



PG 5.6 - CMS 4.3 Agent Integration

February 2010



Table of Contents

Introduction	3
Installation	4
Install a CMS agent if not already installed	4
Install a PG agent if not already installed	4
Configuration	5
Standalone configuration	5
Server Configuration	6
Registry Parameters	7
The CMSInterface registry parameter	7
The "Configuration" registry parameter	8
Data Files	10
Location	10
File Format	11
Example	12



Introduction

This document describes the how to integrate a Performance Guard agent with a CMS agent on a client PC.

The integration enables you to pass performance data measured by a PG agent to a CMS agent.

The PG agent will measure performance data and at regular intervals, specified by the CMS agent, store the performance data in local xml files. The CMS agent will later collect the xml file content and send it to the CMS backend.



Installation

The installation procedure consists of the following steps:

1. Install a CMS agent on the PC, if not already installed.
2. Install a PG agent (version 5.6 and later) on the PC, if not already installed.

Install a CMS agent if not already installed

Please follow the CMS documentation for installing an agent.

Install a PG agent if not already installed

Please follow the PG agent installation documentation if you are deploying an agent that should connect to PG server.

If you are deploying a standalone PG agent you should simply install the agent using the msi file by executing the following command (as an administrator):

```
32-bit: msiexec /i PGuardAgent.msi /qn  
64-bit: msiexec /i PGuardAgent64.msi /qn
```

The agent can of course be deployed using CMS.

The agent will be installed in the following default directory:

```
32-bit: c:\Program Files\Premitech\Performance Guard Agent  
64-bit: c:\Program Files (x86)\Premitech\Performance Guard Agent
```

The agent will be installed as a windows service called "Premitech Performance Guard Agent" and the service is started automatically.



Configuration

Standalone configuration

If the PG agent does not communicate with a PG server you must do the following to configure integration between PG and CMS:

1. Stop the Performance Guard Agent Service
2. Create a text file with agent registry settings. The following is an example text file:

```
Configuration = "ReportInterval=60
DynamicMachineReport=true
DiskIOReport=true
ProcessStatInterval=10
ContextSwitchPerProcess=false
ServerMetrics=true
CollectOSStart=true
DoFullStartupReport=true
StartupTimeoutSecs=300
StartupCondition=<PROCESS; winlogon.exe; started>
LoginStartCondition=<SESSION; logon>
LoginEndCondition=<PROCESS; explorer.exe; started>
CMSEnabled=true
CMSForced=false"

CMSInterface = "CMSMaxFiles=300
CMSDir=CMS\
CMSReportTime=180"
```

3. Open a command prompt in the Performance Guard agent directory.
4. Install the registry settings from the text file (in this case "CMS_PG.cfg") by executing the following command

```
GUARDagent -configfile="CMS_PG.cfg" -updateregistry
```

5. Start the Performance Guard Agent Service

The above steps can be done by the CMS agent at any time to control the behaviour of the PG agent.



Server Configuration

If the PG agent is communicating with a PG server you must enable the "Integration Enabled" option in the CMS section of the agent configuration using the PG Web Console on the PG server. There you can specify all CMS integration parameters.

If you need to control the CMS integration parameters locally on the client you can do the following:

1. Disable the "Always send reports to CMS" option in the CMS section of the agent configuration using the PG Web Console on the PG server.
2. Stop the Performance Guard Agent Service
3. Create a text file with agent registry settings. The following is an example text file:

```
CMSInterface = "CMSMaxFiles=300  
CMSDir=CMS\  
CMSReportTime=180"
```

4. Open a command prompt in the Performance Guard agent directory.
5. Install the registry settings from the text file (in this case "CMS_PG.cfg") by executing the following command

```
GUARDagent -configfile="CMS_PG.cfg" -updateregistry
```

6. Start the Performance Guard Agent Service



Registry Parameters

The PG agent reads/stores its configuration from the following registry key:

32 Bit system: HKLM\SOFTWARE\Premitech\E2EAgent

64 bit system: HKLM\SOFTWARE\wow6432Node\Premitech\E2EAgent

If the PG agent is an OEM version, the registry key will be:

32 Bit system: HKLM\SOFTWARE\Premitech\<OEM-name>_E2EAgent

64 bit system: HKLM\SOFTWARE\wow6432Node\Premitech\<OEM-name>_E2EAgent

The CMSInterface registry parameter

The CMS agent can control the behaviour of the PG agent by setting the value of the "CMSInterface" registry parameter. The value is itself a multi-line string with a name-value pair on each line.

The possible named values are:

Name	Default	Range	Precedence
CMSDir	CMS\	Three valid path types: 1) "." 2) "<subdir>" 3) "<drive>:<path>"	Local directory where data files are stored.
CMSMaxFiles	20	[1;n]	Maximum number of data files in directory. Oldest files are deleted when the number of files reaches CMSMaxFiles.
CMSReportTime	900	ReportInterval x [1;n]	Number of seconds between each data file is stored to disk. The number of seconds should be larger than the ReportInterval value from the Configuration registry parameter - and preferably a multiplum of ReportInterval.



The "Configuration" registry parameter

If the PG agent is connected to a PG server it will initially receive its configuration from the PG server. The configuration is then stored locally in the registry parameter "Configuration". If the PG agent is not connected to a PG server, the value of the "Configuration" parameter will be empty.

The "Configuration" parameter value is a multi-line text string, where each line is a configuration parameter and value pair.

By default the PG agent does not integrate with CMS, so you must set some parameters manually if they are not controlled by a PG server.



The following table is a subset of parameters that can be set in the "Configuration" registry parameter:

Parameter	Default	Range	Description
CMSEnabled	false	true/false	Enables/disables CMS integration When this is enabled, the PG agent will also generate a full Startup report, regardless of any other options.
CMSForced	false	true/false	If set to true, the value of the CMSInterface registry parameter is ignored, i.e. PG agent is in control. If set to false, the value of the CMSInterface registry parameter takes precedence over the Configuration registry parameter, i.e. the CMS agent is in control.
CMSDir	CMS	Three valid types: 1) "." 2) "<subdir>" 3) "<drive>:<path>"	Local directory where data files are stored.
CMSMaxFiles	20	[1;n]	Maximum number of data files in directory. Oldest files are deleted when the number of files reaches CMSMaxFiles.
CMSReportTime	900	ReportInterval x [1;n]	Number of seconds between each data file is stored to disk. The number of seconds should be larger than the ReportInterval value from the Configuration registry parameter - and preferably a multiplum of ReportInterval.
ReportInterval	60	[1;n]	Report period in seconds. See PG documentation
ServerMetrics	false	true/false	Measure context switches etc.. See PG documentation
CollectOSStart	false	true/false	Measure startup/login time. See PG documentation
StartupCondition	-	-	End condition for startup measurement. See PG documentation
StartupTimeOutSecs	300	[1;n]	Timeout value for startup condition. See PG documentation
LoginStartCondition	-	-	Start condition for login measurement. See PG documentation.
LoginEndCondition	-	-	End condition for login measurement. See PG documentation.
DynamicMachineReport	true	true/false	See PG documentation
DiskIOReport	false	true/false	See PG documentation
ProcessStatInterval	10	[1;n]	Sample period in seconds.



			See PG documentation
ContextSwitchPerProcess	false	true/false	Measure context switches per process. See PG documentation

Data Files

Location

The name of the produced data files is PGData<nn>.xml (<nn> is the date in seconds since epoch).

The location of the files is defined by the CMSDir parameter. Be aware that a service like the PG agent is not allowed to write anywhere on the local disk.

If the value of the CMSDir parameter is not a full path, it is considered to be a sub directory of the "%ProgramData%" folder. This is a safe location for services to write files.

The value of the %ProgramData% environment variable depends on the type of operating system:

OS	%ProgramData%
Windows 7	C:\ProgramData
Windows Vista	C:\ProgramData
Windows XP	C:\Documents and Settings\All Users\Application Data

The actual path used by the agent can be seen in the web console of the agent:

http://127.0.0.1:4007/cmd/cmd?group=view&cmd=VIEW_ERRORS

Use Search filter to look for:

Example :

Default	Info	E2EReportAssembler	CMS parameter	Data Files Path='C:\ProgramData \CMS\'
---------	------	--------------------	---------------	--

NOTE: If you can't find the folder in Windows, the reason can be that you don't have the permission to view the folder, or that your Windows Explorer options hides protected and hidden folders from you. Try opening the folder anyway by typing the name of the folder into Windows Explorer.



File Format

- (1) `AggregateRunningSystemReport` :Aggregated values describing the usage of the CPU, Memory, DiskIO, NetIO, and TotalUtilizationMeasure. All the values is described in a 1 to 100 usage measure. Where 1 is very low usage and 100 is maximum usage. NetworkIO and DiskIO usages are calculated from reference values stored in the PG configuration.
The TotalUtilizationMeasure is the worst (highest) of the 4 basic usage measures.
- (2) `RunningSystemReport`: Detailed information about the current performance of the machine
- (3) `StartupBriefReport`: information about the time and duration of an operating system startup event.
- (4) Start time for the measurement of the values in the `StartupReport`
- (5) `Process`: Detailed measurements for a single process.
- (6) `LoginBriefReport`: Information about the time and duration of a user login event. Also contains the name of the user that logged in.

```
<PerformanceGuardData>
  <AggregateRunningSystemReport> (1) </AggregateRunningSystemReport>
  <RunningSystemReport> (2) </RunningSystemReport>
  <StartupBriefReport> (3) </StartupBriefReport>
  <StartupReport>
    <SampleInterval> (4) </SampleInterval>
    <Process> (5) </Process>
    ...
    <Process>...</Process>
  </StartupReport>
  <LoginBriefReport> (6) </LoginBriefReport>
</PerformanceGuardData>
```



Example

```
<PerformanceGuardData>
  <AggregateRunningSystemReport>
    <CPU ValueType='UChar'>6</CPU>
    <Mem ValueType='ULong'>32</Mem>
    <DiskIO ValueType='ULong'>41</DiskIO>
    <NetIO ValueType='ULong'>100</NetIO>
    <TotalUtilizationMeasure ValueType='ULong'>100</TotalUtilizationMeasure>
  </AggregateRunningSystemReport>
  <RunningSystemReport>
    <SampleInterval ValueType='ULong'>1250584440</SampleInterval>
    <PeakCPUUsage ValueType='UChar'>37</PeakCPUUsage>
    <AverageCPUUsage ValueType='UChar'>6</AverageCPUUsage>
    <LowFreePhysicalMemory ValueType='ULong'>8482592</LowFreePhysicalMemory>
    <AverageFreePhysicalMemory ValueType='ULong'>8527056</AverageFreePhysicalMemory>
    <LowFreePagingFile ValueType='ULong'>20646</LowFreePagingFile>
    <AverageFreePagingFile ValueType='ULong'>20680</AverageFreePagingFile>
    <LowVirtualMemory ValueType='ULong'>1984000</LowVirtualMemory>
    <AverageFreeVirtualMemory ValueType='ULong'>1984000</AverageFreeVirtualMemory>
    <AverageDiskWrites ValueType='ULong'>122802</AverageDiskWrites>
    <AverageDiskReads ValueType='ULong'>3159</AverageDiskReads>
    <PeakDiskWrites ValueType='ULong'>432356</PeakDiskWrites>
    <PeakDiskReads ValueType='ULong'>14480</PeakDiskReads>
    <AverageContextSwitches ValueType='ULong'>8393</AverageContextSwitches>
    <PeakContextSwitches ValueType='ULong'>13211</PeakContextSwitches>
    <AverageSendBytes ValueType='ULong'>3099</AverageSendBytes>
    <PeakSendBytes ValueType='ULong'>8073</PeakSendBytes>
    <AverageRecvBytes ValueType='ULong'>54556</AverageRecvBytes>
    <PeakRecvBytes ValueType='ULong'>214949</PeakRecvBytes>
  </RunningSystemReport>
  <StartupBriefReport>
    <SampleTime ValueType='ULong'>3825400163</SampleTime>
    <StartTime ValueType='ULong'>344942282</StartTime>
    <Success ValueType='UChar'>1</Success>
  </StartupBriefReport>
  <StartupReport>
    <SampleInterval ValueType='ULong'>3825400163</SampleInterval>
    <Process>
      <ProcessName ValueType='String'>smss</ProcessName>
      <OwnerName ValueType='String'>SYSTEM</OwnerName>
      <OwnerDomain ValueType='String'>NT AUTHORITY</OwnerDomain>
      <HostedServices ValueType='String'></HostedServices>
      <RunTime ValueType='ULong'>435528</RunTime>
      <TotalCPU ValueType='ULong'>0</TotalCPU>
      <TotalDiskReadIO ValueType='ULong'>11</TotalDiskReadIO>
      <TotalDiskWriteIO ValueType='ULong'>4</TotalDiskWriteIO>
      <TotalOtherIOOps ValueType='ULong'>433</TotalOtherIOOps>
      <TotalOtherIOBytes ValueType='ULong'>9925</TotalOtherIOBytes>
      <TotalPageFaults ValueType='ULong'>531</TotalPageFaults>
      <BasicProcessInfo>
        <ProcessID ValueType='ULong'>516</ProcessID>
        <StartTime ValueType='ULong'>0</StartTime>
        <MajorVersion ValueType='ULong'>64268</MajorVersion>
        <MinorVersion ValueType='ULong'>1199</MinorVersion>
        <Flags ValueType='UChar'>16</Flags>
        <StringLen ValueType='UChar'>4</StringLen>
      </BasicProcessInfo>
    </Process>
    <Process>
      <ProcessName ValueType='String'>wininit</ProcessName>
      <OwnerName ValueType='String'>SYSTEM</OwnerName>
      <OwnerDomain ValueType='String'>NT AUTHORITY</OwnerDomain>
      <HostedServices ValueType='String'></HostedServices>
    </Process>
  </StartupReport>
</PerformanceGuardData>
```



```
<RunTime ValueType='ULong'>435523</RunTime>
<TotalCPU ValueType='ULong'>0</TotalCPU>
<TotalDiskReadIO ValueType='ULong'>3</TotalDiskReadIO>
<TotalDiskWriteIO ValueType='ULong'>1</TotalDiskWriteIO>
<TotalOtherIOOps ValueType='ULong'>3004</TotalOtherIOOps>
<TotalOtherIOBytes ValueType='ULong'>21260</TotalOtherIOBytes>
<TotalPageFaults ValueType='ULong'>1883</TotalPageFaults>
<BasicProcessInfo>
  <ProcessID ValueType='ULong'>644</ProcessID>
  <StartTime ValueType='ULong'>4976</StartTime>
  <MajorVersion ValueType='ULong'>64268</MajorVersion>
  <MinorVersion ValueType='ULong'>1199</MinorVersion>
  <Flags ValueType='UChar'>16</Flags>
  <StringLen ValueType='UChar'>7</StringLen>
</BasicProcessInfo>
</Process>
<Process>
  ...Same data types for all the processing recorded.
</Process>
...Same data for each running process.
</StartupReport>
<LoginBriefReport>
  <SampleTime ValueType='ULong'>3825400163</SampleTime>
  <LoginTime ValueType='ULong'>344942282</LoginTime>
  <LoggedInUser ValueType='String'>SvendBent</LoggedInUser>
  <Success ValueType='UChar'>1</Success>
</LoginBriefReport>
</PerformanceGuardData>
```