



Setting Time On A Cisco Device (Manually)



Why Set The Time On A Cisco Device?

Probably the biggest reason is to be able to correlate events, especially log messages, across a network. If your customers tell you that the network dropped at 5 pm on Tuesday, you're going to want to be able to check your devices for any events that happened around that time. If your devices don't maintain a consistent time, then this task becomes difficult to nearly impossible.

NOTE: This lesson will not cover Network Time Protocol (NTP).



Displaying The Current

While **show time** would be the most logical command, Cisco IOS instead uses the **show clock** command to display the current (date)time:

```
a1#show clock
*00:43:03.791 UTC Mon Mar 1 1993
```

The time showing here is the amount of time (hh:mm:ss) since the device last booted up. The 01 March, 1993 date is less than helpful as well. Why 1993?

In the absence of better information, Cisco IOS devices will initially set their system clocks to epoch start, which will typically be midnight (UTC) March 1, 1993 or 2002.

show clock detail will give you some additional details:

```
a1#show clock detail
*00:46:36.571 UTC Mon Mar 1 1993
No time source
```



Displaying The Current

```
a1#show clock
```

```
*00:43:03.791 UTC Mon Mar 1 1993
```

Notice the asterisk at the beginning of the output? That actually gives us some good information about the time source:

Symbol	Meaning
*	Time is not authoritative: the software clock is not in sync or has never been set.
[blank]	Time is authoritative: the software clock is in sync or has just been set manually
.	Time is authoritative, but NTP is not synchronized: the software clock was in sync, but has since lost contact with all configured NTP servers



Setting The Time

You set the time from privileged EXEC mode with the **clock set** command:

```
a1#clock set ?
```

```
  hh:mm:ss  Current Time
```

```
a1#clock set 17:47:00 ?
```

```
  <1-31>  Day of the month
```

```
  MONTH  Month of the year
```

```
a1#clock set 17:47:00 25 October 2010 ?
```

```
  <cr>
```

```
a1#clock set 17:47:00 25 October 2010
```

```
a1#show clock
```

```
17:47:05.883 UTC Mon Oct 25 2010  <-Note that the asterisk is gone
```

```
a1#show clock detail
```

```
17:47:09.179 UTC Mon Oct 25 2010
```

```
Time source is user configuration
```



Displaying Logging/Debugs

Depending on your device configuration, you may need to configure **service timestamps** in order to see your logging (and/or debugging) in datetime format instead of uptime format.

```
a1#show clock
```

```
18:54:32.995 UTC Mon Oct 25 2010
```

```
a1#clear counters
```

```
Clear "show interface" counters on all interfaces [confirm]
```

```
02:08:30: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console
```

```
a1#conf t
```

```
a1(config)#service timestamps log ?
```

```
  datetime  Timestamp with date and time
```

```
  uptime    Timestamp with system uptime
```

```
<cr>
```

```
a1(config)#service timestamps log datetime
```

```
a1(config)#^Z
```

```
Oct 25 18:58:36: %SYS-5-CONFIG_I: Configured from console by console
```



Clock Persistence

Depending on your device, your clock may not persist after a reload:

```
a1#show clock
19:00:57.115 UTC Mon Oct 25 2010
a1#reload
```

```
a1#show clock detail
*00:01:14.567 UTC Mon Mar 1 1993
No time source
```

Newer platforms have a hardware clock (calendar) which is separate from the software clock. The hardware clock is battery operated, and runs continuously, even if the router is powered off or rebooted.

```
r1#show clock detail
*00:12:21.519 UTC Tue Oct 26 2010
Time source is hardware calendar
```

```
r1#show calendar
00:12:32 UTC Tue Oct 26 2010
```



Clock Persistence

On devices with a hardware clock, the user set clock survives reloads, but it will use the hardware clock setting and not the user configured setting:

```
r1#clock set 23:59:00 31 Dec 1999
r1#show clock detail
23:59:06.435 UTC Fri Dec 31 1999
Time source is user configuration
```

```
r1#show calendar
00:35:31 UTC Tue Oct 26 2010
```

```
r1#reload
```

```
r1#show clock detail
*00:37:58.779 UTC Tue Oct 26 2010
Time source is hardware calendar
```




Clock Persistence

To set the hardware clock to the user configured clock time use **calendar set** command:

```
r1#show calendar
```

```
00:40:59 UTC Tue Oct 26 2010
```

```
r1#calendar set 23:59:00 31 Dec 1999
```

```
r1#show calendar
```

```
23:59:05 UTC Fri Dec 31 1999
```

```
r1#show clock detail
```

```
*00:41:17.547 UTC Tue Oct 26 2010 <-Note
```

```
Time source is hardware calendar
```

```
r1#reload
```

```
r1#show clock detail
```

```
*00:07:42.603 UTC Sat Jan 1 2000
```

```
Time source is hardware calendar
```

```
r1#show calendar
```

```
00:07:53 UTC Sat Jan 1 2000
```



Clock Persistence

There are two other commands for manipulating the relationship between the hardware clock and the user configured clock on a Cisco device, and **clock read-calendar** and **clock update-calendar**

```
r1#clock ?
```

```
  read-calendar  Read the hardware calendar into the clock
  set            Set the time and date
  update-calendar Update the hardware calendar from the clock
```



Clock Persistence

clock read-calendar may be used to manually read the hardware clock setting into the software clock. This command is useful if the **calendar set** command has been used to change the setting of the hardware clock.

```
r1#show calendar
11:00:29 UTC Tue Oct 26 2010
r1#calendar set 23:59:00 31 Dec 1999
r1#show calendar
23:59:03 UTC Fri Dec 31 1999
r1#show clock detail
11:01:12.708 UTC Tue Oct 26 2010
Time source is hardware calendar

r1#clock read-calendar
r1#show clock detail
23:59:58.351 UTC Fri Dec 31 1999
Time source is hardware calendar
r1#show calendar
23:59:59.731 UTC Fri Dec 31 1999
```



Clock Persistence

clock update-calendar does exactly the opposite of the **clock read-calendar** command; it is used to manually read the software clock setting into the hardware clock.

```
r1#show clock detail
00:04:05.531 UTC Sat Jan 1 2000
Time source is hardware calendar
r1#clock set 11:07:00 26 October 2010
r1#show clock detail
11:07:08.267 UTC Tue Oct 26 2010
Time source is user configuration
r1#show calendar
00:04:35 UTC Sat Jan 1 2000

r1#clock update-calendar
r1#show calendar
11:07:26 UTC Tue Oct 26 2010
```



Clock Persistence

If you're using a platform without a hardware clock and you're not running ancient code, you can use the **clock save interval** global configuration command to maintain clock persistence after a reboot:

```
r3#show clock detail
*13:20:29.796 UTC Thu Feb 25 2010
No time source
r3#show calendar
      ^
% Invalid input detected at '^' marker.
r3(config)#clock save interval ?
  <8-24>  clock backup update in hours

r3(config)#clock save interval 8
r3#clock set 20:35:00 25 October 2010

r3#show clock detail
20:35:04.715 UTC Mon Oct 25 2010
Time source is user configuration
```



Clock Persistence

The benefit of using this command is that upon returning from a system reload or power cycle, the system clock will be set to a time and date near the current time and date instead of being reset to the system default time and date. In the absence of better information, Cisco IOS devices will initially set their system clocks to epoch start, which will typically be midnight (UTC) March 1, 1993 or 2002.

When this command is entered, the date and time are saved to NVRAM at the interval specified by this command, and also during any shutdown process. When the system starts up, the system clock is set to the last time and date saved to NVRAM.

```
r3#reload
```

```
r3#show clock detail
```

```
*20:36:08.682 UTC Mon Oct 25 2010
```

```
No time source
```



Clock Persistence

While the **clock save interval** global configuration command is a great option for devices without a hardware clock, it is important to know that there are some limitations.

Whenever you reload your device, the software clock is written to NVRAM. This means that your software clock will “slip” for the amount of time that it takes to reload. This isn’t a huge deal. When your device loses power...the software clock is not written to NVRAM. When the device restores, it will revert to the last software clock value stored in NVRAM. Depending on your clock save interval, this can be as old as 24 hours. Plus your clock still slips for the amount of time that the device is down.



Clock Persistence

```
r3#show clock detail
*12:18:24.812 UTC Tue Oct 26 2010
No time source
```

```
r3#show version | i uptime
r3 uptime is 15 hours, 44 minutes
```

```
r3#show run | i clock save
clock save interval 8
```

Let's power down the router for 30 minutes. The last clock save should have been around 7:45 ago. We should see clock lose that time + 30 minutes for down time.

After power on (about) 30 minutes later:

```
r3#show clock detail
*04:35:30.568 UTC Tue Oct 26 2010
No time source
```




Setting The Time Zone

By default, Cisco devices uses UTC(Coordinated Universal Time, formerly known as Greenwich Mean Time (GMT)). Unless you want to do the mental mathematics to convert UTC to your local time zone, you will want to change the clock to represent your specific time zone (and daylight savings time). Of course, if you have devices in multiple time zones, this makes troubleshooting more difficult and you still have to do the mental mathematics to translate the time on your San Francisco router to that on your London router. Generally you're going to want to either use UTC or standardize on a specific time zone (usually the one used locally by your network team) across your network.

To change the time zone displayed, use the **clock timezone** global configuration command :

```
r1#show clock
16:37:10.786 UTC Tue Oct 26 2010

r1(config)#clock timezone ?
WORD name of time zone
r1(config)#clock timezone CDT ?
<-23 - 23> Hours offset from UTC
r1(config)#clock timezone CDT -5

r1#show clock
11:41:08.702 CDT Tue Oct 26 2010
```



Setting The Time Zone

There are a couple of things to be aware of when setting the time zone. One, the command is going to take the current software and hardware clocks and offset them by the amount specified. So if you had set your calendar and/or clock to the correct local time, IOS is going to assume that value is actually UTC and change the values appropriately:

```
r1#show clock
16:37:10.786 UTC Tue Oct 26 2010 <-Correct CDT time
```

```
r1#show clock
11:41:08.702 CDT Tue Oct 26 2010 <-IOS assumed clock was UTC so CDT time is wrong after adjustment
```

Second, IOS does not verify that your time zone abbreviation is correct.

```
r1(config)#clock timezone POOP -2
```

```
r1#show clock
14:43:51.438 POOP Tue Oct 26 2010
r1#show calendar
14:45:03 POOP Tue Oct 26 2010
```

Third, IOS does not automatically adjust for Daylight Savings Time(DST), but there is a command for that.



Setting The Daylight Savings Time

While IOS does not automatically adjust for Daylight Savings Time(DLT), you can use the global configuration **clock summer-time** command. If the **clock summer-time zone recurring** command is specified without parameters, the summer time rules default to United States rules. Default of the *offset* argument is 60 (minutes).

```
r1(config)#clock timezone CST -6
r1(config)#clock summer-time ?
    WORD  name of time zone in summer
r1(config)#clock summer-time CDT ?
    date      Configure absolute summer time
    recurring  Configure recurring summer time
r1(config)#clock summer-time CDT recurring

r1#show clock
12:08:44.590 CDT Tue Oct 26 2010 <-Date within US DLT

r1#clock set 12:00:00 31 December 2010
r1#show clock
12:00:04.155 CST Fri Dec 31 2010 <-Date outside US DLT, so reverts to CST
```



Setting The Daylight Savings Time

If you live in an area that does not follow the US DLT rules, you can set up DLT to your local specifications (examples are from Cisco command reference):

The following example specifies that summer time starts on the first Sunday in April at 2 a.m. and ends on the last Sunday in October at 2 a.m.:

```
r1(config)#clock summer-time PDT recurring 1 Sunday April 2:00 last Sunday October 2:00
```

If you live in a place where summer time does not follow the pattern in the first example, you can specify the exact date and times. In the following example, daylight saving time (summer time) is configured to start on October 12, 1997 at 2 a.m., and end on April 26, 1998 at 2 a.m.:

```
r1(config)#clock summer-time PDT date 12 October 1997 2:00 26 April 1998 2:00
```



Summary

Cisco devices have a software clock that tracks the time. You'll want to have your devices keep accurate time across your network so that you can correlate events, especially log messages, across a network. If your devices don't maintain a consistent time, then this task becomes difficult to nearly impossible. The preferred method of maintaining consistent time across a network is with the Network Time Protocol (NTP). That method is not discussed in this lesson as it concentrates on "manual" time configuration.

Older Cisco platforms only maintain a software clock, which will not persist device reloads. The **clock save interval** global configuration command mitigates this liability to some extent by saving the clock to NVRAM (every 8 – 24 hours depending on your configuration) and prior to user reload requests. Your software clock will still lose accuracy during downtime and especially during unplanned (e.g. power outage) downtime, but will not reset to *epoch start* (typically midnight (UTC) March 1, 1993 or 2002) as it will without this command.

Newer Cisco platforms have a hardware clock that survives reloads and power outages.

To manually set the time, use the **clock set** privileged EXEC command. To view the time, use the **show clock** privileged EXEC command.