Bitrix Site Manager 4.0

Integration Manual
Quick Start Guide

Version 4.0.5 as of 04.22.2005

Introduction

This document is for use by the IT specialists, web designers and developers; as well as the Internet agency and web design studios staff.

This document depicts common guidelines that should be followed when integrating the Bitrix Site Manager into the web solution design or developing custom solutions.

This paper aims to arm the web specialist with basic knowledge of quicker integration of the product into the design.

New integration technologies in 4.0

The Bitrix Site Manager 4.0 offers a wide range of innovations aimed to speed up the web site building process.

Design templates, the traditionally flexible facility, allow to integrate the software in the design within few hours, start developing the project structure and allocating the business logics software components.

A design template can be uploaded in the form of a single file package using the web interface and applied to one or more sites.

The complete ready-to-go templates can be exported as a single file package and used in other projects or sold to other users of the Bitrix Site Manager.

Templates and sites

The Bitrix Site Manager 4.0 supports the multiple sites concept, which allows creating several sites using a single copy of the product. Each site has its own domain name (or several domain names), design, interface language and content.

The design of a site is the subject of a design template. Each site can be assigned an unlimited number of design templates. The use of templates opens up vast possibilities in the site design customization depending on your needs and various conditions.

The software provides means for flexible design customization for different sites and site sections; for example, it allows using special holiday design for the specified period of time, assign certain design templates to different site visitor groups, switch templates depending on a parameter in the site URL etc (fig. 1.1).

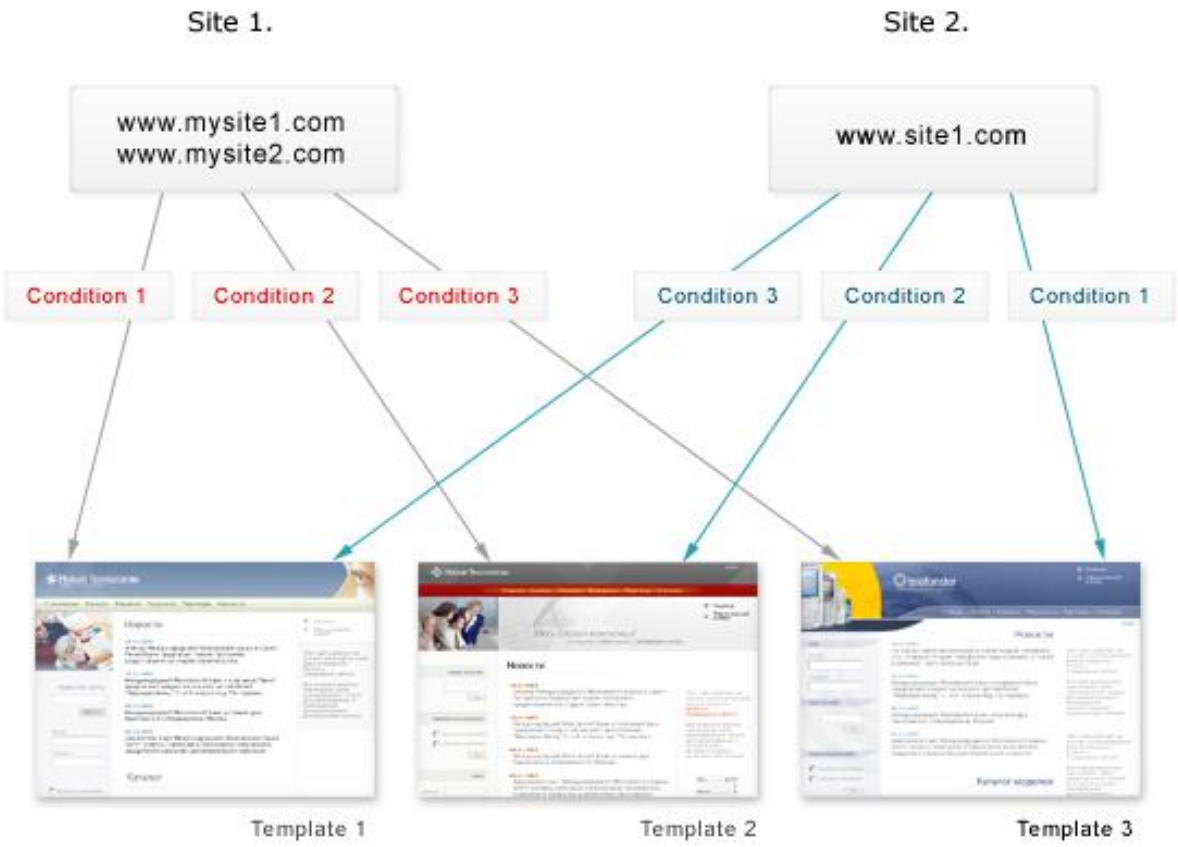

Fig.1.1. Templates and sites

You can select and assign a template to each site listed in the “Settings->Sites” page of the administrative section (fig. 1.2).

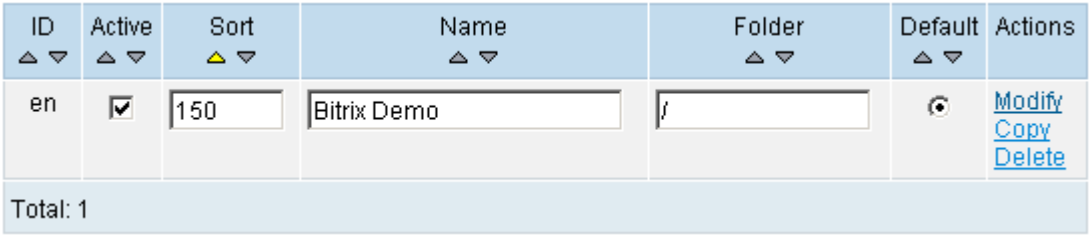

Fig. 1.2. List of sites.

For each site you can define a set of conditions that will regulate the site design selection (fig. 1.3). You may choose to not use conditions at all; in this case, the selected template is used as the default site template.

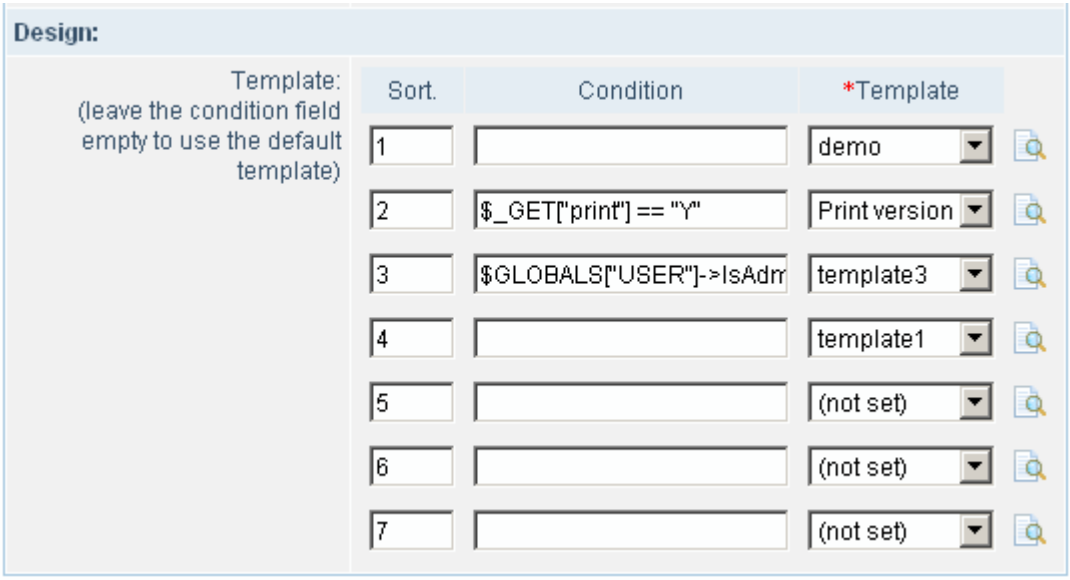

Fig. 1.3. Setting the site templates.

Brief explanation of the screenshot Fig. 1.3:

1. The site will be displayed with the **demo** as a default template. This template will be used first because it is not assigned a condition and its sort weight is 1.
2. The **Print Version** template is used if the URL parameter **print=Y** is specified. For example: if a visitor opens a page using a link like http://www.site2.com/?print=Y, this page will be displayed using the template that enables the correct page printouts.
3. The **template3** is used if a user has been authorized and logged in as an administrator.

Note: a condition is allowed to contain an arbitrary PHP code including the Bitrix Site Manager SDK calls. For details, please see the SDK help section.

You can ensure that the template is configured correctly by clicking the icon  near the templates drop-down list. This will open the *preview mode display* of a site with the selected template applied to it. (Site home page will be used for preview mode.)

An unlimited number of templates can be applied to a single site. A template can be reused for any other sites under different conditions.

Templates Management. Template Structure

A common site design usually includes three main parts (fig. 2.1).

1. Topmost section (**header**).
2. Main section that is used to display the information site content, components or any other program code.
3. Bottom section (**footer**).

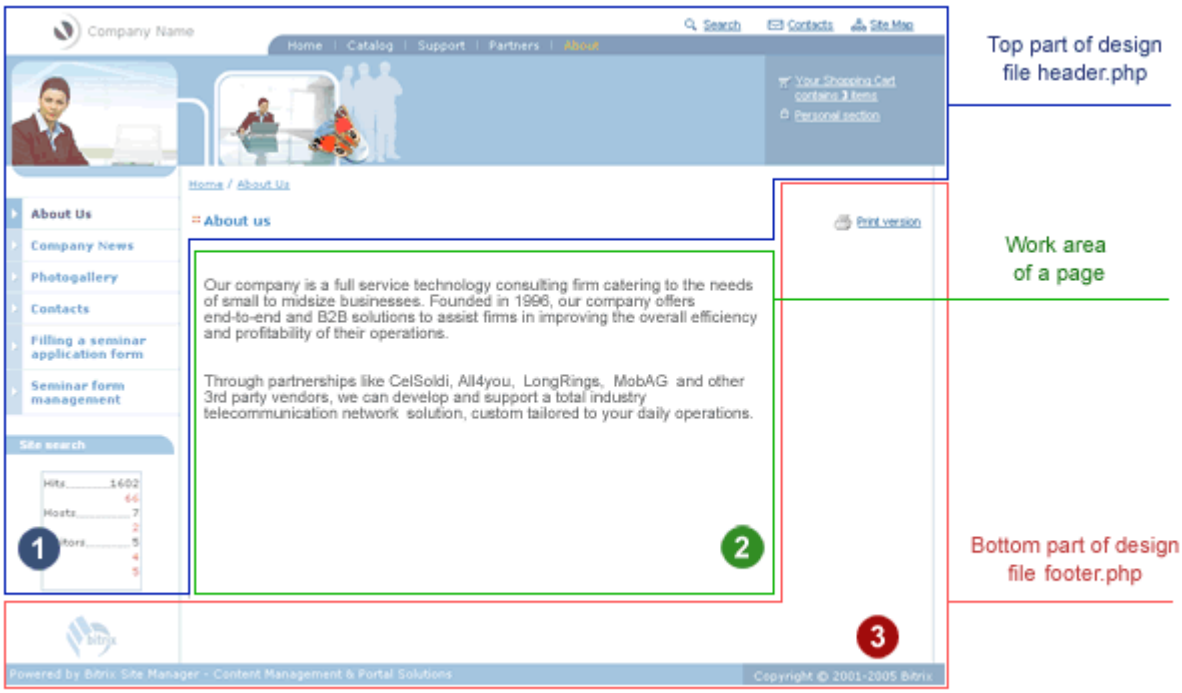

Fig. 2.1. Main design sections.

Let us consider how the design sections can be implemented in a template. We shall elaborate on the template **demo**, which is included in the software installation package.

Templates Management

You manage your site templates in the administrative section *Settings->Site Templates* (fig. 2.2).

ID	Name	Description	Actions
demo 	demo	Default site template	Modify Copy Download Delete
print 	Print version	Print version template	Modify Copy Download Delete
template1 	template1	Mobile Phone shop template	Modify Copy Download Delete

Fig. 2.2. List of templates.

There you can view and edit the existing templates and also add new templates. Click *Modify* to start editing the template **demo**.

The field **Site template design** contains the template code (fig. 2.3):

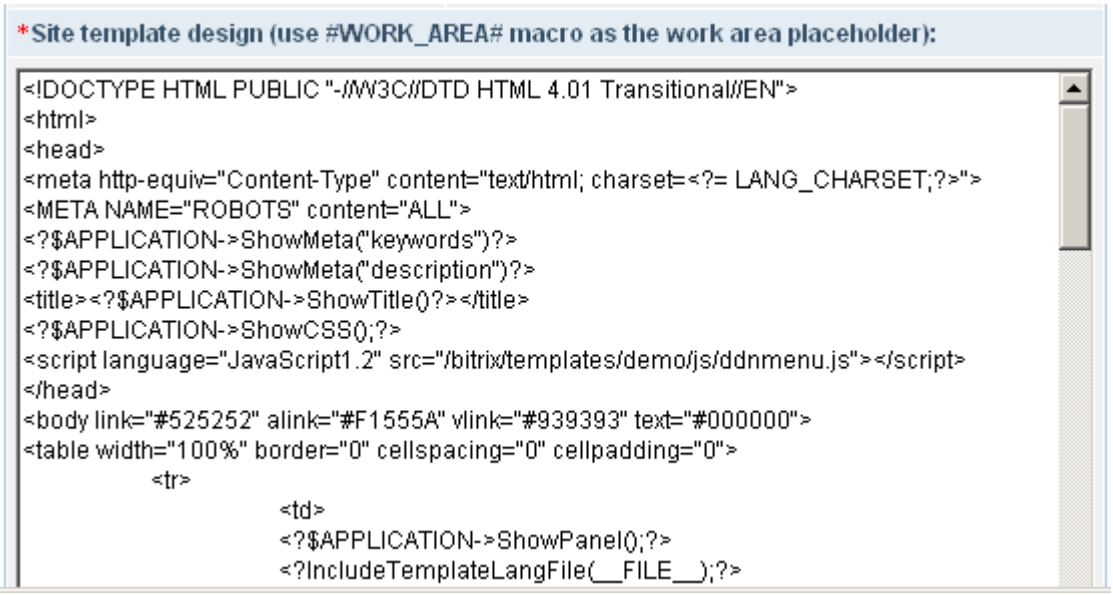

Fig. 2.3. Text field with the site template design.

> **Note**: the template contains a special separator #WORK_AREA# used to delimit the header and the footer sections.

This field displays a template as the joined header and footer parts of the site design in the PHP/HTML format. The code can contain component calls and different functions written in the PHP language that provide displaying of different information: metadata, page header, CSS, administrative toolbar etc. You can add and edit components and functions using the visual editor tools.

The template is stored in files **header.php** and **footer.php**.

> **No limitations on the temlpate composition and site design.**
>
> The use of executable code in templates ensures the flexibility in implementing a web project, allows to develop both simple and complex templates with arbitrary logics and individual design. If you are not quite familiar with the PHP language, you can include software components by simple copying the required code from other templates.

Cascading style sheet (CSS) file used in the template is shown in the field **Style sheet** (fig. 2.4).

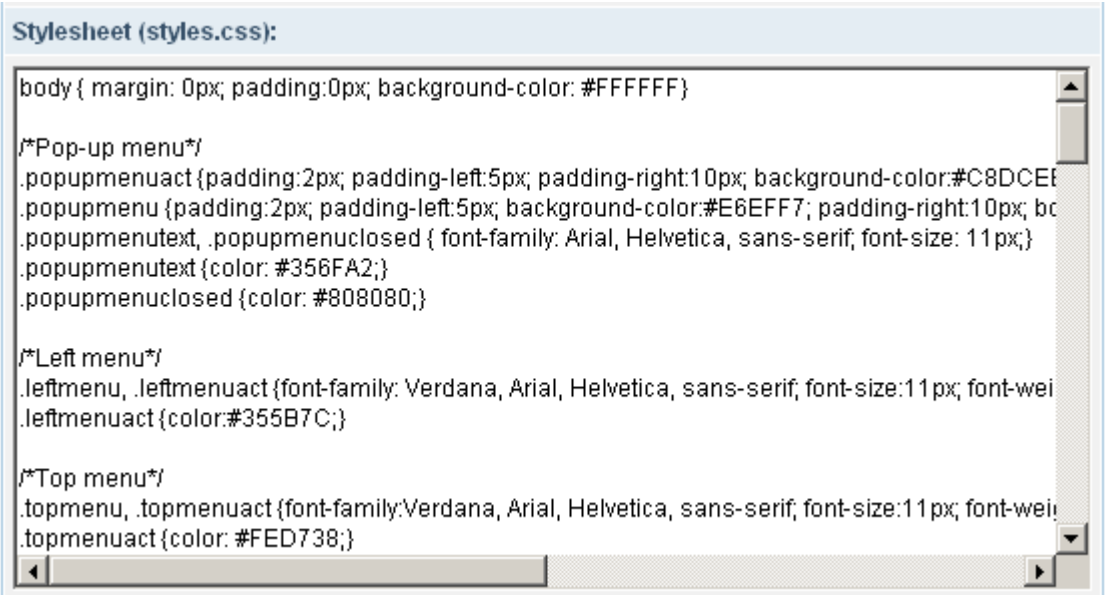

Fig. 2.4. Style entry field.

The CSS is stored in file styles.css.

All templates are stored in directory /bitrix/templates/. Files that comprise the template are stored in subdirectory named by the template identifier. In this example, the template files are located in subdirectory /bitrix/templates/demo/.

When you create a new template using the administrative interface, you provide its identifier, name and description for use in the list of templates. Also you need to provide the site design code, CSS and a set of included components and pictures. When you save the newly created template, the directory /bitrix/templates/<template_identifier> is created automatically.

The file and directory structure of each template is fairly identical. The simplest template may include only the following required files: header.php, footer.php, styles.css, as well as some more menu template files.

All images used in a template are stored in the subdirectory

/bitrix/templates/<template_identifier>/images/.

Any other files and components may also be stored in the template directory. You can view the template structure by clicking the link near the site identifier (fig. 2.5).

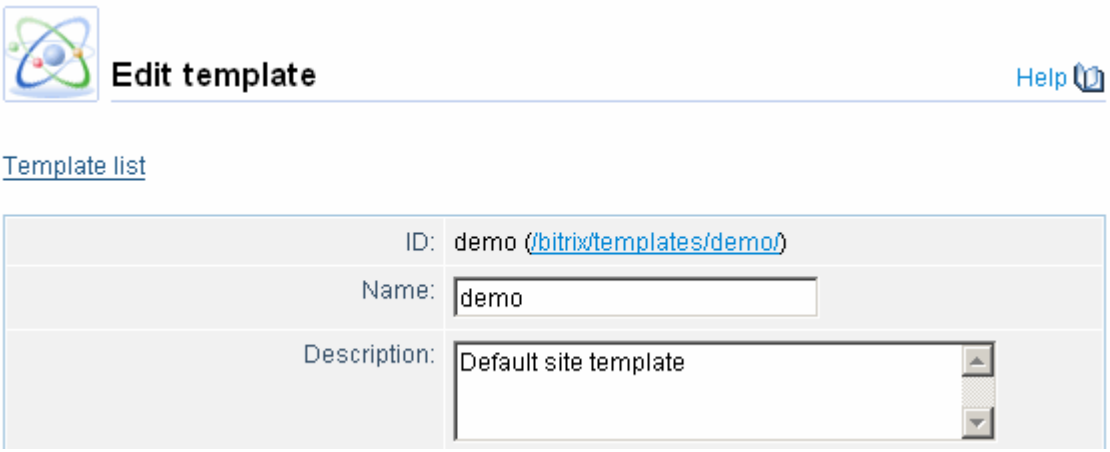

Fig. 2.5. Link to the template structure view.

Template files are shown in the administrative section **Site Explorer** (fig. 2.6):

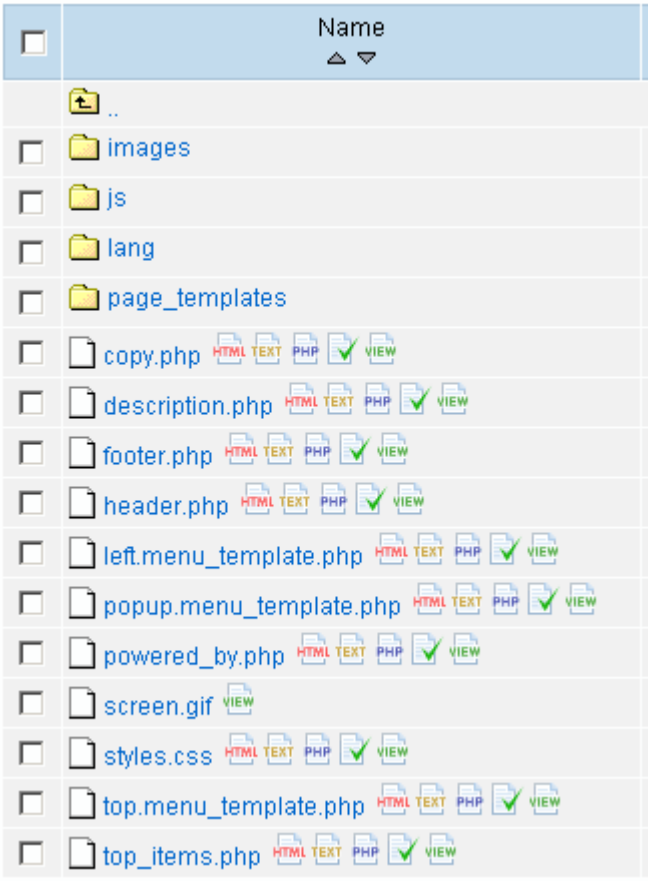

Fig. 2.6. Template structure.

You can split the site template into any number of files representing the site design sections to facilitate access to them. Examples of such sections are: copyrights area, contact information area etc.

Include areas and components

To provide the visual dynamic control over a modern web site, some design parts are implemented as the software components.

Get back to the template **demo** and take a look at the basic components and include areas. The figure 3.1 spotlights components and include areas of this template.

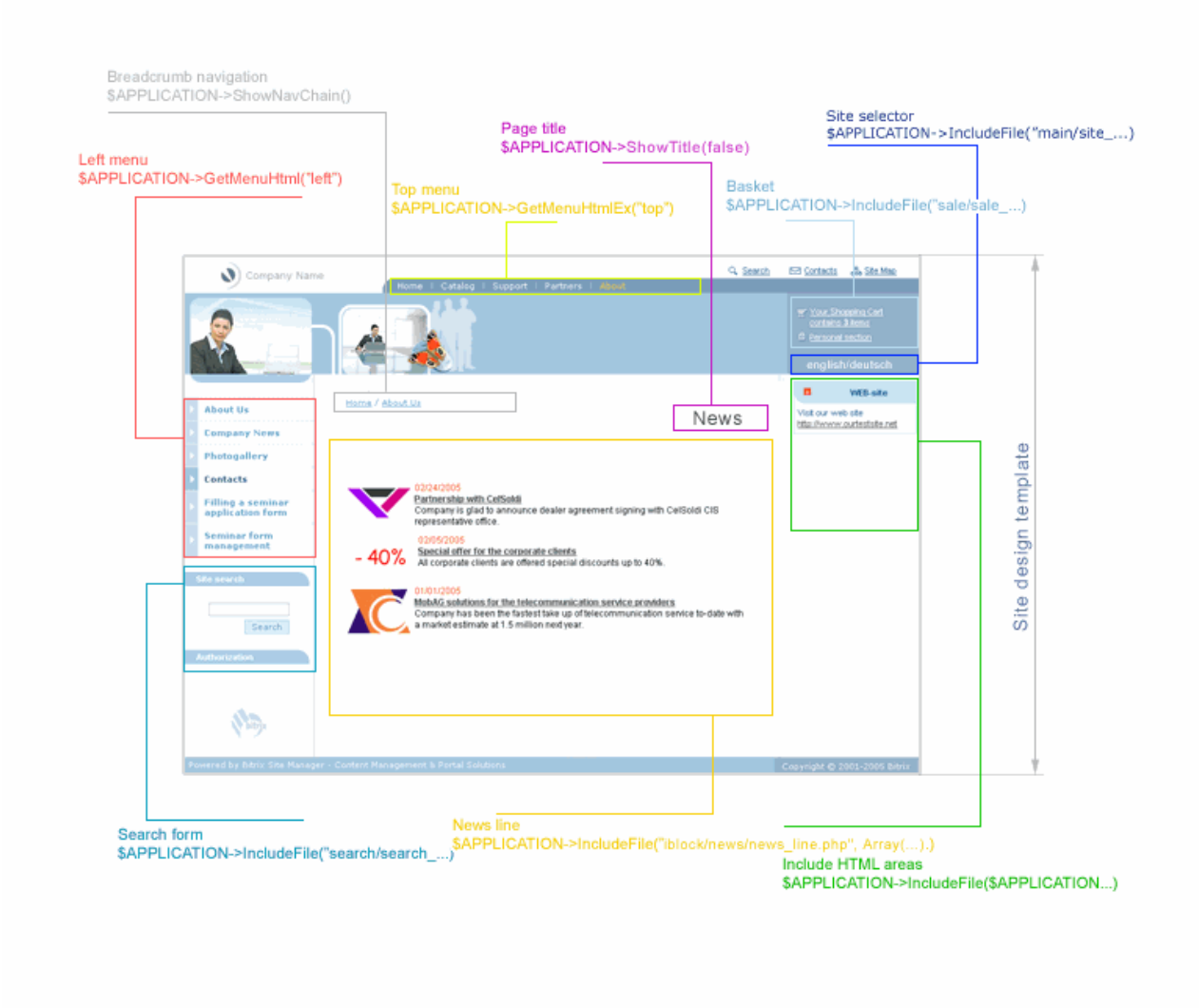

Fig. 3.1. Include areas in template.

When generating the HTML code, the system replaces the spotlighted areas with inclusions of the corresponding components (fig. 3.1).

The generated code includes functions which provide the display of different visual information: metadata, page header, CSS, administrative toolbar.

The following navigational templates are created separately: navigation chain (breadcrumb navigation), site menus..

After an HTML prototype has been created and function calls and component calls were added, you obtain the ready-to-go PHP template of the site design.

*Please refer to the **Integration Manual** for more detailed information on the creation of navigational elements and the appropriate API functions.*

You can turn on the special mode that enable to view the include areas by clicking the button  in the administrative toolbar (fig. 3.2).

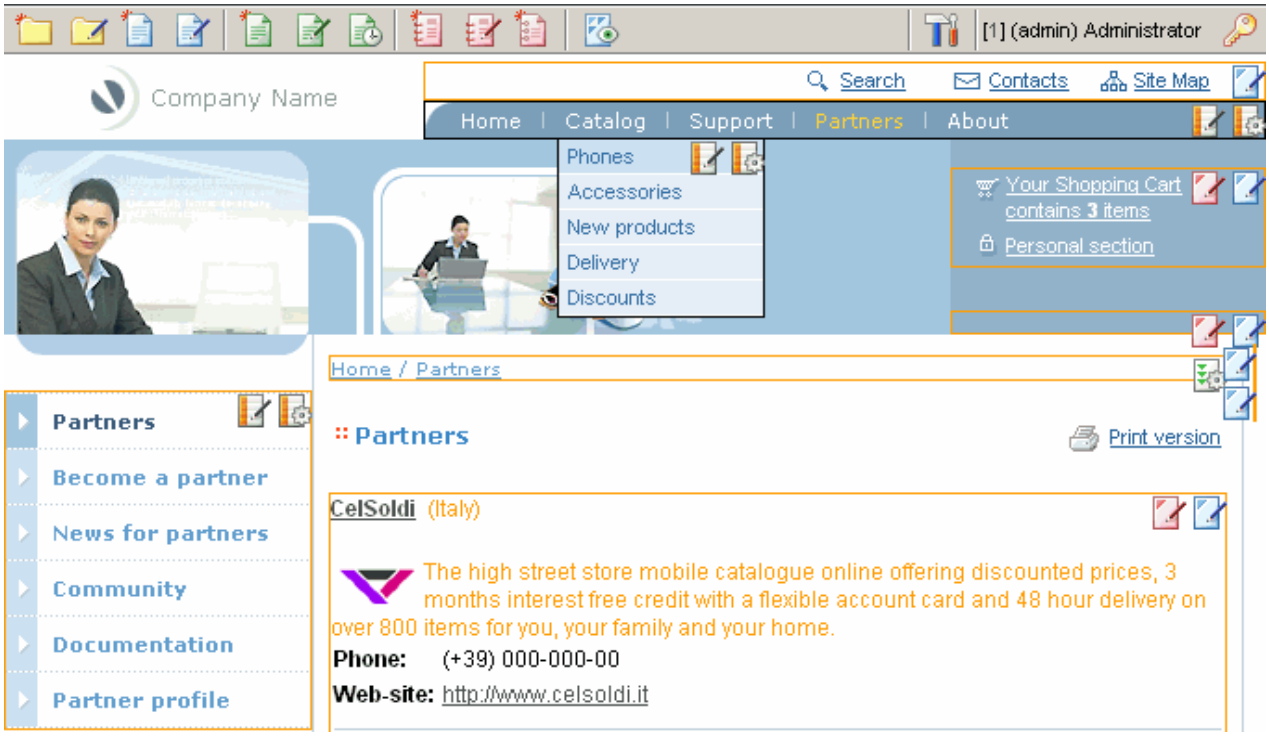

Fig. 3.2. Include areas and components view mode.

Include components may have various controls indicated by icons.

For example, the advertising banner controls provide access to the banner editing facilities. Additionally, it enables to view the list of banners filtered by the selected banner type, in the administrative section.

The menu controls enable to edit the menu items of the selected section or switch to the menu template editing.

You can define and use in your design various include HTML areas stored as individual files. You may want to set these areas to display under certain conditions only, for example in current site section or page. You can use the include area controls to switch to editing immediately.

Software components

It is a good practice to arrange frequently used areas and code as the software components. Virtually any script can be shaped as a component.

A distinctive feature of a component is the ability not only to be controlled on the source code level, but also using parameters that define the component behaviour.

A user can quickly edit the component source code or just change its parameters. For example, the component *Newsfeed* can be assigned a number of news per page or the sort order of elements etc.

Each component is the software script that “exports” special variables by the use of which it can be controlled through the visual interface.

To describe the component parameters, special files are used. These files have a fixed name .description.php and are located in the same or parent directory of the component file.

You can quickly add components to pages using the visual editor (fig. 4.1). To insert a component into a page, simply select it in the list and drag and drop it to the desired position within the page.

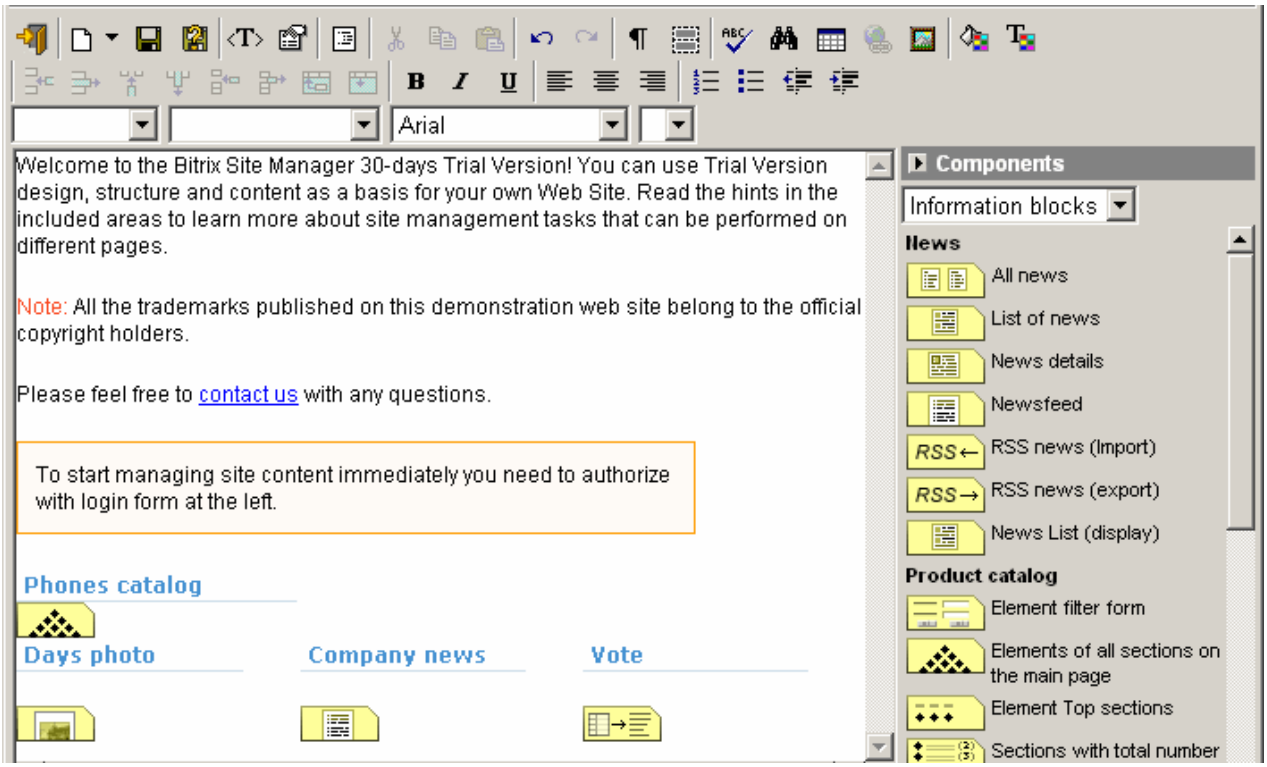

Fig. 4.1. Visual HTML editor.

After you have added a component, you can customize its parameters. To do so, select an existing component in the page; the list of available component parameters will appear in the bottom of the screen (fig 4.2).

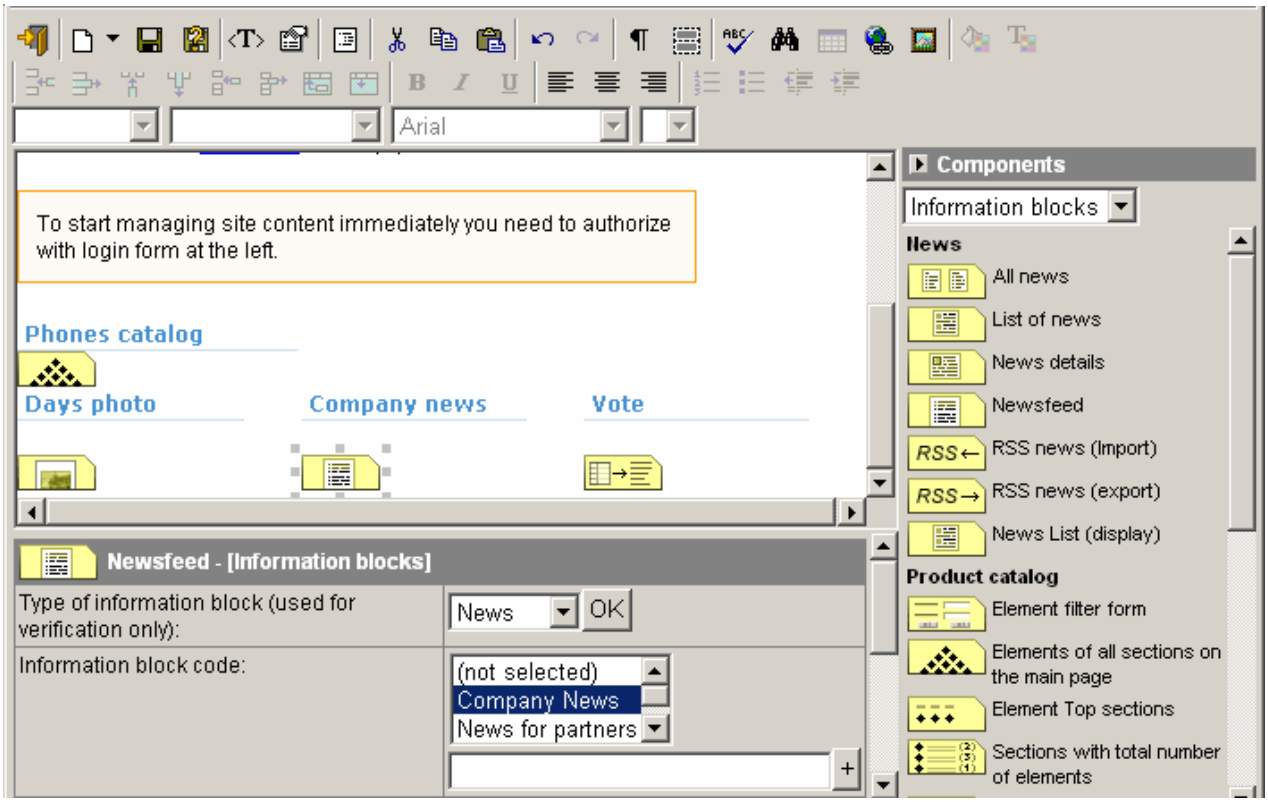

Fig. 4.2. Parameters of component.

You can add an arbitrary PHP code to your page in the similar manner. In this case, you will modify it in the text editor in the bottom of the screen (fig. 4.3).

The *PHP Script* component enables to insert almost any PHP code in a page while leaving the page design undisturbed in the visual editing mode.

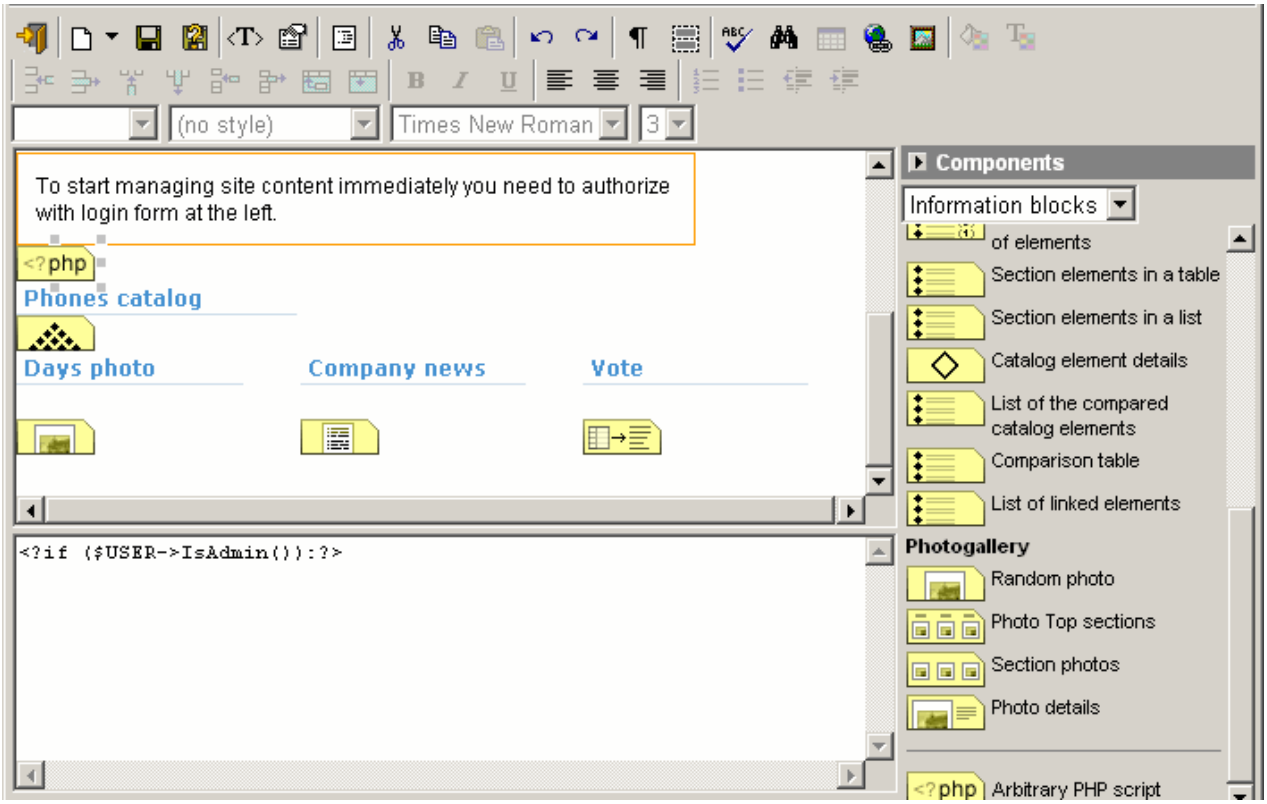

Fig. 4.3 Adding PHP code in the visual mode

The generated code includes the components using the function IncludeFile(). The function parameters include the path to the file with the component code and the component parameters assigned to it in the visual editing mode.

Files pertaining to a component can be stored in the folder of a module to which this component relates.

For example, components used with the *Information Blocks* module are located in the following directories:

/bitrix/modules/iblock/install/templates/iblock/news/

and

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

Alternatively, they can be installed in the default templates folder:

/bitrix/templates/.default/iblock/news/

and

/bitrix/templates/.default/iblock/catalog/ respectively.

Components included in the product installation package have been developed using language resource files containing text messages required by a component. Description files description.php are also included.

For example, descriptions for all components of the *Information Blocks* module can be found in the file

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

If you need a special customization of the component (for example: specific design of a search or authorization form), you can copy the component to directory of the corresponding template.

You can also edit the component code directly in the default template folder. This will affect all templates that do not have a copy of the original component code.

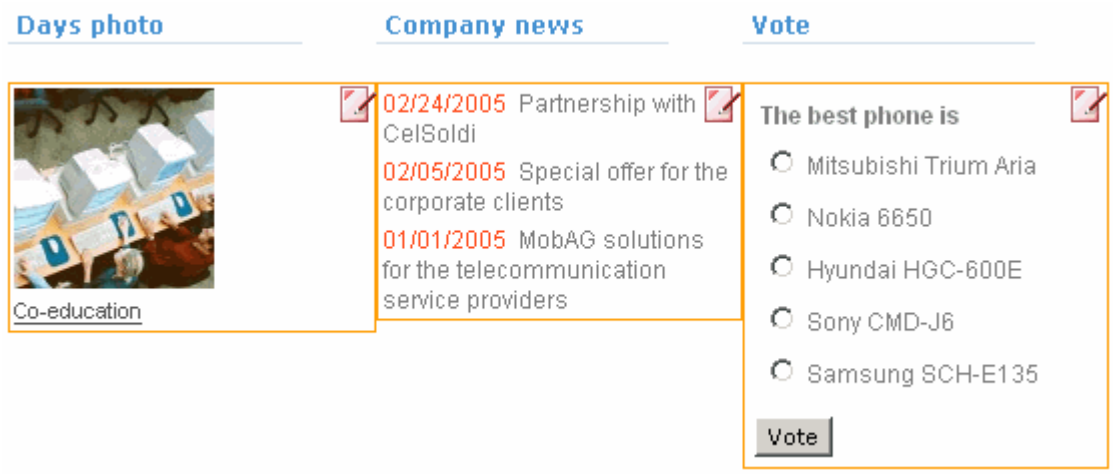

Fig. 4.4 Example of the default template components

Button  switches to editing the component source code directly in the default template directory (/bitrix/templates/.default/).

Button  copies the selected component to the current template folder and opens the editor with the component code.