SNTPC - A Simple Network Time Protocol Client for MS-DOS

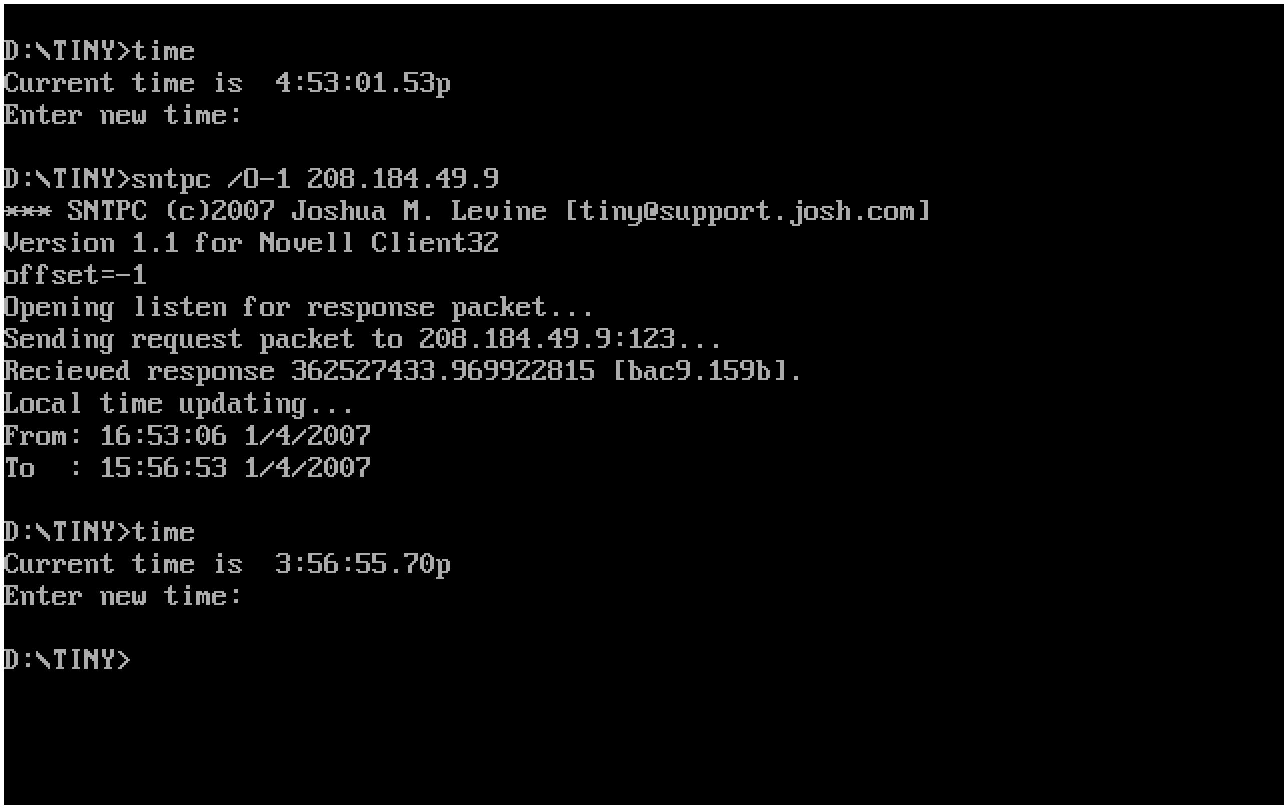
Overview

SNTPC is a very simple command-line utility that will let you sync the clock on a MS-DOS to a remote time server on the internet. There are lots of NTP clients in the world, but I think this is the only one available for MS-DOS machines.

Download

Click here to download SNTPC.ZIP. Note this version is compiled for use with the Novell TCPIP stack.

Screenshot



```
D:\TINY>time
Current time is  4:53:01.53p
Enter new time:

D:\TINY>sntpc /O-1 208.184.49.9
*** SNTPC (c)2007 Joshua M. Levine [tiny@support.josh.com]
Version 1.1 for Novell Client32
offset=-1
Opening listen for response packet...
Sending request packet to 208.184.49.9:123...
Recieved response 362527433.969922815 [bac9.159b].
Local time updating...
From: 16:53:06 1/4/2007
To  : 15:56:53 1/4/2007

D:\TINY>time
Current time is  3:56:55.70p
Enter new time:

D:\TINY>
```

Usage

```
SNTPC [/Ooffset] [/Pport] [/Wwait] [/Ccount] server_address


Where:

     server_address is the TCP/IP address of an SNTP or NTP time server
     (server must be NTP version 3 or higher)


     offset is offset hours to be added/subtracted from received time


     port is the optional TCP/IP port to send to
     (port 123 is used if none specified)
     wait is the optional time in TICKS to delay between requests
     (default wait is 18 ticks which is about 1 second)


     count is the optional number of requests to send before aborting
     (default count is 5 tries)

Returns:

     0=Local clock successfully set
     1=Timeout waiting for response from server (time not set)
     2=Other error (time not set)
```

Example

```
SNTPC /W24 /C5 /O-6 216.254.1.23
```

Will attempt to set the local clock using the server at 216.254.1.23. It will make a maximum of 5 requests, and will wait up to 2 seconds after each request for a response before sending the next request. It will subtract 6 hours from the time returned from the server.

FAQ

Q: How accurate is SNTPC?
A: Not very. SNTPC is very simple- it makes a request to the server and then uses what ever time it gets in the response. It does not even attempt to compute round trip delays or anything like that. Furthermore, the DOS function that sets the local time only has a resolution to one second, so the time set can be off by a maximum of one second plus the network delay between the time server and the local machine.

Q: How can I find a time server to sync to?
A: You can use any NTP sever. You can find databases of public time servers using google. There is also a list of NIST-maintained time servers here.

Q: Which time server should I use?
A: The closer, the better since this will usually have lower delays. Also, it is handy to use a time server that is in your local time zone so you can avoid needing to specify an offset.

Q: Can I use SNTPC to sync the clock of my DOS machine to the clock on my Windows machine?

A: Sure. You'll need to be running an SNTP server on the Windows machine. There are many SNTP servers available for Windows, here is a free one...

http://www.ee.udel.edu/~mills/ntp/html/build/hints/winnt.html

Once you have the SNTP server running on the Windows machine, make sure the DOS machine can ping the Windows machine. If it can, you should be able to just run the SNTPC.EXE program on the DOS machine giving it the IP address of the Windows machine and it should get the time from the Windows machine.

Q: Can I specify the name of the time server rather than the IP address?
A: Unfortunately, the DNS clients on the DOS tcpip stacks don't work so good, so you must specify the IP address explicitly.

Q: How do I know what to use for an offset?
A: The easiest way is to use SMTPC to set the local clock to an NTP server and then look at the time to see how far off it is from your local time. If the time set is an hour later than it really is now for you, use an offset of -1. Note that the offset can change due to daylight savings time changes in your zone or in the server's zone.

Q: I want to use SNTPC on my DOS machine, but don't have TCPIP.
A: There are a few popular TCPIP stacks for DOS. I like the Novell one mostly because it is freely available and works well. Check out the TINY page here to download the Novell stack. SMTPC uses my portable DOS network interface, so it would

be easy to recompile to work the the FTP Software PCTCP stack also.

Q: How does the return value work?
A: This is the value that the DOS *errorlevel* variable is set to when SNTPC finishes. You can check this value in a batch file. Here is an example of a DOS batch file that will attempt to set the time using one NTP server, check to see if it was successful, and if not try a second server...

```
@echo Trying NTP server in Gaithersburg, Maryland...
@SNTPC 192.168.15.1
@if not errorlevel 1 goto success
@echo Trying NTP server in Boulder, Colorado...
@SNTPC 132.163.4.101
@if not errorlevel 1 goto success
@echo ERROR,TIME NOT SET!!!!
@goto done
:success
@Echo Time set successfully!
:done
```

Q: What is the license for SNTPC?
A: You are free to use SNTPC in any non-commercial (educational, hobby, personal) application. If you wish to use SNTPC in a commercial application, please contact me first at the email address below. Full source code is available upon request.

Updates

4/1/2007 - First published

10/18/07 - Added link to Windows SNTP server in the FAQ.

Support

For support, please email me at...

sntpc@support.josh.com

###