

DxSpider update

```
PD9Q de PI1LAP-1 11-Feb-2023 0736Z dxspider >  
check_build  
There is no new build (497 378cbf24)  
PD9Q de PI1LAP-1 11-Feb-2023 0736Z dxspider >
```

Kin EA3CV wrote a fantastic script for updating DxSpider.

Kin Wrote :

If there is a new build, it will update and immediately restart the node.

If there are no changes, the sysop is informed.

To install it, copy the file to **/spider/local_cmd**

From **console.pl** run **load/cmd** the first time.

The **check_build** command can be used like this:

check_build Y	With Backup
check_build N	Without Backup
check_build	Without Backup

If you want to leave it automated, add in **/spider/local_cmd/crontab**

something like this:

```
0 4 * * 1,2,3,4,5 run_cmd("check_build N")
```

It can be downloaded from:

https://www.dropbox.com/s/6ttq5125qrzev67/check_build.pl?dl=0

This utility is not intended to replace the current update procedure,

it is intended for sysops who feel insecure about upgrading their cluster.

(I don't feel insecure when it comes to updating software. I like it when it's easier.)

Basically it does:

```
git reset --hard
git pull
reinicio
```

and a full backup of the `/home/spysop/spider` directory in `/home/sysop/backup`, which will be overwritten each time `check_build` is run.

```
# You need the following package:
# apt install libfile-copy-recursive-perl or
# cpan install File::Copy::Recursive
# Kin EA3CV, ea3cv@cronux.net
#
# 20230201 v1.0
#

use 5.10.1;
use DXDebug;
use File::Copy::Recursive;
use strict;
use warnings;

my $self = shift;
my $backup = shift;

return (1) unless $self->priv >= 9;

if ($backup eq "Y") {
    my $from_dir = "/home/sysop/spider";
    my $to_dir = "/home/sysop/backup";

    if ( !-d $to_dir ) {
        system('mkdir', $to_dir);
    }

    File::Copy::Recursive::rcopy_glob($from_dir, $to_dir);
}

system('cd /spider');

my $act = $main::gitversion;
```

```
$act =~ s/[r\\]//g;
my $res;
my $line = "cd /spider; git reset --hard; git pull";
my @out;

system('git reset --hard');

my $data = qx(git pull);
my $new = `git log --oneline | head -n 1`;

if ($data =~ /Updating/) {
    $res = "There is a new build: $new";
    dbg("DXCron::spawn: $res: $line") if isdbg('cron');
    push @out, $res;
    DXCron::run_cmd("shut");
} elsif ($data =~ /Already|actualizado/) {
    $res = "There is no new build ($main::build $act)";
    push @out, $res;
    dbg("DXCron::spawn: $res.") if isdbg('cron');
}

return (1, @out)
```