ISSPI Support Note 0007

Subject: Reporting on the OAVA database content for troubleshooting purposes

Severity: Important
Classification: Customer viewable
Audience: Customers, Software Support, Consultants

Release history
2015/05/13 First release (thierry.ledent@hp.com)

Get the latest version of this document and two contributed tools on Software Support Online:

https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01570396

ABSTRACT

The OAVA (Operations Agent Virtual Appliance) provides a preinstalled and preconfigured OA (Operations Agent) that collects performance metrics of a vSphere environment directly from vCenter Servers and stores them in a local SQLite database. This data is consumed by other HP software products such as VISPI (Virtualization Infrastructure Smart Plug-in) and SHR (Service Health Reporter).

This support note describes some tools that can be used to examine the content of the database for troubleshooting purposes. Following use cases are described:

Foreword: where to find the tools and how to exchange data with the OAVA

This support note references several support tools. Below two tools are available on a standard OAVA installation:

- easyoa.pl
- sqlite3

Below two tools are available from the Software Support Online at:
https://softwaresupport.hp.com/group/softwaresupport/search-result/-/facetsearch/document/KM01570431

- oava_inst_reports.sh
- oava_list_events.sh

Just copy these tools to the OAVA in directory /opt/OV/contrib/OpC and allow execution permissions. No further installation or configuration steps are required. Both tools provide a short help text:

```
# /opt/OV/contrib/OpC/oava_inst_report.sh -h
Usage:

oava_inst_report.sh [-d ';'] [-c|-m] [-f /tmp/A1copy.db] >/tmp/report.csv

Use option -d to specify an alternate delimiter (semicolon in the example; default is
comma).
Use option -f to specify an alternate dabatase.
Use option -c to create a report of classes instead of instances.
Use option -m to create a report of metrics instead of instances.
```

```
# /opt/OV/contrib/OpC/oava_list_events.sh -h
Usage:

./oava_list_events.sh [-d 'x'] [-e] [-s {ObservedTime|Name|Type} [-r]] [-f <dbpath>]
>/tmp/report.csv

Use option -d to specify a different delimitor (default is ',').
Use option -e to print time in seconds since epoch (default is YYYY-MM-DD hh:mm:ss
UTC).
Use option -s to specify on which field to sort output (default is ObservedTime).
Use option -r to reverse sort order.
Use option -f to specify an alternate dabatase.
```

Note that the above two tools are designed to produce a report that can be uploaded into Excel. The comma is used as default separator, but can be changed with the option '-d'

Finally, the tool SQLite Database Browser is a third-party tool available at http://sqlitebrowser.org that can be installed on a laptop to examine a local copy of an SQLite database such as the OAVA database.

Sometimes there is no direct access possible to the OAVA. In order to exchange data with the OAVA, for instance to upload a contributed support tool to OAVA, download a report created with one of the support tools, or download a copy of the OAVA database, one can use the ovdeploy command on the Operations Management server that manages the OAVA, e.g.:

1. **Upload data from the Operations Manager server to the OAVA**

```
# ovdeploy -upload -file oava_inst_report.sh -host oava.dom.com -td /opt/OV/contrib/OpC
# ovdeploy -cmd "chmod 555 /opt/OV/contrib/OpC/oava_inst_report.sh" -host oava.dom.com
```

Run the above commands from the Operations Manager server to upload oava_inst_report.sh from the current working directory to the OAVA in /opt/OV/contrib/OpC and change the permission to 555.

2. **Download data from the OAVA to the Operations Manager server**

```
# ovdeploy -cmd "cd /var/opt/OV/databases/oa/; cp A1.db A1_copy.db" -host oava.dom.com
# ovdeploy -download -file /var/opt/OV/databases/oa/A1_copy.db -host oava.dom.com -td /tmp
```

Run the above commands from the Operations Manager server to create a copy of the OAVA database on the OAVA and download it to /tmp (specify a Windows path is Operations Manager for Windows).

1. Obtaining a report of all VMware instances collected by OAVA

Obtaining a report of all VMware instances collected by OAVA can be very useful to troubleshoot OAVA collection problems or VISPI problems, especially VISPI discovery problems; or simply to get a better handle on the vSphere environment.

Tool: **/opt/OV/contrib/OpC/oava_inst_reports.sh**

After installing the tool (see "Foreword: where to find the tools and how to exchange data with the OAVA"), run it as below:

```
# /opt/OV/contrib/OpC/oava_inst_report.sh >/tmp/instances.csv
```

Download the file /tmp/instances.csv to a laptop and open it with Excel. The report can be nicely formatted into a table. A sample report (instances_sample.xls) is available from the Software Support Online link mentioned at the top of this document.

This report provides the list of instances along with following information:

- InstanceID: unique ID for this instance in the OAVA database
- Name: display name as known in vSphere client
- Hostname: hostname or FQDN of the instance (if applicable and available)
- IP: IP of the instance (if applicable and available)
- Role: role of the instance (e.g.: datacenter, host, guest, cluster, resource pool...)
- vCenter: IP of the managing vCenter (only for datacenters)
- Datacenter: datacenter hosting the instance (only for clusters, hosts and resource pools)
- Host/Cluster: name of the hosting server or cluster (only for hosts, guests and resource pools)
- ConnectionState: state of the instance (only for hosts, guests and datastores)
- NumberHosts: number of hosts (only for datacenters and clusters)
- NumberGuests: number of guests (only for datacenters)
- FirstObserved: date/time when the instance was first observed (UTC)
- LastObserved: date/time when the instance data was last collected (UTC)

2. Obtaining a description of the metrics collected by OAVA

Obtaining a description of the metrics collected by OAVA helps for a better understanding of the logic of VISPI policies. It is particularly useful when one decides to extract specific metric values from the OAVA database.

Tool: **/opt/OV/contrib/OpC/oava_inst_reports.sh**

After installing the tool (see "Foreword: where to find the tools and how to exchange data with the OAVA"), run it as below:

```
# /opt/OV/contrib/OpC/oava_inst_report.sh -m >/tmp/metrics.csv
```

Download the file /tmp/metrics.csv to a laptop and open it with Excel. The report can be nicely formatted into a table.

This report provides the list of metrics along with following fields:

- Class: class name (e.g. Cluster, Node, Datastore...)
- Metric: metric name (e.g. MemPhysUtil...)
- Class ID: internal ID for this class
- Metric ID: internal ID for this metric within it's class
- Description: short description of the metric

Knowing the Class ID and the Metric ID of a metric is important to browse through the database as the metrics or organized in classes, each of which is stored in a different table of the name dml_<classid> or dml_inst_<classid> (more details on this later). Metrics are organized in classes as follows:

- Class Datastore: metrics collected for datastores
- Class VirtualApp: metrics collected for VMware virtual application
- Class Datacenter: metrics collected for datacenters
- Class Node: metrics collected for hosts and guests
- Class ResourcePool: metrics collected for resource pools
- Class Cluster: metrics collected for clusters
- Class NodeDSRelation: information on node to datastore relationship
- Class DSNodeReletion: information on datastore to node relationship

3. Listing the VMware events and alarms collected in the OAVA database

Obtaining a list of VMware events and alarms collected in the OAVA database is important for troubleshooting problems with the policy VI-VMwareVCEventMonitor. If a VMware event or alarm is not forwarded to the Operations Management server, the first step is to check whether the event was at all collected by OAVA. With this tool, one can list all the events and alarms from the OAVA database. If the event or alarm is missing there, the problem should be addressed as an event or alarm collection problem in OAVA. If the event or alarm is available in the database, the problem should be addressed from a VISPI perspective

Tool: **/opt/OV/contrib/OpC/oava_list_events.sh**

After installing the tool (see "Foreword: where to find the tools and how to exchange data with the OAVA"), run it as below:

```
# /opt/OV/contrib/OpC/oava_list_events.sh >/tmp/events.csv
```

Download the file /tmp/events.csv to a laptop and open it with Excel. The report can be nicely formatted into a table.

This report provides the list of metrics along with following fields:

- ObservedTime: this is the time (UTC) when the event or alarm was collected (not the time when it was generated in VMware)
- InstanceName: this is the name of the instance to which the event or alarm relates
- EventType: this is the type of event
- Description: this is the event or alarm text

4. Checking the current values of individual metrics

The tool easyoa.pl can be used to read the current value of individual metrics from the OAVA database. It also allows to apply some filters to restrict the output to specific instances.

Tool: **/opt/OV/support/easyoa.pl**

This tool is pre-installed on the standard OAVA.
Here a few use cases (assumes /opt/OV/support is in the PATH):

1. **Obtain a list of ESX hosts**

```
# easyoa.pl -dn Infrastructure -c Node -m SystemHostName -f SystemRole=HOST
```

```
  Performing Metric Query ...
    ---------------------------
==
_instance               :       esxhost3.gale2.net
SystemHostName          :       esxhost3.gale2.net
==
_instance               :       esxhost2.gale2.net
SystemHostName          :       esxhost2.gale2.net
==
_instance               :       esxhost4.gale2.net
SystemHostName          :       esxhost4.gale2.net
==
_instance               :       esxhost1.gale2.net
SystemHostName          :       esxhost1.gale2.net
```

2. **Obtain the cpu usage of guests on a particular host**

```
# easyoa.pl -dn Infrastructure -c Node -m SystemState,CPUTotalUtil
            -f SystemRole=GUEST,SystemHostHostName=esxhost1.gale2.net -si 15
```

```
  Performing Metric Query ...
    ---------------------------
==
_instance               :       vm211A
CPUTotalUtil            :       7.256667
SystemState             :       on
==
_instance               :       vm111A
CPUTotalUtil            :       7.133333
SystemState             :       on
==
_instance               :       vm200A
CPUTotalUtil            :       NA
SystemState             :       off
==
_instance                :       vm112A
CPUTotalUtil             :       7.090000
SystemState              :       on
```

Don't be confused by the metric name SystemHostHostName (it is not a typo error); we are indeed filtering on the hostname of the host of the virtual machine. It is important to provide the exact ESX host name in this filter as it will perform an exact match with the content of the database. Use the first example command to check the exact name if in doubt.
The option "-si 15" specifies that the data should be summarized (averaged) over a 15 minutes period. This is recommended because easyoa will otherwise collect the value for the last collection interval which may currently be in progress and therefore return no data for some instances.

3. Obtain the CPU usage of a cluster

```
# easyoa.pl -dn Virtualization -c Cluster -m CPUTotalUtil -si 15
```

```
  Performing Metric Query ...
    ---------------------------
==
_instance                :       Cluster1
CPUTotalUtil             :       2.460000
```

Note that in this particular case we specify the domain "Virtualization" with the option "-dn". All classes are in the "Virtualization" domain, except the class "Node" which is in the "Infrastructure" domain as shown in earlier examples.

For a list of classes and metrics, see "2. Obtaining a description of the metrics collected by OAVA". It is also possible to query the object model with:

```
# easyoa.pl -dn <domainname> -c <classname> -o model
```

5. Sneaking into the database with SQL commands

The command line interface sqlite3 enables to run SQL commands on the OAVA database. Knowledge of SQL is assumed in this section.

Tool: **/opt/OV/bin/sqlite3**

This tool is preinstalled on the standard OAVA.

To avoid unintentionally corrupting the database, it is recommended to work on a copy of the database:

```
# cd /var/opt/OV/databases/oa
# cp A1.db A1_copy.db
```

1. Connect to the database copy

Connect to the database copy with:

```
# /opt/OV/bin/sqlite3 /var/opt/OV/databases/oa/A1_copy.db
SQLite version 3.7.15.2 2013-01-09 11:53:05
```

```
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite>
```

2. Understanding the database model

Let us demonstrate how to query the CPU usage of the guest vm112A. We would like to run a query that would look similar to:

```
> select CPUTotalUtil from <tablename> where SystemHostName='vm112A';      (this will not work)
```

But there is a little challenge since tables are named after the class ID, and metric fields are named after the metric ID. These IDs differ on each OAVA installation. To obtain the class IDs, one can use:

```
sqlite> select Id,Name,Description from ManagedClasses;
3|NodeDSRelation|The NodeDSRelation class represents the association between...
9|DSNodeRelation|The DSNodeRelation class represents the association between...
15|Datastore|The Datastore class represents local, network, or vmfs store mounted...
38|VirtualApp|The VirtualApp class represents a general purpose pool of ...
66|Datacenter|The DataCenter class represents a data center in any IT environment...
75|Node|The Node class represents a general purpose computing system. It could...
406|ResourcePool|The  Resource  pool  class  represents  a  general  purpose  pool...
435|Cluster|The cluster class represents a cluster of nodes for Load Balancing,...
```

Since we want to view metrics from a guest, we are interested in the class Node that has Class ID 75 on this OAVA. Let us now search for the ID of metric CPUTotalUtil and metric SystemHostName in the class 75:

```
sqlite> select Id,Name from Metrics
   ...> where BelongsToClassId=75 and Name in ("CPUTotalUtil","SystemHostName");
129|CPUTotalUtil
386|SystemHostName
```

Note that the tool oava_inst_report.sh can generate a report of class and metric IDs which can considerably speed up above steps (see 2. Obtaining a description of the metrics collected by OAVA).

Now we can figure the table name and field name for these 2 metrics:

- Since CPUTotalUtil is a dynamic metric, it is stored in table dml_<classid>, that is dml_75 and the field name is dml_129.
- Since SystemHostName is a static metric, it is stored in the table dml_inst_<classid>, that is dml_inst_75 and the field name is dml_386.

Remember that class IDs and metric IDs differ on each OAVA installation.

3. Running the SQL query

Finally, the SQL query of our example becomes:

```
sqlite> select datetime(d.CollectionTime,'unixepoch'),d.dml_129
   ...> from dml_75 d, dml_inst_75 s
   ...> where d.InstanceId=s.InstanceId and s.dml_386='vm211A'
   ...> order by d.CollectionTime;
```

```
...
2015-05-13 10:57:03|7.28
2015-05-13 15:05:01|7.09
2015-05-13 15:10:15|7.19
2015-05-13 15:15:18|7.28
2015-05-13 15:20:06|7.21
2015-05-13 15:25:08|7.28
2015-05-13 15:30:11|7.21
2015-05-13 15:35:10|7.13
2015-05-13 15:40:16|7.18
2015-05-13 15:45:04|7.25
```

To exit from the sqlite3 command, use:

```
sqlite> .q
```

6. Browsing through the database with a GUI

The third-party tool SQLite Database Browser enables to browse through the OAVA database without requiring SQL commands.

Tool: **SQLite Database Browser**

This tool is best installed on your laptop. Then make a copy of the OAVA database:

```
# cd /var/opt/OV/databases/oa
# cp A1.db A1_copy.db
```

See Foreword: where to find the tools and how to exchange data with the OAVA for a link to the tool and how to download A1_copy.db.

Start SQLite Database Browser and open A1_copy.db from the File menu.

The tool is very intuitive and enables to browse through the tables, sort the data and apply filters. One challenge is to understand the database model as the table and field names are different on each OAVA installation. For this, check "Understanding the database model" above, but instead of querying the tables ManagedClasses and Metrics through SQL commands, one can directly browse these tables in SQLite Database Browser.