

# The table Command: Subtables



spark, 2010-13-01

With version 10.1, we've given the session table some long-sought functionality, and revamped its iRules interface completely to give users a new, cleaner, full-featured way to keep track of global data. We are very proud to introduce the table command...

1. [Basics](#)
2. [New Ways To Set Data](#)
3. [New Ways To Alter Data](#)
4. [Data Expiration](#)
5. [Advanced Data Expiration](#)
6. **Subtables**
7. [Counting](#)
8. [The Fine Print](#)
9. [Examples](#)

In this sixth part of the series on the usage of the new table command, we discuss how you can create and manage subtables within the session table.

## Subtables

Whew! Still with me? Good, because we're just now getting to some of best stuff.

One of the most powerful changes to the session table in v10.1 is the concept of a *subtable*, which is simply a named set of entries. It lets you organize your entries in groups, and lets you act on those groups. A subtable is basically a table in and of itself, with some special abilities. Subtables only contain entries, though; you can't nest subtables. Here's what they look like:

## The session table

Key	Value	Timeout	Lifetime	touch time	create time
10.1.2.3	UserFoo	180	indefinite	1253830993	1253830993

  

Subtable: DNS					
Key	Value	Timeout	Lifetime	touch time	create time
User77	5254	3600	3600	1253830772	1253830719

  

Subtable: POST					
Key	Value	Timeout	Lifetime	touch time	create time
admin	allowed	30	100	1253830990	1253830910

They are trivial to use: every current table subcommand simply takes a **-subtable <name>** parameter. So you can do things like:

```
table set -subtable $tbl $key $data
table lookup -subtable $othertbl $key
table delete -subtable $thirdtbl $key
```

That's all there is to it! Doesn't look like much, does it? But this is very powerful stuff! Maybe you want a different set of data for each user, or for each client IP address, or for each virtual server? No problem!

```
table set -subtable "$username" $key $data
table add -subtable "[IP::client_addr]" $key $data
table replace -subtable "[virtual]" $key $data
```

Since you can name your subtables however you want, you can now have shared variables with any arbitrary scope! And there's an easy way to clean up:

```
table delete -all -subtable "$username"
```

will remove every entry in a subtable in one fell swoop.

### But wait! There's more!

With subtables comes a new and long-desired ability: the ability to list or count the number of keys:

```
table keys -subtable <name> [-count|-notouch]
```

You'll note that unlike all the other **table** subcommands, the subtable name is not optional here: there's no way to use it on entries that aren't in a subtable. Also note that by default, retrieving the list of keys will touch each one, but getting the count will never touch any entries.

Isn't this AWESOME!?! This feature is going to simplify a lot of your iRules, and give you the ability to do things that simply weren't feasible before. I can't wait to see all the creative uses you come up with!

That covers all of the new session table features. Now we'll cover some ways of putting everything together...

Continue reading part seven in our series: [Counting](#)

---

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](http://f5.com)

F5 Networks, Inc.  
Corporate Headquarters  
[info@f5.com](mailto:info@f5.com)

F5 Networks  
Asia-Pacific  
[apacinfo@f5.com](mailto:apacinfo@f5.com)

F5 Networks Ltd.  
Europe/Middle-East/Africa  
[emeainfo@f5.com](mailto:emeainfo@f5.com)

F5 Networks  
Japan K.K.  
[f5j-info@f5.com](mailto:f5j-info@f5.com)