



1С-Битрикс: Управление сайтом

Курс «Администратор. Базовый»

Примеры операций





Введение

Курс предназначен для базовой подготовки пользователей, осуществляющих администрирование сайтов, созданных на «1С-Битрикс: Управление сайтом». Изучение курса позволяет освоить основные методы администрирования системы, а также пополнить знания по темам, изученным в курсе [«Контент-менеджер»](#).

При условии качественного изучения материалов курса, по его окончании специалист должен уметь:

- устанавливать и настраивать систему;
- осуществлять необходимые процедуры по переносу разработанного сайта на хостинг;
- использовать возможности интерфейса по управлению системой;
- управлять доступом к системе, сайтами, пользователями, группами пользователей;
- работать с модулем **Информационные блоки**;
- работать с модулем **Поиск**;
- работать с инструментами системы;
- выполнять работу по конфигурированию веб-системы для оптимальной работы.

Для прохождения курса:

Обязательно:

- наличие базовых навыков компьютерной грамотности и навыков работы с ОС Windows;
- наличие представлений о WWW и организации доступа к веб-серверу.

Рекомендуется иметь:

- базовые навыки установки и администрирования *NIX-систем



Примеры операций

В главе рассматриваются некоторые типовые операции, которые выполняются при администрировании сайтов, созданных на "1С-Битрикс: Управление сайтом".

Резервное копирование по расписанию

В «1С-Битрикс: Управление сайтом» есть функционал резервного копирования, но для создания резервной копии требуется участие администратора. Создание резервных копий по расписанию не предусмотрено так как есть объективные причины, не позволяющие это сделать достаточно эффективно.

Создание архива - длительный и ресурсоёмкий процесс, на один хит это делать никак нельзя. Варианты решения задачи:

- сделать интерфейс, через который можно запускать создание архива через внешний планировщик (cron);
- использовать системные инструменты.

Рассмотрим последний вариант для разных ОС.

Unix

Имеем *unix* подобную систему (*Linux*, *Freebsd* и т.д.) и возможность ставить задачи на cron. Традиционно *unix* предоставляет утилиту *tar* для создания архивов и *bzip* для сжатия, а база данных MySQL - утилиту создания дампа базы *mysqldump*.

Создаем скрипт

Архив сайта по сути представляет дамп базы данных (содержимое БД в виде файла) и все файлы сайта (включая публичную часть и ядро *Bitrix Framework*).

Значит, перейдём в корень сайта и дамп базы создадим командой:

```
mysqldump -hhost -uuser -ppass dbname > dump.sql
```

А архив создадим командой:

```
tar -czf archive.tar.gz
```

Теперь если это оформить в простейший скрипт, получим:

```
#!/bin/sh

##### Config #####

doc_root=/home/denis/www/test
```



```
host=localhost
username=username
password=secret
database=sitemanager_demo_start
charset=cp1251
#####
cd $doc_root &&
mysqldump -h$host -u$username -p$password --default-character-
set=$charset $database > $doc_root/bitrix/backup/archive.sql &&
tar -czf $doc_root/bitrix/backup/archive.tar.gz . &&
echo OK
```

В коде вводятся параметры конфигурации, необходимо указать соответственно свои:

- путь к корневой папке портала (значение DOCUMENT_ROOT в phpinfo),
- имя хоста,
- имя пользователя,
- пароль и имя БД,
- кодировку базы данных. Для портала в UTF-8 надо написать utf8, иначе оставить как есть.

 **Обратите внимание, что пробелы ни перед, ни после равно не ставятся.**

Этот вариант скрипта переходит в корень портала, создаёт дамп базы в кодировке, которую мы указали, затем создаёт в папке bitrix/backup архив портала.

Конструкция && позволяет прервать выполнение в случае ошибки (например, не удалось подключиться к базе).

Это будет работать, но в архив попадут архивы портала, созданные ранее, кеш, да и часть текущего архива тоже, что, прямо скажем, не очень красиво. Поэтому скрипт лучше модифицировать.

Как улучшить

Можно воспользоваться параметром исключения **--exclude** для папки *bitrix/backup*, добавив дамп базы, который лежит в этой папке, отдельным вызовом. Заодно можно упростить настройку скрипта, адаптировав его под *Bitrix Framework*, ведь мы же знаем, что все данные подключения к базе хранятся в *bitrix/php_interface/dbconn.php*. Тогда в качестве настройки можно будет использовать только путь к корневой папке портала.



```
#!/bin/sh

doc_root=$1
name=$2

if [ -z $doc_root ]; then
    echo Usage: $0 /path/to/document/root [backup_name]
    exit
fi

if [ -z $name ]; then
    name=backup
fi

dbconn=$doc_root/bitrix/php_interface/dbconn.php

readcfg() {
    grep $1 $dbconn | sed 's/.*"\(.*\)".*/\1/'
}

host=`readcfg DBHost`
username=`readcfg DBLogin`
password=`readcfg DBPassword`
database=`readcfg DBName`

utf=`grep 'BX_UTF' $dbconn | grep true`

if [ -z "$utf" ]; then
    charset=cp1251
else
    charset=utf8
fi
```



```
fi

backup_dir=$doc_root/bitrix/backup

if [ ! -e $backup_dir ]; then
    mkdir $backup_dir
fi

cd $doc_root &&

mysqldump -h$host -u$username -p$password --default-character-
set=$charset $database > $backup_dir/$name.sql &&

tar -cf $backup_dir/$name.tar $backup_dir/$name.sql &&

rm $backup_dir/$name.sql &&

tar -rf $backup_dir/$name.tar --exclude '*bitrix/tmp/*' --exclude
'*bitrix/updates/*' --exclude '*bitrix/backup/*' --exclude
'*bitrix/*cache/*' . &&

gzip -f -9 $backup_dir/$name.tar &&

echo OK && exit

echo Error
```

Этот вариант скрипта читает параметры **dbconn.php**, подключается к базе данных, делает дамп. Затем делает архив проекта, сжимает его. Дамп удаляет. В случае необходимости создаёт папку *bitrix/backup*.

На вход скрипт принимает два параметра: путь к корню портала и имя архива без расширения (не обязательный, по умолчанию **backup**).

В случае успеха пишет **OK**, иначе **Error**.

Удаление старых архивов

В описанном варианте, если имя архива не меняется - он каждый раз переписывается. Если меняется имя архива, можно легко сделать удаление старых бэкапов. Например, удаляем архивы старше недели:

заменяем строку

```
echo OK && exit
```



на

```
find $backup_dir -ctime +7 -exec rm {} \; && echo OK && exit
```

Автоматическое восстановление

Технически с помощью скрипта можно производить и автоматическое восстановление. Но автоматическое восстановление портала - процедура не безопасная. Что-то пойдёт не так и портал «упадёт». Это следует делать с большой осторожностью.

Хотя само по себе восстановить базу несложно. Замените

```
mysqldump -h$host -u$username -p$password --default-character-set=$charset $database > $backup_dir/$name.sql
```

на

```
mysql -h$host -u$username -p$password --default-character-set=$charset $database < $backup_dir/$name.sql
```

и удалите всё что ниже.

⚠ Важно! *И всё же мы настоятельно не рекомендуем это делать на живом проекте в автоматическом режиме.*

Как пользоваться?

- Сохраняем код скрипта в файле на хостинге, желательно за пределами корня сайта, например backup.sh, делаем исполняемым:

```
chmod +x backup.sh
```

- Проверяем из корня сайта:

```
./backup.sh
```

Должно отобразиться ОК, а в папке bitrix/backup появиться файл backup.tar.gz.

- С целью безопасности лучше задать имя архива, иначе злоумышленник сможет скачать его по прямой ссылке: <http://portal.ru/bitrix/backup/backup.tar.gz/> Например:

```
/home/denis/bin/backup.sh /var/www/test f43tg245g2_backup
```

- Добавляем задачу на cron. Если мы хотим делать резервную копию регулярно в час ночи, пишем:

```
crontab -e
```

и



```
0 1 * * * /home/denis/bin/backup.sh /var/www/test
f43tg245g2_backup
```

- А если надо, чтобы каждый раз создавался новый архив с именем даты, запись может иметь вид:

```
0 1 * * * /home/denis/bin/backup.sh /var/www/test
f43tg245g2_backup_`date +%Y%m%d`
```

Если на хостинге не действуют ограничения на время работы скриптов через cron (или они достаточно щадящие), то этот вариант должен нормально работать в большинстве случаев.

Windows

На взгляд создателя скрипта, использовать *shell* скрипты в этом случае не очень удобно. Рассмотрим аналог скрипта выше, написанный на *PHP*.

Для создания дампа скрипт использует утилиту *mysqldump*, соответственно, она должна лежать в стандартных путях запуска (или укажите в коде скрипта полный путь).

Архив **tar.gz** создаётся на базе класса *Bitrix Framework* (того же, что и в штатной системе резервного копирования). Параметры запуска те же: первый обязательный - это путь к корню сайта, второй - имя архива.

Предположим, что корпоративный портал работает через «*Битрикс: Веб-окружение*», размещенное в папке *E:\WEB*. Архив делается командой:

```
e:\web\apache\php e:\web\www\test\backup.php e:\web\www\test
```

В этом случае код скрипта следующий:

```
<?
$_SERVER['DOCUMENT_ROOT'] = $argv[1];

if (!$_SERVER['DOCUMENT_ROOT'])
    die('Usage: '.basename(__FILE__).' /path/to/document/root.'"");

$_SERVER['DOCUMENT_ROOT'] =
str_replace('\\', '/', $_SERVER['DOCUMENT_ROOT']);

$name = $argv[2] ? $argv[2] : 'backup';
```



```
require($_SERVER['DOCUMENT_ROOT'].'/bitrix/php_interface/dbconn.php');
ini_set('max_execution_time',0);
$charset = defined('BX_UTF') && BX_UTF ? 'utf8' : 'cp1251';
$backup_dir = $_SERVER['DOCUMENT_ROOT'].'/bitrix/backup';
if (!file_exists($backup_dir))
    mkdir($backup_dir);

shell_exec("mysqldump -h$DBHost -u$DBLogin -p$DBPassword --default-
character-set=$charset $DBName > $backup_dir/$name.sql");

require($_SERVER['DOCUMENT_ROOT'].'/bitrix/modules/main/classes/genera
l/tar_gz.php');

function getmicrotime()
{
    list($usec, $sec) = explode(" ", microtime());
    return ((float)$usec + (float)$sec);
}

function Store($path)
{
    global $arch;

    $path = str_replace('\\', '/', $path);

    if
(preg_match('#^' . $_SERVER['DOCUMENT_ROOT'] . '/bitrix/backup#', $path) ||
preg_match('#^' . $_SERVER['DOCUMENT_ROOT'] . '/bitrix/[^/]*cache/#', $path
))

        return;

    $arch->AddFile($path, '', $_SERVER['DOCUMENT_ROOT']);
```



```
if (is_dir($path))
{
    $dir = opendir($path);
    while(false !== $file=readdir($dir))
    {
        if ($file=='.' || $file=='..')
            continue;
        Store($path.'/'.$file);
    }
    closedir($dir);
}

}

$arch = new CArchiver($backup_dir.'/'.$name.'.tar.gz', true);
$arch->_openWrite();
$arch->AddFile("$backup_dir/$name.sql", '', $_SERVER['DOCUMENT_ROOT']);
Store($_SERVER['DOCUMENT_ROOT']);
$arch->_close();
unlink("$backup_dir/$name.sql");
?>
```

Строго говоря, этот же скрипт будет работать и на unix платформе, но он работает медленнее и ограничений на php скрипты обычно действует больше.

Установка продукта на IIS 6

В главе приводятся подробные инструкции по установке PHP в виде ISAPI-фильтра под операционную систему Microsoft Windows с IIS версии 6.0. Для функционирования системы «1С-Битрикс: Управление сайтом» необходима установка PHP версии 4.1.2 или выше.

При использовании дистрибутива PHP в версии 5 рекомендуется использовать IIS версии 7.



Установка PHP

Перед началом установки убедитесь, что PHP поддерживается вашим сервером.

Для установки PHP выполните следующие действия:

- Загрузите дистрибутив PHP версии 4.1.2 или выше, поставляемый в архиве. Дистрибутив доступен на сайте <http://www.php.net/downloads.php>.
- Распакуйте полученный архив. Например, в папку **x:\php**.

⚠ Примечание: В документации использован вид пути **x:**, так как нам не известно, на каком диске установлен Windows.

- Переименуйте файл **php.ini-recommended** в **php.ini**.
- Скопируйте файл **php.ini** в директорию **x:\Windows**.
- Откройте файл **php.ini** для редактирования в тестовом редакторе.
- Найдите строку:

```
extension_dir =
```

и измените ее на:

```
extension_dir = x:\php\extensions
```

- Найдите строку:

```
doc_root =
```

измените ее на:

```
doc_root = "x:\Inetpub\wwwroot"
```

- Найдите строку:

```
cgi.force_redirect = 1
```

измените ее на:

```
cgi.force_redirect = 0
```

- Сохраните внесенные изменения.
- Скопируйте файл **php4ts.dll**, расположенный в каталоге **x:\php**, в директорию **x:\Windows\System32**.
- Перейдите в меню **Start > Settings > Control Panel > Administrative Tools > Internet Information Services (IIS) Manager**.



- Нажмите правой кнопкой мыши на ваш сайт в дереве каталога **Web Sites** (в большинстве случаев он может быть обозначен как **Default Web Site**). Для того чтобы открыть окно настроек свойств сайта, в появившемся контекстном меню выберите пункт **Properties**.
- Откройте вкладку **Home Directory**.
- В разделе **Application settings** для поля со списком **Execute Permissions** выберите значение **Scripts and Executables**.
- Далее, в том же разделе (Application settings) нажмите кнопку **Configuration**.
- Перейдите на вкладку **Mappings**.

Настройка расширений для всего сайта на вкладке **Mapping** позволит использовать эти расширения для всех виртуальных каталогов, расположенных на данном сайте.

- Нажмите кнопку **Add**, чтобы настроить путь к **ISAPI DLL** и указать расширение php-модуля.
- Заполните поля открывшейся формы следующим образом:
 - **Executable**: при помощи кнопки **Browse** найдите исполняемый файл библиотеки **php4isapi.dll** или введите строку с адресом самостоятельно:
x:\php\sapi\php4isapi.dll;
 - **Extension**: введите в поле расширение **.php**;
 - Снимите флажок в поле **Verify that file exists**. Установите флажок только в поле **Script Engine**.
- Нажатием кнопки **OK** сохраните внесенные изменения в каждом окне.
- Нажмите правой кнопкой мыши на **Web Service Extensions** в дереве каталога IIS. В появившемся контекстном меню выберите пункт **Add a New Web service extension**.
- В разделе **Required file** нажмите кнопку **Add** для того, чтобы задать имя нового фильтра (**.php**) и путь к ISAPI-файлу **php4isapi.dll**. Данный файл должен быть расположен в **x:\php\sapi\php4isapi.dll**.
- Поставьте флажок в поле **Set extension status Allowed**.
- Нажмите кнопку **OK**.

Чтобы использовать страницу **index.php** по умолчанию при загрузке сайта сделайте следующее:

- Нажмите правой кнопкой мыши на ваш сайт в дереве каталога **Web Sites**. В появившемся контекстном меню выберите пункт **Properties**.
- Перейдите на вкладку **Documents**.



- Нажмите кнопку **Add**.
- Откроется окно **Add content Page**. В поле данного окна введите имя страницы **index.php**.
- Нажмите кнопку **OK**.
- По умолчанию страница **index.php** будет помещена в конец списка. Используя кнопку **Move Up**, переместите страницу в начало списка.
- Подтвердите сделанные настройки, нажав кнопку **OK**.

Для того чтобы внесенные изменения вступили в силу, необходимо остановить IIS, а затем запустить его снова.

Проверка работы PHP

- Создайте файл с именем **test.php**.
- Поместите в созданный файл строку

```
<? phpinfo(); ?>
```

- Поместите файл в корневой каталог вашего сайта.
- Откройте в браузере указанную страницу **http://localhost/test.php**, чтобы проверить работу PHP.
- В окне браузера должна появиться таблица с настройками PHP.

Ошибки, возникающие в работе

Проблема:

Нарушение целостности баз данных MySQL с типом таблиц **MyISAM**.

Решение:

В случае возникновения ошибок с базой данных MySQL с типом таблиц **MyISAM** рекомендуется воспользоваться встроенным инструментом системы для проверки и восстановления базы данных на странице **Проверка БД** ([Настройки](#) > [Инструменты](#) > [Проверка БД](#)). Использование скрипта проверки и восстановления базы данных позволит оперативно восстановить работу сайта.

Для проверки:

- Перейдите на страницу **Проверка БД** ([Настройки](#) > [Инструменты](#) > [Проверка БД](#)).



- Нажмите на кнопку **Проверить/Восстановить таблицы**. Запустится процесс проверки, результатом которого будет сводная таблица результатов проверки и исправлений.

Таблица результатов проверки:

Название таблицы	Количество записей	Размер таблицы (байт)	Результат проверки	Результат восстановления
b_agent	2	200	[status] OK	
b_captcha	0	0	[status] OK	
b_event	0	0	[status] OK	
b_event_message	2	940	[status] OK	
b_event_message_site	2	14	[status] OK	
b_event_type	2	496	[status] OK	
b_favorite	0	0	[status] OK	
b_file	94	7 476	[status] OK	
b_group	5	424	[status] OK	
b_group_subordinate	0	0	[status] OK	
b_group_task	2	40	[status] OK	
b_iblock	11	1 876	[status] OK	
b_iblock_cache	0	0	[status] OK	

При выполнении процесса проверки и восстановления обратите внимание на следующее:

- Скрипт проверки и восстановления базы данных может быть использован только для MySQL с типом таблиц **MyISAM**;
- Если повреждены таблицы статистики и нет возможности перейти в административный раздел, сбор статистики может быть временно отключен с помощью параметра `?no_keep_statistic_LICENSE-KEY=Y`. В параметре указывается лицензионный ключ сайта.
- Существует возможность использования скрипта проверки и восстановления базы данных без перехода в административный раздел. Для этого при обращении к странице восстановления необходимо указать два параметра: имя (**login**) и пароль (**password**) на доступ к базе данных.

Например:

http://www.mysite.ru/bitrix/admin/repair_db.php?login=DB_Login&password=DB_Password.

По умолчанию значения данных параметров хранятся в файле **/bitrix/php_interface/dbconn.php**. Иногда возникает ситуация, когда сайт перестает отвечать, и посетителям отображается пустая страница. В этом случае рекомендуется открыть этот файл, содержащий параметры соединения с базой данных, и установить значение параметра `$DBDebug = true;`.



В результате будет получен код ошибки, содержащий, как правило, названия поврежденных таблиц базы данных. В случае если целостность таблиц действительно нарушена, следует воспользоваться скриптом восстановления.

Проблема:

Выводится ошибка:

```
DB query error. Please try later.
```

Решение:

В файле `/bitrix/php_interface/dbconn.php` установите значение переменной `$DBDebug=true;` и пришлите текст ошибки в службу **Технической поддержки** компании «1С-Битрикс»: <http://dev.1c-bitrix.ru/support/>.

Проблема:

На экран выводится ошибка:

```
MySQL Query Error: LOCK TABLE b_agent WRITE[Access denied for user: 'user @%' to database 'dab_test']
```

Решение:

Начиная с 4-ой версии MySQL права на **LOCK** назначаются отдельно. Необходимо дополнительно назначить права на **LOCK TABLES** для данного пользователя базы данных.