

## AdiIRC - \$totp - # 6

Added in 3.3

**\$totp(<key>, [time], [hash], [digits], [timestep], [encoding])**

Returns a TOTP (Time-based One-time Password) based on the specified parameters.

*TOTP is designed for hashes no shorter than 160 bits, so using md5 hash is not secure enough and should never be used with \$totp.*

See also [\\$hotp](#), [\\$hmac](#).

### Parameters

key - The key to hash. (Auto-detected between text/hex/base32 as described below)

time - A [\\$ctime](#) timestamp to indicate current start time. (default is [\\$ctime](#)) valid range 0-2^64-1

hash - Hash method to hash the key with. (sha1, sha256, sha384, sha512, md5, sha1 is default)

digits - Number of digits to return. (3 - 10, default is 6) **(AdiIRC only: digits=0 returns entire internal HMAC string)**

timestep - Duration of the time interval. (default is 30) valid range = 1-3600

encoding - Sets encoding method for 'key'. (t = UTF8 plain text, x = hex, a = base32) **(AdiIRC only)**

Default when 'encoding' is not used, attempts to support 'key' in several formats as follows:

if (key length excluding spaces is any of lengths 40|64|128 and \$remove(key,\$chr(32)) is hex ) encoding = x

if (key length excluding spaces is any of lengths 16|26|32 and \$remove(key,\$chr(32)) is base32) encoding = a

These assume hex keys of 160|256|512 bits, or base32 keys of 80|128|160 bits. all other cases: encoding = t

encoding 'x' or 'a' ignore all spaces padding, but 't' does not. All encoding formats reject key being \$null or entirely consisting of spaces.

Note: definition of base32 is case-insensitive [a-zA-Z2-7] after removing spaces, and '=' padding is NOT allowed

Note: It's recommended that the secret 'key' contain entropy no less than the bit length of the 'hash' used. Also, hash blocklength is 64 except for sha512|sha384 having 128. Key is shortened to hash(key) if key is longer than 'blocklength'. i.e. Using key longer than 512 bits with hash=sha1 shortens key to 160 bits:

```
//var -s %key $regsubex(foo,$str(x,65),/x/g,$rand(a,z)) | echo -a $totp(%key,123) same as $totp($sha1(%key),123)
(Equivalence is due to the length 40 string seen as encoding=x)
```

If interval longer than 1 hour is needed, must use timestep=1 with altered time value. i.e. for 9-digit daily code which uses sha256 and changes each day at 6pm. Change the 18 if the rollover hour-of-day should be different:

```
//var %interval $calc( ($ctime - $timezone -3600*18) // 86400) | echo -a $totp(key,%interval,sha256,9,1,t)
```