

# Alibaba Cloud CDN

Tools

# Tools

## CDN Log Merge Tool

This tool allows you to conveniently download all-day log files of a CDN domain by specifying the CDN domain, log date, and download path

### Tool Download

- For Windows: [cdnlog-tool-win.zip](#)
- For Linux: [cdnlog-tool-linux.zip](#)

### Installation Instructions

#### For Windows

- Environment requirements

Windows XP, Windows 7, or Windows 8

Installation and use

Download the installation package

- Double-click the installation file and click "Next" successively



After the installation is complete, click the "cdnlog" icon on the desktop to start the program

Example of use

Start the program to enter the login page, enter the "Access Key" and "Access Key Secret" (Immediately Obtaining Access Key Secret), and click "Login"



- Select a domain name, download start/end date, and download path, and then click "Start Download"



- After download is complete, the downloaded and merged logs can be viewed in the local download path

## For Linux

- Environment requirements

All mainstream Linux versions are supported

### Parameter description

```

$./cdnlogdownloader -h
Usage of ./cdnlogdownloader:
-date="": log date as YYYY-MM-DD
-domain="": the domain name want to download log
-host="cdn.aliyuncs.com": server to fetch log
-port=80: server port
-id="": Access Key Id
-secret="": Access Key Secret

```

### Example of use

```

./cdnlogdownloader -domain="www.leonwongphoto.com" -date="2015-07-27
2015-07-28" -id="rHIY87bG5NAAj7gW" -secret="0ucYoSsKTnqMj5tCnXV5xAcCz3dA64"

```

## Sample Authentication Code Overview

For URL authentication rules, see [URL Authentication Document](#). Through the demo, you can easily perform URL authentication. The following Python demo contains three authentication methods: Method A, Method B and Method C, and describes the composition of requested URLs and hash strings for each of the methods

## Python Demo

```
import re
import time
import hashlib
import datetime

def md5sum(src):
    m = hashlib.md5()
    m.update(src)
    return m.hexdigest()

def a_auth(uri, key, exp):
    p = re.compile("^(http://|https://)?(?:[^\?/]+)/(?:[^\?]*)?(?:\?.*)?$")
    if not p:
        return None
    m = p.match(uri)
    scheme, host, path, args = m.groups()
    if not scheme: scheme = "http://"
    if not path: path = "/"
    if not args: args = ""
    rand = "0" # "0" by default, other value is ok
    uid = "0" # "0" by default, other value is ok

    sstring = "%s-%s-%s-%s-%s" %(path, exp, rand, uid, key)
    hashvalue = md5sum(sstring)
    auth_key = "%s-%s-%s-%s" %(exp, rand, uid, hashvalue)

    if args:
        return "%s%s%s%s&auth_key=%s" %(scheme, host, path, args, auth_key)
    else:
        return "%s%s%s%s?auth_key=%s" %(scheme, host, path, args, auth_key)

def b_auth(uri, key, exp):
    p = re.compile("^(http://|https://)?(?:[^\?/]+)/(?:[^\?]*)?(?:\?.*)?$")
    if not p:
        return None
    m = p.match(uri)
    scheme, host, path, args = m.groups()
    if not scheme: scheme = "http://"
    if not path: path = "/"
    if not args: args = ""

    # convert unix timestamp to "YYmmDDHHMM" format
    nexptime = datetime.datetime.fromtimestamp(exp).strftime('%Y%m%d%H%M')
    sstring = key + nexptime + path
    hashvalue = md5sum(sstring)
```

```
    return "%s%s/%s/%s%s%s" %(scheme, host, nexp, hashvalue, path, args)

def c_auth(uri, key, exp):
    p = re.compile("^(http://|https://)?([^\/?]+)(/[^?]*)?(\\?.*)?$")
    if not p:
        return None
    m = p.match(uri)
    scheme, host, path, args = m.groups()
    if not scheme: scheme = "http://"
    if not path: path = "/"
    if not args: args = ""

    hexexp = "%x" %exp
    sstring = key + path + hexexp
    hashvalue = md5sum(sstring)

    return "%s%s/%s/%s%s%s" %(scheme, host, hashvalue, hexexp, path, args)

def main():

    uri = "http://xc.cdnpe.com/ping?foo=bar"      # original uri
    key = "<input private key>"                 # private key of authorization
    exp = int(time.time()) + 1 * 3600           # expiration time: 1 hour after current itme

    authuri = a_auth(uri, key, exp)             # auth type: a_auth / b_auth / c_auth

    print("URL : %s\nAUTH: %s" %(uri, authuri))

if __name__ == "__main__":
    main()
```