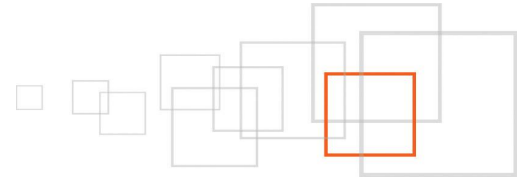


# **eZ Find Demystified: Installing and configuring a multi-core Solr/eZ Find 2.6 instance with eZ Tika**

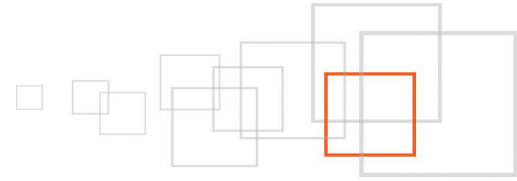
---

**By Daniel Arrrup-Øien**  
**Open Concept SA – [www.openconcept.no](http://www.openconcept.no)**



## Table of Contents

1 Goal description.....	3
2 Introduction.....	3
3 Pre-requisites and target population.....	3
4 Conventions.....	4
5 Step 1: Install Java runtime.....	4
6 Step 2: Install eZ Find and eZ Tika.....	4
Installing eZ Find.....	4
Update the databases.....	5
Installing eZ Tika.....	5
Setting file permissions.....	5
7 Step 3: Prepare for multi-core and future upgrades.....	5
8 Step 4: Configure Solr and the cores.....	6
9 Step 5: Adjust the schema.xml files.....	7
10 Step 6: Configure eZ Find.....	7
11 Step 7: Configure eZ Tika.....	8
12 Step 8: Activate the extensions.....	8
13 Step 9: Start up Solr.....	9
14 Step 10: Index your sites.....	9
15 Step 11: Edit your search template and start searching.....	10
16 Step 12: Create a startup script.....	11
17 Adding more cores.....	15
18 Upgrading.....	15
19 Conclusion.....	16
20 Resources.....	16
21 About the author: Daniel Arnrup-Øien.....	16
22 License choice.....	16



## 1 Goal description

After completing this tutorial, you should be able to set up a single, multi-core eZ Find instance with eZ Tika. This will allow you to index multiple eZ Publish sites along with virtually any file type, as well as take advantage of eZ Find's extended searching and sorting functionality.

The multi-core setup variant is useful even if you just have a single eZ Publish site – it doesn't require any extra resources, and you have the ability to easily add more sites to the search server when the need arises.

## 2 Introduction

eZ Find is a search extension that integrates the content management features of eZ Publish with the powerful, Lucene-based Apache Solr search server. Once installed and configured, it provides faster and more extensive search facilities as well as functionality to overcome some limitations in eZ Publish, such as selecting and sorting on an attribute common to more than one content class.

The eZ Tika extension is a binary file plugin and wrapper for the Apache Tika toolkit, which detects and extracts metadata and structured text content from various document types such as PDF, Word, Excel, ODF, and many more. Used together with eZ Find, virtually all content in an eZ Publish system can be indexed, categorised and searched.

The result is a search server that features enterprise-grade speed and features – the open source way!

It can be tricky to install and configure eZ Find properly, since several technologies are involved and there are some pitfalls to avoid. This tutorial aims to provide a step-by-step guide to the process, as well as give some glimpses into the workings of Solr and how eZ Find integrates it with eZ Publish.

## 3 Pre-requisites and target population

You should be familiar with the Linux command line and package administration, and have a good working knowledge of setting up and using eZ Publish version 4.x (or Community Project releases from 2011 and onward).

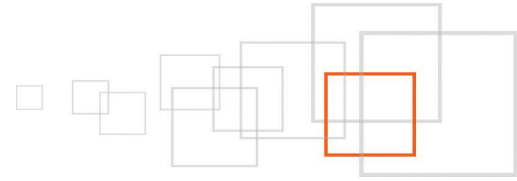
Technical requirements:

- A running Linux server, preferably Debian 6, **with command line and root or sudo access**. The system should be configured with the prerequisites for eZ Publish (PHP, MySQL, web server etc.).
- Two or more separate eZ Publish instances containing some content to index. (You can get by with just one eZ Publish instance if you skip the parts related to adding a second Solr core.)

Reference platform for this tutorial:

- Debian 6 ("Squeeze") Virtual Machine with 4GB RAM / 2 x 1.5 Ghz CPU / 200GB Hard Drive
- eZ Publish Community Project 2012.3
- MySQL 5.1.61
- PHP 5.3.10

Although this tutorial uses Debian 6 as a reference platform, it should be applicable to other Linux distributions



with only minor adaptations. **All commands are assumed to be issued as root unless otherwise noted.**

## 4 Conventions

Command-line commands are surrounded by black borders and set in italic type. Bold italic indicates a variable string that may need to be adjusted to your local environment.

Code sections are surrounded by black borders and use regular text (not italic). Bold indicates a variable string that may need to be adjusted to your local environment.

## 5 Step 1: Install Java runtime

Solr and eZ Find require the Java Runtime Environment (JRE) version 1.6. It comes in a few different packages, but since we don't need a GUI we might as well get the lean and mean "headless" OpenJDK version. Run these commands:

```
apt-get update
apt-get install openjdk-6-jre-headless
```

## 6 Step 2: Install eZ Find and eZ Tika

**IMPORTANT NOTE:** from this point on, the tutorial assumes you have two eZ Publish instances running in the following directories:

```
/var/www/ezpublish/ezfindexample1/
```

```
/var/www/ezpublish/ezfindexample2/
```

You should replace these paths with those in your own environment, as necessary. They are highlighted in **bold text** in the command/code boxes throughout this tutorial.

### Installing eZ Find

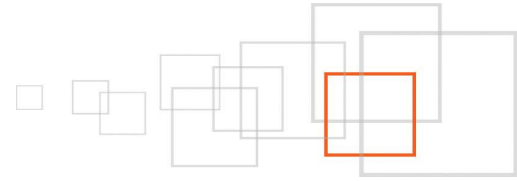
eZ Find is currently bundled with Community Project (CP) releases, so you can download the 2012.3 CP release of eZ Publish for this tutorial. See <http://share.ez.no/downloads/downloads>.

If you're setting up an eZ Publish instance for the purpose of this tutorial, just install eZ Publish according to the official documentation. **Choose the "Plain Site" install option.** If you don't you can still do the tutorial but you'll have more files to wade through and you'll have to substitute more directories.

If you are installing or upgrading eZ Find in an existing eZ Publish instance, unpack the 2012.3 release and copy the `ezfind/` directory from the `extension/` directory to the `extension/` directory of your existing eZ Publish instance. Repeat for each instance on which you want eZ Find to run.

Example:

- `/var/www/ezpublish/ezfindexample1/extension/ezfind`
- `/var/www/ezpublish/ezfindexample2/extension/ezfind`



## Update the databases

Now update the databases using the following commands (replace **paths** and **user/database** info as necessary):

```
cd /var/www/ezpublish/ezfindexample1/  
mysql -u user -p database_name < extension/ezfind/sql/mysql/mysql.sql  
  
cd /var/www/ezpublish/ezfindexample2/  
mysql -u user -p database_name < extension/ezfind/sql/mysql/mysql.sql
```

## Installing eZ Tika

eZ Tika can be downloaded from <http://projects.ez.no/eztika/downloads>. Installation instructions are included with the download, but at this stage it's as simple as extracting the downloaded archive to the extension folder in your eZ Publish instance(s).

Example:

```
/var/www/ezpublish/ezfindexample1/extension/eztika
```

```
/var/www/ezpublish/ezfindexample2/extension/eztika
```

## Setting file permissions

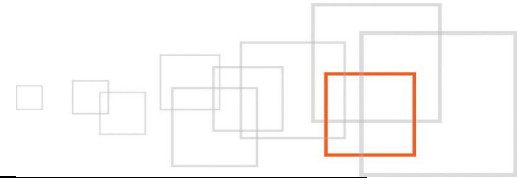
Before we forget, we set the permissions for our extensions. Replace your **paths** and **web server user name** as necessary.

```
cd /var/www/ezpublish/ezfindexample1/  
chmod -R og+rwx design extension settings var  
chown -R www-data:www-data design extension settings var  
cd /var/www/ezpublish/ezfindexample2/  
chmod -R og+rwx design extension settings var  
chown -R www-data:www-data design extension settings var
```

## 7 Step 3: Prepare for multi-core and future upgrades

Now we're going to move the payload of our two extensions so that they are independent of our eZ Publish instances, to enable Solr's multi-core mode and to facilitate future upgrades. Issue these commands, replacing **paths** as necessary:

```
cd /  
mkdir -p srv/{tika,solr/cores}  
cp -rp /var/www/ezpublish/ezfindexample1/extension/eztika/bin /srv/tika/  
cp -rp /var/www/ezpublish/ezfindexample1/extension/ezfind/java /srv/solr/  
ln -s /srv/solr/java/solr/lib /srv/solr/cores  
mkdir -p /srv/solr/cores/ezfindexample1/data
```



```
mkdir -p /srv/solr/cores/ezfindexample2/data
cp -rp /var/www/ezpublish/ezfindexample1/extension/ezfind/java/solr/conf
/srv/solr/cores/ezfindexample1/
cp -rp /var/www/ezpublish/ezfindexample2/extension/ezfind/java/solr/conf
/srv/solr/cores/ezfindexample2/
```

Essentially, we have copied the eZ Find and eZ Tika binaries to their own directories, and created separate directories to hold the configuration and data directories for each Solr core. When you are done, you should have a directory structure like the following:

```
srv/
|-- solr
|   |-- cores
|       |-- ezfindexample1
|           |-- conf
|           |-- `-- data
|       |-- `-- ezfindexample2
|           |-- conf
|           |-- `-- data
|       |-- `-- java
|-- `-- tika
|-- `-- bin
```

Create a solr.xml file for global configuration of Solr cores:

```
touch /srv/solr/cores/solr.xml
```

Now set permissions:

```
chmod -R 755 /srv
```

## 8 Step 4: Configure Solr and the cores

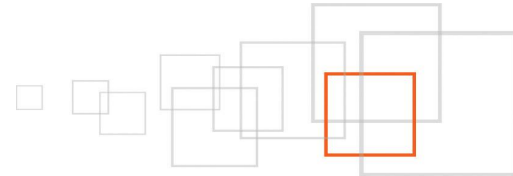
Edit /srv/solr/cores/solr.xml to look like the following, substituting **instanceDir** and **name** as necessary:

```
<?xml version="1.0" encoding="UTF-8" ?>
<solr persistent="true" sharedLib="lib">
<cores adminPath="/admin/cores">
    <core name="ezfindexample1" instanceDir="ezfindexample1" />
    <core name="ezfindexample2" instanceDir="ezfindexample2" />
</cores>
</solr>
```

Edit /srv/solr/cores/**ezfindexample1**/conf/solrconfig.xml and look for the following element:

```
<dataDir>${solr.data.dir:./solr/data}</dataDir>
```

Replace this element with:



```
<dataDir>/srv/solr/cores/ezfindexample1/data</dataDir>
```

Repeat for the second core - edit `/srv/solr/cores/ezfindexample2/conf/solrconfig.xml` and look for the following element:

```
<dataDir>${solr.data.dir:./solr/data}</dataDir>
```

Replace this element with:

```
<dataDir>/srv/solr/cores/ezfindexample2/data</dataDir>
```

## 9 Step 5: Adjust the schema.xml files

eZ Find version 2.6 uses Solr version 3.1, which has an internal schema that is different from previous versions. At the time of this writing, the schema.xml file distributed with eZ Find can cause problems when indexing objects with multiple locations and during wildcard searches on non-ASCII content (non-English characters).

To address these problems, we need to edit the schema.xml file for each core. The following files must be modified in our example:

```
/srv/solr/cores/ezfindexample1/conf/schema.xml
```

```
/srv/solr/cores/ezfindexample2/conf/schema.xml
```

First fix the multiple location problem. Find the section beginning with this comment:

```
<!-- Location information ( may be multiple ) -->
```

Add the following element to the field definitions in this section – **but ONLY if it doesn't already exist**:

```
<field name="meta_priority_si" type="sint" indexed="true" stored="true" multiValued="true"/> <!-- Priority -->
```

Now to fix the wildcard search problem (you may skip this step if you only have English-language content). Find the following element, which occurs **twice** in the file:

```
<filter class="solr.ASCIIFoldingFilterFactory"/>
```

Comment out **both** occurrences of this element (they appear around line 255 and 268):

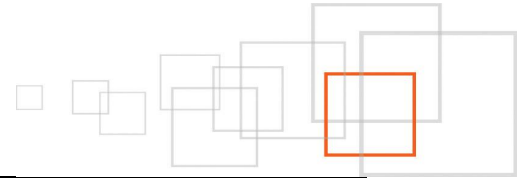
```
<!--  
<filter class="solr.ASCIIFoldingFilterFactory"/>  
-->
```

## 10 Step 6: Configure eZ Find

Edit `/var/www/ezpublish/ezfindexample1/extension/ezfind/settings/solr.ini` and update the SearchServerURI setting:

```
SearchServerURI=http://localhost:8983/solr/ezfindexample1
```

Edit `/var/www/ezpublish/ezfindexample2/extension/ezfind/settings/solr.ini` and update the SearchServerURI setting:



```
SearchServerURI=http://localhost:8983/solr/ezfindexample2
```

**NOTE:** If you are curious and decide to look at the file extension/ezfind/settings/ezfind.ini, **please note that the MultiCore setting under [LanguageSearch] has NOTHING to do with our multi-core setup.** It is used when splitting content in different languages into its own cores, and is beyond the scope of this tutorial. Leave ezfind.ini alone for now.

## 11 Step 7: Configure eZ Tika

Out of the box eZ Tika amazingly requires no configuration, but since we have split off the binaries we need to update two files in each eZ Publish instance.

Edit the following files:

```
/var/www/ezpublish/ezfindexample1/extension/eztika/settings/binaryfile.ini.append.php  
/var/www/ezpublish/ezfindexample2/extension/eztika/settings/binaryfile.ini.append.php
```

In the above files, update the TextExtractionTool setting under [HandlerSettings]:

```
TextExtractionTool=/srv/tika/bin/eztika
```

Now edit the following files:

```
/var/www/ezpublish/ezfindexample1/extension/eztika/settings/binaryfile.ini.append.withxpdf.php  
/var/www/ezpublish/ezfindexample2/extension/eztika/settings/binaryfile.ini.append.withxpdf.php
```

In the above files, update the TextExtractionTool setting under [HandlerSettings]:

```
TextExtractionTool=/srv/tika/bin/eztika
```

In the same files, update the TextExtractionTool setting under [PDFHandlerSettings]:

```
[PDFHandlerSettings]  
TextExtractionTool=/srv/tika/bin/ezpdfTOText
```

Edit the file /srv/tika/bin/eztika and update the TIKA\_JAR variable:

```
TIKA_JAR=/srv/tika/bin/tika.jar
```

## 12 Step 8: Activate the extensions

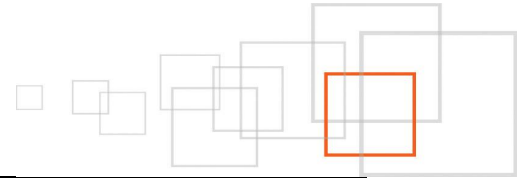
Edit the following files:

```
/var/www/ezpublish/ezfindexample1/settings/override/site.ini.append.php  
/var/www/ezpublish/ezfindexample2/settings/override/site.ini.append.php
```

Under [ExtensionSettings], add the following in the order shown:

```
[ExtensionSettings]  
ActiveExtensions[]
```





```
ActiveExtensions[]=eztika
ActiveExtensions[]=ezfind
```

Regenerate autoloads and clear cache:

```
su - www-data
cd /var/www/ezpublish/ezfindexample1/settings/override
php bin/php/ezpgenerateautoloads.php
php bin/php/ezcache.php -clear-all
exit
```

## 13 Step 9: Start up Solr

Before proceeding with this step, make sure that:

- Your sites have some content that can be indexed
- You have added some file attachments such as PDF's, Word documents etc
- The relevant content classes have their attributes marked searchable.

**NOTE: The default eZ Publish setting for the “file” attribute in the “File” class is NOT searchable!**  
Make sure attributes using the “file” datatype are set to searchable before you start indexing or adding file attachments, otherwise you might spend hours trying to find out why eZ Find indexes your text fields but not your files.

Now we are ready to start the Solr engine. For the first run we're going to do it via a command line to check that everything is in order and to monitor the indexing process. Later we're going to create a startup script to handle the stopping and starting of Solr.

Therefore, you should open up a separate terminal window and connect to your server to run Solr. After this, issue the following commands as root:

```
cd /srv/solr/java
java -Dezfind -Dsolr.solr.home=/srv/solr/cores -server -d64 -Xmx768m -Xms768m -XX:
+UseParallelGC -XX:+AggressiveOpts -XX:NewRatio=5 -jar start.jar
```

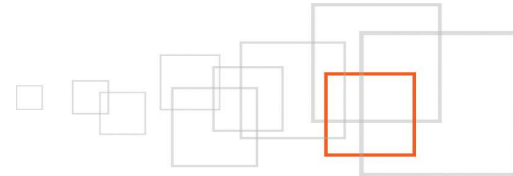
The parameters can be tweaked, and if you have little memory you can reduce the `-Xmx768m` and `-Xms768m` parameters to lower numbers, such as `-Xmx256m -Xms256m`.

If everything went well, you should see a cascade of output in the java terminal, ending with something like `2012-04-15 11:57:04.640:INFO::Started SocketConnector@0.0.0.0:8983`

## 14 Step 10: Index your sites

Run the following commands as the web user, substituting `your_admin_siteaccess` as necessary.

```
su - www-data
cd /var/www/ezpublish/ezfindexample1/
php extension/ezfind/bin/php/updatesearchindexsolr.php -s your_admin_siteaccess
```



```
cd /var/www/ezpublish/ezfindexample2/  
php extension/ezfind/bin/php/updatesearchindexsolr.php -s your_admin_siteaccess  
exit
```

You should see some indexing feedback as Solr indexes your sites, such as threads started/stopped and total time to index. If you have Solr running in a separate terminal, you can examine its output in case of any errors.

By default, eZ Find indexes new content as it is added. You can also defer indexing to a cronjob – see the official eZ Find documentation.

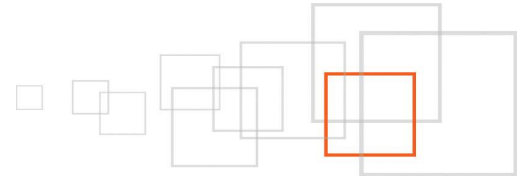
## 15 Step 11: Edit your search template and start searching

Create or edit the search template for each of your sites, substituting **path** and **design directory** as necessary:

```
/var/www/ezpublish/ezfindexample1/design/plain_site/templates/content/search.tpl  
/var/www/ezpublish/ezfindexample2/design/plain_site/templates/content/search.tpl
```

To just get started, the files should contain the code below. You can of course do lots of further customisation, but this is enough to test that Solr search is working and show off Solr's relevancy sorting and keyword highlighting.

```
{let search=false()}  
{section show=$use_template_search}  
{set page_limit=10}  
{set search=fetch(ezfind, search,  
    hash(query,$search_text,  
    section_id,$search_section_id,  
    subtree_array,$search_subtree_array,  
    offset,$view_parameters.offset,  
    limit,$page_limit))}  
{set search_result=$search['SearchResult']}  
{set search_count=$search['SearchCount']}  
{set stop_word_array=$search['StopWordArray']}  
{set search_data=$search}  
{/section}  
<h1>{"Search"|i18n("design/standard/content/search")}</h1>  
<form action="{/content/search/"|ezurl} method="get">  
<p>  
<input type="text" class="textshort" name="SearchText" value="{ $search_text|wash}" />  
<input name="SearchButton" type="submit" class="submit" value="{ 'Search'|  
i18n('global/labels')}" />  
</p>  
{switch name=Sw match=$search_count}  
{case match=0}  
<div class="warning">  
<h2>{'No results were found when searching for "%1"|  
i18n("design/standard/content/search",, array($search_text|wash))}</h2>  
</div>  
{/case}
```



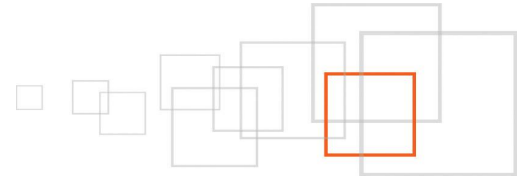
```
{case}
<div class="feedback">
<h2>{'Search for "%1" returned %2 matches'|
i18n("design/standard/content/search",,array($search_text|wash,$search_count))}</h2>
</div>
{/case}
{/switch}
{foreach $search_result as $snode}
<h2><a href="{ $snode.url_alias|ezurl(no) }">{$snode.name}</a></h2>
<p>{$snode.highlight}</p>
{/foreach}
{include name=Navigator
uri='design:navigator/google.tpl'
page_uri='/content/search'
page_uri_suffix=concat('?SearchText=', $search_text|urlencode, $search_timestamp|gt(0) |
choose(' ', concat('&SearchTimestamp=', $search_timestamp)))
item_count=$search_count
view_parameters=$view_parameters
item_limit=$page_limit}
</form>
{/let}
```

Clear caches in your site admin, then go to your site and do some searches. Try both regular and wildcard (\*) searches, and searching on non-English characters if needed. Check that results are returned from both content objects and uploaded files with text content.

## 16 Step 12: Create a startup script

Below is a sample startup script for Debian 6, customised for this tutorial. The eZ Find installation also contains example startup scripts for Debian, Gentoo and Red Hat. Note that the PARAMETERS variable is customised for a 64-bit environment. If this doesn't work for you, you can adapt one of the scripts in the bin/scripts directory in the eZ Find extension. Remember to set SOLR\_HOME to the correct path for your installation.

```
#!/bin/sh -e
#
# eZ Find init script for debian.
#
# Usage:
#
# Set the correct SOLR_HOME value, and copy this file to /etc/init.d
# Symlink to /etc/init.d/solr to /etc/rc3.d/S70solr and /etc/rc5.d/S70solr
#
# Example:
# cp solr /etc/init.d/solr# cd /etc/init.d && chmod 755 solr
# cd /etc/rc2.d && ln -s ../init.d/solr S70solr
# cd /etc/rc5.d && ln -s ../init.d/solr S70solr
# cd /etc/rc2.d && ln -s ../init.d/solr K70solr
```



```
# cd /etc/rc5.d && ln -s ../init.d/solr K70solr

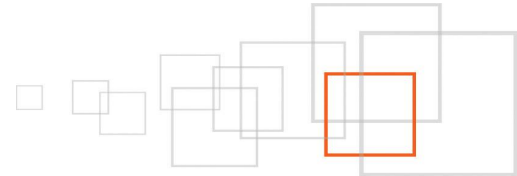
. /lib/lsb/init-functions

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DESC="Solr indexing server"
NAME=solr
SOLR_HOME=/srv/solr/java
PARAMETERS="-Dezfind -Dsolr.solr.home=/srv/solr/cores -server -d64 -Xmx768m -Xms768m
-XX:+UseParallelGC -XX:+AggressiveOpts -XX:NewRatio=5 -jar start.jar"
DAEMON=/usr/bin/java
PIDFILE=/var/run/$NAME.pid
SCRIPTNAME=/etc/init.d/$NAME

# Gracefully exit if the package has been removed.
test -x $DAEMON || exit 0

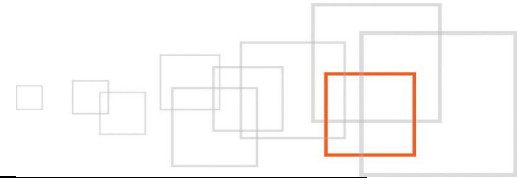
#
# Function that starts the daemon/service.
#
d_start() {
    start-stop-daemon --start --pidfile $PIDFILE --chdir $SOLR_HOME --background
--make-pidfile --exec $DAEMON -- $PARAMETERS
}
#
# Function that stops the daemon/service.
#
d_stop() {
    start-stop-daemon --stop --quiet --pidfile $PIDFILE --name java
    rm -f $PIDFILE
}
#
# Function that checks if solr is running
# Returns success (0) if solr is running, failure (1) if not running
#
d_status() {
    if [ -f "$PIDFILE" ] && ps `cat $PIDFILE` >/dev/null 2>&1; then
        return 0 # EXIT_SUCCESS
    else
        return 1 # EXIT_FAILURE
    fi
}

case "$1" in
    start)
        echo " * Starting $DESC: $NAME"
        if d_status; then
```



```
        echo "    ...$NAME is already running."
        return 1
    fi
    d_start
    if d_status; then
        echo "    ...done."
    else
        echo "    ...failed to start solr"
        exit 1
    fi
    ;;
stop)
    echo " * Stopping $DESC: $NAME"
    if d_status; then
        d_stop
        if d_status; then
            echo "    ...$NAME is still running."
            exit 1
        else
            echo "    ...done."
        fi
    else
        echo "    ...$NAME is not running."
    fi
    ;;
status)
    if d_status; then
        echo " * $NAME is running (PID: `cat $PIDFILE`)"
    else
        echo " * $NAME is not running."
    fi
    ;;
restart|force-reload)
    echo -n "Restarting $DESC: $NAME"
    d_stop
    sleep 1
    d_start
    echo "."
    ;;
*)
    echo "Usage: $SCRIPTNAME {start|stop|restart|force-reload}" >&2
    exit 1
    ;;
esac
exit 0
```

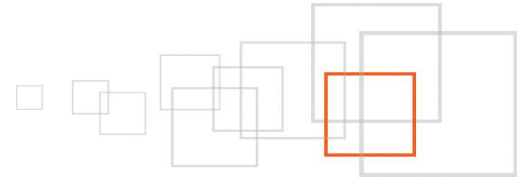
You can put the above script in the file `/etc/init.d/solr` and set it to autostart:



```
cd /etc/init.d  
chmod 755 solr  
update-rc.d solr defaults
```

You can then also stop, start, reload or restart Solr using:

```
/etc/init.d/solr start|stop|restart|force-reload
```



## 17 Adding more cores

Adding more sites to be indexed by eZ Find and Solr is a simple matter. Let's assume that the new site for which we want to add a core is installed in the directory `/var/www/ezpublish/ezfindexample3/`, that the eZ Tika and eZ Find extensions are installed in the site's `extensions/` directory, and that the database has been readied for eZ Find (see **step 2** of this tutorial).

1. Perform the following commands:

```
cd /srv/solr/cores
cp -rp ezfindexample2 ezfindexample3
```

2. Edit `/srv/solr/cores/solr.xml` and add the new core:

```
<?xml version="1.0" encoding="UTF-8" ?>
<solr persistent="true" sharedLib="lib">
<cores adminPath="/admin/cores">
  <core name="ezfindexample1" instanceDir="ezfindexample1" />
  <core name="ezfindexample2" instanceDir="ezfindexample2" />
  <core name="ezfindexample3" instanceDir="ezfindexample3" />
</cores>
</solr>
```

3. Edit `/srv/solr/cores/ezfindexample3/conf/solrconfig.xml` and update the `dataDir` element:

```
<dataDir>/srv/solr/cores/ezfindexample3/data</dataDir>.
```

4. Restart Solr

5. Perform **steps 6-8** and **10-11** of this tutorial, substituting the example paths with paths to the new site.

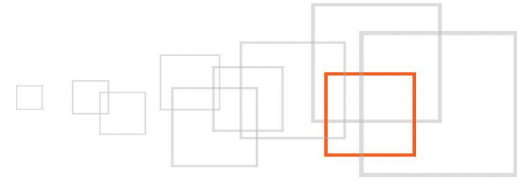
That's it!

## 18 Upgrading

When upgrading eZ Find, perform this upgrade on each site as per normal instructions, keeping your `solr.ini` file intact. Then copy the `extension/ezfind/java` directory into `/srv/solr/` to update the binaries. Be mindful of any changes in `schema.xml` that may need to be updated in each core's `schema.xml` file.

The same principle applies to eZ Tika. Be sure to keep your `customised binaryfile.ini.append.php` and `binaryfile.ini.append.withxpdf.php`, then copy the new `extension/eztika/bin` directory into `/srv/tika/`.

**NOTE:** since eZ Tika is mainly a wrapper for the Apache Tika, you can download the latest tika jar file from <http://tika.apache.org/download.html>. Then rename it `tika.jar` and put it in the `/srv/tika/bin/` directory. You don't need to upgrade the actual eZ Tika extension in this case, but you'll enjoy the benefits of newer versions of Tika.



## 19 Conclusion

I hope this tutorial has helped you both install a working multi-core instance of Solr and eZ Find, along with some insight into how the different technologies fit together. The hard labour should be worth it!

Advanced configuration and development with eZ Find and Solr is a large subject matter. See the resources section below for some more reference and learning materials.

## 20 Resources

- eZ Find official documentation – <http://doc.ez.no/Extensions/eZ-Publish-extensions/eZ-Find>
- eZ Find development tutorial - <http://share.ez.no/learn/ez-publish/advanced-development-with-ez-find-part-1-datatypes-in-solr-and-ez-find>
- Apache Tika home - <http://tika.apache.org/>
- Apache Solr and Lucene home - <http://lucene.apache.org/solr/>

## 21 About the author: Daniel Arnrup-Øien



Daniel Arnrup-Øien is Board Director and developer at Open Concept SA, a Norwegian cooperative corporation dedicated to providing complete open source web solutions with high quality and performance. He has been developing in eZ Publish since 2005. He can be reached at [daniel.oien@openconcept.no](mailto:daniel.oien@openconcept.no).

## 22 License choice

This work is licensed under the Creative Commons Attribution-ShareAlike (CC BY-SA) license.