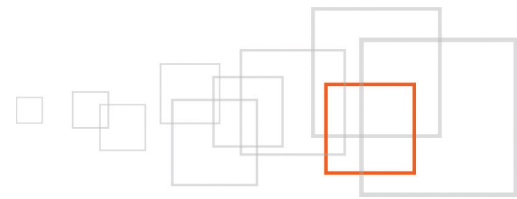


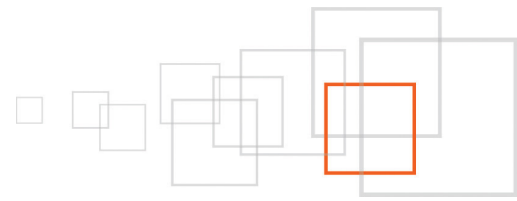
Running eZ Publish daemons in shared virtual hosting environment

By Marko Žmak
<http://www.studioartlan.com/>



Index

1 Goal description.....	3
2 Introduction.....	3
3 Pre-requisites and target audience.....	3
4 Situation and demands.....	3
Situation.....	3
Demands.....	4
5 The idea.....	4
6 Step by step.....	4
Install pre-requisites.....	4
Create a user for starting daemons and Install OpenOffice conversion macro.....	4
Extract the eZP installation and set file permissions.....	5
Create the startup scripts.....	5
Install the scripts as daemons.....	5
Run the daemons.....	5
Setup the extensions.....	5
7 Solr startup script.....	6
8 Open Office startup script.....	10
9 Tweaks.....	14
Running OpenOffice without the XWindows Server.....	14
Running on non-default ports.....	14
10 Conclusion.....	15
11 Resources.....	15
12 Credits.....	15
13 About the author : Marko Žmak.....	15
14 License choice.....	15



1 Goal description

At the end of this tutorial, you should have enough knowledge for setting up the eZ Find Solr daemon, and ezodf (OpenOffice.org) conversion daemon in a shared virtual hosting environment.

2 Introduction

The recommended way of using the ezfind and ezodf extensions on a server with multiple eZP installations is to have only one instance of Solr server and one instance of OpenOffice conversion server for all the eZP installations.

Running single instances of this servers for multiple eZP installations is not a problem if you have full access to your server, but if you happen to have many eZP installation in a limited virtual hosting environment, setting these two servers can be a little bit tricky. This tutorial describes how to do it.

3 Pre-requisites and target audience

- an eZ Publish 4.x installation
- ability to run shell scripts on the server

To fully understand this tutorial, you need:

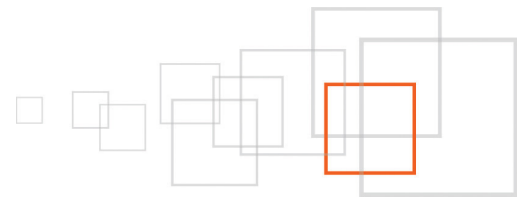
- knowledge and experience in setting up the ezfind and ezodf extensions
- experience in setting up Solr server and OpenOffice conversion daemon
- some knowledge of bash script programming
- basic knowledge of linux administration and understanding of shared virtual hosting environment

Since this tutorial contains step-by-step instructions, and some parts should be done by the system administrator of the server, it can also be understood by anyone whit some basic knowledge of eZ Publish and shared virtual hosting concepts.

4 Situation and demands

Situation

- A server in a virtual hosting (chroot) environment with many eZP installations on different user accounts (each for a different client),
- No direct access to the root user



Demands

- Only one Solr server and only one OpenOffice conversion server instance for all eZP installations,
- Start both servers as daemons at boot time,
- Being able to start/stop/restart the daemons,
- Don't run daemons as root (for security reasons),
- Being able to read and reset daemons' log files

5 The idea

In order for this to work we have to run the OpenOffice conversion server as a user which has access to all the eZP installations, and write access to their "var" folders - this is because the OpenOffice conversion server writes the converted file inside the "var" folder of the eZP instance that requested the conversion. So the idea goes like this:

- Run daemons under the user of the web server (e.g. www, nobody, apache) - it is the only one that for sure has access to all the eZP installations . Also the www user is usually restricted enough to avoid security issues
- Have another user for starting/stopping daemons
- For each daemon create a startup script to start as www user or stop it

6 Step by step

To setup the Solr and OpenOffice conversion server as daemons follow these step by step instructions.

Install pre-requisites

Install the usual pre-quisites for Solr and OpenOffice conversion daemons that are described in their respective documentations.

Note: There's a way for setting up and running OpenOffice and our OpenOffice conversion server without installing an XWindows Server on the server. It's described later in the "Tweaks" section of this tutorial.

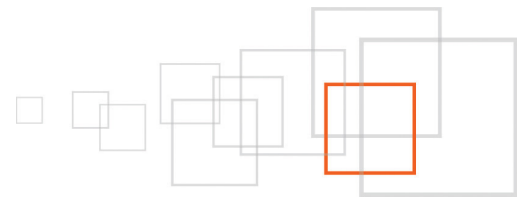
Create a user for starting daemons and Install OpenOffice conversion macro

Create a new user with a shell access (let's call it ezdaemon), or ask your server admin to do it.

Install the needed conversion macro for ezdaemon user. The instructions for installation of the macro can be found here: <http://doc.ez.no/Extensions/ODF-Import-Export/eZODF-extension>

Note: If you didn't install an XWindows Server you can still install the conversion macro using the OpenOffice shared user install. This install requires root acces so you should ask your server administrator to do it. The documentation can be found here:

http://wiki.services.openoffice.org/wiki/Documentation/Administration_Guide/Using_Package_Manager



(section: "To Add an Extension for All Users ")

Extract the eZP installation and set file permissions

In the home folder of ezdaemon user unpack the eZP installation to the "ezpublish" subfolder. You don't have to run the web based eZP setup wizard, we only need this to get the servers in ezfind and ezodf extensions.

Note that the versions of this two extensions unpacked here must match (or be compatible with) the versions of the same extensions used in all of your eZP installations.

In order for the Solr server to work you should give the "nobody" user read and write permissions to the folder "extension/ezfind/java/solr/data" (and its subfolders) inside the extracted "ezpublish" folder. It's needed because this is the place where the Solr server will write its index files.

Note: In a normal situation when we are **not** in shared virtual hosting environment, and we **have** root access, the solr data folder should be moved to a global system location. However in our case leaving it in this "dummy" eZP installation is a better solution because you can change the configuration files and upgrade the ezfind extension without having to bug the system administrator.

Create the startup scripts

Create the scripts for starting the daemons as the user your web server is running as. Let's say it is "nobody" user. I have started from the RHEL script that comes with the ezfind extension, and modified it to suit this specific needs. I've attached the full scripts along with this tutorial.

Install the scripts as daemons

Ask your server admin to install your scripts as daemons. If running RHEL or similar OS, just put the scripts in the "/etc/init.d/" folder.

Also, ask the server admin to give the user ezdaemon sudo privileges for this two scripts.

Run the daemons

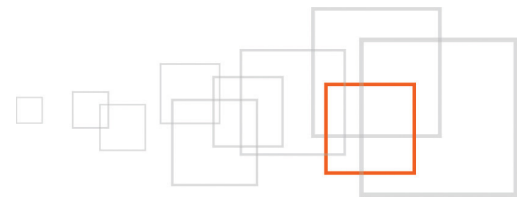
Run the daemons using the startup script. For example like this:

```
sudo /etc/init.d/ezodf start
sudo /etc/init.d/solr start
```

Prior to running them make sure that the needed ports are open on the server (the ports are described in the docs for both servers).

Setup the extensions

Setup the ezfind and ezodf extensions in your eZP installations as described in the docs. Once the daemons started and working it should not take more than 5-10 minutes to set it up for each eZP installation.



7 Solr startup script

This is the startup script for running solr server on RHEL or CENTOS. It's based on the original startup script that comes with ezfind installation and modified to suit this specific case.

The main modifications consist of:

- start the daemon as the "nobody" user
- redirect the stdout and stderr of daemon to log files

Usage: /etc/init.d/solr {start|stop|restart|reload|status}

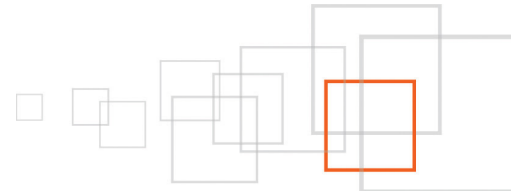
Code:

```
#!/bin/bash
#
# eZ Find init script for RHEL and CENTOS.
#
# 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/rc3.d && ln -s ../init.d/solr S70solr
# cd /etc/rc5.d && ln -s ../init.d/solr S70solr
# cd /etc/rc3.d && ln -s ../init.d/solr K70solr
# cd /etc/rc5.d && ln -s ../init.d/solr K70solr

DESC="Solr indexing server"
NAME=solr
JAVA_HOME=

SOLR_HOME=/home/ezdaemon/ezpublish/extension/ezfind/java
HOME=/home/ezdaemon
LOGDIR=$HOME/log
LOG_STDOUT=$LOGDIR/${NAME}_stdout.log
LOG_STDERR=$LOGDIR/${NAME}_stderr.log

source /etc/rc.d/init.d/functions
```



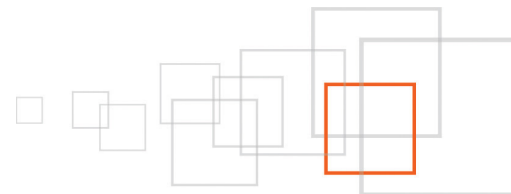
```
for JAVA in "$JAVA_HOME/bin/java" "/usr/bin/java" "/usr/local/bin/java"
do
    if [ -x $JAVA ]
    then
        break
    fi
done
if [ ! -x $JAVA ]
then
    echo "Unable to locate java. Please set JAVA_HOME environment variable."
    exit
fi

RETVAL=0

d_start() {
    CURRENT_DIR=`pwd`
    daemon --user=nobody --check $NAME --pidfile $HOME/var/run/solr.pid "cd $SOLR_HOME
    && nohup $JAVA -jar $SOLR_HOME/start.jar" >> $LOG_STDOUT 2>> $LOG_STDERR &
    RETVAL=$?
    sleep 1 # Sleep 1 second, to make sure java is registered.
    pid=`pidof java`
    echo $pid > $HOME/var/run/solr.pid
    cd $CURRENT_DIR
    [ $RETVAL -eq 0 ] && su - nobody -c "touch $HOME/var/lock/subsys/$NAME"
    return $RETVAL
}

d_stop() {
    killproc -p $HOME/var/run/solr.pid $NAME >> /dev/null 2&>1
    RETVAL=$?
    [ $RETVAL -eq 0 ] && rm -f $HOME/var/lock/subsys/$NAME
    return $RETVAL
}

d_restart() {
    d_stop >> /dev/null 2&>1
    sleep 2
}
```

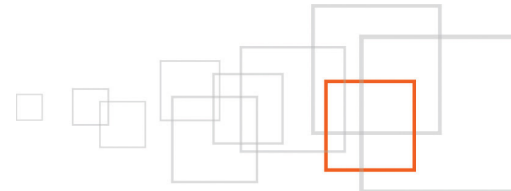


```
d_start >> /dev/null 2&>1
}

d_reload() {
    killproc -p $HOME/var/run/solr.pid $NAME -HUP 2&>1
    RETVAL=$?
    return $RETVAL
}

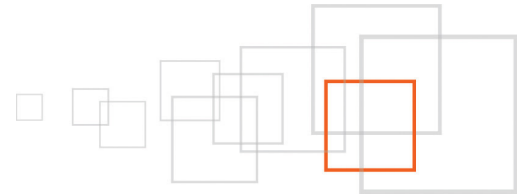
d_status() {
    status -p $HOME/var/run/solr.pid $NAME >> /dev/null 2&>1
    return $?
}

case "$1" in
start)
    echo " * Starting $DESC ($NAME)"
    d_status
    if [ $? -eq 0 ]
    then
        echo " ...already running."
    else
        if d_start
        then
            echo " ...done."
        else
            echo " ...failed."
            RETVAL=1
        fi
    fi
    ;;
stop)
    echo " * Stopping $DESC ($NAME)"
    if d_stop
    then
        echo " ...done."
    else
        echo " ...failed."
        RETVAL=1
    fi
    ;;
*)
    echo " * Invalid argument: $1"
    RETVAL=1
    ;;
esac
```

```
fi
;;
restart)
echo " * Restarting $DESC ($NAME)"
d_status
if [ $? -ne 0 ]
then
echo "   ...not running."
RETVAL=1
else
if d_restart
then
echo " * ...done."
else
echo " * ...failed."
RETVAL=1
fi
fi
;;
reload)
echo " * Reloading $DESC ($NAME): "
d_reload
echo "   ...done."
;;
status)
d_status
if [ $? -eq 0 ]
then
echo " * $DESC ($NAME) is running"
else
echo " * $DESC ($NAME) is not running"
fi
;;
*)
echo "$Usage: $0 {start|stop|restart|reload|status}"
RETVAL=1
esac

exit $RETVAL
```



8 Open Office startup script

To start the OpenOffice conversion server, we use the same script, modified to run the conversion daemon.

Usage: /etc/init.d/ezodf {start|stop|restart|reload|status}

Code:

```
#!/bin/bash
#
# eZ ODF Conversion daemon init script for RHEL and CENTOS.
#
# Usage:
#
# Set the correct folder variables values, and copy this file to /etc/init.d
# Symlink to /etc/init.d/ezodf to /etc/rc3.d/S70ezodf and /etc/rc5.d/S70ezodf
#
# Example:
# cp ezodf /etc/init.d/ezodf
# cd /etc/init.d && chmod 755 ezodf
# cd /etc/rc3.d && ln -s ../init.d/ezodf S70ezodf
# cd /etc/rc5.d && ln -s ../init.d/ezodf S70ezodf
# cd /etc/rc3.d && ln -s ../init.d/ezodf K70ezodf
# cd /etc/rc5.d && ln -s ../init.d/ezodf K70ezodf

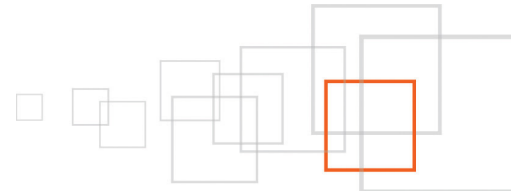
DESC="eZ ODF Conversion daemon"
NAME=ezodf

HOME=/home/ezdaemon
VARDIR=$HOME/var
LOGDIR=$HOME/log

LOG_STDOUT=$LOGDIR/${NAME}_stdout.log
LOG_STDERR=$LOGDIR/${NAME}_stderr.log

SCRIPT=$HOME/ezpublish/extension/ezodf/scripts/daemon.php
PHP=/usr/local/bin/php
PIDFILE=$VARDIR/run/ezodf.pid

source /etc/rc.d/init.d/functions
```



```
RETVAL=0

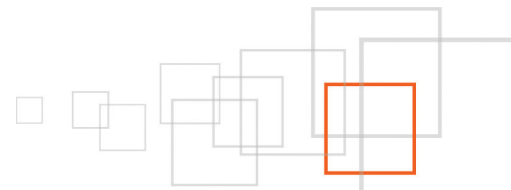
d_start() {
    CURRENT_DIR=`pwd`
    daemon --user=nobody --check $NAME --pidfile $PIDFILE "nohup $PHP $SCRIPT" >>
    $LOG_STDOUT 2>> $LOG_STDERR &
    RETVAL=$?
    sleep 1
    pid=`pgrep -u nobody php`
    echo $pid > $PIDFILE
    cd $CURRENT_DIR
    [ $RETVAL -eq 0 ] && su - nobody -c "touch $VARDIR/lock/subsys/$NAME"
    return $RETVAL
}

d_stop() {
    killproc -p $PIDFILE $NAME >> /dev/null 2&>1
    RETVAL=$?
    [ $RETVAL -eq 0 ] && rm -f $VARDIR/lock/subsys/$NAME
    return $RETVAL
}

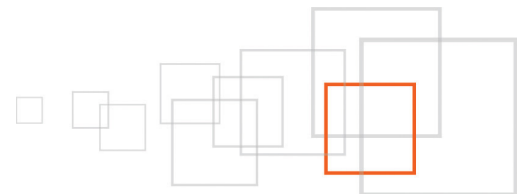
d_restart() {
    d_stop >> /dev/null 2&>1
    sleep 1
    d_start >> /dev/null 2&>1
}

d_reload() {
    killproc -p $PIDFILE $NAME -HUP 2&>1
    RETVAL=$?
    return $RETVAL
}

d_status() {
    status -p $PIDFILE $NAME >> /dev/null 2&>1
    return $?
}
}
```

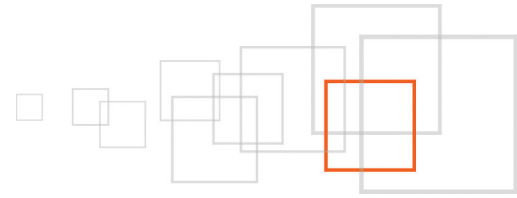


```
case "$1" in
start)
  echo " * Starting $DESC ($NAME)"
  d_status
  if [ $? -eq 0 ]
  then
    echo "   ...already running."
  else
    if d_start
    then
      echo "   ...done."
    else
      echo "   ...failed."
      RETVAL=1
    fi
  fi
  ;;
stop)
  echo " * Stopping $DESC ($NAME)"
  if d_stop
  then
    echo "   ...done."
  else
    echo "   ...failed."
    RETVAL=1
  fi
  ;;
restart)
  echo " * Restarting $DESC ($NAME)"
  d_status
  if [ $? -ne 0 ]
  then
    echo "   ...not running."
    RETVAL=1
  else
    if d_restart
    then
      echo " * ...done."
    fi
  fi
  ;;
*)
  echo " * Invalid command: $1"
  RETVAL=1
  ;;
esac
```



```
    else
        echo " * ...failed."
        RETVAL=1
    fi
fi
;;
reload)
    echo " * Reloading $DESC ($NAME): "
    d_reload
    echo "   ...done."
    ;;
status)
    d_status
    if [ $? -eq 0 ]
    then
        echo " * $DESC ($NAME) is running"
    else
        echo " * $DESC ($NAME) is not running"
    fi
    ;;
*)
    echo "$Usage: $0 {start|stop|restart|reload|status}"
    RETVAL=1
esac

exit $RETVAL
```



9 Tweaks

Running OpenOffice without the XWindows Server

If you want to run OpenOffice without the XWindows Server (which can spare you some headaches) you have to run it with the “-headless” parameter. To add this parameter you will have to edit the *extension/ezodf/scripts/daemon.php* script inside the eZP installation of the ezdaemon user. Just change this line:

```
$convertShellCommand = escapeshellcmd( $ooexecutable . " -writer -invisible" ) . " " .
```

to this:

```
$convertShellCommand = escapeshellcmd( $ooexecutable . " -writer -invisible -headless" ) . " " .
```

This is actually a hack, and it is not the “clean” way to do it, but I have also proposed an enhancement that would allow to do this without hacking the script: <http://issues.ez.no/17308>

Running on non-default ports

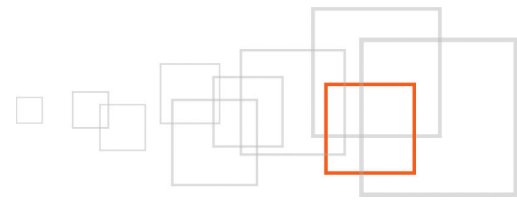
It is recommended to run the Solr and OpenOffice conversion server on non-default ports. First, it is generally more secure not to use default ports. And second, in this way you leave the default port free for some quick testing of new eZP installations.

To change the port on which the OpenOffice conversion server runs, you'll have to edit *extension/ezodf/scripts/daemon.php* inside the eZP installation for ezdaemon user and change the value of the \$port variable.

The official way of changing the default Solr server port is to add the port parameter (for example “-Djetty.port=12345”) to the Solr startup command. However, this requires changing the solr startup script from the chapter 8 which requires root access. This means you would have to bug your system administrator every time you want to change the port for some reason. So in this situation the best way to change the port for Solr server is to edit *extension/ezfind/java/etc/jetty.xml* in the eZP installation of ezdaemon user. It is defined in the line that goes like this:

```
<Set name="port"><SystemProperty name="jetty.port" default="8983"/></Set>
```

Note: If you start the servers on non-default ports you should also change the corresponding settings in the ini files of the ezfind and ezodf extensions of every eZP Installation.



10 Conclusion

Running a single instance of Solr and OpenOffice conversion servers speeds up deployment of new eZP sites and makes setting ezfind and ezodf extensions a piece of cake.

Setting this in a virtual hosting environment requires a little bit more work, but it is worth the effort. It can be very useful to provide a quality eZP shared hosting.

In this tutorial we gave step-by-step instructions to set it up on RHEL/CENTOS, but also a general idea that allows to set it up on other operating systems.

11 Resources

- **eZFind (Solr-based, enterprise-grade search extension for eZ publish) docs:**
<http://doc.ez.no/Extensions/eZ-Find/2.1/>
- **eZODF (Open-office extension for eZ Publish) docs:** <http://doc.ez.no/Extensions/ODF-Import-Export/eZODF-extension>
- **OpenOffice.org administration guide :**
http://wiki.services.openoffice.org/wiki/Documentation/Administration_Guide

12 Credits

Special thanks to I.T: Plus d.o.o. and Neomedia digital agency for help in testing and setting it up.

13 About the author : Marko Žmak



Marko Žmak is a Croatian web developer with a great experience in eZ Publish platform. He has specialised in creating innovative eZP solutions tailored and optimized for specific customers' needs.

Marko is an active member of the eZ Publish community and owner of eZ community partner company Studio Artlan : <http://share.ez.no/directory/companies/studio-artlan>

14 License choice



Attribution-NonCommercial-ShareAlike Creative Commons licence (cc by-nc-sa)

<http://creativecommons.org/about/licenses/>