

# Cluster synchronisation with csync2

## *Managing distributed config files and application images*

Clifford Wolf

ROCK Linux - <http://www.rocklinux.org>

CNGW - <http://www.cngw.org>

LINBIT - <http://www.linbit.com>

Introduction

- The Problem
- The Solution

[Csync2 Overview](#)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

# Introduction

Introduction

● The Problem

● The Solution

Csync2 Overview

Csync2 Configuration

Bootstrapping the Databases

URLs and References

- Multiple hosts are sharing the same files
  - ◆ Configuration files
  - ◆ Application images
  - ◆ Other data such as htdocs
  
- Changes are rare but possible
  - ◆ Changes can happen on any cluster node
  - ◆ Changes might also be file removals or moves
  - ◆ Conflicts are possible and should be detected
  
- Setups can be very complex
  - ◆ High number of nodes and some might be down
  - ◆ File-groups which are only synced between some hosts
  - ◆ Actions to be triggered after updating some files

## Introduction

● The Problem

● The Solution

Csync2 Overview

Csync2 Configuration

Bootstrapping the Databases

URLs and References

- Csync2 targets this problems
  
- Csync2 has some requirements:
  - ◆ A GNU C-Compiler (alloca, etc)
  - ◆ GNU Flex and GNU Bison (config file parser)
  - ◆ The sqlite and librsync libraries
  - ◆ Autoconf+Automake
  
- Csync2 has been developed for and tested in large cluster environments at LINBIT Information Technologies.

[Introduction](#)

---

**Csync2 Overview**

- Building and Installing (1)
- Building and Installing (2)
- Csync2 Files
- Check Mode
- Update Mode
- Simple Mode
- Example configuration
- Csync2 usage (1)
- Csync2 usage (2)
- Csync2 usage (3)

[Csync2 Configuration](#)

---

[Bootstrapping the Databases](#)

---

[URLs and References](#)

---

# Csync2 Overview

Introduction

Csync2 Overview

● Building and Installing (1)

● Building and Installing (2)

● Csync2 Files

● Check Mode

● Update Mode

● Simple Mode

● Example configuration

● Csync2 usage (1)

● Csync2 usage (2)

● Csync2 usage (3)

Csync2 Configuration

Bootstrapping the Databases

URLs and References

## ■ ROCK Linux

- ◆ `rocket build csync2`
- ◆ `rocket create_gem csync2`
- ◆ `mine -i csync2-1.13.gem`

## ■ Debian GNU/LINUX

- ◆ `tar xzf csync2-1.13.tar.gz`
- ◆ `cd csync2-1.13; debuild`
- ◆ `dpkg -i csync2_1.13-1_i386.deb`

## ■ Others

- ◆ `tar xzf csync2-1.13.tar.gz`
- ◆ `cd csync2-1.13; ./configure`
- ◆ `make && make install`

# Building and Installing (2)

Introduction

Csync2 Overview

● Building and Installing (1)

● Building and Installing (2)

● Csync2 Files

● Check Mode

● Update Mode

● Simple Mode

● Example configuration

● Csync2 usage (1)

● Csync2 usage (2)

● Csync2 usage (3)

Csync2 Configuration

Bootstrapping the Databases

URLs and References

## ■ /etc/services

◆ `csync2 30865/tcp`

## ■ /etc/inetd.conf

◆ `csync2 stream tcp nowait root  
/usr/sbin/csync2 csync2 -i`

## ■ /etc/csync2\*.cfg

◆ Add your configs and synchronisation groups

## ■ /var/lib/csync2/\*

◆ Bootstrap the local status databases

[Introduction](#)

[Csync2 Overview](#)

● [Building and Installing \(1\)](#)

● [Building and Installing \(2\)](#)

● [Csync2 Files](#)

● [Check Mode](#)

● [Update Mode](#)

● [Simple Mode](#)

● [Example configuration](#)

● [Csync2 usage \(1\)](#)

● [Csync2 usage \(2\)](#)

● [Csync2 usage \(3\)](#)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

- `/usr/sbin/csync2`  
csync2 executable
- `/usr/share/csync2/csync2_locheck.sh`  
example csync2 logout check
- `/etc/csync2.cfg`  
config file for default config
- `/var/lib/csync2/hostname.db`  
database file for default config
- `/etc/csync2_foobar.cfg`  
config file for *foobar* config
- `/var/lib/csync2/hostname_foobar.db`  
database file for *foobar* config



Introduction

Csync2 Overview

- Building and Installing (1)
- Building and Installing (2)
- Csync2 Files
- **Check Mode**
- Update Mode
- Simple Mode
- Example configuration
- Csync2 usage (1)
- Csync2 usage (2)
- Csync2 usage (3)

Csync2 Configuration

Bootstrapping the Databases

URLs and References

- `csync2 -crv /`
- Compares local filesystem with local status database
- Changed files are updated in the status database
- `..` and are added to the dirty database
- Needs aprox 1.5 sec for 15.000 files (117.023 syscalls) on an AMD Athlon(tm) XP 1800+ Server on Linux 2.4.25.
- When no filename is given, the hint database (`csync2 -h`) is used.

[Introduction](#)

**Csync2 Overview**

- Building and Installing (1)
- Building and Installing (2)
- Csync2 Files
- Check Mode
- **Update Mode**
- Simple Mode
- Example configuration
- Csync2 usage (1)
- Csync2 usage (2)
- Csync2 usage (3)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

- `csync2 -uv`
- Update remote hosts according to the dirty database
- Conflicts are detected automatically
- .. and can be resolved using `csync2 -f`
- Failed updates are kept in the dirty database for later re-tries.
- A dry-run is done by `csync2 -ud`
- When no filename is given, the entire dirty-db is used.

[Introduction](#)

[Csync2 Overview](#)

- Building and Installing (1)
- Building and Installing (2)
- Csync2 Files
- Check Mode
- Update Mode
- **Simple Mode**
- Example configuration
- Csync2 usage (1)
- Csync2 usage (2)
- Csync2 usage (3)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

- `csync2 -xv`
- Check and update. This is what most people want to do.
- As with the other modes, this automatically resolves relative paths. So e.g. this is possible: `csync2 -xr .`
- All options for `-c` and `-u` are accepted.
- When no filename is given, `-r /` is implied.

# Example configuration

[Introduction](#)

[Csync2 Overview](#)

- Building and Installing (1)
- Building and Installing (2)
- Csync2 Files
- Check Mode
- Update Mode
- Simple Mode

[● Example configuration](#)

- Csync2 usage (1)
- Csync2 usage (2)
- Csync2 usage (3)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

```
group mygroup
{
    host host1 host2 host3;
    host host4@host4-eth2;

    key /etc/csync2.key_mygroup;

    include /etc/apache;
    include /home/bob;
    exclude /home/bob/temp;
    exclude *~ .*;

    action {
        pattern /etc/apache/httpd.conf;
        pattern /etc/apache/sites-available/*;
        exec "/usr/sbin/apache2ctl graceful";
        logfile "/var/log/csync2_action.log";
        do-local;
    }
}
```

# Csync2 usage (1)

[Introduction](#)

**Csync2 Overview**

- Building and Installing (1)
- Building and Installing (2)
- Csync2 Files
- Check Mode
- Update Mode
- Simple Mode
- Example configuration

**● Csync2 usage (1)**

- Csync2 usage (2)
- Csync2 usage (3)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

```
csync2 v1.12 - cluster synchronisation tool, 2nd generation
LINBIT Information Technologies GmbH <http://www.linbit.com>
Copyright (C) 2004 Clifford Wolf <clifford@clifford.at>
This program is free software under the terms of the GNU GPL.
```

```
Usage: csync2 [-v..] [-C config-name] [-D database-dir] [-N hostname] ..
```

With file parameters:

```
-h [-r] file..      Add (recursive) hints for check to db
-c [-r] file..      Check files and maybe add to dirty db
-u [-d] [-r] file.. Updates files if listed in dirty db
-f file..           Force this file in sync (resolve conflict)
-m file..           Mark files in database as dirty
```

Simple mode:

```
-x [-d] [[-r] file..] Run checks for all given files and update
remote hosts.
```

---

**Introduction**

---

**Csync2 Overview**

- Building and Installing (1)
- Building and Installing (2)
- Csync2 Files
- Check Mode
- Update Mode
- Simple Mode
- Example configuration
- Csync2 usage (1)
- **Csync2 usage (2)**
- Csync2 usage (3)

---

**Csync2 Configuration**

---

---

**Bootstrapping the Databases**

---

---

**URLs and References**

---

Without file parameters:

- c Check all hints in db and eventually mark files as dirty
- u [-d] Update (transfer dirty files to peers and mark as clear)
  
- H List all pending hints from status db
- L List all file-entries from status db
- M List all dirty files from status db
  
- S myname peername List file-entries from status db for this synchronisation pair.
  
- T Test if this node is in sync with all peers.
  
- T myname peername Test if this synchronisation pair is in sync.
  
- T myname peer file Show difference between file on peer and local.
  
- The modes -H, -L, -M and -S return 2 if the requested db is empty.  
The mode -T returns 2 if both hosts are in sync.
  
- i Run in inetd server mode.
- R Remove files from database which don't match config entries.

[Introduction](#)

**Csync2 Overview**

- Building and Installing (1)
- Building and Installing (2)
- Csync2 Files
- Check Mode
- Update Mode
- Simple Mode
- Example configuration
- Csync2 usage (1)
- Csync2 usage (2)
- **Csync2 usage (3)**

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

Modifiers:

- r Recursive operation over subdirectories
- d Dry-run on all remote update operations
  
- B Don't block everything into big SQL transactions. This slows down csync2 but allows multiple csync2 processes to access the database at the same time. Use e.g. when slow lines are used or huge files are transferred.
  
- I Init-run. Use with care and read the documentation first! You usually don't need this option unless you are initializing groups with really large file lists.
  
- G Group1,Group2,Group3,...  
Only use this groups from config-file.
  
- P peer1,peer1,...  
Only update this peers (still mark all as dirty).

Creating key file:

```
csync2 -k filename
```

[Introduction](#)

[Csync2 Overview](#)

**[Csync2 Configuration](#)**

- Configs, Groups and Keys
- Host definitions
- Include/exclude pattern
- Action definitions (1)
- Action definitions (2)

[Bootstrapping the Databases](#)

[URLs and References](#)

# Csync2 Configuration



# Configs, Groups and Keys

[Introduction](#)

[Csync2 Overview](#)

[Csync2 Configuration](#)

● **Configs, Groups and Keys**

● Host definitions

● Include/exclude pattern

● Action definitions (1)

● Action definitions (2)

[Bootstrapping the Databases](#)

[URLs and References](#)

- Multiple configurations can exist in parallel:  
`/etc/csync2.cfg, /etc/csync2_foobar.cfg, ...`
- A configuration can be selected using the `-C` option.
- A configuration contains of multiple groups.
- A group(-list) can be selected using the `-G` option.

```
group gallien {
    host asterix obelix;
    key /etc/csync2.key_gallien;
}
group entenhausen {
    host dagobert donald daisy duesentrieb;
    key /etc/csync2.key_entenhausen;
}
```

- Every group must have a key-file (shared secret).
- This key-file can be generated with `csync2 -k keyfile`

[Introduction](#)[Csync2 Overview](#)[Csync2 Configuration](#)[● Configs, Groups and Keys](#)[● Host definitions](#)[● Include/exclude pattern](#)[● Action definitions \(1\)](#)[● Action definitions \(2\)](#)[Bootstrapping the Databases](#)[URLs and References](#)

- Every group has a host list assigned.
- Groups which do not have the local hostname in them are ignored.
- The hostname may be different from the interface name.
- .. e.g. if crossover cables are used.
- This must be written as `host-name@interface-name`.

```
group gallien {  
    host asterix@asterix-x obelix@obelix-x;  
    key /etc/csync2.key_gallien;  
}
```

[Introduction](#)[Csync2 Overview](#)[Csync2 Configuration](#)

● Configs, Groups and Keys

● Host definitions

● **Include/exclude pattern**

● Action definitions (1)

● Action definitions (2)

[Bootstrapping the Databases](#)[URLs and References](#)

- Include/exclude pattern are used to specify which files should be synced.
- Pattern can be specified for full path names and basenames.
- The full path name pattern list and the basename pattern list must accept (include) the file. The last matching pattern for each of both lists is chosen.

```
group gallien {
    host asterix@asterix-x obelix@obelix-x;
    key /etc/csync2.key_gallien;

    include /home/bob;
    exclude /home/bob/temp;
    include *.txt
    exclude ~*;
}
```

# Action definitions (1)

[Introduction](#)

[Csync2 Overview](#)

[Csync2 Configuration](#)

● [Configs, Groups and Keys](#)

● [Host definitions](#)

● [Include/exclude pattern](#)

● [Action definitions \(1\)](#)

● [Action definitions \(2\)](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

- It's often useful to define commands to be executed when a group of files has been updated.
- The output is written to `/dev/null` unless a logfile is specified.
- It's also possible to let `csync2` execute the command locally when a change has been detected.
- The command is only executed once - also if multiple files have been updated.

# Action definitions (2)

[Introduction](#)[Csync2 Overview](#)[Csync2 Configuration](#)

- Configs, Groups and Keys
- Host definitions
- Include/exclude pattern
- Action definitions (1)
- **Action definitions (2)**

[Bootstrapping the Databases](#)[URLs and References](#)

```
group gallien {
    host asterix@asterix-x obelix@obelix-x;
    key /etc/csync2.key_gallien;

    include /etc/apache;

    action {
        pattern /etc/apache/httpd.conf;
        pattern /etc/apache/sites-available/*;
        exec "/usr/sbin/apache2ctl graceful";
        logfile "/var/log/csync2_action.log";
        do-local;
    }
}
```

[Introduction](#)

[Csync2 Overview](#)

[Csync2 Configuration](#)

**[Bootstrapping the Databases](#)**

- The boring default way
- Fast Status-DB Creation
- Bootstrapping Procedures

[URLs and References](#)

# Bootstrapping the Databases

# The boring default way

[Introduction](#)

[Csync2 Overview](#)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

● **The boring default way**

● Fast Status-DB Creation

● Bootstrapping Procedures

[URLs and References](#)

- Csync2 and config installed on all hosts
- `csync2 -cr /` marks everything as dirty (added)
- `csync2 -u` want's to sync everything to every peer host
- `..` and then discards all dirty entries after detecting that the files are already in sync.

- Number of syncs for  $n$  files on  $m$  hosts in the worst case:

$$n \frac{m(m-1)}{2}$$

- This is much wasted time for doing nothing if the hosts have been already in sync.

[Introduction](#)[Csync2 Overview](#)[Csync2 Configuration](#)[Bootstrapping the Databases](#)[● The boring default way](#)[● Fast Status-DB Creation](#)[● Bootstrapping Procedures](#)[URLs and References](#)

- `csync2 -cIr` / just updates the local database without adding anything to the list of dirty files.
- This is great for initially creating the databases if the hosts are already in sync.
- But: If the hosts have different files, noone will recognise.
- So `csync2 -T myname peername` can check if two hosts are in sync.
- And `csync2 -TI myname peername` marks files as dirty.
- Note that `-TI` can impossibly detect deletions or moves.
- It's possible to cycle thru all hosts with `-IT` to check the entire cluster.



[Introduction](#)[Csync2 Overview](#)[Csync2 Configuration](#)[Bootstrapping the Databases](#)[● The boring default way](#)[● Fast Status-DB Creation](#)[● \*\*Bootstrapping Procedures\*\*](#)[URLs and References](#)

- For small groups, just run `csync2 -cr /` and `csync2 -u` on all hosts.
  
- For big groups this would take too much time, so:
  - ◆ Manually sync all nodes (e.g. using `rsync`)
  - ◆ Run `csync2 -cIr /` on all nodes
  - ◆ Compare the nodes with `csync2 -T`
  
- For big groups where hard-syncing is impossible:
  - ◆ Run `csync2 -cIr /` on all nodes
  - ◆ Find and resolve conflicts with `-TI` and `-m`

[Introduction](#)

[Csync2 Overview](#)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

**[URLs and References](#)**

- [Related Projects](#)
- [Credits](#)

# URLs and References

[Introduction](#)

[Csync2 Overview](#)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

● [Related Projects](#)

● [Credits](#)

- **librsync**: A library for rsync-like binary diffing  
<http://librsync.sourceforge.net/>
  
- **sqlite**: A small embeddable SQL database engine  
<http://www.sqlite.org/>
  
  
  
  
  
  
  
  
  
  
- **rsync**: popular file synchronisation tool  
<http://rsync.samba.org/>
  
  
  
  
  
  
  
  
  
  
- **unison**: another file synchronisation tool  
<http://www.cis.upenn.edu/~bcpierce/unison/>

[Introduction](#)

[Csync2 Overview](#)

[Csync2 Configuration](#)

[Bootstrapping the Databases](#)

[URLs and References](#)

● [Related Projects](#)

● [Credits](#)

- LINBIT Information Technologies GmbH:  
<http://www.linbit.com/>

- Clifford Wolf:  
<http://www.clifford.at/>

<http://oss.linbit.com/>