

Bitrix Site Manager 4.0

Multisite system configuration guide

Version 4.0.0 as of 02.18.2005



Bitrix Site Manager 4.0.....	1
Preface.....	2
Multisite version features	2
Mode 1. Single web server for all sites.....	2
Mode 2. Separate web servers for each site.....	5
Conclusion	8
Further information.....	8

Preface

The multisite version of the Bitrix Site Manager introduces creating of unlimited number of sites using only one copy of the product.

This document is for use by system administrators who are involved in configuring the Bitrix Site Manager and server software so that they function altogether without collisions.

All recommendations concern UNIX or Windows based systems running the Apache web server software.

Multisite version features

The multisite version of the product allows the following two configuration modes.

Mode 1: the software and sites are operated by a single instance of the Apache web server.

Mode 2: each site runs on a separate instance of the Apache web server or a virtual web server.

The product distribution package is shipped configured to operate in the first configuration mode.

We shall consider creating the multiple site system using both configuration modes. The example will explaining how to create a system consisting of the two sites:

#1 www.site1.com – company corporate web site;

#2 www.site2.com – company on-line store.

Mode 1. Single web server for all sites

The first mode presumes the use of a single Apache web server configured to contain the sites in `/home/www/allsites/`

Let us install the Bitrix Site Manager 4.0 in this catalog.

The first mode requires that each site reside in a separate subdirectory of the common directory, for example:

#1 - `/home/www/allsites/s1/`

#2 - `/home/www/allsites/s2/`

You can choose any other names for catalogs *s1* and *s2*, for example *shop* and *company*, or *en* and *de* respectively. You can also make one of the sites to reside in the root directory, whereas the other will be placed in its subdirectory.

You have to keep in mind that it is extremely important to separate one site from the other and ensure that the file structure does not interfere.

The configuration file `httpd.conf` of an Apache web server should contain the following lines, which will reflect the selected:

```
<VirtualHost *:80>
  ServerAdmin admin@site1.com
  DocumentRoot "/home/www/allsites/"
  ServerName www.site1.com
  ErrorLog logs/allsite.log
  CustomLog logs/allsite.log common
</VirtualHost>
```

Please note that the record DocumentRoot "/home/www/allsites/" explicitly defines the directory containing the installed software. The variable DocumentRoot has the same value for all sites.

The line <VirtualHost *:80> means that the web server will reply to requests for any domain name and any IP address. That is, having the DNS configured properly, the web server will support both www.site1.com and www.site2.com.

The next step requires that we properly configure and specify both sites in the installed Bitrix software.

You can customize your sites in the administrative section of any site, for example www.site1.com/bitrix/

Select the menu command *System Settings -> Sites*.

Click the link *Modify* of the site #1 and specify the following parameters:

Name: site1

Domain name: site1.com

Site folder: /s1/

Server URL: www.site1.com

Site name: site1

Path to web server root folder of this site: leave blank

It is always a good idea to specify the domain name without *www*. You can specify as many as needed domain names that users can indicate in their browsers to access your site; each domain name in the list should start from the new line. All domains of third level and lower are considered belonging to this site; the Bitrix Site Manager will open the site #1 being indicated by names both www.site1.com and site1.com.

Please bear in mind that the values you specify in the field *Domain name* are used by the system to deliver the user information to these domains through the **UserMultiSiteTransfer** technology. It might be as well to specify a full list of registered domain names by which your site can be accessed.

Do not include in this list sites not running on the current instance of the Bitrix Site Manager. Invalid domains will slow down the system response to users; user information will not be transferred to sites running on other instances of the software.

The site folder should indicate the existing directory containing the site. The path to web server root folder is not used in this mode; this field should be blank for all sites.

Now set the preferences of the site #2.

Name: site2

Domain name: site2.com

Site folder: /s2/

Server URL: www.site2.com

Site name: site2

Path to web server root folder of this site: leave blank

The system is now almost ready to go. The only thing to do is customize the site selection preferences on the main project page /index.php

When site visitors type www.site1.com or www.site2.com in their browsers, they actually open the page /index.php. In the multiple site world, this file has slightly different function: we have to place the site selection algorithm in it.

A good start for creating the special index page is examining the supplied index file, which can be found here: /bitrix/modules/main/install/public/index.php

This document contains an example of the main page for the plural site configuration:

```
<?
require($_SERVER['DOCUMENT_ROOT']. "/bitrix/modules/main/include/mainpage.php
");

/*CMainPage::RedirectToSite(CMainPage::GetSiteByAcceptLanguage());*/

if($page =
    CmainPage::GetIncludeSitePage(CMainPage::GetSiteByHost()))
    require_once($page);

require($_SERVER['DOCUMENT_ROOT']. "/bitrix/header.php");?>

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

The table below contains the short description on functions used.

CMainPage::GetSiteByHost()	Returns the site code defined by the host.
CMainPage::GetSiteByAcceptLanguage()	Returns the site code defined by the <i>Accept-Language</i> variable in the client browser software.
CMainPage::GetIncludeSitePage(\$site)	Returns the fully qualified path to the root file of a site defined by \$site for further connection by calling require().
CMainPage::RedirectToSite(\$site)	Redirects to the root of a site defined by \$site

See the product help section for more detailed description of functions.

The default algorithm checks the domain name that the visitor typed in the browser to view the site, by calling the function CMainPage::GetSiteByHost(); searches the list of domain names (the field *Domain name*) for this domain name to get the site identifier and includes the real code of the site by calling the function CMainPage::GetIncludeSitePage(\$site).

In our case, this means that a visitor who typed www.site1.com in the browser will get the document /s1/index.php, whereas a visitor who typed www.site2.com in the browser will get /s2/index.php.

Please note that if the function `CMainPage::GetSiteByHost()` is unable to find the domain name, the default site will be used.

The described algorithm allows to avoid using redirects for both visitors and search engine robots as well as makes handling multiple sites more convenient. This algorithm is recommended for use with multiple sites, though other variations are possible.

If your sites represent language mirrors of a single web resource, you can use the code in comments:

```
CMainPage::RedirectToSite(CMainPage::GetSiteByAcceptLanguage());
```

The function `CMainPage::GetSiteByAcceptLanguage()` checks the variable *Accept-Language* in the client browser, compares it to the site identifier (a good idea is using site identifiers named after the language identifier: en, de etc.) and returns the appropriate web site. If the function cannot determine the required language, the default site content is transferred to a visitor. If a browser specifies more than language, the returned site is the one with higher priority in the client settings.

In the example above, the function `CMainPage::RedirectToSite($site)` will perform redirect (HTTP response code 302) and take a visitor to the main page of a site, for example www.site1.com/en/ or www.site2.com/de/

You can combine the described examples. For example, you can use page inclusion instead of redirect:

```
if($page =  
CMainPage::GetIncludeSitePage(CMainPage::GetSiteByAcceptLanguage()))  
    require_once($page);
```

Or you can perform redirect after defining the site by host:

```
CMainPage::RedirectToSite(CMainPage::GetSiteByHost());
```

Choose the desired method of switching and include the corresponding code in the file `/index.php`. Your sites are now ready to operate in the mode 1.

Mode 2. Separate web servers for each site

The second mode requires that you make special adjustments to both web server and the installed software.

We shall consider configuration of the two sites:

#1 www.site1.com – company corporate web site;

#2 www.site2.com – company on-line store.

Each of the sites needs a separate Apache web server to be configured:

#1 - `/home/www/site1/`

#2 - `/home/www/site2/`

The configuration file `httpd.conf` of an Apache web server should contain the following two records:

```

<VirtualHost *:80>
    ServerAdmin admin@site1.com
    DocumentRoot "/home/www/site1/"
    ServerName site1.com
    ServerAlias *.site1.com
    ErrorLog logs/site1.log
    CustomLog logs/site1.log common
</VirtualHost>
<VirtualHost *:80>
    ServerAdmin admin@site2.com
    DocumentRoot "/home/www/site2/"
    ServerName site2.com
    ServerAlias *.site2.com
    ErrorLog logs/site2.log
    CustomLog logs/site2.log common
</VirtualHost>

```

Please note that the records *DocumentRoot* points to different directories. Virtually, each of the sites will have the document root "/", but physically they will reside in different directories.

Lines `<VirtualHost *:80>` denote that the web server will respond to any domain name and IP address. But the variable *ServerAlias* tells the web server to respond to a specific domain name only. That is, requests to www.site1.com are processed by the web server instance working with the directory `/home/www/site1/`, but www.site2.com is managed by the web server that is responsible for the directory `/home/www/site1/`

You can configure your web servers to serve specific IP addresses. Example below shows how to configure an Apache web server to support different IP addresses:

```

<VirtualHost 192.168.0.1:80>
    ServerAdmin admin@site1.com
    DocumentRoot "/home/www/site1/"
    ServerName site1.com
    ErrorLog logs/site1.log
    CustomLog logs/site1.log common
    Options +FollowSymLinks
</VirtualHost>
<VirtualHost 192.168.0.2:80>
    ServerAdmin admin@site2.com
    DocumentRoot "/home/www/site2/"
    ServerName site2.com
    ErrorLog logs/site2.log
    CustomLog logs/site2.log common
    Options +FollowSymLinks
</VirtualHost>

```

Having the DNS server properly configured for different domain names, each virtual web server will run on a separate IP address and respond to a specific IP address or domain name.

Next step is to install the product with this configuration. The console commands illustrate how to install the system on UNIX-based systems.

Install the Bitrix Site Manager to the directory of the first site `/home/www/site1/`;

Create the folder `/home/www/shared/`. It will contain files common to all sites.

```
mkdir /home/www/shared
```

Move the entire directory `/home/www/site1/bitrix/` to `/home/www/shared/bitrix/`

```
mv /home/www/site1/bitrix /home/www/shared/bitrix
```

Move the entire directory `/home/www/site1/upload/` to `/home/www/shared/upload/`

```
mv /home/www/site1/upload /home/www/shared/upload
```

Create the symbolic link to the directory `/bitrix/` for each site:

```
ln -s /home/www/shared/bitrix /home/www/site1/
```

```
In -s /home/www/shared/upload /home/www/site1/
```

```
In -s /home/www/shared/bitrix /home/www/site2/
```

```
In -s /home/www/shared/upload /home/www/site2/
```

Please ensure that the directory `/home/www/shared/` of a web server has write permissions. This is essential for the update system and image uploads;

Copy the public pages of the second site to the directory `/home/www/site2/`

Windows based web servers require third-party tools for the symbolic link creation. For example, *Far Manager* or Sysinternals *Junction*.

Next step is to properly configure sites in the installed Bitrix software.

You can perform adjustments in the administrative section of any of the sites; for example in www.site1.com/bitrix/

Select the menu command *System Settings -> Sites*.

Click the link *Modify* of the site #1 and specify the following parameters:

Name: site1

Domain name: site1.com

Site folder: /

Server URL: www.site1.com

Site name: site1

Path to web server root folder of this site: `/home/www/site1/`

It is always a good idea to specify the domain name without *www*. You can specify as many as needed domain names that users can indicate in their browsers to access your site; each domain name in the list should start from the new line.

Please bear in mind that the values you specify in the field *Domain name* are used by the system to deliver the user information to these domains through the **UserMultiSiteTransfer** technology. It might be as well to specify a full list of registered domain names by which your site can be accessed.

Do not include in this list sites not running on the current instance of the Bitrix Site Manager. Invalid domains will slow down the system response to users; user information will not be transferred to sites running on other instances of the software.

Use the following settings for the second site:

Name: site2

Domain name: site2.com

Site folder: /

Server URL: www.site2.com

Site name: site2

Path to web server root folder of this site: `/home/www/site2/`

Please note that both sites has the same site folder `/`. This is possibly due to the fact that the sites run on separate web servers with different physical files location.

The configuration is now ready to run.

Unlike the first configuration mode, this mode does not require any special site selection algorithm.

Conclusion

The described recommendations allow configuring the Bitrix Site Manager to work in the multiple site mode.

The Bitrix Site Manager does not give preference to any of the versions.

The first mode is generally used if you use the virtual hosting or feel uncertainty at configuring web servers.

The second variant can be recommended if you use a dedicated server or the shared hosting allows several web server instances. This mode allows to exclude catalogs like /s1/ or /s2/ from URL's and start addresses from the root directory “/”.

Further information

Please do not hesitate to contact the Bitrix company to get any additional information on how to configure your web sites.

Web: <http://www.bitrixsoft.com/>

Technical support: <http://www.bitrixsoft.com/support/>

e-Mail: support@bitrixsoft.com