхнего меню

echo $APPLICATION->GetMenuHtml("top");

?>

...

<?

//---Код включения для левого меню

echo $APPLICATION->GetMenuHtml("left");

?>
```

⚠ Примечание: В демонстрационной версии продукта приведен пример создания выпадающего JavaScript меню. Построение выпадающего меню выполняется на основе двух

типов используемых в системе меню: верхнего и левого меню. При необходимости данное меню может быть скопировано и использовано в других проектах.

Настройка дополнительных редактируемых областей



Как это выглядит

Шаблоны документов и шаблоны включаемых областей являются обыкновенными HTML-документами, которые содержат predetermined элементы форматирования: таблицы, картинки, стили и т.п. Созданные шаблоны копируются в папки `/page_templates/` либо для всех шаблонов (`/bitrix/templates/.default/page_templates/`), либо для каждого шаблона в отдельности (`/bitrix/templates/<идентификатор шаблона>/page_templates/`).

В системе существует возможность создания различных типов включаемых областей, например:

- § включаемая область для конкретного файла – будет выводиться только при просмотре текущего документа;
- § включаемая область для конкретной секции – будет выводиться для всех документов в данном разделе (секции);
- § другие типы включаемых областей



Рис. 4.15. Пример включаемой области.



Как это работает

Включаемые области сохраняются в отдельных файлах с некоторым суффиксом, например, `"_inc"`. Для конкретного файла этот суффикс добавляется к имени файла. В поставляемом дистрибутиве продукта включаемый файл для раздела (секции) сайта сохраняется с именем `sect_inc.php`. Количество редактируемых областей для каждого файла или раздела может быть расширено. При этом потребуются некоторая модификация шаблона.

Создание и редактирование включаемых областей для текущего документа может быть выполнено непосредственно из публичного раздела сайта в режиме показа включаемых областей. В тех местах, где предполагается вывод дополнительных редактируемых областей, будут показаны иконки для быстрого перехода к созданию или редактированию этих областей.

При создании включаемых областей можно использовать настроенные шаблоны показа для этих областей. Список шаблонов заполняется в файле `.content.php`.

Перечисленные в файле шаблоны будут доступны при создании новой страницы в виде выпадающего списка.

Следует обратить внимание, что имя шаблона может быть передано параметром при подключении редактируемой области в шаблоне сайта (см. выделение синим цветом в примере ниже).



Как это реализуется

Код для включения дополнительных редактируемых областей:

```
<?
$APPLICATION->IncludeFile(substr($APPLICATION->GetCurPage(), 0, strlen($APPLICATION->GetCurPage())-4)."_inc.php", Array(), Array("MODE"=>"html",
"NAME"=>GetMessage("PAGE_INC"), "TEMPLATE"=>"page_inc.php"));
?>

<?
$APPLICATION->IncludeFile($APPLICATION->GetCurDir()."sect_inc.php", Array(),
Array("MODE"=>"html", "NAME"=>GetMessage("SECT_INC"), "TEMPLATE"=>"sect_inc.php"));
?>
```

Работа с компонентами

Часто используемые области на сайте могут быть оформлены как программные компоненты. Практически любой программный скрипт может быть представлен компонентом.

Основная особенность компонентов – возможность управлять не только программным кодом, но также определенными параметрами, которые определяют поведение компонента.

Пользователь может быстро перейти к редактированию самого кода компонента или же просто изменить его параметры, например, для компонента «Лента новостей» задать количество выводимых новостей на странице или изменить порядок сортировки элементов и т.п.

Каждый компонент представляет собой программный скрипт, в котором выделены отдельные переменные для внешнего управления через визуальный интерфейс.

Для описания параметров компонента используются специальные файлы [.description.php](#), расположенные в той же или вышележащей директории, что и файл компонента.

Для быстрого добавления компонентов разработан специальный интерфейс на базе визуального редактора (рис. 7.1). При редактировании страницы можно выбрать компонент из списка и переместить его в рабочую область страницы. Для этого достаточно просто перетащить на страницу иконку компонента.

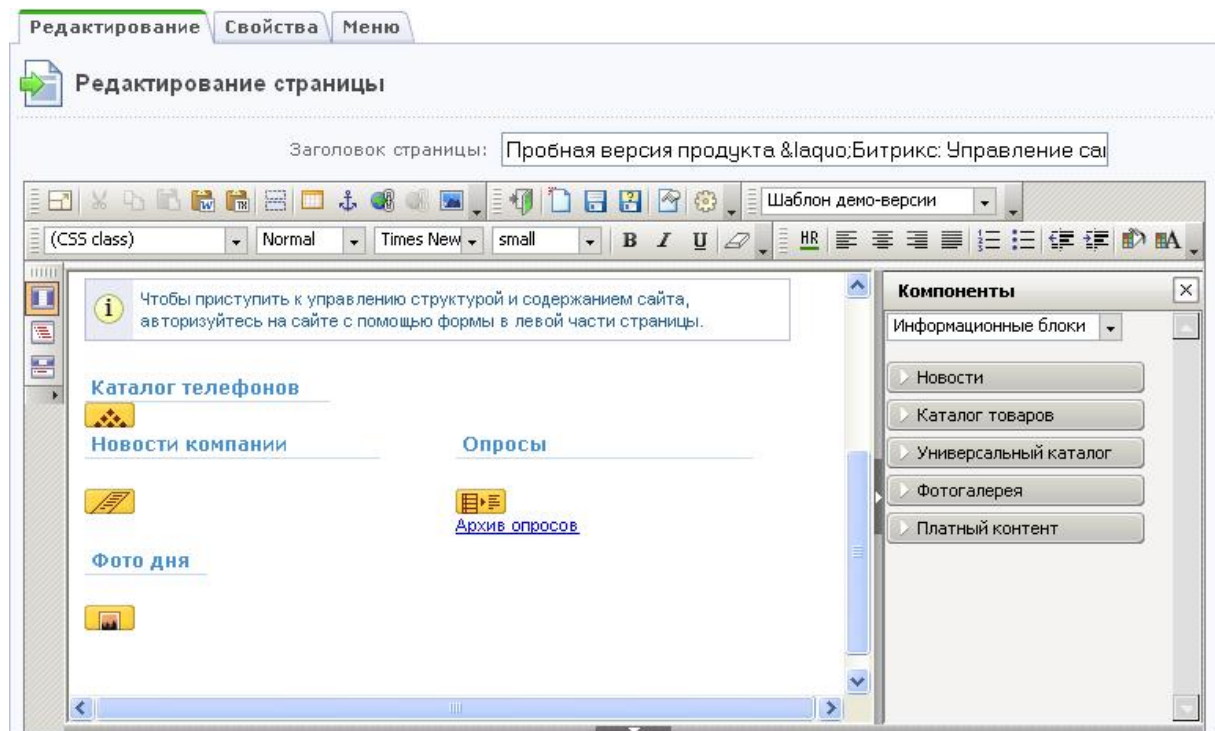


Рис. 7.1. Визуальный редактор.

После помещения на страницу компонента его параметры могут быть дополнительно настроены. При установке курсора на значок компонента в рабочей области в нижней части экрана будет выведен список параметров, доступных для данного компонента (рис 7.2).

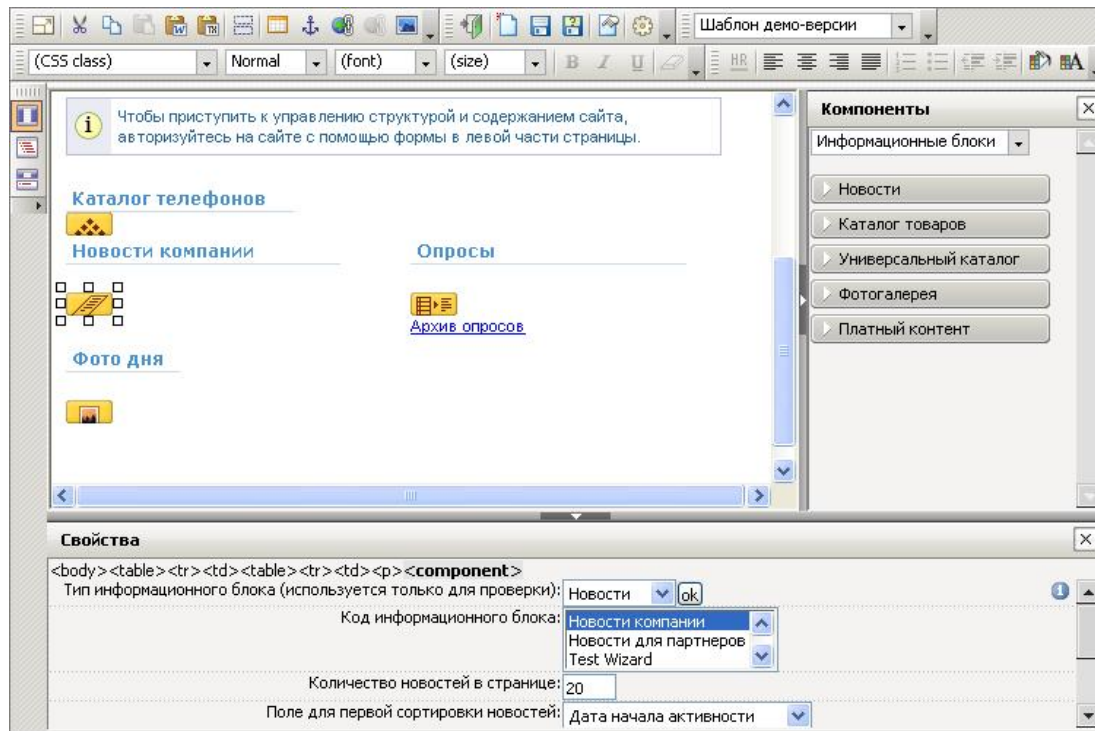


Рис. 7.2. Параметры компонента.

Аналогичным образом в рабочую область страницы может быть добавлен произвольный PHP код, модификация которого также выполняется через диалог в нижней части экрана (рис. 7.3).

⚠ Примечание: Возможность использования программного кода в рабочей области страницы позволяет разместить практически любой PHP код без ущерба для визуального режима редактирования содержимого страницы.

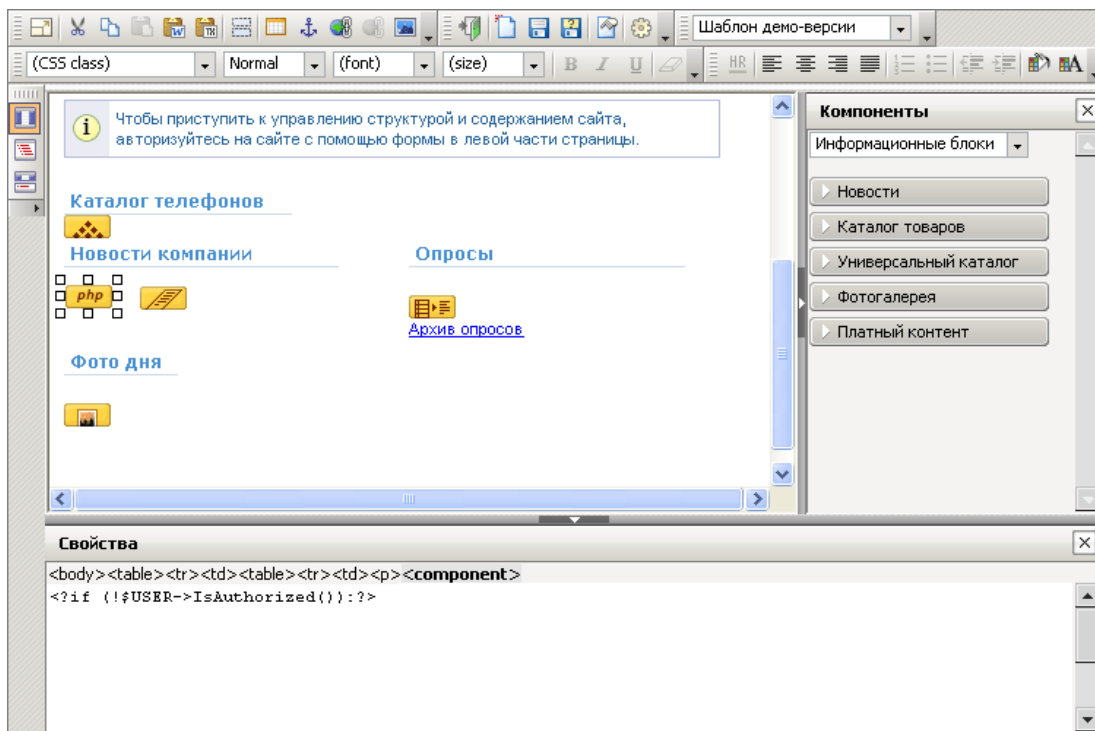


Рис. 7.3 Пример добавления PHP кода в визуальном редакторе

Непосредственно в коде страницы подключение компонентов производится с использованием функции IncludeFile(). В качестве параметров функции указываются путь к файлу с кодом компонента, а также параметры компонента, заданные при его добавлении на страницу.

Файлы компонента, поставляемые вместе с системой, могут размещаться непосредственно в папке с файлами модуля, к которому они относятся.

Например, компоненты для модуля информационных блоков находятся в следующих директориях:

</bitrix/modules/iblock/install/templates/iblock/news/>

и

</bitrix/modules/iblock/install/templates/iblock/catalog/>.

Либо они могут быть установлены в папку шаблона по умолчанию: </bitrix/templates/.default/iblock/news/>

и

</bitrix/templates/.default/iblock/catalog/> соответственно.

Компоненты, входящие в поставку продукта, разработаны с использованием специальных языковых файлов, в которых содержатся все текстовые сообщения необходимые для показа компонента, а также его описание в файле [description.php](#)

Например, описания всех компонентов модуля информационных блоков хранятся в файле:

</bitrix/modules/iblock/install/templates/iblock/.description.php>

В случае если требуется дополнительная настройка внешнего вида компонента, например формы поиска или формы авторизации, компонент может быть скопирован в директорию соответствующего шаблона сайта и отредактирован.

Либо компонент может быть изменен непосредственно в папке шаблона по умолчанию. В этом случае, его изменение отразится на всех шаблонах, в которые указаный компонент не был скопирован.

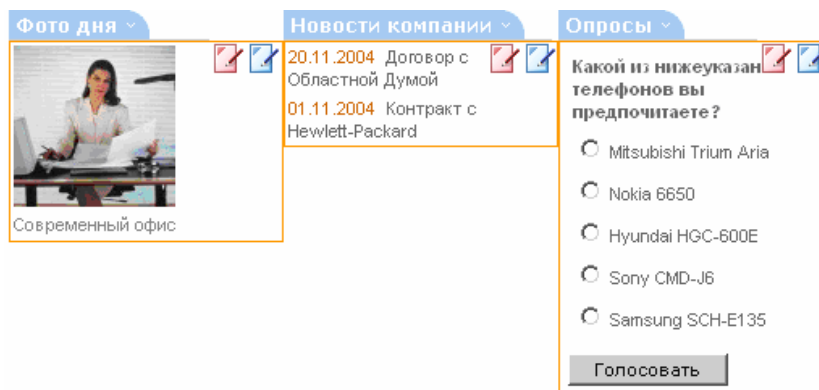



Рис. 7.4 Пример элементов управления компонентами

Иконка  позволяет перейти к редактированию кода компонента непосредственно в папке шаблона по умолчанию (</bitrix/templates/.default/>).

Иконка  позволяет скопировать компонент в папку текущего шаблона и перейти к редактированию кода компонента.

Ниже описаны некоторые особенности работы с основными компонентами, поставляемыми в составе дистрибутивного комплекта.

Модуль подписки

Компонент «Форма подписки»



Как это выглядит

Стандартная форма подписки включает динамически формируемый список рубрик подписки, поле для ввода электронного адреса подписчика, кнопку для подтверждения действия и ссылку для перехода на страницу управления подпиской. После заполнения формы, посетитель сайта будет переведен на форму упрощенной регистрации и затем будет добавлен в список подписчиков. Впоследствии он сможет изменить параметры подписки или отказаться от рассылки сайта. Страница упрощенной регистрации будет доступна в том случае, если в настройках сайта установлен флаг "Позволять ли пользователям регистрироваться самостоятельно?".

Рис. 8.1. Форма подписки.



Как это работает

Управление рубриками подписки, списком подписчиков и рассылкой сообщений осуществляется из административного раздела: Сервисы -Рассылки.

Администратор сайта заводит различные рубрики подписки. Посетители сайта подписываются на них. Затем администратор сайта создает рассылки, которые отправляются по списку подписчиков.



Как это реализуется

Шаблон для показа формы подписки находится в файле:

/bitrix/templates/<имя шаблона>/subscribe/subscr_form.php

В общем случае, если шаблон не требует дополнительной настройки дизайна, он может быть включен из

/bitrix/templates/.default/subscribe/subscr_form.php

Включение формы подписки выполняется следующим образом:

```
<?
$APPLICATION->IncludeFile( "subscribe/subscr_form.php", Array( "PAGE" =>
SITE_DIR."personal/subscr_edit.php" ) );
?>
```

Компонент «Страница подписки»

Компонент позволяет создать стандартную страницу подписки.

В параметрах можно задать путь к странице управления параметрами подписки, а также указать – нужно ли выводить количество подписчиков для рубрик подписки.

Компонент «Управление параметрами подписки»

Компонент позволяет создать страницу управления параметрами подписки. В параметрах компонента можно указать – разрешать ли анонимную подписку и показывать ли ссылку на авторизацию при анонимной подписке.

Модуль поиска

Форма поиска



Как это выглядит

Посетителям сайта доступна стандартная форма поиска:

The image shows a simple search form with a light blue border. At the top left, there is a small orange pushpin icon. Below it, the text 'поиск по сайту' is centered. Underneath, there is a white text input field with a light blue border, followed by a blue button with the white text 'Поиск'.

Рис. 8.2. Стандартная форма поиска.

После выполнения запроса на поиск посетитель сайта будет перенаправлен на страницу с результатами поиска.



Как это работает

Поиск производится по статическим материалам, информационным блокам (новостям, каталогу, вакансиям и т.п.) и форуму.



Как это реализуется

Включение формы в основной шаблон показа выполняется следующим образом:

```
<?
$APPLICATION->IncludeFile( "search/search_form.php", Array( "SEARCH_PAGE" =>
SITE_DIR."search/index.php" ) );
?>
```

В параметрах компонента можно указать путь к странице результатов поиска.

Компонент «Страница поиска»

В параметрах компонента можно указать значения для выпадающего списка «Где искать», количество результатов поиска, путь к шаблону дополнительной цепочки навигации и время кэширования значений выпадающего списка «Где искать».

Главный модуль

Формы авторизации



Как это выглядит

Стандартная форма авторизации позволяет посетителям сайта быстро ввести свои параметры входа и перейти в закрытые разделы при наличии соответствующих прав.



Как это работает

Форма авторизации позволяет проверить права пользователя для доступа к определенным разделам и страницам сайта. Права доступа могут быть заданы для страниц сайта, разделов сайта и информационных блоков.



Как это реализуется

Включение формы в основной шаблон показа производится следующим образом:

```
<? $APPLICATION->IncludeFile("main/auth/auth_form.php");?>
```

Панель переключения сайтов

Выводится панель переключения сайтов. Включение панели в основной шаблон выполняется следующим образом:

```
<? $APPLICATION->IncludeFile("main/site_panel.php");?>
```

Компонент «Карта сайта»

Компонент предназначен для вывода карты сайта в публичной части. Не имеет дополнительных параметров настройки.

Компонент «Параметры пользователя»

Компонент предназначен для показа и редактирования информации профайла текущего пользователя.

Модуль статистики

Размещение таблицы статистики



Как это выглядит

Посетителям сайта может быть показана таблица со статистическими данными по посещаемости сайта. В таблице представлена статистика по общему количеству посетителей сайта и по количеству посетителей за текущую дату. Для администратора сайта подписи представлены ссылками, по которым он может перейти из данной таблицы непосредственно в раздел просмотра детализированной статистики.



Как это работает

Система сбора статистики позволяет собрать и обработать данные по посетителям сайта, рекламным кампаниям, хитам поисковиков, ссылающимся сайтам, сессиям и т.п.



Как это реализуется

Включение таблицы статистики в основной шаблон показа выполняется следующим образом:

```
<? $APPLICATION-> IncludeFile("statistic/stat_table/default.php");?>
```

Модуль информационных блоков



Как это выглядит

Система "Битрикс: Управление сайтом" позволяет управлять как статической, так и динамической информацией. К динамической информации на сайте могут быть отнесены регулярно обновляющиеся материалы: новости, пресс-релизы, фотогалерея, вакансии, каталог товаров.

Управление динамической информацией в системе осуществляется на базе так называемых информационных блоков или информационных разделов, которые состоят из определенного количества элементов. Например, раздел новостей "Новости компании" является информационным блоком, а каждая новость является элементом информационного блока.



Как это работает

Управление информационными блоками осуществляется из административного раздела в секции Контент – Информационные блоки.

В системе могут быть представлены различные типы информационных блоков: новости, фотогалерея, вакансии, каталог. Для каждого информационного блока могут быть заданы свойства, которые затем будут унаследованы всеми элементами внутри данного информационного блока. Так, например, могут быть заданы специфические свойства товаров в каталоге: масса, габариты, цвет и т.п.

Свойства могут иметь предустановленное значение для всех элементов внутри информационного блока. Важными настройками для информационных блоков являются настройки пути к странице со списком элементов данного информационного блока, пути к странице со списком элементов раздела информационного блока и пути к странице детального просмотра элементов блока.

При выборке элементов блока важным параметром является не только ID блока, но также и мнемоническое имя блока, задаваемое символами латинского алфавита.

Для различных типов информационных блоков уровень вложенности элементов может определяться двумя уровнями: информационный блок – группы – элементы.



Как это реализуется

В новой версии продукта информация из информационных блоков выводится с использованием специализированных компонентов.

На базе отдельных компонентов выполнены страницы показа списка новостей, детального показа новостей, ленты новостей, страницы показа импортированных данных в формате RSS и страницы для экспорта данных информационных блоков в формате RSS.

В составе продукта поставляется большой комплект компонентов для вывода информации из каталога товаров.

Отдельная группа компонентов предложена для организации на сайте фотогалереи.

Пример организации показа новостей на сайте

Создание страниц для показа новостей начинается с заведения соответствующего информационного блока в административном интерфейсе и заполнения его элементов – новостей.

При создании нового информационного блока необходимо указать путь к странице показа списка элементов информационного блока (новостей), путь к странице со списком элементов раздела (если информационный блок содержит разделы) и путь к странице детального просмотра элемента (новости).

Указанные выше страницы должны быть созданы в структуре сайта. Например, для показа списка новостей </about/news/index.php>, а для детального показа новости </about/news/detail.php>.

Далее, на вновь созданной странице следует разместить соответствующие компоненты. Для страницы показа списка новостей в визуальном редакторе нужно перетащить на страницу иконку компонента «Список новостей» из раздела «Информационные блоки», а для страниц детального просмотра новости – иконку компонента «Новость детально».

Для того чтобы организовать экспорт новостей с этой страницы в RSS, в данном случае нужно вставить на страницу картинку RSS 2.0 и поставить на ней ссылку на некоторую страницу, например, </about/news/rss.php>. Указанную страницу следует также создать средствами продукта и в визуальном редакторе разместить на ней компонент экспорта в RSS формате.

Предварительно нужно убедиться, что в настройках соответствующего информационного блока разрешен экспорт данных в формате RSS.

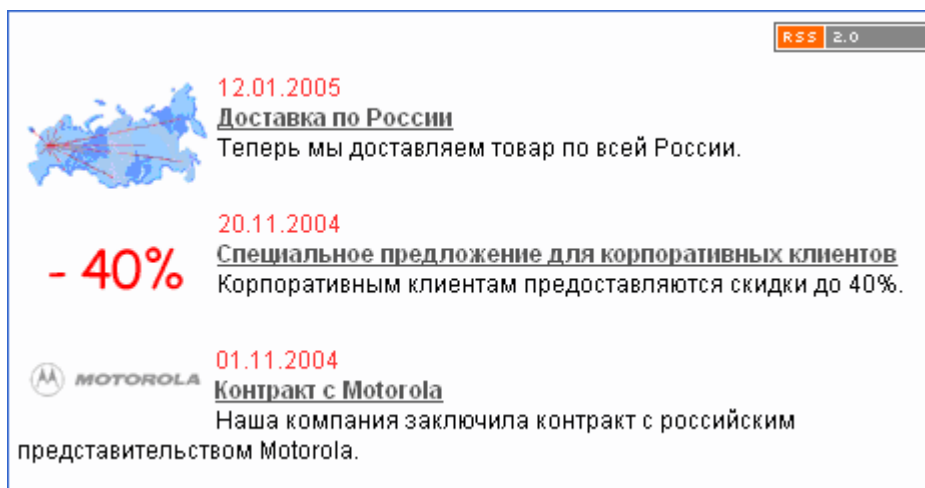


Рис. 10.1 Показ новостей со ссылкой на страницу экспорта новостей в формате RSS

Модуль Веб-форм

Показ веб-формы в публичной части сайта осуществляется следующим образом:

- § создается веб-форма в административном разделе сайта: Сервисы - Веб-формы - Настройка форм;
- § в публичной части сайта на некоторой странице в визуальном редакторе добавляется компонент «Заполнение веб-формы (шаблон по умолчанию)»;
- § в настройках компонента выбираются название формы, а также названия страниц для показа списка результатов и редактирования результата веб-формы;
- § последовательно, средствами продукта добавляются указанные страницы и на них помещаются соответствующие компоненты;
- § для всех компонентов выполняются настройки.

Модуль Опросов

Размещение опросов производится аналогичным образом. В настройках компонентов дополнительно указываются названия страниц с отображением результатов опросов.

В системе представлен компонент для вывода опросов, например, на главную страницу сайта с последующим отображением результатов. Для не голосовавших посетителей – показывается опросы, сразу после голосования – показываются результаты опросов; для уже проголосовавших ранее посетителей – показываются результаты опросов.

В настоящий момент компоненты разработаны для всех основных модулей системы, и список их будет пополняться как со стороны нашей компании, так и со стороны партнеров компании «Битрикс».

Использование файлов языковых сообщений

При выделении общего шаблона для нескольких языков, следует обратить внимание на текстовые элементы дизайна: заголовки, альт-текст к рисункам, подписи на кнопках и т.п.

Шаблон показа настраивается таким образом, что HTML-код является общим для всех языков, а все текстовые элементы в нем заменены на текстовые сообщения. Языковые сообщения хранятся отдельно в папке `/bitrix/templates/<идентификатор шаблона>/lang/ru/` для русского языка и, соответственно, в папке `...../lang/en/` для английского языка (хранение языковых сообщений для других языков выполняется аналогично).

Например, языковые сообщения для редактируемой области с копирайтами (файл `/bitrix/templates/<идентификатор шаблона>/copy.php`) хранятся соответственно в файлах:

§ `/bitrix/templates/<идентификатор шаблона>/lang/ru/copy.php` для русского интерфейса

§ и в файле `/bitrix/templates/<идентификатор шаблона>/lang/en/ copy.php` для английского интерфейса.

Файл с языковыми сообщениями для русского языка выглядит следующим образом:

```
<?
$MESS ['SEARCH_TITLE'] = "Поиск";
?>
```

В файле `/bitrix/templates/<идентификатор шаблона>/lang/ru/copy.php` языковые сообщения подключаются в самом начале:

```
<?
IncludeTemplateLangFile(__FILE__);
?>
```

В самом документе языковое сообщение вызывается следующим кодом (например, для кнопки «Копировать»):

```
<font class="copy"><?echo GetMessage("COPY");?></font>
```

Настройка продукта на техподдержку со стороны партнеров

Партнеры компании «Битрикс» могут произвести некоторые изменения в административной части продукта.

Для того чтобы настроить сайт на техподдержку со стороны партнеров следует произвести следующие действия:

1. Изменение адреса сайта и ссылки на техподдержку в правом нижнем углу экрана административной части.

Создать файл `/bitrix/php_interface/this_site_support.php`, который будет содержать необходимый текст. Тогда будет подключаться именно он, а не стандартный текст.

Пример файла `/bitrix/php_interface/this_site_support.php`:

```
<a href="http://www.flowersweb.info" class="pagebottomtext">www.flowersweb.info</a>
 
  | 
<a class="pagebottomtext"
href="http://www.flowersweb.info/support/">Техподдержка</a>
```

2. Встраивание логотипа или текста в левый нижний угол экрана (под меню) административной части.

Создать файл `/bitrix/php_interface/this_site_logo.php`, который будет содержать необходимый текст.

Если этот файл существует, то именно его содержимое будет показано. Содержимое файла создается произвольно.

3. Изменение почтового адреса, на который направляются письма в техподдержку со страницы Проверка сайта - Техподдержка (`/bitrix/admin/site_checker.php`)

Необходимо определить константы

```
define("THIS_SITE_SUPPORT_EMAIL", "E-Mail системы техподдержки");
```

```
define("THIS_SITE_SUPPORT_CHARSET", "Кодировка");
```

Определять кодировку не обязательно, в этом случае она будет устанавливаться автоматически в "windows-1251" для русского языка и в "iso-8859-1" для остальных.

Определить константы можно в файлах

`/bitrix/php_interface/dbconn.php` (подключается в самом начале)

или

`/bitrix/php_interface/after_connect.php` (подключается после соединения с базой)

или

`/bitrix/php_interface/<код сайта>/init.php` (подключается после определения сайта, для определенного сайта).

Формат файла описания шаблонов компонентов

Файлы описания шаблонов компонентов могут располагаться в следующих директориях:

- § в папке с шаблоном компонента, например:
`/bitrix/modules/iblock/install/templates/iblock/news/.description.php`,
- § в корневой папке для шаблонов модуля, к которому относится компонент, например:
`/bitrix/modules/iblock/install/templates/iblock/.description.php`.

Обратите внимание, что в корне шаблонов модуля (в общем случае это папки первого уровня в шаблоне) в любом случае должен присутствовать файл `.description.php`, содержащий название раздела (модуля) в переменной `$$sSectionName`.

Сами шаблоны описываются в массиве `$arTemplateDescription`, имеющем следующий формат:

- § "индекс массива" - путь к шаблону относительно папки, в которой находится файл `.description.php`.
- § "значение массива" - массив формата:
 - "NAME"** - название шаблона (обязательный),
 - "DESCRIPTION"** - описание шаблона,
 - "ICON"** - путь к иконке шаблона,
 - "PARAMS"** - массив параметров шаблона в формате:
 - "NAME"** - название параметра
 - "TYPE"** - тип параметра (STRING или LIST)
 - "MULTIPLE"** - множественность, если "Y", то будут передаваться в шаблон как массив
 - "VALUES"** - значения для свойства типа LIST. Массив "код"=>"значение"
 - "DEFAULT"** - значение по умолчанию. Для множественных – массив значений.
 - "SIZE"** - размер поля select для свойств типа LIST
 - "CNT"** - количество полей для ввода значений множественных свойств.
 - "COLS"** - количество столбцов в полях для ввода
 - "ROWS"** - количество строк в полях для ввода
 - "ADDITIONAL_VALUES"=>"N"** – используется для отключения поля ручного ввода значений для свойств типа "LIST"

В самих полях для ввода обычный текст (например, "news") интерпретируется как строка.

Для того чтобы записать выражение PHP необходимо все значение поля ввода обрмить в `"={...}"`.

Например:

```
={$_REQUEST["BID"]}
```

или

```
={{(isset($_REQUEST["BID"])?$_REQUEST["BID"]:"15")}}
```

Ниже приводится пример файла `.description.php` в корне шаблонов модуля (пример не реальный; как правило, текст в файле должен быть заменен на `GetMessage()`):

```
<?
IncludeTemplateLangFile(__FILE__);
$$sSectionName = "Информационные блоки";
```

```

$arTemplateDescription =
    Array(
        "iblock/news/index.php" =>
            Array(
                "NAME"           => "Список новостей",
                "DESCRIPTION"    => "Показывает список новостей, разделяя по информационным
блокам",
                "ICON"          => "/bitrix/images/iblock/components.gif",
                "PARAMS"        =>
                    Array(
                        "IBLOCK_TYPE"      => Array("NAME"=>"Код типа инфоблока",
"TYPE"=>"LIST", "MULTIPLE"=>"N",
"VALUES"=>Array("news"=>"Новости"), "DEFAULT"=>""),
                        "IBLOCK_ID"       => Array("NAME"=>"Код инфоблока", "TYPE"=>"LIST",
"MULTIPLE"=>"Y",
"VALUES"=>Array("32"=>"Новости компании", "12"=>"Новости продукта"),
"DEFAULT"=>"12"),
                        "COUNT_FOR_COLUMN" => Array("NAME"=>"Количество строк в каждом
столбце", "TYPE"=>"STRING", "DEFAULT"=>"10"),
                        "COLUMN_COUNT"    => Array("NAME"=>"Количество столбцов",
"TYPE"=>"STRING", "DEFAULT"=>"1")
                    )
            )
    );
?>

```

Приложение 1. Рекомендации по подготовке HTML шаблона

Подготовка базового графического дизайна сайта должна проводиться с учетом дополнительных требований, которые накладываются использованием продукта "Битрикс: Управление сайтом".

- § При подготовке графического дизайна следует заранее разметить линию раздела дизайна на "пролог" (**header.php**) и "эпилог" (**footer.php**);
- § Следует выделить основные элементы дизайна, для последующей модификации таблицы стилей: шрифты, цвета заливки и т.п.;
- § Разрабатывая дизайн меню различных уровней, желательно выделять повторяющиеся элементы - для упрощения создания шаблона меню и дальнейшего управления этими меню.
- § Для облегчения сопровождения различных языковых версий сайта по возможности следует использовать вместо графических элементов текстовые.
- § При нарезке графического дизайна и подготовке HTML шаблона, необходимо заранее предусмотреть место расположения основных компонентов системы управления сайтом. Выделить области меню, рекламные области, области размещения дополнительных форм.
- § Рекомендуется производить подготовку шаблона с учетом последующей табличной сборки. Одновременно допускается использование слоев.
- § При нарезке графического дизайна выделяются однотонные области. При сборке шаблона эти области могут быть представлены ячейками таблиц со сплошной заливкой цвета.

Приложение 2. Настройка внешнего вида дополнительных элементов

Настройка сообщений об ошибках

Часто возникает необходимость выполнить настройку сообщений об ошибках, чтобы сообщение об ошибке было аккуратно показано в дизайне сайта.

Для того чтобы произвести настройку внешнего вида сообщения об ошибке соединения с базой данных, следует отредактировать файл:

/bitrix/php_interface/dbconn_error.php.

Для того чтобы произвести настройку внешнего вида сообщения об ошибке в запросе к базе данных, следует отредактировать файл:

/bitrix/php_interface/dbquery_error.php.

Настройка файла подключаемого при закрытии сайта

Для того чтобы произвести настройку внешнего вида файла, подключаемого при закрытии публичной части сайта, следует скопировать файл:

/bitrix/modules/main/include/site_closed.php

и поместите его в

/bitrix/php_interface/<язык>/

или в

/bitrix/php_interface/include/

Настройка внешнего вида навигации постраничного просмотра

Постраничный показ информации организуется с использованием PHP функции [NavPrint\(\)](#) – функции вывода ссылок для постраничной навигации. Для управления внешним видом постраничной навигации могут быть использованы следующие параметры:

`NavPrint($title, $show_allways=false, $StyleText="text", $template_path)`

где

\$title – название выводимых элементов,

\$show_allways – если значение параметра "false", то функция не будет выводить навигационные ссылки, если все записи уместятся на одну страницу. Если "true", то ссылки для постраничной навигации будут выводиться всегда,

\$StyleText – CSS класс шрифта для вывода навигационных ссылок;

\$template_path – путь к шаблону показа навигационных ссылок.

Шаблон показа постраничной навигации формируется по умолчанию следующим образом (можно вынести в отдельный файл, скорректировать и указать путь к шаблону в вызове функции):

```
<?
echo('<font class="'. $StyleText. '">'. $title. ' ');
echo(( $this->NavPageNomer-1) * $this->NavPageSize+1);
echo(' - ');
```

```

if($this->NavPageNomer != $this->NavPageCount)

    echo($this->NavPageNomer * $this->NavPageSize);

else

    echo($this->NavRecordCount);

echo(' '.GetMessage("nav_of").' ');

echo($this->NavRecordCount);

echo("\n<br>\n</font>");

echo('<font class="'.StyleText.'">');

if($this->NavPageNomer > 1)

    echo('<a href="'.UrlPath.'?PAGEN_'. $this->NavNum.'=1'. $strNavQueryString. '#nav_start'. $add_anchor.' ">'. $sBegin.' </a>&nbsp;|&nbsp;
sp;<a href="'.UrlPath.'?PAGEN_'. $this->NavNum.'='.( $this->NavPageNomer-1).

$strNavQueryString. '#nav_start'. $add_anchor.' ">'. $sPrev.' </a>');

else

    echo($sBegin.' &nbsp;|&nbsp;'. $sPrev);

echo('&nbsp;|&nbsp;');

$NavRecordGroup = $nStartPage;

while($NavRecordGroup <= $nEndPage)

{

    if($NavRecordGroup == $this->NavPageNomer)

        echo(' '. $NavRecordGroup.' &nbsp;');

    else

        echo('<a href="'.UrlPath.'?PAGEN_'. $this->NavNum.'='.$NavRecordGroup.$strNavQueryString. '#nav_start'. $add_anchor.' ">'. $NavReco
rdGroup.' </a>&nbsp;|&nbsp;');

        $NavRecordGroup++;

}

echo(' |&nbsp;');

if($this->NavPageNomer < $this->NavPageCount)

    echo ('<a href="'.UrlPath.'?PAGEN_'. $this->NavNum.'='.( $this->NavPageNomer+1).

$strNavQueryString. '#nav_start'. $add_anchor.' ">'. $sNext.' </a>&nbsp;|&nbsp;
bsp;<a

```

```
href="'. $sUrlPath.'?PAGEN_'. $this->NavNum.'='.' $this->NavPageCount.' $str
NavQueryString.' #nav_start'. $add_anchor.'">'.' $sEnd.'</a>&nbsp;');

else

    echo ($sNext.'&nbsp;|&nbsp;'. $sEnd.'&nbsp;');

//"All" section is being show with the code below

if($this->bShowAll)

    echo ($this->NavShowAll? '|&nbsp;<a href="'. $sUrlPath.'?SHOWALL_'. $this-
>NavNum.'=0'. $strNavQueryString.'#

nav_start'. $add_anchor.'">'.' $sPaged.'</a>&nbsp;': '|&nbsp;<a
href="'. $sUrlPath.'?SHOWALL_'. $this->NavNum.'=1'. $strNavQueryString.'#
nav_start'. $add_anchor.'">'.' $sAll.'</a>&nbsp;');

echo('</font>');

?>
```