

# rsyncをWindow 10で使う

rsyncを Windows 10 で使いたい。

リモートサーバーとローカルを同期させるのならWinSCPで十分だ。だがWordPressのようにファイル数が多いと同期に時間がかかるrsyncなら圧縮オプションもあって転送時間が短くなる・・・と期待してやってみた。

SSHを利用してrsyncでファイル同期

<https://monoworks.co.jp/post/2018-07-31-rsync-with-ssh/>

SSHの公開鍵認証の設定とrsyncの設定まとめ

<https://2inc.org/blog/2012/06/26/1667/>

## cwRsyncのダウンロード

残念ながらrsyncはWindowsに対応していない。Itefix Consultingが、cwRsync (cygwin + rsync)という製品を出しているが、1年間のアップグレード + サポート付きで\$19という価格は購入をためらわせるものがある。

過去にはFree Editionがあったが、公開をやめてしまったようだ。

### cwRsync Free Edition 5.4.1

CNETに過去のFree Versionがミラーされていた (5.4.1, 2014/10/16) ので、これを使わせてもらう。内蔵されているrsyncのバージョンは3.1.1だ。

ダウンロードしたZIPファイルを展開する。

2011/04/02	17:00	1,756	cwrsync.cmd
2014/08/07	18:33	1,816,576	cygcrypto-1.0.0.dll
2014/08/16	10:24	103,975	cyggcc_s-1.dll
2011/10/17	02:20	1,008,654	cygiconv-2.dll
2014/08/16	10:29	12,839	cygssp-0.dll
2014/08/14	06:04	3,197,390	cygwin1.dll
2013/05/10	06:21	74,269	cygz.dll
2014/10/17	06:19	1,108	README.cwrsync.txt
2008/04/16	00:12	4,170	README.rsync.txt
2014/06/25	05:31	439,310	rsync.exe
2014/10/17	05:59	0	rsync.html
2014/10/17	05:59	0	rsyncd.conf.html
2014/10/08	07:04	446,488	ssh-keygen.exe
2014/10/08	07:04	684,056	ssh.exe
14個のファイル		7,790,591	バイト

このうち\*.exeと\*.dllをPATHの通っているフォルダにコピーしておく。

## cwRsync Free Edition 5.5.0

archive.orgにFree Versionが残っていた。

[5.5.0, 2015/12/27](#)のrsyncのバージョンは3.1.2だ。おそらくこれが、最終のFree Versionだろう。

2019/07/17	00:00	<DIR>	bin
2015/12/27	21:08		1,679 cwrsrcsync.cmd
2015/12/27	20:48		1,108 README.cwrsrcsync.txt
2008/04/16	00:12		4,170 README.rsync.txt
3個のファイル		6,957	バイト

binフォルダの内容は

2015/12/19	06:29	2,028,061	cygcrypto-1.0.0.dll
2015/12/19	06:29	110,109	cyggcc_s-1.dll
2015/12/19	06:29	1,034,269	cygiconv-2.dll
2015/12/19	06:29	42,013	cygintl-8.dll
2015/12/27	18:02	41,501	cygpopt-0.dll
2015/07/03	06:10	12,829	cygssp-0.dll
2015/11/15	07:45	3,488,471	cygwin1.dll
2015/12/19	06:29	84,519	cygz.dll
2015/12/27	17:05	429,582	rsync.exe
2015/08/22	20:20	413,710	ssh-keygen.exe
2015/08/22	20:20	694,286	ssh.exe
11個のファイル		8,379,350	バイト

このbinフォルダの内容をPATHの通っているフォルダにコピーしておく。

## .sshフォルダの準備

Windows 10のOpenSSHは、C:\Users\<ユーザー名>\.ssh に鍵を保存して参照する。

cwRsyncの場合には、/home/<ユーザー名>/.ssh/ を参照する。これは C:\home\<ユーザー名>\.ssh だ。

5.4.1では、環境変数HOMEが定義されていれば、%HOME%\.ssh を参照してくれた（例：HOME=D:\であれば、D:\.ssh を参照した）。だが、5.5.0では%HOME%の存在は無視されてしまうようだ

できればWindowsのOpenSSHと鍵を共有できた方がよい（ような気がする）ので、

C:\Users\<ユーザー名>\.ssh

C:\home\<ユーザー名>\.ssh

この二つが同一になるようにしたい。そこで、ジャンクションを作成する。

```
mklink /j C:\home C:\Users
```

これでC:\にhomeというジャンクションができた（ちなみに、このジャンクションを削除するには rmdir home ）

## Windowsのシンボリックリンクとジャンクションとハードリンクの違い

<https://www.atmarkit.co.jp/ait/articles/1306/07/news111.html>



環境変数CWRSYNCHOMEはどちらのバージョンでも参照していないようだ。

## 鍵の生成と転送

### 鍵の生成

ECDSA 384bitで鍵を生成する。

```
ssh-keygen -t ecdsa -b 384 -C "Comment"
```

Enter file in which to save the key (C:\Users\<username>\.ssh/id\_ecdsa):  
というのは、鍵の保存先（ホームの下の.sshフォルダ）を尋ねてきているが、変更の必要が無ければEnterで。

Enter passphrase (empty for no passphrase):  
パスフレーズを聞いてきているがrsyncを実行するたびにパスフレーズを入力するのが面倒なので、省略することにして、そのままEnter

ホームの下.sshフォルダに id\_ecdsa と id\_ecdsa.pub の二つのファイルができているはずだ（秘密鍵と公開鍵）。

-C “Comment” は鍵の識別用のコメントなので、自分のメールアドレスなどの連絡先、自分で識別できる文字列、あるいは空欄にすると良いようだ      -C “” とか。

### 公開鍵をリモートサーバーに転送

scpを使うんだったら、

```
scp %HOMEDRIVE%%HOMEPATH%\.ssh\id_ecdsa.pub username@ftp.server.ne.jp:
```

あるいはFTPで転送する。  
（リモートサーバーのホームディレクトリに転送したことにして進める）。

リモートサーバーにSSHでログインして.sshフォルダがなければ作っておく）

```
cat id_ecdsa.pub >> .ssh/authorized_keys
```

セキュリティのために、他のユーザーから見えないようにパーミッションを変えておく。

```
chmod 600 .ssh/authorized_keys  
chmod 700 .ssh
```

アップロードした id\_ecdsa.pub は削除しておこう。

rm id\_ecdsa.pub

Gowをインストールしている環境では scp がエラーになってしまった。

Gowではscpコマンドはscp.batというバッチファイルによってpscp.exeにリダイレクトされているが、そのあたりが良くないのかもしれない。



Windows 10の新しいバージョンではOpenSSHが標準でサポートされているのでSCPコマンドも標準で実装されている。

C:\Program Files (x86)\Gow\bin から

scp.bat sftp.bat ssh.bat

を削除してしまえば%SystemRoot%\System32\OpenSSH\ の

scp.exe sftp.exe ssh.exe

が実行されるようになる（荒業）。

## 秘密鍵のパーミッション変更

このままだとrsyncを実行したときに

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@          WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0770 for '/home/username/.ssh/id_ecdsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/home/username/.ssh/id_ecdsa": bad permissions
```

というエラーが出てしまうUNIX環境なら単に `chmod 600 ~/.ssh/id_ecdsa` でオッケーなのだがWindows環境だとちょっと面倒な手順がいる（cygwinでchmodしてもダメだった）。

### cwrsync permissions on private key

<https://stackoverflow.com/questions/26186429/cwrsync-permissions-on-private-key>

chmodにはrsyncコマンドそのものを使う。

```
pushd %HOMEDRIVE%%HOMEPATH%\ssh
rsync --chmod=700 id_ecdsa id_ecdsa.bak
move id_ecdsa.bak id_ecdsa
popd
```

## Windows 10のOpenSSHでも使えるように

ここまでの設定で、cwRsyncのrsyncや同梱のsshは使えるようになったはず。だがWindows 10のOpenSSHを使おうとすると・・・

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@          WARNING: UNPROTECTED PRIVATE KEY FILE!          @
```

```
Permissions for 'C:\\Users\\username\\.ssh\\id_ecdsa' are too open.  
It is required that your private key files are NOT accessible by others.  
This private key will be ignored.
```

とこの症状が再現してしまう。こちらは秘密鍵ファイルのACLの変更をしなければならない。

### Windows SSH: Permissions for 'private-key' are too open\

<https://superuser.com/questions/1296024/windows-ssh-permissions-for-private-key-are-too-open>

1. id\_ecdsaファイルを右クリックして、「プロパティ」を選択。
2. 「セキュリティ」タブを開く。
3. 「グループまたはユーザー名」をチェックする（自分のユーザー名とSYSTEM以外に、EveryoneとかUsersとかAdministratorsとか「なし」がリストに入っていたら、それを削除しなくてはならない）。
4. 「編集(E)」をクリック。
5. 自分のユーザー名とSYSTEM以外のエントリを、一つ一つクリックしては、「削除(R)」で削除していく。
6. SYSTEMは「特殊なアクセス許可」のみチェックされていることを確認。
7. 自分は変更、読み取りと実行、読み取り、書き込みにひととおりチェックが入っているのを確認。
8. [OK]で閉じる。

もし、所有者が自分以外だったり、継承が有効になっているとうまくいかない。

```
icacls %HOMEDRIVE%%HOMEPATH%\.ssh\id_ecdsa /inheritance:r  
icacls %HOMEDRIVE%%HOMEPATH%\.ssh\id_ecdsa /grant:r "%username%": "(R)"
```

を実行してから、セキュリティタブを開いて再度行ってみる。

参考までにコマンドライン版。

```
pushd %HOMEDRIVE%%HOMEPATH%\.ssh  
:: Remove Inheritance ::  
cmd /c icacls id_ecdsa /c /t /inheritance:d  
:: Set Ownership to Owner ::  
cmd /c icacls id_ecdsa /c /t /grant %username%:F  
:: Remove All Users, except for Owner ::  
cmd /c icacls id_ecdsa /c /t /remove Administrator "Authenticated Users"  
BUILTIN\Administrators BUILTIN\Everyone System Users  
:: Verify ::  
cmd /c icacls id_ecdsa  
popd
```

## rsyncの実行

rsyncコマンドでファイルやディレクトリの同期を行う

<https://onoredekaiketsu.com/rsync-command/>

rsync コマンド (その1) ファイルやディレクトリを同期する

<https://www.atmarkit.co.jp/ait/articles/1702/02/news031.html>

rsync -arv /dir/from/ /dir/to/ □fromをtoに同期させる)

rsync -arv /dir/from /dir/to/ □同期先に/dir/to/fromというディレクトリが作成される)

-a, --archive	アーカイブモード(-rlptgoD オプションと同義)
-r, --recursive	ディレクトリで再帰的に実行する
-l, --links	ソフトリンクを維持する
-p, --perms	パーミッションを維持する
-t, --times	タイムスタンプを維持する
-g, --group	グループを維持する
-o, --owner	オーナーを維持する (root のみ)
-D, --devices	デバイスを維持する (root のみ)
-u, --update	同期先のファイルの方が新しい場合はスキップする
--delete	同期元にはないファイルを同期先から削除する
-n, --dry-run	実際の同期はせずに実行時の動作だけを表示□DryRun□
-z, --compress	転送中のデータを圧縮する
-c, --checksum	更新日とサイズではなく、チェックサムで変更の有無をチェックする
-v, --verbose	動作内容を表示する



-n をつけて十分テストを行ってから、本番に挑もう！  
-z でデータを圧縮転送できる。



転送元や転送先のフォルダを指定する場合、リモートはusername@ftp.server.ne.jp:folder/subfolderという指定になる。コロンが(:) リモートのサーバー名とフォルダのセパレータとして使われているので□Windowsのドライブ名指定のコロンが使えない。  
そこで、ローカルフォルダについてはC:\folderではなく、/cygdrive/c/folderというcygwinスタイルのフォルダ指定を行う必要がある。



cwRsyncにはcygwinのDLLが含まれているため□cygwin□やGow□をインストールしていなくても動作する。

From:

<https://ragi.mokuren.ne.jp/> - らぎめも

Permanent link:

<https://ragi.mokuren.ne.jp/windows/rsync?rev=1563380564>

Last update: **2019/07/18**

