

Automated Gomez Performance Monitoring Part 2 – DataCenter Identification



Joe Pruitt, 2010-07-06

In my previous article titled “Automated Gomez Performance Monitoring”, I gave an overview of Gomez and the partnership between F5 Networks and Gomez to assist users of both technologies to integrate the two together.

In that article, I illustrated how to use iRules to serve up the Gomez bootstrap JavaScript code eliminating the need to install in on a backend web server. The iRule is also used to automatically inject the client side JavaScript into the response stream for your webpages.

The iRule is completely standalone and requires no knowledge of how your web applications are configured or where they are located. You simply apply the iRule to the virtual server fronting your applications and Gomez takes care of the rest by organizing the requests by the URLs of your pages. The Gomez JavaScript looked like this:

```
1: <SCRIPT LANGUAGE="JavaScript"><!--
2: var gomez={gs: new Date().getTime(), acctId:'XXXXXX', pgId:'', grpId:''};
3: //--></SCRIPT>
4: <script src="/js/axfTag.js" type="text/javascript"></script>
```

The only configuration change needed in the iRule was to set the Gomez Account Id in the “acctId” parameter. But, you may notice there are two additional parameters in there that I didn’t use. The first one is the Page Identifier (pgId) which I will discuss in a future article. What I will discuss here is the Group Identifier (grpId).

The Group Identifier

On problem with performance testing when you are running a multi-datacenter infrastructure using the same domain name, like we are here at devcentral.f5.com, is that when you look at your reports, all application requests will show up on the same graph. If you are having a datacenter specific issue, it is not easy to narrow down where the bottleneck occurs as the reports will treat all requests as a single application.

The Group Identifier property allows you to give one additional search criteria to the Gomez reporting system. It would make perfect sense to use this feature to mark the application request to a given datacenter, or even to a given virtual server and that is exactly what we did on our implementation. We run DevCentral.f5.com out of multiple data centers for redundancy. And top top that off, within the datacenters, we have multiple virtual servers running the application. We have production virtuals with WebAccelerator and development virtuals without the web optimization features enabled. Since the iRule is running in the network layer and has information about the client side of the connection, it was very straightforward to build a lookup table that mapped the client side address to a specific virtual server.

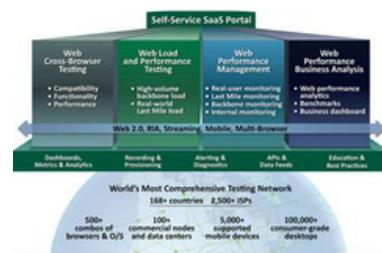
```
1: set GOMEZ_GROUPID_LIST [list \
2: "xxx.xxx.xxx.1 DC_PRD_DC1" \
3: "xxx.xxx.xxx.2 DC_PRD_DC2_C1" \
4: "xxx.xxx.xxx.3 DC_PRD_DC2_C2" \
5: "xxx.xxx.xxx.4 DC_PRD_DC2_C2_NO_WAM" \
6: "xxx.xxx.xxx.5 DC_PRD_DC3" \
7: "xxx.xxx.xxx.6 DC_DEV_DC1" \
8: "xxx.xxx.xxx.8 DC_DEV_DC2" \
9: "xxx.xxx.xxx.9 DC_STG_DC1" \
10: ];
11:
12: set gomez_group_id [findclass [IP::local_addr] $::GOMEZ_GROUPID_LIST " " ]
```

In addition to targeting production virtuals, we were also able to target internal development systems so that they run the Gomez injection tests and then be excluded from the final reports.

The above code builds a lookup table (replace xxx.xxx.xxx.xxx with your local ip addresses for your given virtual servers). Then the findclass method is used to lookup the address and return a user friendly name for the data center.

Once the user friendly group name is found, it can be included with the above JavaScript by the following code:

```
1: set gomez_client [subst {<SCRIPT LANGUAGE="JavaScript"><!--
2: var gomez={gs: new Date().getTime(), acctId:'XXXXXX', pgId:'', grpId:'$gomez_group_id';
```



```
3: //--></SCRIPT>
4: <script src="/js/axfTag.js" type="text/javascript"></script>
5: } ];
```

You'll notice I had to throw a "subst" command around the code snippet so that TCL would correctly replace the value of the \$gomez_group_id variable in the gomez_client variable. After this, the iRule is the same as in [the previous example](#).

Conclusion

If you are running a multi-datacenter environment and would like to enable Gomez performance monitoring to your applications, this iRule be all you need to help you target down datacenter specific issues with your application.

Download

You can download the full script in the [iRules CodeShare](#) under [GomezInjectionWithGroupId](#).

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