```
[05] 12/22/18 T(9192) _VMR 16:25:51.220948 DEBUG needAttn.C[1566]: START
handleVIOResponse scan [ 39 ].
```

Inside the **doNeedAttn** call, we recognize a `krest` HMC REST API job request pattern, which is handled by **kriSubmitNeedAttn** and **krigetJobResult()** calls. This pattern is similar to the case covered in “QuickQuery asynchronous job example” on page 105, so we do not need to get into details. The NA request, as it is delivered at the VIOS level, together with its corresponding response, obtained from the HSDB, are shown in Example 2-233.

Example 2-233 NeedsAttention request and response in the vioservice log

```
# alog -f /home/ios/logs/viosvc.log -o > viosvc.log.NAshortMsg.txt
# more viosvc.log.NAshortMsg.txt
...
START 12058942 62259487 12/22/18-15:20:45.144 vioservice.c 1.18 320]
/usr/ios/sbin/vioservice lib/libviopass/passthru
[0 12058942 62259487 12/22/18-15:20:45.145 viosvc_res.c 1.26 456] stdin pipe
input:[<?xml version="1.0"?>
<VIO xmlns="http://ausgsa.austin.ibm.com/projects/v/vios/schema/vioHADR2.00"
version="2.00" author="LIBKREST" title="Req Needs Attention">
  <Request action_str="VIO_HS_NEEDS_ATTENTION" dataType="GLOBAL_DATA"/>
</VIO>
]
[0 12058942 62259487 12/22/18-15:20:45.240 viosvc_res.c 1.26 464]
vio_response.result=[<VIO><Response>
<needsAtt hsTag="-3026381694404356594">
<vmStatList>
<vmList machine_type="8286" model="42A" serial="21E0B2V">
<vmStat uuid="4e395e99-fffb-4345-8067-2d5296ba7d91"><VM state="STARTED"
shortMsg="0x28" >
</VM>
</vmStat>
</vmList>
</vmStatList>
</needsAtt>
<needsAtt hsTag="-3026381694404356594">
<vmStatList>
<vmList machine_type="8286" model="42A" serial="2100DEW">
</vmList>
</vmStatList>
</needsAtt>
</Response></VIO>
]
[END 12058942 62259487 12/22/18-15:20:45.241 vioservice.c 1.18 323] exited with
rc=0
```