



! Here are some templates for Tcl scripts that you can use for ping and traceroute.

!
!
!
!
!

- Email info@configtoolbox.com if you have any questions.

Cisco Tcl (pronounced "tickle") is a powerful scripting language which can streamline configuration, troubleshooting, and routine tasks on a router or switch. It is supported on Cisco IOS, IOS-XE, IOS-XR, and NX-OS.

It is common to write a Tcl script in a standard ASCII editor (aka plain text file, .txt) and then paste it into your router or switch CLI, however you can also save the Tcl script to local flash for execution or a centralized server to execute over TFTP, FTP, or SCP. It can also be typed directly into CLI line by line, although that is not very efficient. Tcl requires you to enter a shell, input variables to run the script, see the results, then exit the shell.

A common Tcl script is a basic ping of multiple addresses. This can save you time since you can script it once and then run the script as often as needed. So, if you are needing to ping many addresses frequently, a Tcl script is the way to go. This is a nice feature to have for daily network administration to run a quick check on connectivity, or you could script traceroutes to verify your network paths, etc.

Here's an example script that will ping multiple addresses in Cisco IOS —

Let's first do a quick check to see if your router or switch supports Tcl. Login to your router and enter the following from privileged EXEC mode.

```
Router#tclsh <press enter>
```

You should see the prompt change once you press <enter>. If your IOS does not enter the Tcl shell (seen below) then your IOS version does not support Tcl.

```
Router(tcl)#
```

You can now start entering (or paste from ASCII file) commands for your script. Below are example IP addresses, of course —

```
foreach address {  
1.1.1.1  
1.1.1.2  
1.1.1.3  
1.1.1.4  
1.1.1.5  
2.1.1.1  
2.1.1.2  
2.1.1.3
```

```
2.1.1.4
2.1.1.5
} { ping $address
}
```

The above script will ping all ten addresses, display the results, and then bring you back to the 'Router(tcl)#' prompt. At this point you could run another script or type 'tclquit' <enter> to exit the Tcl shell. The word 'address' in the script above is arbitrary and can be named anything.

You could put the script in an ASCII file and save it to your local flash or nvram if you plan to run it frequently. Or keep it on a centralized server that runs TFTP, FTP, or SCP so you can have a single Tcl script to launch from many routers or switches.

As an example, let's say you have a TFTP server with address 1.1.1.50 and you placed a Tcl script on it named 'script.tcl' —

```
Router#tclsh tftp://1.1.1.50/script.tcl
```

Or you can copy 'script.tcl' to your local flash or nvram and execute —

```
Router#tclsh flash:script.tcl
```

or

```
Router#tclsh nvram:script.tcl
```

As another example, let's say you want a Tcl script that will ping multiple addresses with parameters —

```
Router#tclsh
Router(tcl)#foreach address {
1.1.1.1
1.1.1.2
1.1.1.3
1.1.1.4
1.1.1.5
2.1.1.1
2.1.1.2
2.1.1.3
2.1.1.4
2.1.1.5
} { ping $address repeat 50 size 100
}
```

Or maybe you want to traceroute to each address —

```
Router#tclsh
Router(tcl)#foreach address {
1.1.1.1
1.1.1.2
1.1.1.3
1.1.1.4
1.1.1.5
2.1.1.1
```

```
2.1.1.2
2.1.1.3
2.1.1.4
2.1.1.5
}{ traceroute $address
}
```

Remember that 'tclquit' will exit the Tcl shell —
Router(tcl)#tclquit

Let us know if you have any questions or need config guidance.
- Config Toolbox @ <https://configtoolbox.com/contact-us>

PRIVACY POLICY

Config Toolbox is mindful of your privacy. Please see our website for Disclaimer, Privacy Policy, and Terms and Conditions. (<https://configtoolbox.com/config-toolbox-blog>)

