Contents

Setup:Installation Guide/Advanced

- Backup
- Configuration Folder settings.d
- ExtendedSearch Configuration
- Folder structure under Windows
- Maintenance scripts
- batchStabilize
- dumpPDF
- executeTransfer
- export
- Performance Optimization
- ManualRecache
- MySQL
- PHP
- VisualEditor Configuration
- Windows Folder Structure

Setup:Installation Guide/Advanced/Backup

Contents

Manual backup

Backup of the database

To back up your database, use the native program `mysqldump` of MySQL/MariaDB: Open a console, switch to the directory `<installpath-bluespice>` and backup your database with the following command:

```
mysqldump -u <username> -p --lock-tables <database> > database.sql
```

For `<username>` , use the administrative database user, and for `<database>` , use the database of your MediaWiki/BlueSpice installation.

> ! On Windows, make sure that your environment variables are set correctly, so that you can access the command `mysqldump` .

Backup of the file system

Back up the complete directory `<installpath-bluespice>`, which now also includes the database image, to a different location on your server.

For security reasons, immediately delete the file `<installpath-bluespice>/database.sql`, since it can be accessed from the browser.

Using the `mediawiki-adm` tool

To eas backup and restore one can also use the `mediawiki-adm` tool. It can be downloaded here: https://github.com/hallowelt/misc-mediawiki-adm

Example usage:

```
mediawiki-adm wiki-backup \
    --mediawiki-root <installpath-bluespice> \
    --dest /mnt/backup/
```

The tool will leave out all unnecessary data (like temporary files and cache database tables)and produce much smaller backup files that the manual method described above. It also allows to create versioned backups.

Setup:Installation Guide/Advanced/Configuration Folder settings.d

> **Hinweis:** This document provides background information for advanced users. We recommend not to make any changes on BlueSpice, if you are not familiar with the following information.

- For editing the configuration files with a text editor, the files must be saved in UTF-8 coding without BOM (Byte Order Mark).
- The placeholder `<installpath-bluespice>` stands for the path to your BlueSpice installation, e.g `C:\inetpub\wwwroot\bluespice` (Windows) or `/var/www/bluespice` (Linux).

Explanation For the Folder settings.d

Starting with BlueSpice version 2.27.1, all default settings as well as the modules of BlueSpice are outsourced. All configurations can now be found in separate files in the /settings.d folder.

The files in this folder integrate the standard MediaWiki extensions required for BlueSpice, perform recommended default configurations, and also integrate all BlueSpice-related extensions (free or pro).

The files in the /settings.d folder are automatically integrated in alphabetical order and are therefore prefixed with numbers.

If you want to add additional configurations yourself, you can save your own files here. For this, it is recommended to use the prefix 001-.

Additional configurations, which are not packed per default but can be used, can be found at auf github.com.

Update-safe changing of default configuration files

If you want to change the files delivered by default in the settings.d folder, we strongly recommend that you choose the following method so as not to lose this configuration after an update:

- Create a copy of the respective configuration file with the extension `.local.php` - e.g. *020-VisualEditor.local.php*
- Only edit this file

Files with this file extension are generally preferred in the loading process.

ExtendedSearch configuration

If the ElasticSearch service is up and running, you can enable BlueSpice Extended Search.

- Go to your BlueSpice install path. For example:

```
cd /var/www/bluespice
```

- Go to the folder *settings.d*:

```
cd settings.d
```

- Create a file in this folder and name it *020-BlueSpiceExtendedSearch.local.php*.
- Put the following content in the newly created file.

```
<?php
wfLoadExtension( 'BlueSpiceExtendedSearch' );
$GLOBALS['wgSearchType'] = 'BS\\ExtendedSearch\\MediaWiki\\Backend\\BlueSpiceSearch';
```

- Go to your BlueSpice install path again. For example:

```
cd /var/www/bluespice
```

- Run the following scripts to build the search index.

```
php extensions/BlueSpiceExtendedSearch/maintenance/initBackends.php --quick
php extensions/BlueSpiceExtendedSearch/maintenance/rebuildIndex.php --quick
php maintenance/runJobs.php
```

Done! The extended search functionality is now active.

Folder structure under Windows

Contents

Preface

Microsoft Windows distributes program installations and configurations, the Webroot under IIS etc. to various folders in the file system. Since BlueSpice is a web application based on PHP, it is not always easy to use the Windows default system paths in the application.

For this reason, we recommend creating your own folder structure for BlueSpice, all related data and additional program installations.

In the following we describe a proven folder structure. This is recommended by us.

Folder structure

In the following, "\bluespice\" is defined as the base directory for the installation in the root directory of the drive (e.g. C:\bluespice or D:\bluespice).

The folder structure is based on Unix operating systems:

```
bluespice/
│
├─ backup/           # for backups
│
├─ bin/              # for additional programs and own scripts
│
├─ etc/              # for configuration files
│
├─ opt/
│  └─ bluespice/
│     ├─ cache/         # BlueSpice cache
│     ├─ images/        # File uploads
│     └─ widgets/       # Rendered widgets
│
├─ tmp/              # Temporary files
│
└─ var/
   ├─ log/           # LOG files of the web server
   └─ www/           # Actual BlueSpice application
```

Folder names for program installations

When installing programs in \bluespice\, please make sure to install these programs in subfolders that do not contain special characters and/or spaces.

Setup:Installation Guide/Advanced/Maintenance scripts

- createAndPromote.php:
  creates a new user or modifies an existing user. For the corresponding user, the script can then grant additional rights (e.g. sysop, bureaucrat or bot rights) and change the user's password.

- dumpBackup.php:
creates an XML dump which contains the content of the wiki (wiki pages with all their revisions), without the site-related data.

- dumpPDF.php:
exports pages from all content namespaces to PDF.

- importDump.php:
imports XML dump files produced from*Special:Export* or *dumpBackup.php* and saves them into the current wiki.

- importImages.php:
uploads images to MediaWiki from the same computer where the wiki is configured. It does a remote upload if configuration parameter$wgForeignFileRepos is configured correctly.

- executeTransfer.php:
transfers files from one wiki to another. Dependent on the extensionContentTransfer

- export.php:
exports a defined list of pages to PDF.

- contentStabilization v4.3+ :
changes the page status of pages.

Setup:Installation Guide/Advanced/Maintenance scripts /batchStabilize

Contents

Mass approval using a script

The script `ContentStabilization/maintenance/batchStabilize.php` can be used to approve all drafts in the wiki starting with BlueSpice 4.3+. This affects both first drafts, drafts of pages with already approved versions and drafts for embedded resources.

This script allows the approval of all pages in a namespace or, alternatively, a list of pages defined in a text file.

Options

Option	Description
`--namespace`	Id of namespace to flag entirely
`--user`	Required. The user name of an existing user that is shown as page approver
`--comment`	Adds a stabilization comment to the revision history of the page
`--verbose`	Runs the script in verbose mode to show exceptions

Examples

Whole namespace

```
php extensions/ContentStabilization/maintenance/batchStabilize.php --user WikiSysop --
namespace 3000
```

List of page names

Starting with BlueSpice 4.3.2, a file with a newline-separate list of pages (as an arg or as stdin) can be used to define the pages that need to be approved:

```
php extensions/ContentStabilization/maintenance/batchStabilize.php --verbose /tmp/page-
titles-to-review.txt
```

```
php extensions/ContentStabilization/maintenance/batchStabilize.php --verbose < /tmp
/page-titles-to-review.txt
```

In case the page titles contain special characters (like ä, ö, ü and so on), make sure to save the file in UTF-8 encoding. Otherwise the script may not be able to find the according page names inside the database!

Setup:Installation Guide/Advanced/Maintenance scripts /dumpPDF

→ all maintenance scripts

Extension

BlueSpice Extension: BlueSpiceUEModuleBookPDF

Usage

This script allows to export the wiki pages **from all content namespaces** to PDF.

To run the script, execute:

```
php {MW_ROOT}/extensions/BlueSpiceUEModuleBookPDF/maintenance/dumpPDF.php --file:
<localpath> --limit:200 --mail:admin@mymail.com
```

Options

- `--file:` Defines the output path and the base filename.
  This name will be appended with the namespace number and, if `--limit` is set, with the split group.
  Expamle: `--file=/tmp/test.pdf` will result in `/tmp/test_0.pdf` or `/tmp/test_0_ABC.pdf`,
- `--limit:` Sets a limit for wikipages in a namespace included in one pdf. If the limit is exceed, the namspace will be split into pdfs depending on the first character of the page name. If parameter `--limit` is not set, all pages of a namespace will be exported to one pdf file.
- `--verbose`: Runs the script in verbose mode.
- `--mail:` E-mail address for a notification email.

You can see the list of possible options with descriptions by using the following command:

```
php {MW_ROOT}/extensions/BlueSpiceUEModuleBookPDF/maintenance/dumpPDF.php --help
```

Setup:Installation Guide/Advanced/Maintenance scripts /executeTransfer

all maintenance scripts

Contents

Reference

BlueSpice Extension: ContentTransfer

Script location: `extensions/ContentTransfer/maintenance/executeTransfer.php`

Usage

There are three ways to specify pages which pages to transfer:

- **`--category`** : This option lets you transfer based on category:

```
php extensions/ContentTransfer/maintenance/executeTransfer.php --category=GlobalPage
 --targets=Target1,Target2,Target3
```

- **`--namespace`** : This option lets you transfer based on namespace ID:

```
php extensions/ContentTransfer/maintenance/executeTransfer.php --namespace=10 --targ
ets=Target1,Target2,Target3
```

- **`--pages`** This option lets you explicitly specify which pages to transfer:

```
php extensions/ContentTransfer/maintenance/executeTransfer.php --pages=Page1,Demo:Pa
ge2,Page3 --targets=Target1,Target2,Target3
```

Configuration

The maintenance script can consume a JSON file with the necessary configuration. A sample JSON file is located at this path: `extensions/ContentTransfer/docs/transfer-config.json.example`. This file can be used to create a JSON configuration file somewhere, so that it is not necessary to type all options in the CLI each time.

For that, `--json-config` can be used. There can be a specified path to the file with the JSON configuration.

If you need to create your own JSON configuration file, copy the sample JSON file somewhere and edit it to set the necessary configuration.

For example, you have the file `/home/some_user/config/transfer-config.json`. Then it can be used like that:

```
php extensions/ContentTransfer/maintenance/executeTransfer.php --json-config=/home
/some_user/config/transfer-config.json
```

You can specify either an absolute file path from the root or a relative path from the present working directory.

Note!

It is also possible to use a JSON file and additionally pass options from the CLI. In that case, the options from CLI will have greater priority and **will override** the same options from JSON file.

Example:

```
php extensions/ContentTransfer/maintenance/executeTransfer.php --json-config=/home
/some_user/config/transfer-config.json --pages=Page1,Demo:Page2,Page3 --
targets=Target1,Target2,Target3
```

Here, the script will read the configuration from the specified JSON file , but the **“pages”** and **“targets”** options **will be overridden** from the CLI (if they are presented in JSON). The same is true for the other options.

Options

- `--user` : The context user of the "sending" wiki who will be used to transfer wiki pages. This is just about recording the transfer in the DB. Default: **“MediaWiki default”**
- `--only-modified` : If the page should be transferred only if it was modified since the last transfer. **It's just a flag, it can be passed or not.**
- `--modified-since` :Transfer the page only if it was modified since the specified date. The date must be specified in the format **"DD.MM.YYYY"**. Default: **“”**
- `--include-related` :If all related wiki pages should also be transferred (templates, files, links, and so on, which are used on the page to be transferred. **It's just a flag, it can be passed or not.**
- `--force` :Transfer the page even if it is protected on the receiving wiki. **It's just a flag, it can be passed or not.**

Also there is a `--dry` option (it's a flag too), which may be useful if a user just wants to take a look at transferring titles, but without actual changes in “receiving” wikis.

You can see a complete list of possible options with descriptions by the following command:

```
php extensions/ContentTransfer/maintenance/executeTransfer.php --help
```

If there is a need to set a specific user for creating pages in the target wiki , it can be done that way:

```
php extensions/ContentTransfer/maintenance/executeTransfer.php ... --
targets=Target1=User1,Target2=User2,Target3...
```

Setup:Installation Guide/Advanced/Maintenance scripts/export

Contents

Running the script

The script to export wiki pages to PDF is executed from:

```
php {MW_ROOT}/extensions/BlueSpiceUniversalExport/maintenance/export.php --
specification-file=spec.json
```

The specification file `spec.json` can have the following content:

```
{
    "module": "pdf",
    "attachments": 1,
    "recursive": 1,
    "title": "MyPDF",
    "target": "localfilesystem",
    "target-file-name": "All.pdf",
    "target-file-path": "/path/to/target-file"
}
```

Key	Value	Description
module	pdf	Output type
attachments	1 0	Attachments will be included in the export
recursive	1 0	Linked pages (one level deep) will be included.
title	string	Title of the wiki page that contains the list of pages to be exported. **The pages need to be links.**
target	localfilesystem	The file will be exported to a local file system.
target-file-name		Name of the exported PDF file
target-file-path		Target file location

Creating a page list for the export

The export list is defined in the wiki page that is shown as the `title` in the json-file. (In the example above, this is the page `MyPDF` .)

The page `MyPDF` now needs to list all pages that will be exported to PDF.

This list can be created with an SMW inline query (BlueSpice free) or with a DPL3 list (BlueSpice free). Example of an SMW query to create an export for all pages in the categories `Process` and `Work instruction`:

You would add the following to your `MyPDF` page in the wiki:

```
==All processes==
{{#ask:[[Process:+]]|format=ul|limit=99999}}

==All work instructions==
{{#ask:[[Work instruction:+]]|format=ul|limit=99999}}
```

The same list can be created with the following DPL3 syntax:

```
==All processes==
{{#dpl:category=Process}}

==All work instructions==
{{#dpl:category=Work instruction}}
```

Setting up a CronJob

To automate the export, the following CronJob is available:

```
php {MW_ROOT}/extensions/BlueSpiceUniversalExport/maintenance/export.php --
specification-file={MW_ROOT}/extensions/BlueSpiceFoundation/data/spec.json
```

Setup:Installation Guide/Advanced/Mathoid

Redirect to:

- Setup:Installation Guide/Webservices/Mathoid

Setup:Installation Guide/Advanced/Performance Optimization

- Installation Guide/Advanced/Performance Optimization/ManualRecache
- Installation Guide/Advanced/Performance Optimization/MySQL
- Installation Guide/Advanced/Performance Optimization/PHP

Performance Optimization: ManualRecache of LanguageCache

To prevent Rebuilding LocalizationCache with every page call, apply the following configuration:

> **Important!** When following these instructions, make sure that the $wgCacheDirectory variable is set in advance. This is already the case in the BlueSpice standard delivery in the file settings.d/005-Directories.php.

> **Note:** This performance optimization achieves the best performance by enabling and configuring opcache in PHP.

In the `settings.d` folder of your codebase, create the file 006-ManualRecache.php with the following contents:

```
<?php
$wgLocalisationCacheConf = [
'class' => LocalisationCache::class,
'store' => 'array',
'storeClass' => false,
'storeDirectory' => $wgCacheDirectory,
'manualRecache' => true,
];
```

Then delete all existing files in the `cache` folder of your codebase. Then manually re-create the LocalizationCache by entering the following on your console:

```
php /pfad/zur/installation/maintenance/rebuildLocalisationCache.php --force
```

Finally, make sure that this is done regularly in the background via cronjob ("Task Scheduler" under Windows). We recommend this twice a day, for example at 6am and 6pm.

Performance Optimization: MySQL (MariaDB)

Match MySQL or MariaDB directly to your memory and CPU cores. To do this, apply the following configuration in your my.ini:

```
[mysqld]

; with 16GB RAM
innodb_buffer_pool_size=4096M
tmp-table-size=1024M
max-heap-table-size=1024M
query_cache_size=1024M

; with 12 CPU cores
innodb-buffer-pool-instances=12
max_connections=12000

; Only use "127.0.0.1", not "localhost" when accessing MySQL
; Remember to also make these changes in the $wgDBserver variable in LocalSettings.php.
skip-name-resolve
```

Performance Optimization: PHP

Contents

opcache

Enable the Zend extension opcache in PHP. In the php.ini, we recommend the following configuration:

```
opcache.enable=1
opcache.memory_consumption=512
opcache.max_accelerated_files=100000
opcache.validate_timestamps=1
opcache.revalidate_freq=2
opcache.optimization_level=0x7FFF9FFF
```

Up to BlueSpice 4.2.x only -v4.2.x : In addition, you should include the BlueSpice configuration files (`extensions/BlueSpiceFoundation/config/*)` in the Opcache blacklist (`opcache.blacklist_filename`). You can find more information in the official PHP documentationfor configuration.php.

This config-file is no longer available from version 4.3.

zlib

Activate the extension zlib. In the php.ini we recommend the following configuration:

```
zlib.output_compression = On
zlib.output_compression_level = 9
```

You can find more information in the official PHP documentation for zlib.

VisualEditor configuration

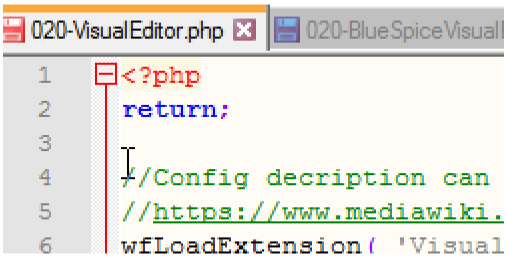
Contents

Server settings

Before you can use VisualEditor in BlueSpice, you have to activate it in your installation:

In the BlueSpice installation directory, locate the following files inside the folder „var\www\settings.d“:

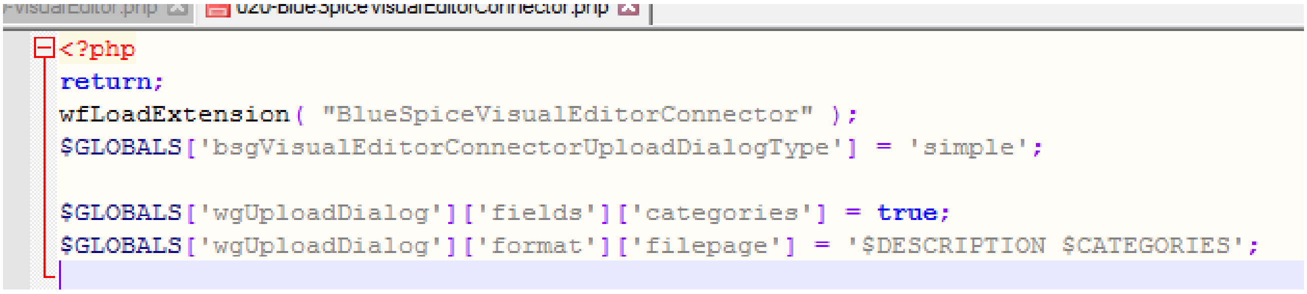
020-VisualEditor.php:



```
<?php
return;

//Config decription can
//https://www.mediawiki.
wfLoadExtension( 'Visual
```

020-BlueSpiceVisualEditorConnector.php:



```
<?php
return;
wfLoadExtension( "BlueSpiceVisualEditorConnector" );
$GLOBALS['bsgVisualEditorConnectorUploadDialogType'] = 'simple';

$GLOBALS['wgUploadDialog']['fields']['categories'] = true;
$GLOBALS['wgUploadDialog']['format']['filepage'] = '$DESCRIPTION $CATEGORIES';

```

1. Remove the line containing the „return;“ command in both files and save them.
2. Restart your web server (recommended).

Checking the Configuration manager

In case the VisualEditor is still not appearing, check if VisualEditor is shown as activated in the configuration manager of your wiki:

1. Go to *Global actions > Config manager*
2. Make sure the checkbox "Enable VisualEditor for BlueSpice extensions" is activated:

File:ve-config.png
By default, VisualEditor is only activated in the main (Pages) and the User namespaces.

Activating VisualEditor for a namespace

1. Go to Global actions > Namespace manager In the column "VisualEditor", you can see which namespaces have VisualEditor activated. They have a green check mark.
2. Click the wrench icon in the "Actions" column. The wrench appears, when you hover over the row of the namespace:

File:ve-config-namespace.png

1. Activate the checkbox for VisualEditor in the dialog window and click "Done":

File:ve-config-namespace-edit.png

1. Refresh the Namespace manager page (F5) to confirm the change.

VisualEditor should now be working correctly in all namespaces that actively use it.

Customizing the link to the help pages

By default, the help link for VisualEditor links to the help page for VisualEditor on the BlueSpice helpdesk.

File:Setup:installation-ve-helplink.png

VisualEditor help link

The link can be changed to go to a custom URL:

To configure the url, add:

```
`$GLOBALS['bsgVisualEditorConnectorHelpUrl'] = 'https://your_url';`
```

to 'LocalSettings.php' or the appropriate config file.

Windows Folder Structure

Contents

Introduction

Microsoft Windows distributes program installations and configurations, the Webroot under IIS etc. to various folders in the file system. Since BlueSpice is a web application based on PHP, it is not always easy to use the Windows default system paths in the application.
For this reason, we recommend creating a separate folder structure for BlueSpice, all related data and additional program installations.
In the following we describe a proven folder structure that is also recommended by us.

Folder structure

In the following, "\bluespice\" is specified as the base directory for the installation in the root directory of the drive (e.g. C:\bluespice or D:\bluespice).

The folder structure based on Unix operating systems is:

```
bluespice/
│
├─ backup/           # für Backups
│
├─ bin/              # für Zusätzliche Programme sowie eigene Scripte
│
├─ etc/              # für Konfigurationsdateien
│
├─ opt/
│  └─ bluespice/
│     ├─ cache/         # BlueSpice-eigener Cache
│     ├─ images/        # Datei-Uploads
│     └─ widgets/       # Gerenderte Widgets
│
├─ tmp/              # Temporäre Dateien
│
└─ var/
   ├─ log/           # LOG-Dateien des Webservers
   └─ www/           # Die eigentliche BlueSpice-Applikation
```

Folder names for program installations

When installing programs in \bluespice\bin, please always ensure that these programs are installed in subfolders that do not contain any special characters and/or spaces.