

Listing of Data sources in Performance Grapher [PMi] - BSM 9.1x Vs BSM 9.2x

BSM 9.1x

In order to communicate with any selected CI, Performance Grapher needs the CI information like - System name, Associated Graph templates, Instances- from BSM. OMi provides this information to Performance Grapher.

On selecting a CI, Performance Grapher makes Web service request to OMi to get the CI information. The Web-service response will contain the information required by Performance Grapher to connect to the CI and display the respective datasources.

The OMi Web-service response would look like :

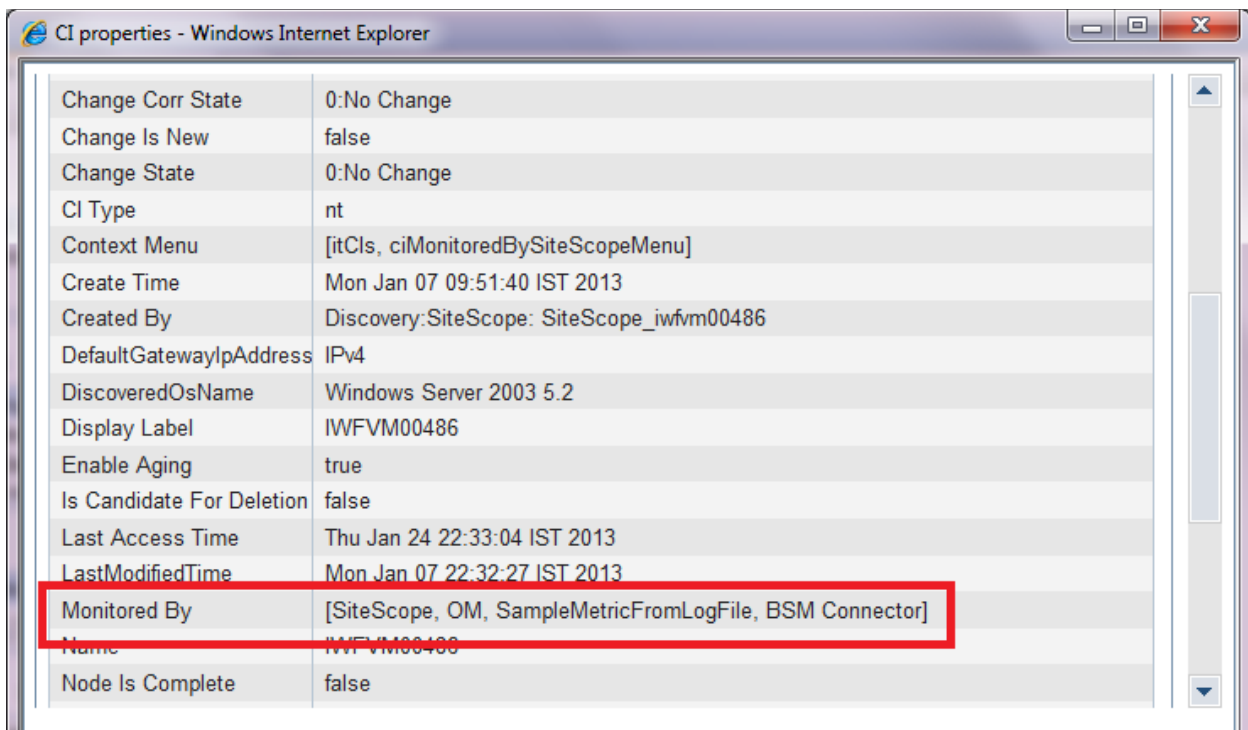
```
<?xml version="1.0"?> <graphParameters>
  <graphTemplates>
    <graphUuid>b771e39e-60ea-4571-bc75-89ce38b962bb</graphUuid>
    <graphUuid>c546e39e-54ea-4901-bc75-89ce38b962eb</graphUuid>
  </graphTemplates>
  <systemNames>
    <systemName>abc.def.hi.com</systemName>
  </systemNames>
  <instanceNames>
    <instance>
      <instanceName>SiteScope:4:600212</instanceName>
      <monitoredBy>lmn.op.qr.com</monitoredBy>
    </instance>
    <instance>
      <instanceName> abc.def.hi.com </instanceName>
    </instance>
  </instanceNames>
  <canDesignGraphs> true</canDesignGraphs>
</graphParameters>
```

In the above Web-service response:

- a. **<graphTemplates>** represents the associated graph templates for the CI. In case no graph templates are associated with the CI, then Web-service will return error response.
- b. Connects to AGENT [PA/EPC/RTM] data source for the system, **abc.def.hi.com**
- c. Creates AGENTLESS data source since Web-service output contains **<monitoredBy>** information. **<monitoredBy>** represents the SiteScope server that monitors this system. Performance Grapher will connect to the SiteScope server to get data for the corresponding **<instanceName>**, which is the CMDB_HINT.

BSM 9.2x

The key change in BSM 9.20 is that the dependency of Performance Grapher on OMi has been removed. Now, Performance Grapher can directly get the CI information from UCMDB. Listing of data sources depends on “monitored by” parameter of CI properties. [Right Click on CI > Show > Properties]



Change Corr State	0:No Change
Change Is New	false
Change State	0:No Change
CI Type	nt
Context Menu	[itCIs, ciMonitoredBySiteScopeMenu]
Create Time	Mon Jan 07 09:51:40 IST 2013
Created By	Discovery:SiteScope: SiteScope_iwfv00486
DefaultGatewayIpAddress	IPv4
DiscoveredOsName	Windows Server 2003 5.2
Display Label	IWFVM00486
Enable Aging	true
Is Candidate For Deletion	false
Last Access Time	Thu Jan 24 22:33:04 IST 2013
LastModifiedTime	Mon Jan 07 22:32:27 IST 2013
Monitored By	[SiteScope, OM, SampleMetricFromFile, BSM Connector]
Name	IWFVM00486
Node Is Complete	false

As per the above illustration, on selecting CI, IWFVM00486, the data sources - SiteScope, Agent [OM] and BSM Connector – would be listed in Performance Grapher

Performance Grapher-SiteScope Integration

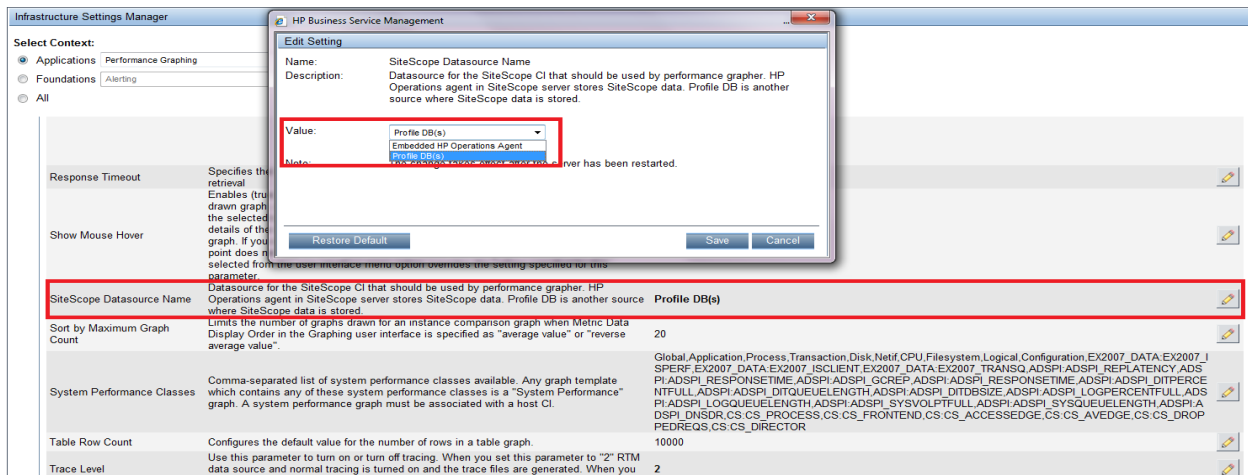
SiteScope integrated with BSM will log data into BSM Profile DB for the created monitors. Until BSM 9.21, Performance Grapher could not communicate with BSM profile DB to get SiteScope data. For this reason, SiteScope would push data into Operations Agent [OA], installed in the SiS server, as AGENTLESS datasource. Performance Grapher will query the CODA DB and get AGENTLESS data for the respective SiteScope monitors.

Performance Grapher can get SiteScope data in either of the 2 ways:

1. SiteScope data from OA [AGENTLESS datasource]
2. SiteScope data from BSM Profile DB [SiteScope datasource] – Added since BSM 9.21

In BSM 9.21, the SiteScope integration is configurable. By default, Performance Grapher would get SiteScope data from OA. To change the integration to Profile DB,

Admin > Platform > Infrastructure Settings > Performance Graphing Application



The screenshot shows the 'Infrastructure Settings Manager' application with a dialog box titled 'Edit Setting' open. The dialog box has the following fields:

- Name:** SiteScope Datasource Name
- Description:** Datasource for the SiteScope CI that should be used by performance grapher. HP Operations agent in SiteScope server stores SiteScope data. Profile DB is another source where SiteScope data is stored.
- Value:** A dropdown menu with three options: 'Profile DB(s)', 'Embedded HP Operations Agent', and 'Profile DB(s)'. The 'Profile DB(s)' option is selected and highlighted with a red box.
- Notes:** A text area containing: 'The user interface menu option overrides the setting specified for this parameter.' Below this, a note states: 'When you select Profile DB(s), the HP Operations agent server has been restarted.'

At the bottom of the dialog box are buttons for 'Restore Default', 'Save', and 'Cancel'. Below the dialog box, in the background settings window, the 'SiteScope Datasource Name' setting is highlighted with a red box, showing its description and the value 'Profile DB(s)'.

To enable Performance Grapher to query SiteScope data from Coda db, select the option – “Embedded HP Operations Agent”

To enable Performance Grapher to query SiteScope data from BSM Profile DB, select the option – “Profile DB(s)”

SiteScope data from OA [AGENTLESS data source]

Performance Grapher can seamlessly collect data stored in Coda DB. Performance Grapher-SiteScope with Operations Agent (OA) integration works on this principle. SiteScope as a product bundles Operations Agent (OA). While installing software, user can enable the SiteScope-OA integration for SiteScope server.

While creating a SiteScope monitor, user needs to select the option to say that the data should be logged to HP Operations Agent, without which this integration would not work.

For any monitor created, SiteScope logs data into Operations Agent (OA) as well as the BSM Profile DB. SiteScope data is logged in to CODA db as “AGENTLESS” data source. In Performance Grapher, each SiteScope monitor is identified by the key metric, “CMDB_HINT”. CI may have one or many SiteScope monitors. Based on the value for “CMDB_HINT”, Performance Grapher would graph the data.

When a CI is selected, Performance Grapher passes on CI Id to get the CI information, which would look like :

```
<graphParameters>
  <graphTemplates>
    <graphUuid>b771e39e-60ea-4571-bc75-
      89ce38b962bb</graphUuid>
  </graphTemplates>
  <systemNames>
    <systemName>abc.def.ghi.com</systemName>
  </systemNames>
  <instanceNames>
    <instance>
      <instanceName>SiteScope:4:600212</instanceName>
    </instance>
  </instanceNames>
</graphParameters>
```

```
        <monitoredBy>lmn.op.qr.com</monitoredBy>
    </instance>
</instanceNames>
<canDesignGraphs>true</canDesignGraphs>
<ciType>unix</ciType>
</graphParameters>
```

In the above example, for the selected CI, **abc.def.ghi.com**, “**SiteScope:4:600212**” is the value for CMDB_HINT metric and the CI is monitored by the SiteScope server “**lmn.op.qr.com**”. Performance Grapher will connect to “**lmn.op.qr.com**” and get the data for “**SiteScope:4:600212**” instance.

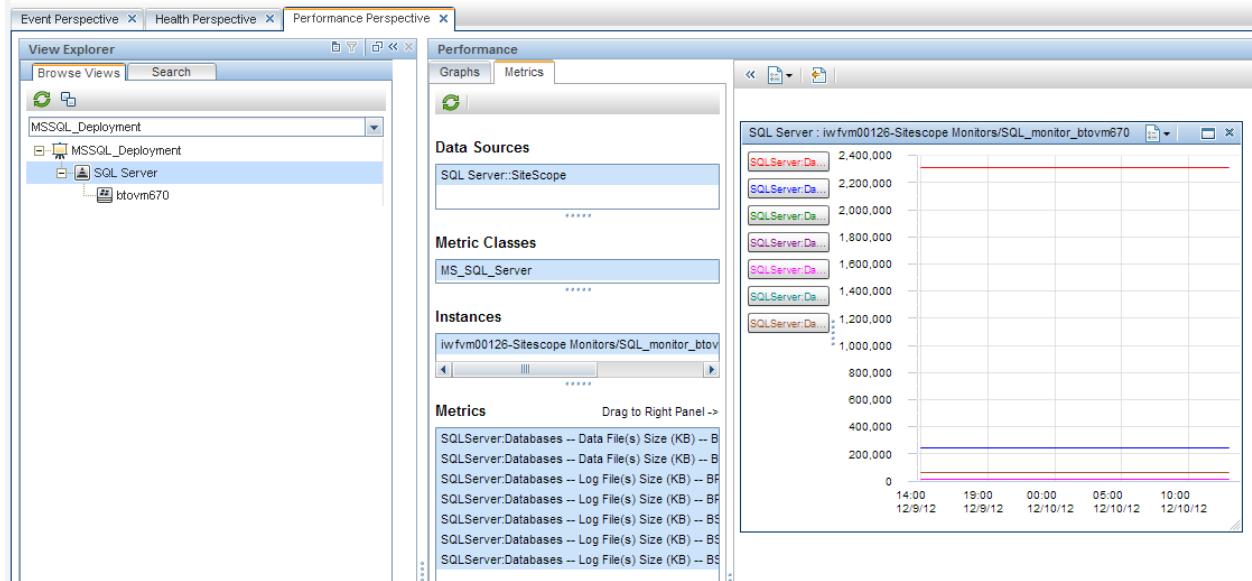
Note : For Performance Grapher – SiteScope server communication to be successful, the default port - **383** should be open in the SiteScope server.

Performance Grapher makes CODA API calls to retrieve the Classes, Metrics and data [values for Metrics] from the CODA db. From this, Performance Grapher would get data for the CMDB_HINT value(s) with respect to the selected CI and display the corresponding data on the Graph.

SiteScope data from Profile DB [SiteScope data source]

As in the SiteScope-OA integration, Performance Grapher would get the SiteScope server that monitors the selected CI from the CI information. Performance Grapher would fetch Metric list from Profile DB using TMC queries. Performance data is fetched from Profile DB using GDE queries.

When Performance Grapher collects data from Profile DB, the data source would be listed as “Selected CI::SiteScope”



From the above screenshot, **Metric Classes** would list the SiteScope Monitor Type Name without special characters. **Instances** lists the full path of the monitor and the SiteScope server monitoring the CI.

SiteScope data source has instance name in few metric names. Performance Grapher supports wildcard in metric name to create templates with such metrics.