

v11: iRules Data Group Updates



Jason Rahm, 2012-11-01

Several months ago I wrote up the v10 formatting for internal and external datagroups: [iRules Data Group Formatting Rules](#). In v11, however, there is a change to the format of the internal data group and the data group reference to external class files (the formatting in the external class file itself is unchanged). The formatting rules in v11 for data groups more closely resembles the tmsh commands necessary to build the class at the CLI (these command attributes are masked if you are using the GUI). I'll follow the same format as the original write-up in showing the various data group types. The format is the same among internal data group types. If there is no value associated with the key, there is a curly bracket pair trailing the key on the same line. If there is an associated value with a key, the curly bracket opens the value, followed by a newline with the keyword data and the value, then another newline with the closing curly bracket. After the records are listed, the type is specified. For external data groups, the file name and the type are specified. If the filename is in /var/class, the path is omitted from the filename reference.

Address Data Groups

Internal Data Group

```
ltm data-group internal addr_testclass {
  records {
    192.168.1.1/32 { }
    192.168.1.2/32 {
      data "host 2"
    }
    192.168.2.0/24 { }
    192.168.3.0/24 {
      data "network 2"
    }
  }
  type ip
}
```

External Data Group

```
ltm data-group external addr_testclass_ext { external-file-name addr_testclass.class type ip }
```

Integer Data Groups

Internal Data Group

```
ltm data-group internal int_testclass {
  records {
    1 {
      data "test 1"
    }
    2 {
      data "test 2"
    }
  }
  type integer
}
```

```
}
```

External Data Group

```
ltm data-group external int_testclass_ext {  
    external-file-name int_testclass  
    type integer  
}
```

String Data Groups

Internal Data Group

```
ltm data-group internal str_testclass {  
    records {  
        str1 {  
            data "value 1"  
        }  
        str2 {  
            data "value 2"  
        }  
    }  
    type string  
}
```

External Data Group

```
ltm data-group external str_testclass_ext {  
    external-file-name str_testclass.class  
    type string  
}
```

External Datagroup File Management

Beginning in v11, external datagroups are imported into a local filestore rather than simply existing someplace on the file system (/config/filestore). However, this filestore is not meant to be edited manually. Please follow the steps below for creating or modifying external datagroups. Thanks to hoolio for the steps in this external datagroup section.

Create a New External Datagroup from the CLI Non-Interactively

1. Create or copy over to LTM a temporary file containing the external data group contents. If copying, make sure the line terminators are `\n` only not `\r\n`.

```
# cat /var/tmp/string_name_value_external_dg.txt  
"name1" := "value1",  
"name2" := "value2",  
"name3" := "value3",
```

2. Create the new external data group file

```
tmssh create /sys file data-group string_name_value_external_dg_file separator "://" source-path
file:/var/tmp/string_name_value_external_dg.txt type string
```

3. Create the external data group referencing the file

```
tmssh create /ltm data-group external string_name_value_external_dg external-file-name
string_name_value_external_dg_file
```

Modify the External Datagroup File for Existing Datagroup

1. Create a new temporary file containing the updated external data group contents

```
# cat /var/tmp/string_name_value_external_v2_dg.txt
"name1" := "valueA",
"name2" := "valueB",
"name3" := "valueC",
```

2. Import the new data group file

```
tmssh create /sys file data-group string_name_value_external_v2_dg_file separator "://" source-path
file:/var/tmp/string_name_value_external_v2_dg.txt type string
```

3. Modify the data group definition to reference the new external data group file

```
tmssh modify /ltm data-group external string_name_value_external_dg external-file-name
string_name_value_external_v2_dg_file
```

4. Delete the old data group file if it's unneeded

```
tmssh delete sys file data-group string_name_value_external_dg_file
```

Handling Line Terminator Discrepancies

Whether importing external datagroups in the GUI or from the CLI, the system does not accept files with `\r\n` line terminators, it only accepts `\n`. If you copy files over from windows, most likely you have the wrong terminator in your file format. To check, you can use the **od** command.

Datagroup Contents created in vi on LTM

```
[root@golgotha:Active] data_group_d # od -c /var/tmp/string_name_value_external_dg.txt
0000000  "  n  a  m  e  1  "      :  =      "  v  a  l  u
0000020  e  1  "  ,  \n  "  n  a  m  e  2  "      :  =
0000040  "  v  a  l  u  e  2  "  ,  \n  "  n  a  m  e  3
0000060  "      :  =      "  v  a  l  u  e  3  "  ,  \n
0000077
```

Datagroup Contents created in Notepad on Windows

```
[root@golgotha:Active] data_group_d # od -c /var/tmp/notepad_dg.txt
0000000  "  n  a  m  e  1  "      :  =      "  v  a  l  u
0000020  e  1  "  ,  \r  \n  "  n  a  m  e  2  "      :  =
0000040  "  v  a  l  u  e  2  "  ,  \r  \n  "  n  a  m
0000060  e  3  "      :  =      "  v  a  l  u  e  3  "  ,
0000100
```

If your line terminators are incorrect, you can use the tr command to remove the \r's.

```
[root@golgotha:Active] tmp # cat /var/tmp/notepad_dg.txt | tr -d '\r' > /var/tmp/notepad_dg_update.tx
[root@golgotha:Active] tmp # od -c /var/tmp/notepad_dg_update.txt
00000000 " n a m e 1 " : = " v a l u
0000020 e 1 " , \n " n a m e 2 " : =
0000040 " v a l u e 2 " , \n " n a m e 3
0000060 " : = " v a l u e 3 " ,
0000076
```

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com