



DataScript: Troubleshooting Rules

Avi Technical Reference (v16.3)

Copyright © 2018

DataScript: Troubleshooting Rules

[view online](#)

DataScripts may fail in one of the following two ways.

Configuration Error

All DataScripts are parsed during configuration save / edit. Incorrect DataScript syntax will result in a failure when attempting to commit the DataScript. The configuration time error is reported as an aid in debugging and will catch syntax errors, not logic errors.

The following DataScript has extra characters. Lua interprets this as a variable which has not yet been declared, such as `abc = 1`.

```
if avi.vs.client_ip() == "10.1.1.1" then
  abc
  avi.http.close_conn()
  def
end
ghi
```

```
Failed to compile the datascript "Drop_Client_on_http_req.ds": luac: Drop_Client_on_http_req.ds:3: '=' expected near 'av
```

There are three syntax errors in the rule above, but only the first error encountered will be reported. In this example, `abc` is an undeclared variable, but the determination that it is noticed until reading the next line. Hence, the error is reported on line 3, not line 2: `http_req.ds:3` and the error location is reported *near 'avi'*. DataScripts are very tolerant of spaces and carriage returns. The variable declaration `abc = 2` is valid.

```
abc
=
2
```

The following DataScript is missing an `end` to complete the `if` logic.

```
if avi.vs.client_ip() == "10.1.1.1" then avi.http.close_conn()
```

```
Failed to compile the datascript "Drop_Client_on_http_req.ds": luac: Drop_Client_on_http_req.ds:1: 'end' expected near '
```

Execution Error

Even though a DataScript may be saved without error, it may still have logic or other issues that may pop up later. When a DataScript execution encounters a failure, the script execution aborts for that request or response. An HTTP 500 Internal Server Error is sent to the client. A client log is generated with the stack trace of the aborted script to aid in debugging.

Use the following Virtual Service > Logs filter to search for client's that encountered a DataScript error.

```
significance="Request ended abnormally: Datascript failed to execute"
```

Open a client log and click *All Headers* to see the more info about the error. The following is an example of an error from the logs:

Datascript Error

```
[string "Drop Client"]:2: attempt to call field 'close_conn' (a nil value)
```

Datascript Error Stack Trace

```
stack traceback:
coroutine 0:
  [string "Drop Client"]:2: in function <[string "Drop Client"]:1>
```

The stack trace for this error may not provide the reason, but it does provide a useful hint. The error is in line two of the rule *"Drop Client"]:2* and has something to do with the *close_conn* function. The issue may still not be immediately obvious looking at the rule...

```
if avi.vs.client_ip() == "10.1.1.1" then
  avi.close_conn()
end
```

Searching the Avi documentation for more help on the *avi.close_conn* returns the [avi.http.close_conn](#) article. After changing the function to the correct name, test again and validate the issue is resolved.

In the above example, most clients would not be impacted by the broken rule. This same logic can be used to isolate an issue and test without impacting other users.