

---

# Updating Koha

Joshua Ferraro <jmf at liblime dot com>

Pierrick Le Gall <pierrick at koha-fr dot org>

Copyright © 2004-2006 Joshua Ferraro

This document is related to Koha and is licensed to you under the GNU General Public License version 2 or later (<http://www.gnu.org/licenses/gpl.html>).

Koha-related documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies.

You may create a derivative work and distribute it provided that you:

1. License the derivative work with this same license, or the Linux Documentation Project License (<http://www.tldp.org/COPYRIGHT.html>). Include a copyright notice and at least a pointer to the license used.
2. Give due credit to previous authors and major contributors.

Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

No liability for the contents of this document can be accepted. Use the concepts, examples and information at your own risk. There may be errors and inaccuracies, that could be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products or brands should not be seen as endorsements.

2006-03-17

## Revision History

Revision 2.2.0	2004-12-13	jmf
----------------	------------	-----

Changes made by Joshua Ferraro. Original Document

Revision 2.2.0p1	2005-01-09	jmf
------------------	------------	-----

Minor change to cvs command following official release of 2.2.0

Revision 2.2.5	2006-03-13	plg
----------------	------------	-----

Bug fixes on symlinks. New script to create symlinks, with comand-line parameters. Precision on CVS repository : use of CVS working directory when talking about the checked out directory.

## Table of Contents

1. Introduction .....	2
2. Before You Begin .....	2
2.1. Understand What You Will Be Doing .....	2
3. Checking Out Koha from Savannah .....	2
4. Creating the Symlinks .....	3
4.1. Automating the Symlink Creation .....	4
5. Updating Your httpd.conf File .....	6
6. Updating Your Database .....	6

## 1. Introduction

At the beginning of the project, Koha project developers used `sourceforge.net` as their centralized repository for source code; in December 2005 the source code was moved to `savannah.nongnu.org`. Savannah's repositories use CVS (Concurrent Versions System), a tool used to manage changes within a source code tree.

Some in the Koha community have expressed interest in migrating Revision Management Systems (RMS) from CVS to a more distributed system like Arch. If project developers move all Koha development to an Arch model a different guide would be necessary, but for now, I can describe the basics of how to keep your Koha 2.2 system up-to-date using CVS and Savannah. At this time, this document does not discuss how to upgrade a system from an older version of Koha to a newer version.

## 2. Before You Begin

In order to access a CVS repository, you must first install a CVS client (if one is not already installed on your system). General information about CVS client software (and it's use with `sourceforge.net`) can be found in the following SourceForge.net document: Basic Introduction to CVS and SourceForge.net (SF.net) Project CVS Services [[http://sourceforge.net/docman/display\\_doc.php?docid=14033&group\\_id=1](http://sourceforge.net/docman/display_doc.php?docid=14033&group_id=1)]. Information more specific to Savannah can be found in the Savannah Frequently Asked Questions [<http://savannah.nongnu.org/faq/>].

### 2.1. Understand What You Will Be Doing

If you are unfamiliar with the concept of symlinks take a look at this helpful introduction: <http://librenix.com/?inode=5164>. You will be "checking out" a Koha CVS repository and then symlinking your installed Koha directory tree to the CVS repository which will enable you to update to the latest stable version of Koha by issuing a simple CVS update command.

The resulting directory structure will also afford you an easy centralized backup point for your Koha directory tree -- you should make sure to regularly backup your database and Koha directory tree to preserve your data as well as any customizations you've made to Koha templates, modules or scripts. More info on backing up your Koha system can be found in that section of the Koha documentation. For now, choose a location that is easy for you to remember. I have a directory `/build` on a separate partition that I use as the root directory for my Koha installation as well as custom library-specific scripts, backups, etc., so in the examples I give, replace `/build` with the location that you have chosen.

It's best to perform your update immediately after installing Koha to ensure that you will be running the very latest version. Also, if you setup the symlinks upon the initial install you may find that it's easier to roll-back an update if something should go wrong.

## 3. Checking Out Koha from Savannah

The checking out process is quite simple. Change your directory to the directory you have selected as

```
the root directory for your CVS working copy (mine is /build) and type: cvs -z3 -d:pserver:anonymous@cvs.savannah.nongnu.org:/sources/koha co -r rel_2_2 koha
```

At this point you should see a long listing of all the files that are being checked out, and eventually it will finish and you will be back at the command line:

```
...
cvs checkout: Updating koha/z3950
U koha/z3950/processz3950queue
U koha/z3950/search.pl
U koha/z3950/z3950-daemon-launch.sh
U koha/z3950/z3950-daemon-shell.sh
U koha/z3950/z3950import.pl
cvs checkout: Updating koha/z3950/encodingfix
U koha/z3950/encodingfix/Biblio.pm
U koha/z3950/encodingfix/README-english
U koha/z3950/encodingfix/README-polish
U koha/z3950/encodingfix/search.pl
cvs checkout: Updating koha/z3950/server
U koha/z3950/server/zed-koha-server.pl
[root@frodo]#
```

So now you have a new directory "koha" which contains the very latest files and bugfixes for that version of Koha. But this directory does not update your installed version of Koha. To do that you will need to link the installed directory tree to locations within the new directory you just created. One confusing element of this step is the fact that the CVS directory tree is quite different than the installation tree. This is because the organization of the documents after an installation doesn't make sense as the organization for developers to use while programming.

## 4. Creating the Symlinks

The first step you'll want to take is to backup your existing Koha directory tree. If you performed a default install of Koha without customizing your directory locations the following directories need to be backed up:

- /usr/local/koha/intranet/cgi-bin
- /usr/local/koha/intranet/htdocs/intranet-tmpl
- /usr/local/koha/intranet/modules/C4
- /usr/local/koha/opac/cgi-bin
- /usr/local/koha/opac/htdocs

This is easily accomplished by using the **mv** command. For example, to backup the /usr/local/koha/intranet/cgi-bin directory, change directories to /usr/local/koha/intranet and issue:

```
[root@frodo]# mv cgi-bin cgi-bin.bak
```

If for some reason you are having trouble with the symlinking or you want to revert back to the default directories you can always use this backup to restore things (simply reverse the above command).

After backing up all of the above directories you'll be ready to create the symlinks. This is accomplished using the **ln** command with the **-s** option. For more info on **ln**, see the **man** page.

`$KOHA_WORKING_DIR` is where you "checked out" Koha, `$KOHA_INSTALL_DIR` is where you installed Koha (`/usr/local/koha` by default) Here are the symlink mappings we're going to create:

- `$KOHA_WORKING_DIR ---> $KOHA_INSTALL_DIR/intranet/cgi-bin`
- `$KOHA_WORKING_DIR/koha-tmpl/intranet-tmpl --->`  
`$KOHA_INSTALL_DIR/intranet/htdocs/intranet-tmpl`
- `$KOHA_WORKING_DIR/C4 ---> $KOHA_INSTALL_DIR/intranet/modules/C4`
- `$KOHA_WORKING_DIR/opac ---> $KOHA_INSTALL_DIR/opac/cgi-bin`
- `$KOHA_WORKING_DIR/koha-tmpl/opac-tmpl --->`  
`$KOHA_INSTALL_DIR/opac/htdocs/opac-tmpl`

For example, to link the first directory listed above you would type the following:

```
[root@frodo]# ln -s /koha /usr/local/koha/intranet/cgi-bin
```

## 4.1. Automating the Symlink Creation

Automating the symlink creation is a good idea if you want to have a record of exactly what you did to create the symlinks. Automating also makes it easier if you need to restore a Koha installation quickly. Here's a simple Perl script that I use to create my symlinks:

```
#!/usr/bin/perl -w

use strict;
use Getopt::Long;

my %opt = ();
GetOptions(
    \%opt,
    qw/working_dir=s install_dir=s no-backup help/
) or die "\nHouston, we got a problem\n";

if (exists $opt{help}) {
    print <<FIN;

Create symbolic links between a Koha installation and a Koha CVS working
directory.

Usage: koha_symlinks.pl --install_dir=<installation directory>
                        --working_dir=<working directory>
                        [--no-backup]
                        [--help]

--install_dir: is the directory where Koha is installed. By default,
               /usr/local/koha

--working_dir: where you checked out Koha. No default.

--no-backup: you really don't want to backup your installation directory

--help: show this help
```

```

FIN

    exit(0);
}

$opt{install_dir} = exists $opt{install_dir}
? $opt{install_dir}
: '/usr/local/koha';

# Configurable Variables
foreach my $option (qw/working_dir install_dir/) {
    if (not exists $opt{$option}) {
        die 'option "', $option, '" is mandatory', "\n";
    }

    if (not -d $opt{$option}) {
        die '"', $opt{$option}, '" must be an existing directory', "\n";
    }

    if (not $opt{$option} =~ m{^/}) {
        die '--', $option, ' must be an absolute path', "\n";
    }
}

unless (exists $opt{'no-backup'}) {
    my @directories = (
        'intranet/cgi-bin',
        'intranet/htdocs/intranet-tmpl',
        'intranet/modules/C4',
        'opac/cgi-bin',
        'opac/htdocs',
    );

    foreach my $directory (map {$opt{install_dir}.'/'.'$_} @directories) {
        system(
            'mv',
            $directory,
            $directory.'.bak'
        );
    }
}

system(
    'ln',
    '-s',
    $opt{working_dir},
    $opt{install_dir}.'/intranet/cgi-bin'
);

system(
    'ln',
    '-s',
    $opt{working_dir}.'/koha-tmpl/intranet-tmpl',
    $opt{install_dir}.'/intranet/htdocs/intranet-tmpl'
);

system(
    'ln',
    '-s',
    $opt{working_dir}.'/C4',
    $opt{install_dir}.'/intranet/modules/C4'
);

```

```

system(
    'ln',
    '-s',
    $opt{working_dir}.'/opac',
    $opt{install_dir}.'/opac/cgi-bin'
);

system(
    'ln',
    '-s',
    $opt{working_dir}.'/koha-tmpl/opac-tmpl',
    $opt{install_dir}.'/opac/htdocs'
);

```

## 5. Updating Your httpd.conf File

Depending on your settings you may need to add the following line to both your intranet and opac virtual hosts configuration section of your `httpd.conf` or `/etc/koha-httpd.conf` file:

```
Options +FollowSymLinks
```

This ensures that Apache will allow connections to symlinked directories.

## 6. Updating Your Database

The next step in updating your Koha installation is to update your database. Over time, when bugs are fixed or when new features are added, the Koha database is subject to change. The `updatedatabase` script will modify your database to the latest version. It's located in the `koha/updater` directory of the CVS working copy. It is strongly recommended that you backup your database before running the `updatedatabase` script. For more information on how to backup your system see that section in the Koha documentation.

In order to run this script you may need to export the environment variable so that Perl knows where `Context.pm` is. You can do this by running:

```
[root@frodo]# export PERL5LIB="your_koha_cvs_working_copy"
```

or the path to your modules (usually `/usr/local/koha/intranet/modules`). Alternatively you can simply specify the path while executing the script:

```
[root@frodo]# perl -I your_koha_cvs_working_copy updatedatabase
```

You may see some messages that look like this:

```

localhost:/build/koha/updater# perl -I /build/koha updatedatabase
connected to your DB. Checking & modifying it
Could convert to MyISAM database tables...
  Alter z3950random in marc_breeding
  Alter bookfundid in aqbookfund
  Alter aqbudgetid in aqbudget
  Alter id in z3950servers
Creating index on z3950results
Checking for data required in table userflags...
Checking for data required in table systempreferences...

```