Automic
Professional Services

Automic - AppLink

Sysload Studio – Open Agent for StreamCore (SGM v6.x)

(February 2015 – Build 8)

Automic
Professional Services

Content

Update

Version	Date	Autor	Update
0.1	14/08/2013	Laurent LAO (Automic / Orsyp)	Created
1.0	16/10/2013	Laurent LAO (Automic / Orsyp)	Release
1.2	27/05/2014	Laurent LAO (Automic / Orsyp)	Correction Script (v1.20)
1.3	17/07/2014	Laurent LAO (Automic / Orsyp)	Correction Script (v1.30)
1.4	14/08/2014	Laurent LAO (Automic / Orsyp)	build7 Correction Script (v1.40)
1.5	05/09/2014	Laurent LAO (Automic / Orsyp)	build8 Correction Script (v1.50) New metrics : application_shapping_rule Compatibility with SGM 6.2
1.5	11/02/2015	Laurent LAO (Automic / Orsyp)	New Setup Kit v5.80 (Industrialization option)

References

Version	Date	Autor	Name

1. Introduction

Sysload Studio help you to create your own Monitoring Agents. Thus, custom scripts may beings integrated Monitoring Agents to expand the scope of the monitoring done by Sysload and / or correlate metrics and ability to use third-party servers with metrics (eg number of users connected to a business application, number of transactions, the response time of a request).

For example, agents have already been put in place for our customers to collect metrics
- Network (routers, switchs, ...)

- Storage (backup robot, temperature sensors, ...)

- Databases (MySQL, SyBase)

- Application (WebLogic, Apache, Oracle Application Server, ...)

- Business (number of users of a web portal, response time, ...)

The web interface allows Sysload Studio set up metrics and combine these metrics in monitoring scripts (shell, Perl, Powershell, etc..) That will collect the values of these metrics. Monitoring scripts and schedules are encapsulated by the custom Monitoring Agent. Custom Monitoring Agents benefit from standard features Sysload Agents: Alerting, real-time monitoring, historical tracking, projections, centralized deployment, easy maintenance, etc).

This document describes the integrating process for **StreamCore SGM** and prerequisites.

2. Environment StreamCore SGM

2.1 Streamcore Device

- **StreamGroomer Manager (SGM)** : Administration, Database management, configuration management, and StreamGroomer management.
- **StreamGroomer (SG)** : Appliance collect QoS metrics.

The Monitored Objects will collect statistics data on the SGM.

	Environment tested
Version	SGM 6.0 SGM 6.1 SGM 6.2

2.2 SGM directory

- **Sites** :
- **Categories** :

The Monitored Objects will collect onty Sites statistics

2.3 SGM Rules

There are 3 rules domains :

- **Network** : Acces Link, Shapping et Grooming.
- **Application** : Intermediaire and terminated
- **VOIP** : Signalization and Media

All rules ares collected

3. Sysload Studio : oaScSGM

3.1 Agent Properties

Properties	Value
Agent Code (Type)	oaScSGM
Agent Code Number (Type ID)	1107 (automatic)
Agent Name (Label)	oaScSGM
Default Collector Port	9581
Default Real-time Period (seconds)	30 (default)
Default History Period (minutes)	5 (default)
Value Interpolation	Yes
Target Agent Environment - MS Windows	Yes
Target Agent Environment - Linux / Unix	Yes

3.2 Domains / Metrics

Following Domain and metrics :

Domain	Type Domain	Metric ID	Label FR	Label EN	Unit
Site_configuration	Globale	site_config_lat	Site latence	Site latency	-
		site_config_lng	Site lng	Site lng	-
		site_config_maxRateWanLtoR	Débit Max vers WAN (Traffic sortant)	Max Rate to WAN	bps
		site_config_maxRateWanLtoR2	Débit Max vers WAN 2 (Traffic sortant)	Max Rate to WAN 2	bps
		site_config_maxRateWanRtoL	Débit Max depuis WAN (Traffic entrant)	Max Rate from WAN	bps
		site_config_maxRateWanRtoL2	Débit Max depuis WAN 2 (Traffic entrant)	Max Rate from WAN 2	bps
Site_stats	Globale	site_stat_ruleQosType	Site - Type de QoS	Site – Qos Type	-
		site_stat_ruleNbCnx	Site – Nb de connexions	Site – Nb Cnx	nb
		site_stat_ruleRttWan	Site – Délai A/R WAN (RTT/RTD)	Site – WAN Round-Trip-Time Delay (RTT/RTD)	ms
		site_stat_ruleNewTcpDelayTotal	Site – Temps de réponse Total	Site – TCP Delay Total	ms
		site_stat_ruleNewTcpDelayNet	Site – Temps de transfert	Site – TCP Delay Net	ms
		site_stat_ruleOperStat	Site – Etat	Site – Status	0/1

		us			
		site_stat_ruleLtoRavgRate	Site – Débit moy. Local vers Distant	Site – Local to Remote AVG Rate	bps
		site_stat_ruleLtoRmaxRate	Site - Débit max. Local vers Distant	Site - Local to Remote MAX Rate	bps
		site_stat_ruleRtoLavgRate	Site - Débit moy. Distant vers Local	Site - Remote to Local AVG Rate	bps
		site_stat_ruleRtoLmaxRate	Site - Débit max. Distant vers Local	Site - Remote to Local MAX Rate	bps
Rules_configuration	Instance	qosType	Type de QoS	QoS Type	-
		history	Activation historique	Rule history	0/1
		ruleType	Type de règle	Rule Type	-
		voipOptionStatic	Rule voipOptionStatic	Rule voipOptionStatic	-
		isTerm	Règle terminale	Rule isTerm	-
		netflow	Activation netflow	Rule netflow	0/1
Network_Local_traffic	Instance	local_traffic_ruleQosType	Type de QoS	Qos Type	-
		local_traffic_ruleNbCnx	Nb de connexions	Nb Cnx	nb
		local_traffic_ruleRttWan	Délai A/R WAN (RTT/RTD)	WAN Round-Trip-Time Delay (RTT/RTD)	ms
		local_traffic_ruleNewTcpDelayTotal	Temps de réponse Total	TCP Delay Total	ms
		local_traffic_ruleNewTcpDelayNet	Temps de transfert	TCP Delay Net	ms

		local_traffic_ruleOperStatus	Etat	Status	0/1
		local_traffic_ruleLtoRavgRate	Débit moy. Local vers Distant	Local to Remote AVG Rate	bps
		local_traffic_ruleLtoRmaxRate	Débit max. Local vers Distant	Local to Remote MAX Rate	bps
		local_traffic_ruleRtoLavgRate	Débit moy. Distant vers Local	Remote to Local AVG Rate	bps
		local_traffic_ruleRtoLmaxRate	Débit max. Distant vers Local	Remote to Local MAX Rate	bps
Network_Access_link	Instance	access_link_ruleQosType	Type de QoS	Qos Type	-
		access_link_ruleNbCnx	Nb de connexions	Nb Cnx	nb
		access_link_ruleRttWan	Délai A/R WAN (RTT/RTD)	WAN Round-Trip-Time Delay (RTT/RTD)	ms
		access_link_ruleNewTcpDelayTotal	Temps de réponse Total	TCP Delay Total	ms
		access_link_ruleNewTcpDelayNet	Temps de transfert	TCP Delay Net	ms
		access_link_ruleOperStatus	Etat	Status	0/1
		access_link_ruleLtoRavgRate	Débit moy. Local vers Distant	Local to Remote AVG Rate	bps
		access_link_ruleLtoRmaxRate	Débit max. Local vers Distant	Local to Remote MAX Rate	bps
		access_link_ruleRtoLavgRate	Débit moy. Distant vers Local	Remote to Local AVG Rate	bps
		access_link_ruleRtoL	Débit max. Distant vers	Remote to Local	bps

		maxRate	Local	MAX Rate	
Network_Shaping	Instance	shaping_ruleQosType	Type de QoS	Qos Type	-
		shaping_ruleNbCnx	Nb de connexions	Nb Cnx	nb
		shaping_ruleRttWan	Délai A/R WAN (RTT/RTD)	WAN Round-Trip-Time Delay (RTT/RTD)	ms
		shaping_ruleNewTcpDelayTotal	Temps de réponse Total	TCP Delay Total	ms
		shaping_ruleNewTcpDelayNet	Temps de transfert	TCP Delay Net	ms
		shaping_ruleOperStatus	Etat	Status	0/1
		shaping_ruleLtoRavgRate	Débit moy. Local vers Distant	Local to Remote AVG Rate	bps
		shaping_ruleLtoRmaxRate	Débit max. Local vers Distant	Local to Remote MAX Rate	bps
		shaping_ruleRtoLavgRate	Débit moy. Distant vers Local	Remote to Local AVG Rate	bps
		shaping_ruleRtoLmaxRate	Débit max. Distant vers Local	Remote to Local MAX Rate	bps
Network_Groomng	Instance	grooming_ruleQosType	Type de QoS	Qos Type	-
		grooming_ruleNbCnx	Nb de connexions	Nb Cnx	nb
		grooming_ruleRttWan	Délai A/R WAN (RTT/RTD)	WAN Round-Trip-Time Delay (RTT/RTD)	ms
		grooming_ruleNewTcpDelayTotal	Temps de réponse Total	TCP Delay Total	ms

		grooming_ruleNewTcpDelayNet	Temps de transfert	TCP Delay Net	ms
		grooming_ruleOperStatus	Etat	Status	0/1
		grooming_ruleLtoRavgRate	Débit moy. Local vers Distant	Local to Remote AVG Rate	bps
		grooming_ruleLtoRmaxRate	Débit max. Local vers Distant	Local to Remote MAX Rate	bps
		grooming_ruleRtoLavgRate	Débit moy. Distant vers Local	Remote to Local AVG Rate	bps
		grooming_ruleRtoLmaxRate	Débit max. Distant vers Local	Remote to Local MAX Rate	bps
Network_Remote_Shaping	Instance	terminal_shaping_ruleQosType	Type de QoS	Qos Type	-
		terminal_shaping_ruleNbCnx	Nb de connexions	Nb Cnx	nb
		terminal_shaping_ruleRttWan	Délai A/R WAN (RTT/RTD)	WAN Round-Trip-Time Delay (RTT/RTD)	ms
		terminal_shaping_ruleNewTcpDelayTotal	Temps de réponse Total	TCP Delay Total	ms
		terminal_shaping_ruleNewTcpDelayNet	Temps de transfert	TCP Delay Net	ms
		terminal_shaping_ruleOperStatus	Etat	Status	0/1
		terminal_shaping_ruleLtoRavgRate	Débit moy. Local vers Distant	Local to Remote AVG Rate	bps
		terminal_shaping_ruleLtoRmaxRate	Débit max. Local vers Distant	Local to Remote MAX Rate	bps
		terminal_shaping_rule	Débit moy. Distant vers	Remote to Local	bps

		RtoLavgRate	Local	AVG Rate	
		terminal_shaping_ruleRtoLmaxRate	Débit max. Distant vers Local	Remote to Local MAX Rate	bps
Application_Shapping_rule	Instance	application_shapping_ruleNbCnx	Nb de connexions	Nb Cnx	nb
Application_Intermediaire_rule	Instance	intermediaire_ruleQosType	Type de QoS	Qos Type	-
		intermediaire_ruleNbCnx	Nb de connexions	Nb Cnx	nb
		intermediaire_ruleRttWan	Délai A/R WAN (RTT/RTD)	WAN Round-Trip-Time Delay (RTT/RTD)	ms
		intermediaire_ruleNewTcpDelayTotal	Temps de réponse Total	TCP Delay Total	ms
		intermediaire_ruleNewTcpDelayNet	Temps de transfert	TCP Delay Net	ms
		intermediaire_ruleOperStatus	Etat	Status	0/1
		intermediaire_ruleLtoRavgRate	Débit moy. Local vers Distant	Local to Remote AVG Rate	bps
		intermediaire_ruleLtoRmaxRate	Débit max. Local vers Distant	Local to Remote MAX Rate	bps
		intermediaire_ruleRtoLavgRate	Débit moy. Distant vers Local	Remote to Local AVG Rate	bps
		intermediaire_ruleRtoLmaxRate	Débit max. Distant vers Local	Remote to Local MAX Rate	bps
Audio_Video_Terminal_ rule	Instance	terminal_av_ruleQosType	Type de QoS	Qos Type	-
		terminal_av_ruleNbCn	Nb de connexions	Nb Cnx	nb

		x			
		terminal_av_ruleRttWan	Délai A/R WAN (RTT/RTD)	WAN Round-Trip-Time Delay (RTT/RTD)	ms
		terminal_av_ruleNewTcpDelayTotal	Temps de réponse Total	TCP Delay Total	ms
		terminal_av_ruleNewTcpDelayNet	Temps de transfert	TCP Delay Net	ms
		terminal_av_ruleOperStatus	Etat	Status	0/1
		terminal_av_ruleLtoRavgRate	Débit moy. Local vers Distant	Local to Remote AVG Rate	bps
		terminal_av_ruleLtoRmaxRate	Débit max. Local vers Distant	Local to Remote MAX Rate	bps
		terminal_av_ruleRtoLavgRate	Débit moy. Distant vers Local	Remote to Local AVG Rate	bps
		terminal_av_ruleRtoLmaxRate	Débit max. Distant vers Local	Remote to Local MAX Rate	bps
		terminal_av_ruleMOSCQ	MOSCQ	MOSCQ	
		terminal_av_ruleMOSLQ	MOSLQ	MOSLQ	
		terminal_av_ruleVoipLatence	VOIP Latence	VOIP Latency	
		terminal_av_ruleVoipLoss	VOIP Loss	VOIP Loss	
		terminal_av_ruleVoipDiscard	VOIP Discard	VOIP Discard	
		terminal_av_ruleVoipJitter	VOIP Guige	VOIP Jitter	

		terminal_av_ruleVoipConv	VOIP Conv	VOIP Conv	
Application_Terminal_rule	Instance	terminal_data_ruleQosType	Type de QoS	Qos Type	-
		terminal_data_ruleNbCnx	Nb de connexions	Nb Cnx	nb
		terminal_data_ruleRttWan	Délai A/R WAN (RTT/RTD)	WAN Round-Trip-Time Delay (RTT/RTD)	ms
		terminal_data_ruleNewTcpDelayTotal	Temps de réponse Total	TCP Delay Total	ms
		terminal_data_ruleNewTcpDelayNet	Temps de transfert	TCP Delay Net	ms
		terminal_data_ruleOperStatus	Etat	Status	0/1
		terminal_data_ruleLtoRavgRate	Débit moy. Local vers Distant	Local to Remote AVG Rate	bps
		terminal_data_ruleLtoRmaxRate	Débit max. Local vers Distant	Local to Remote MAX Rate	bps
		terminal_data_ruleRtoLavgRate	Débit moy. Distant vers Local	Remote to Local AVG Rate	bps
		terminal_data_ruleRtoLmaxRate	Débit max. Distant vers Local	Remote to Local MAX Rate	bps
Fallback	Instance	terminal_fallback_ruleQosType	Type de QoS	Qos Type	-
		terminal_fallback_ruleNbCnx	Nb de connexions	Nb Cnx	nb
		terminal_fallback_ruleRttWan	Délai A/R WAN (RTT/RTD)	WAN Round-Trip-Time Delay (RTT/RTD)	ms

		terminal_fallback_rule NewTcpDelayTotal	Temps de réponse Total	TCP Delay Total	ms
		terminal_fallback_rule NewTcpDelayNet	Temps de transfert	TCP Delay Net	ms
		terminal_fallback_rule OperStatus	Etat	Status	0/1
		terminal_fallback_rule LtoRavgRate	Débit moy. Local vers Distant	Local to Remote AVG Rate	bps
		terminal_fallback_rule LtoRmaxRate	Débit max. Local vers Distant	Local to Remote MAX Rate	bps
		terminal_fallback_rule RtoLavgRate	Débit moy. Distant vers Local	Remote to Local AVG Rate	bps
		terminal_fallback_rule RtoLmaxRate	Débit max. Distant vers Local	Remote to Local MAX Rate	bps

3.3 Agent Variable

Agent variables used to define a specific setting for a Monitored Object (MO).

These variables will be measured at the installation and / or in the configuration file of the OpenAgent (File <Agent Name>.ini : oaScSGM.ini).

These variables are used by:

- monitoring task

- Plugins Sysload Studio (Perfmon, SNMP ...)

Agent Variable	Description	Default Value	Encrypted Variable	Using by
%(_moname)	Agent Name		-	
%(_installdir)	Agent Installation Directory		-	Monitoring Task
%(mo.SGM_host)	Host or IP SGM		-	Monitoring Task
%(mo.DB_name)	Database SGM		-	Monitoring Task
%(mo.DB_user)	User DB SGM	global	-	Monitoring Task
%(mo.&DB_password&)	Password User DB SGM	none	Yes	Monitoring Task
%(mo.Site)	Site name		-	Monitoring Task
%(mo.SGM_port)	Port CLI SGM	9001	-	Monitoring Task
%(mo.debug)	Activate debug	0	-	Monitoring Task

3.4 Monitoring Scripts and Tasks

Scripts collection of metrics must be:

- Imported in SP Studio

- Attached to the metric

Scheduled task data collection can then be configured.

Monitoring Script	Collected Metrics	Language du Script
GetSCMetrics.pl	All metrics	PERL

Monitoring Tasks	Schedule cycle	Command Line (Windows)	Command Line (Linux/Unix)
GetMetric_SGM	30 secondes	perl "%(_installdir)\GetSCMetrics.pl" -u %(mo.DB_user) -p %(mo.&DB_password&) -h %(mo.SGM_host) -port %(mo.SGM_port) -db %(mo.DB_name) -s %(mo.Site) -debug %(mo.debug) -lib "%(_installdir)" -period %(mo.period_collect)	/usr/bin/perl "%(_installdir)\GetSCMetrics.pl" -u %(mo.DB_user) -p %(mo.&DB_password&) -h %(mo.SGM_host) -port %(mo.SGM_port) -db %(mo.DB_name) -s %(mo.Site) -debug %(mo.debug) -lib "%(_installdir)" -period %(mo.period_collect)

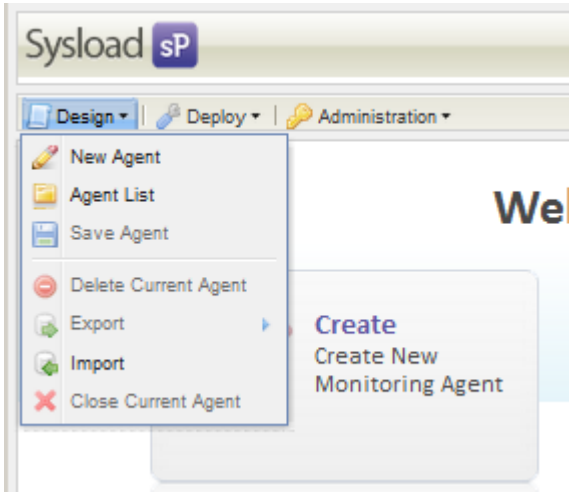
4. Setup

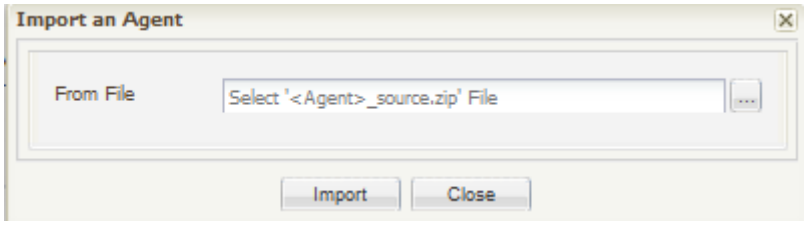
4.1 Load Project source / Generate resource file

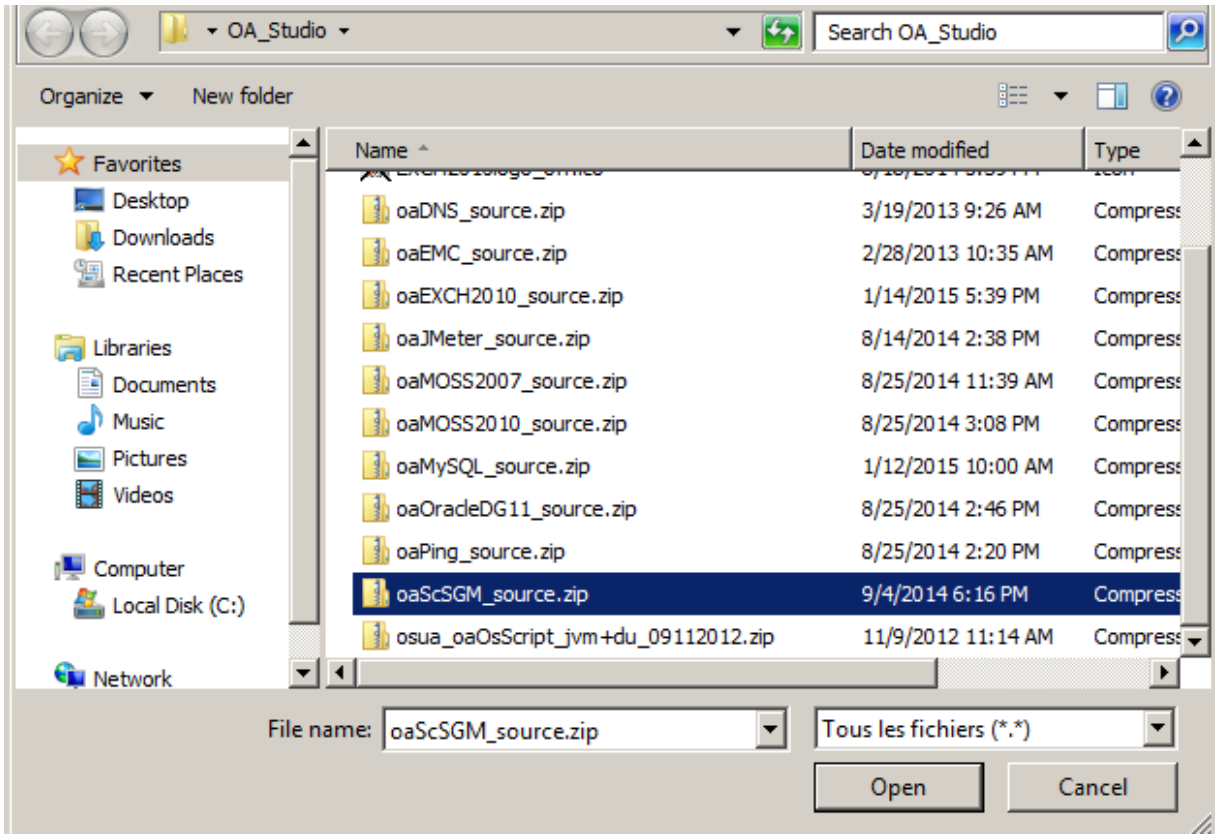
1. Connect on the Web portal Sysload Studio Server (http://<SpsHost>:<SpsPort>)

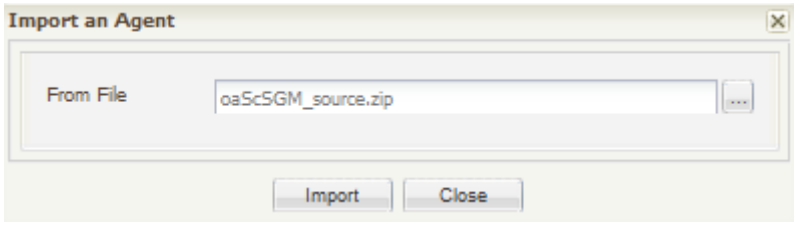
Default password : sysload

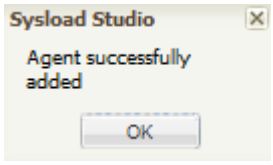
2. Import Open Agent source



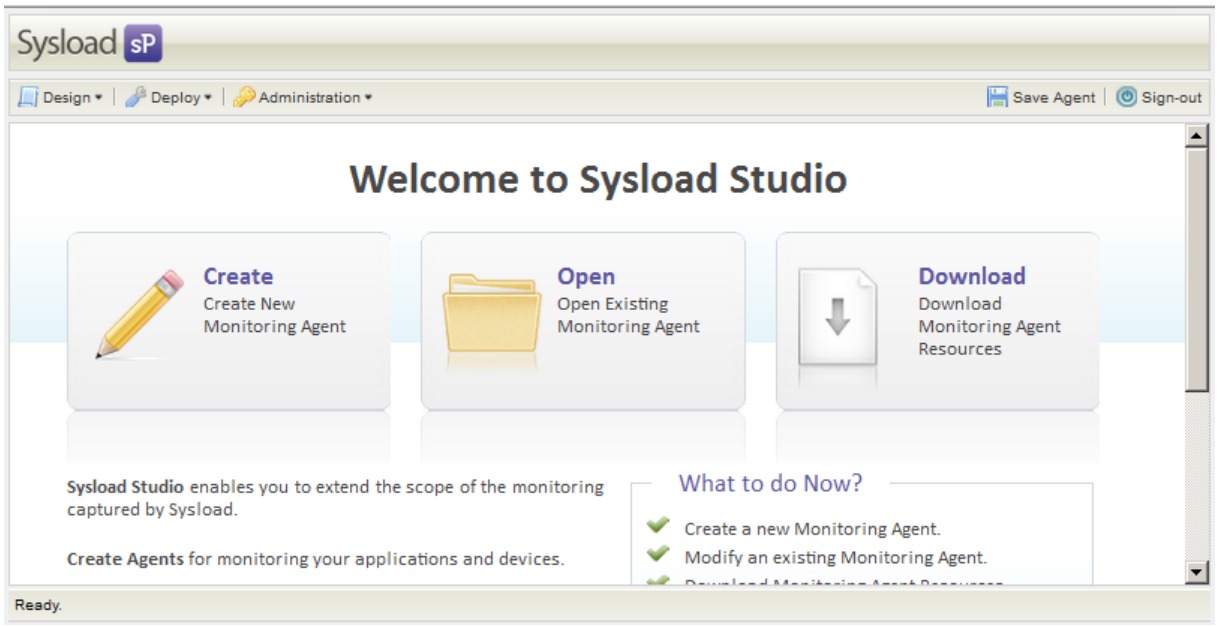




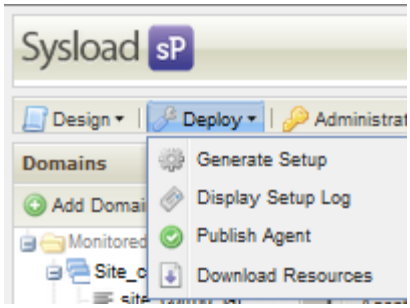


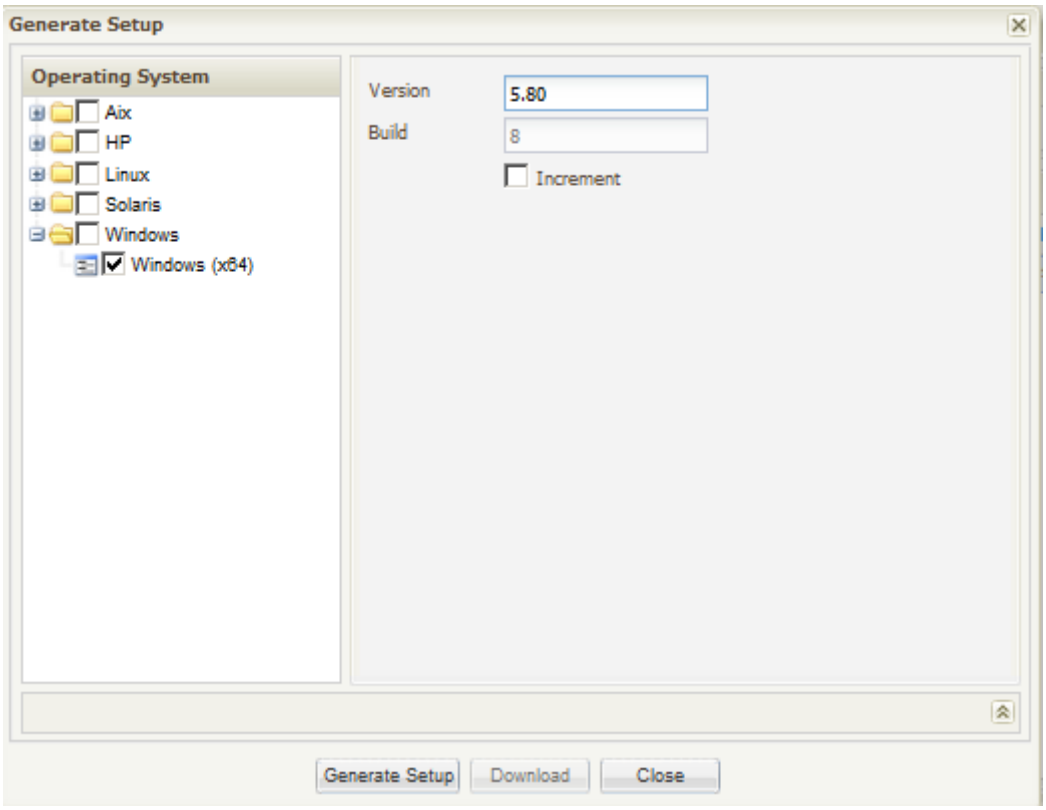


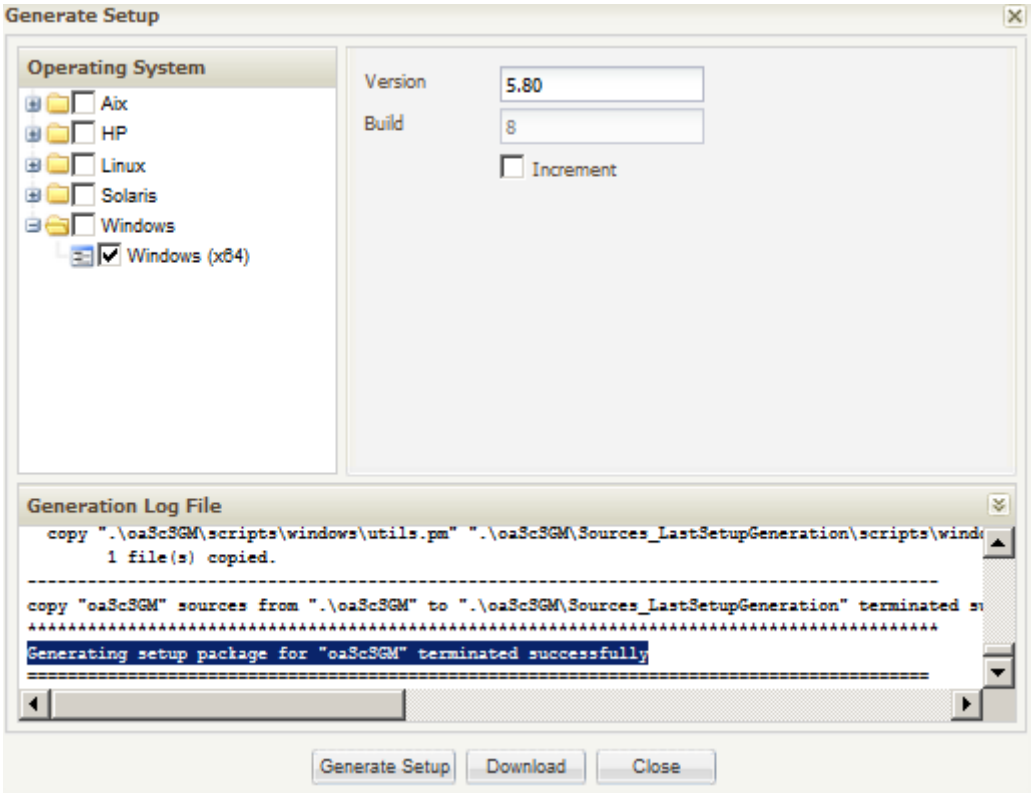
3. Open the project



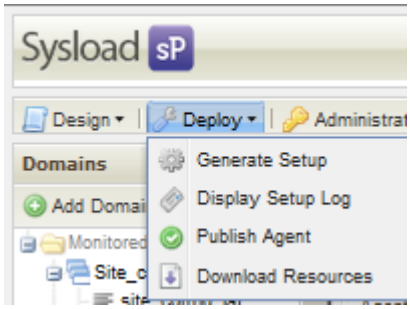
4. Generate the setup

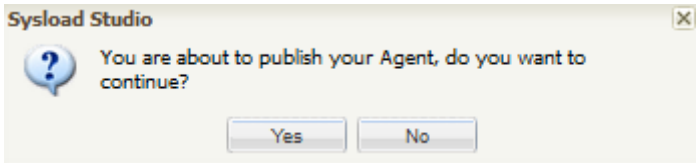


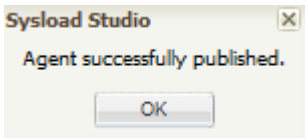




5. Publish the Open agent







4.2 Requirements

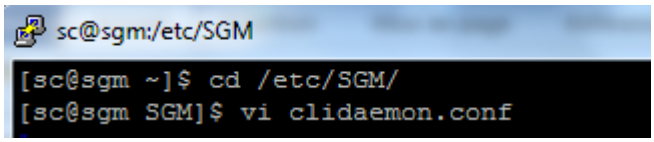
Activate SGM Daemon CLI

Using default configuration, the daemon CLI listener on a local socket (Socket Unix).

1. Open Putty and connect on the SGM as «**sgm** » user:

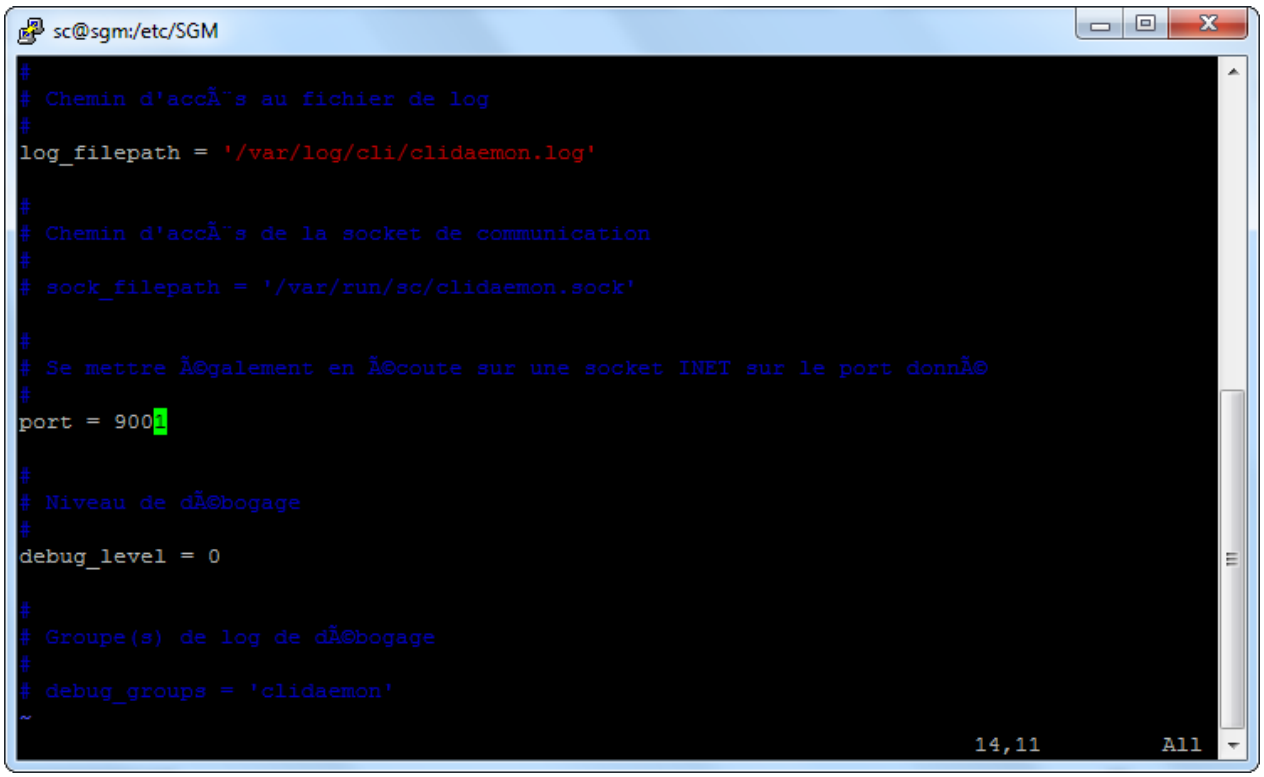
2. Go to Daemon CLI configuration directory, create a backup and edit configuration file : « clidaemon.conf »

```
sgm > cd /etc/SGM

sgm > vi clidaemon.conf
```



3. Uncomment the parameter « port » :

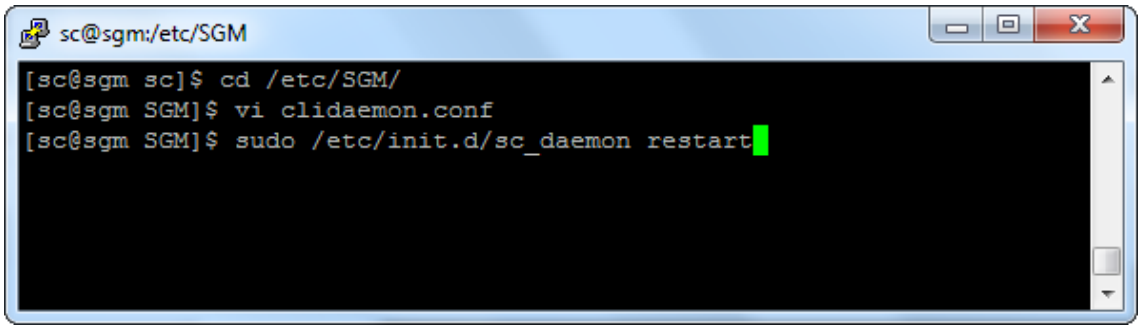
```
# Activation Deamon Cli Port
Port = 9001
```



4. Save the file

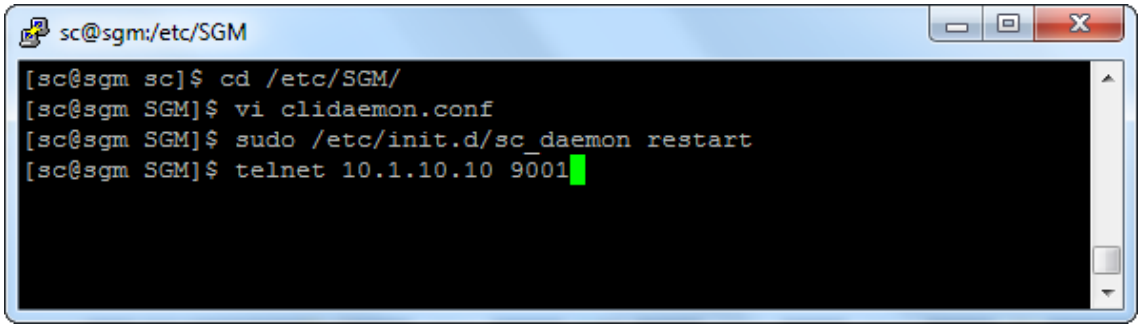
5. Restart the daemon cli

```
sgm > sudo /etc/init.d/sc_daemon restart
```



6. Checking control daemon CLI is listening.

```
sgm > telnet <IP SGM> <port SGM>
```



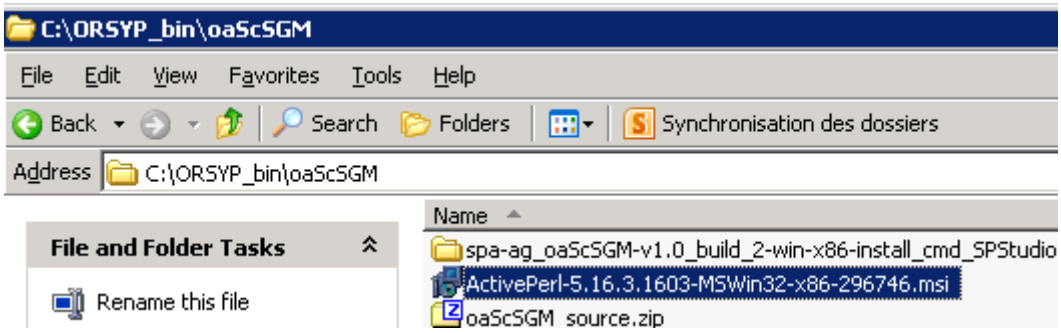
7. Can check Daemon CLI log file

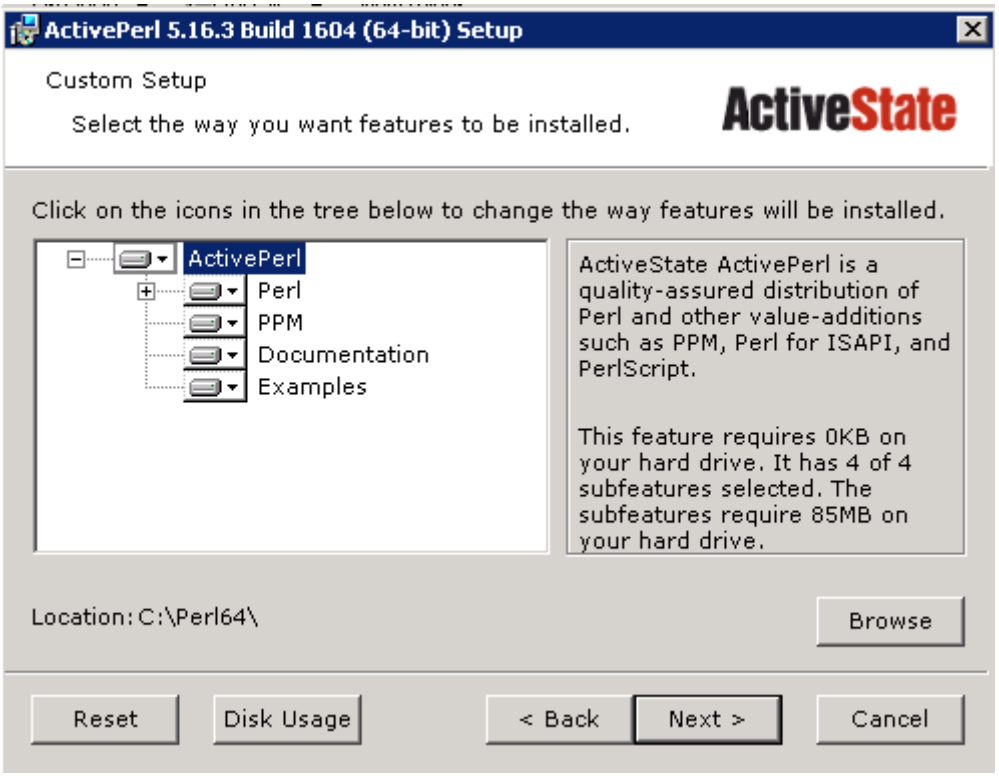
```
sgm > tail –f /var/log/cli/clidaemon.log
```

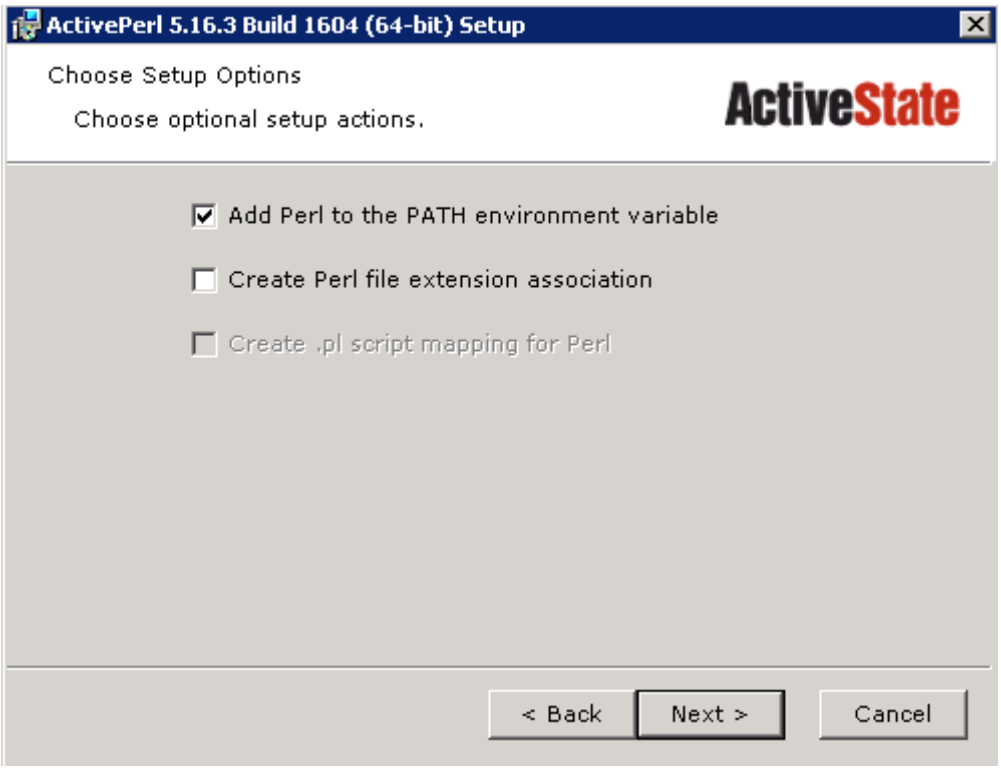
Component: Perl

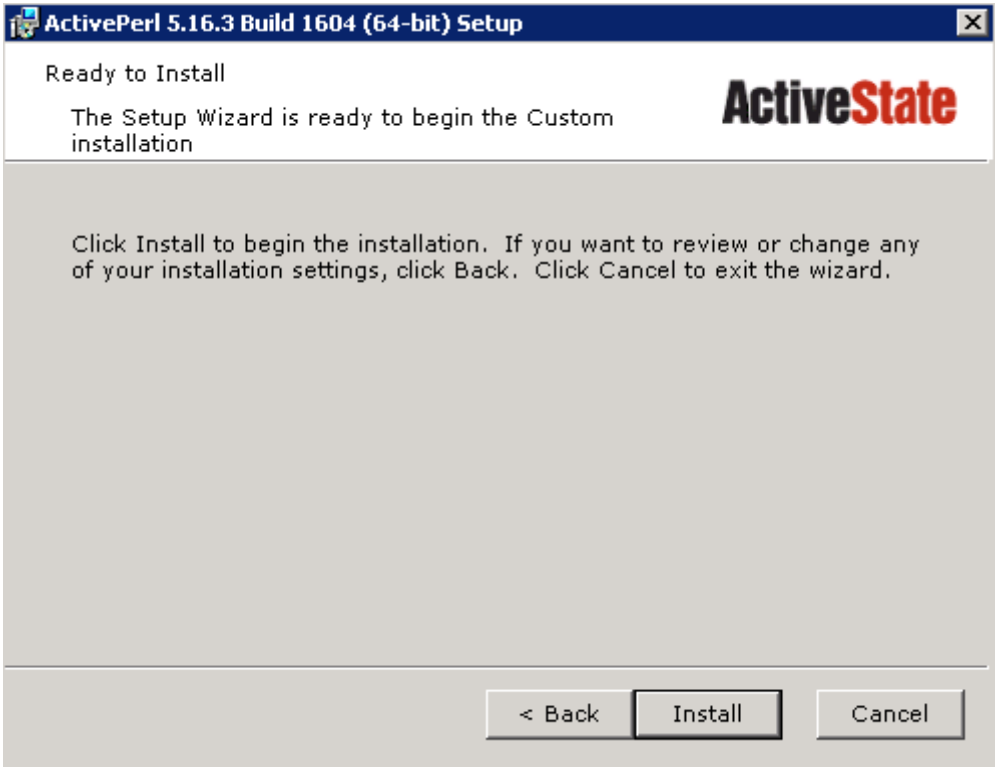
The monitoring script use a script written in Perl language. Perl must be installed. For example : ActivePerl.

Some mandatory library must be added.

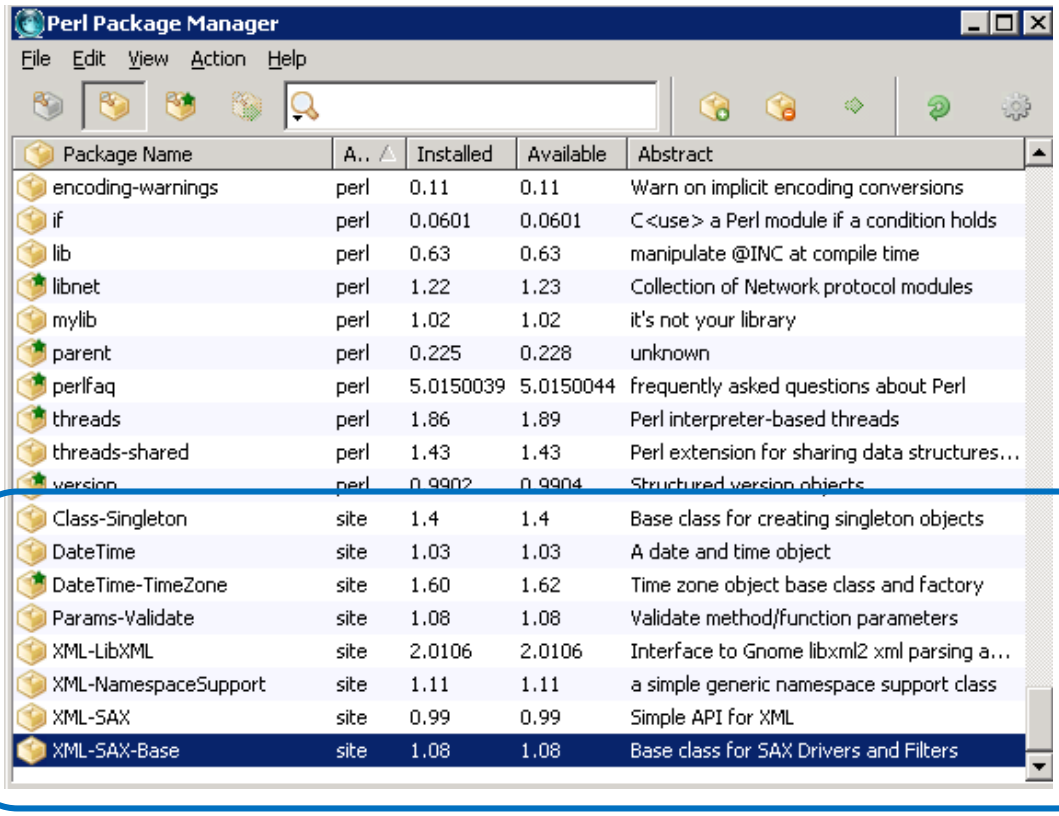






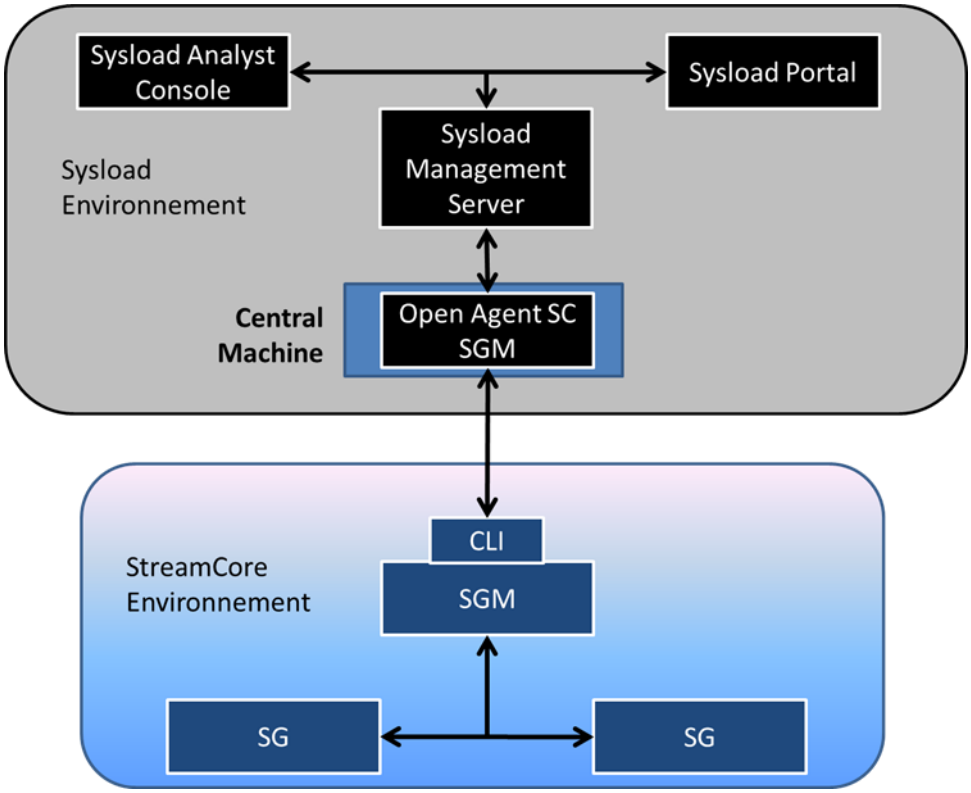


Library Perl	
Add mandatory lib	Class-Singleton DateTime DateTime-Locale DateTime-TimeZone Params-Validate XML-LibXML XML-NamespaceSupport XML-SAX



4.3 Setup OpenAgent oaScSGM

Where install the agent

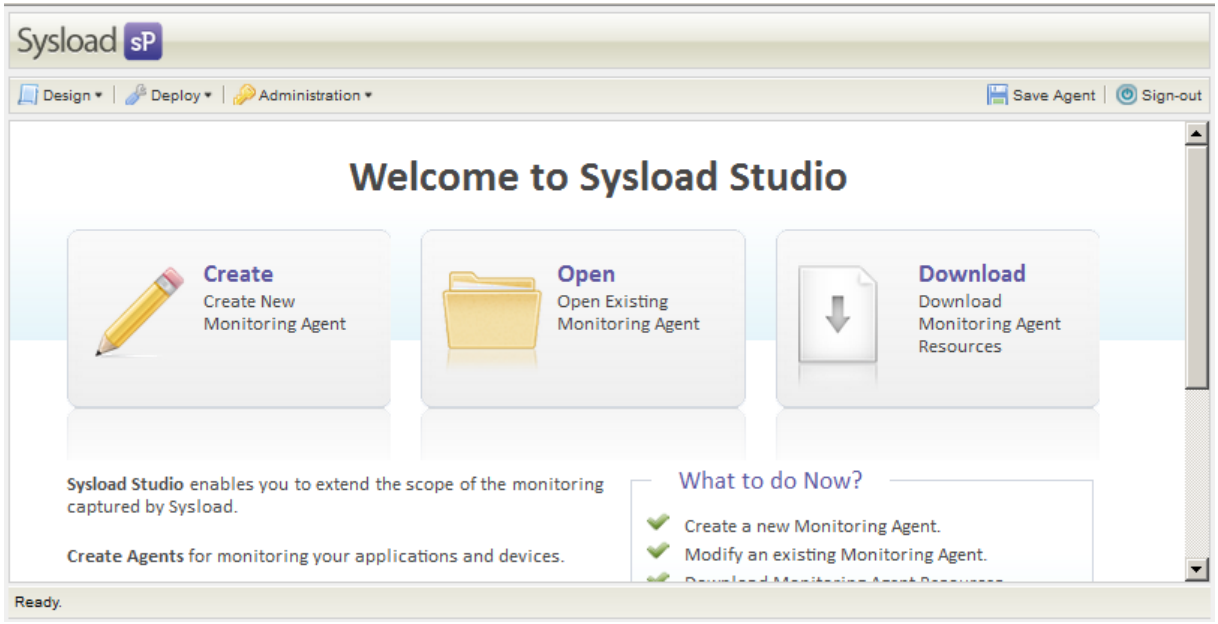


Get installation kit

1. Connect on Sysload Studio Server (http://<SpsHost>:<SpsPort>)

Default password : sysload

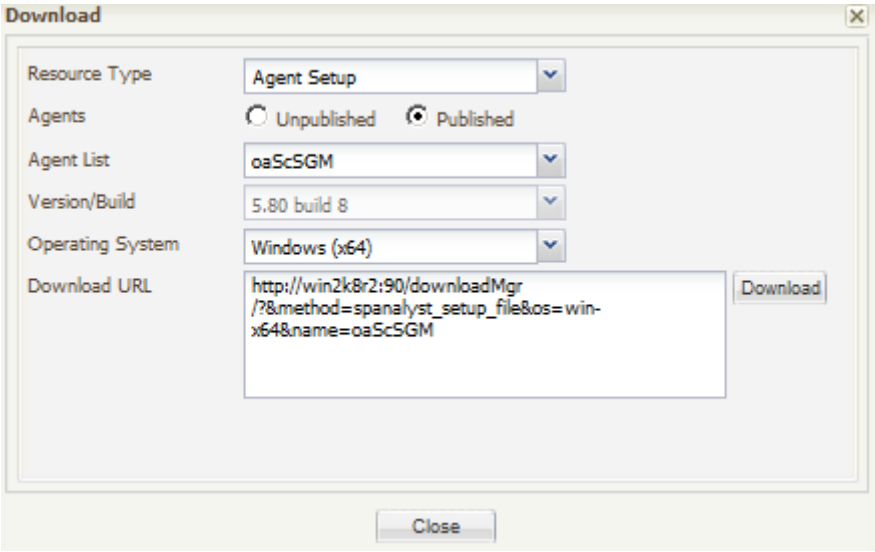
2. Click on « Download » section



3. Select oaScSGM Agent and click on « Download »

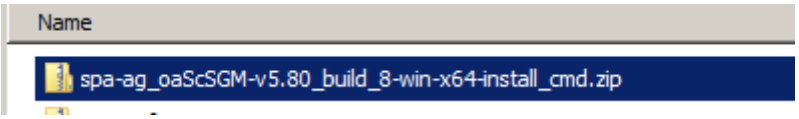
Parameter	Value

Resource type	Agent Setup
Agents	Published
Agent List	oaScSGM
Operating System	Windows (x64)



Setup on MS Windows Platform

1. Check the rules for opening ports on your network device (Firewall and Gateway).
2. Check your token server, if a SA – OPENAGENT token is available
3. Copy the setup package to a temporary directory on the target host



4. Unzip the setup package ***spa-ag_oaScSGM-v5.80_build_8-win-x64-install_cmd.zip***

5. Run command Prompts as **Administrator** and go to the temporary directory:

```
cd /d c:\Sources\spa-ag_oaScSGM-v5.80_build_8-win-x64-install_cmd
```

6. Run Setup command with install parameter

```
.\setup.exe -task install
```

Follow the instruction : <CR> equivalent to carriage return

```
SP Analyst Agent oaScSGM 5.80 (monitoring core version) :Installation (first
install) ...

* Sysload temp directory (parameter 'tmpdir')? [ C:\Program
Files\Sysload\tmp ] : <CR>

    . Checking tmpdir:'C:\Program Files\sysload\tmp'...

* Sysload admin directory (parameter 'admindir')? [ C:\Program Files\Common
Files\sysload\admin ] : <CR>

    . Checking admindir:'c:\Program Files\Common Files\sysload\admin'...

* Sysload home directory (parameter 'homedir')? [ C:\Program Files\Sysload ]
: <CR>

    . Checking homedir:'C:\Program Files\Sysload'...


ORSYP SOFTWARE USER LICENSE
--Enter: '': Display more / 'y': I agree with the License Agreement / 'n': I
do not agree with the License Agreement-- : y

* Sysload Token Server name or IP (parameter 'tkserver')? : <TksHost>

* Sysload Token Server port number (parameter 'tkport')? [ 9555 ] : <CR>

* Enable Automation (y/n) (parameter 'enableautomation')? [ y ] : <CR>

* Sysload File Server ini path (parameter 'FileServerIniFilePath')? [
C:\Program Files\Common Files\sysload\sldfsd\sldfsd.ini ] : <CR>

* Sysload Management Server name or IP ? : <MgtsHost>

* Sysload Management Server port number for Agents ? [ 9901 ] : <CR>
```

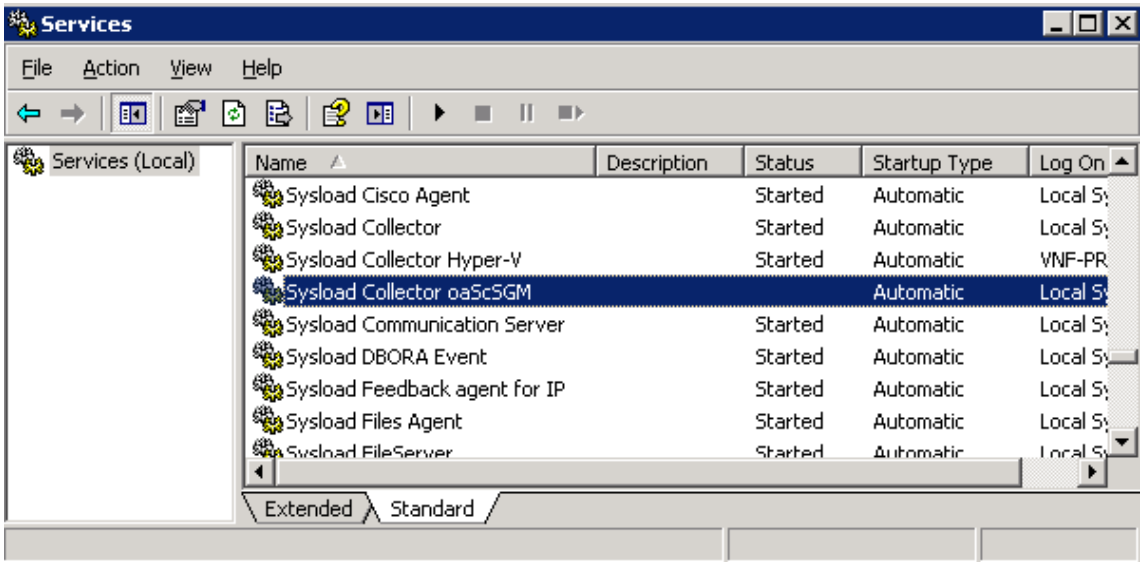
```
* Agent Host for Monitored Object registration {IP Adress, FQDN or "-" if
empty}? [ - ] : <CR>

* MO Groups {"class1.name1|class2.name2| …." Where class=category, system
role or zone}? [ - ] : <CR>
```

```
* List of Monitored Objects to monitor (parameter 'molist')? :
<DB_SGM>_<Site1>,<DB_SGM>_<Site2>

    . Checking user...

* AD Account for service (syntax: LocalSystem | <userdomain>\<username>) [
LocalSystem ] : <USERDOMAIN>\<USERAD>

* Enter Password : *****

* Re-enter Password : *****

* Do you want to configure Sysload services to automatic start mode (y/n) [
y ] : <CR>


* Do you want to configure Sysload services to start at the end of the setup
(y/n)
 [ y ] : <CR>

    . Work in progress...

    . Installing/Updating Sysload files...

* [Monitored object '<DB_SGM>_<Site1>'] Value for parameter 'SGM_Host'? :
<HOST SGM>

* [Monitored object '<DB_SGM>_<Site1>'] Value for parameter 'DB_name'? :
<DB_SGM>

* [Monitored object '<DB_SGM>_<Site1>'] Value for parameter 'DB_user'? :
global

* [Monitored object '<DB_SGM>_<Site1>'] Value for parameter 'DB_password'?
(encrypted) : ****

* [Monitored object '<DB_SGM>_<Site1>'] Value for parameter 'Site'? : <Site
1>

* [Monitored object '<DB_SGM>_<Site1>'] Value for parameter 'SGM_port'? :
9001

* [Monitored object '<DB_SGM>_<Site1>'] Value for parameter 'debug'? : 0

* [Monitored object '<DB_SGM>_<Site2>'] Value for parameter 'SGM_Host'? :
<HOST SGM>

* [Monitored object '<DB_SGM>_<Site2>'] Value for parameter 'DB_name'? :
<DB_SGM>
```

```
* [Monitored object '<DB_SGM>_<Site2>'] Value for parameter 'DB_user'? :
global

* [Monitored object '<DB_SGM>_<Site2>'] Value for parameter 'DB_password'?
(encrypted) : ****

* [Monitored object '<DB_SGM>_<Site2>'] Value for parameter 'Site'? : <Site
2>

* [Monitored object '<DB_SGM>_<Site2>'] Value for parameter 'SGM_port'? :
9001

* [Monitored object '<DB_SGM>_<Site2>'] Value for parameter 'debug'? : 0

    . Starting Sysload services...

SP Analyst Agent oaScSGM 5.80 (monitoring core version) :Installation
(firstinstall) done
```

4.4 Configuration

1. Open the Services Manager
2. Stop the Collector agent service
    - Sysload Collector oaScSGM



3. Go to the Agent install directory, create a backup and edit the agent configuration file ***oaScSGM.ini***

4. Modify following parameters :

- Activate purge agent logs

```
[LogParam]
#Default is cyclic log not activated
```

```
EnableCyclicLog=1

#Size (in KB) of the program log file when the log file is set to be
circular.
CyclicLogMaxFileSizeKb=100000
```

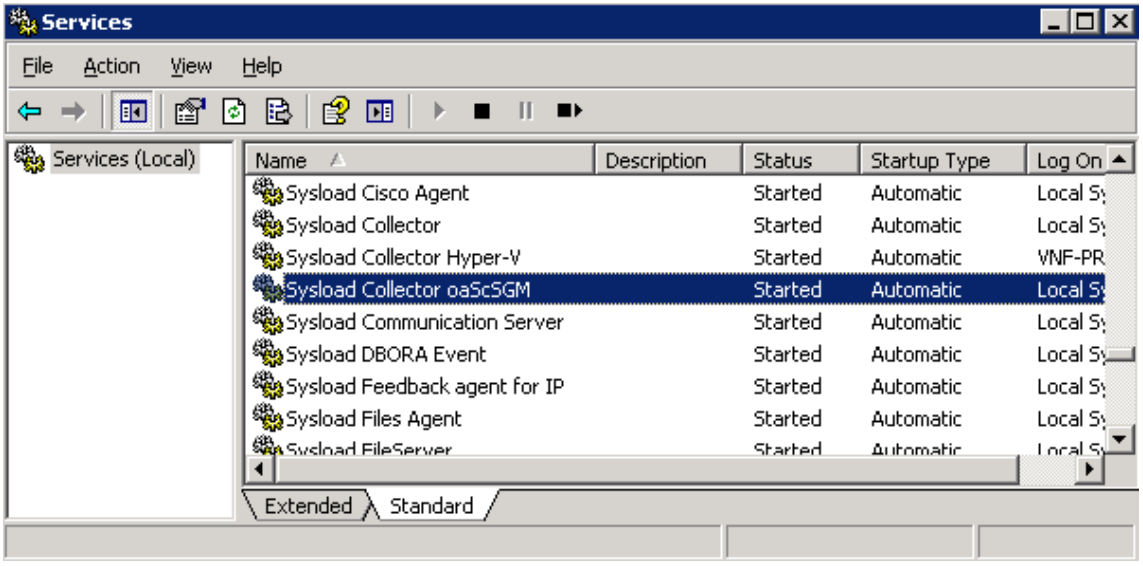
```
[LogParam]
#Default is cyclic log not activated
EnableCyclicLog=1

#Size (in KB) of the program log file when the log file is set to be circular.
CyclicLogMaxFileSizeKb=100000
```

- Activate purge histored data file

```
[LONG_TERM_HIST_PURGE_MONTHLY]
EnableLongTermHistPurge=1
NumberOfMonthsToKeep=13
ExecPurge_DayOfMonth=15
ExecPurge_HourMin=04:00
```

5. Start the Collector agent service

    - Sysload Collector oaScSGM



4.5 Check control

To check the OpenAgent open a webpage with the following address : http://<hostname>:<port listener http>

URL Module OpenAgent : http://<AgentHost>:9581

5. Update the Sysload Server ressource file

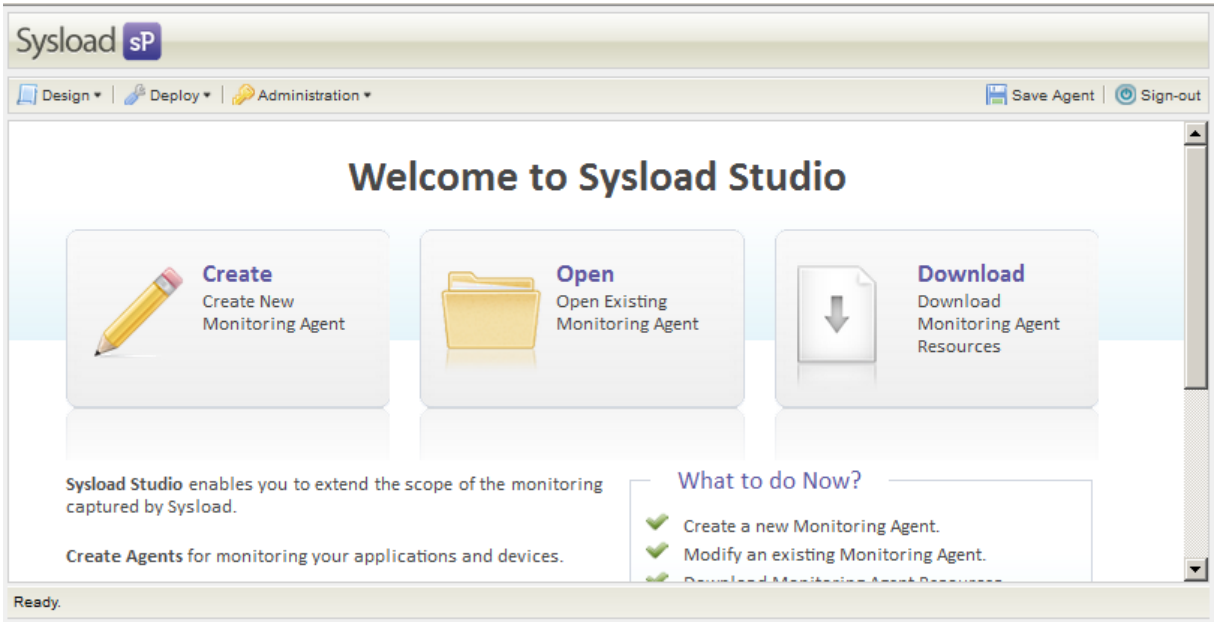
5.1 Get Ressources kits

Get Ressources Files for Management Server

1. Connect on the Sysload Studio Server (http://<SpsHost>:<SpsPort>)

Default password : sysload

2. Click on « Dowload » Section



3. Select oaScSGM Agent and click on « Download »

Parameter	Value
Resource type	Management Server
Agents	Published
Agent List	oaScSGM

Get Ressources Files for Sysload Analyst Console

1. Select oaScSGMAgent and click on « Download »

Parameter	Value
Resource type	Sysload Analyst Console

Agents	Published
Agent List	oaScSGM

Get Ressources Files for Sysload Portal

1. Select oaScSGMAgent and click on « Download »

Parameter	Value
Resource type	Sysload Portal
Agents	Published
Agent List	oaScSGM

5.2 Update process

Update Sysload Management Server ressources files

1. Copy the package in a temporary directory on the Sysload Management Server host

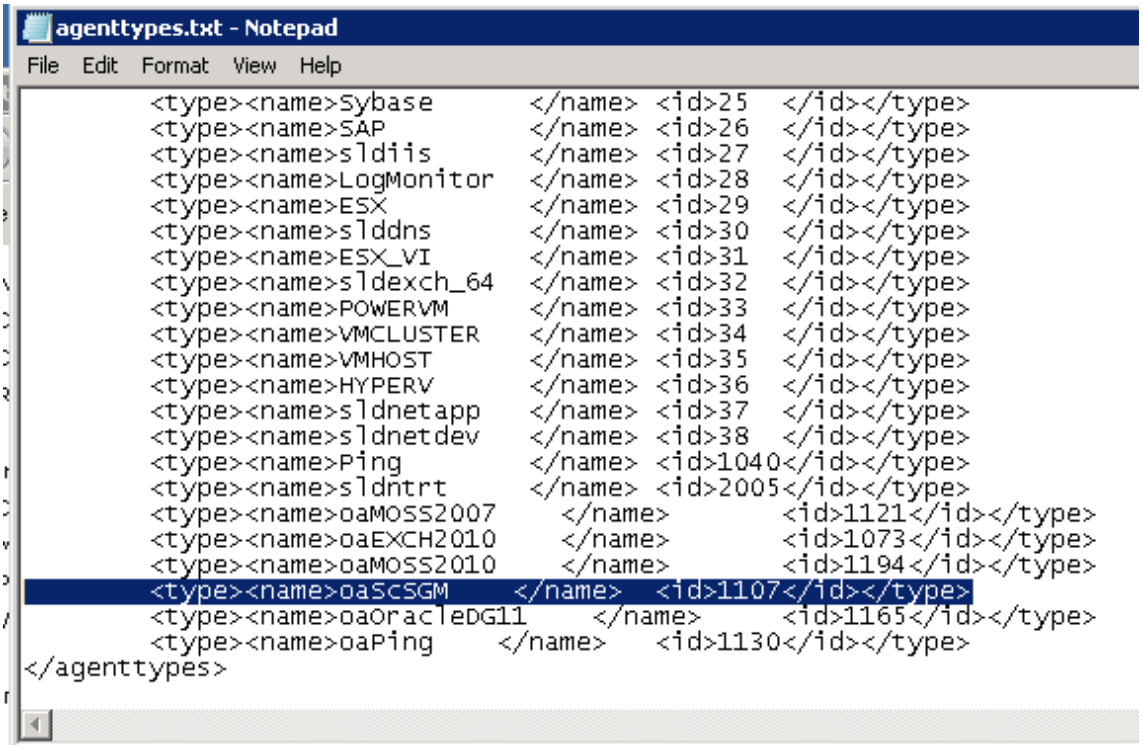
2. Unzip the package ***oaScSGM-v5.80_build_8-addons-mgts.zip***

3. Copy files in the Sysload Management Server ressources agents directory

Example : C:\Program Files\Sysload\sldmgts\resources\agents

4. Modify the Sysload Management Server ressources agents type file

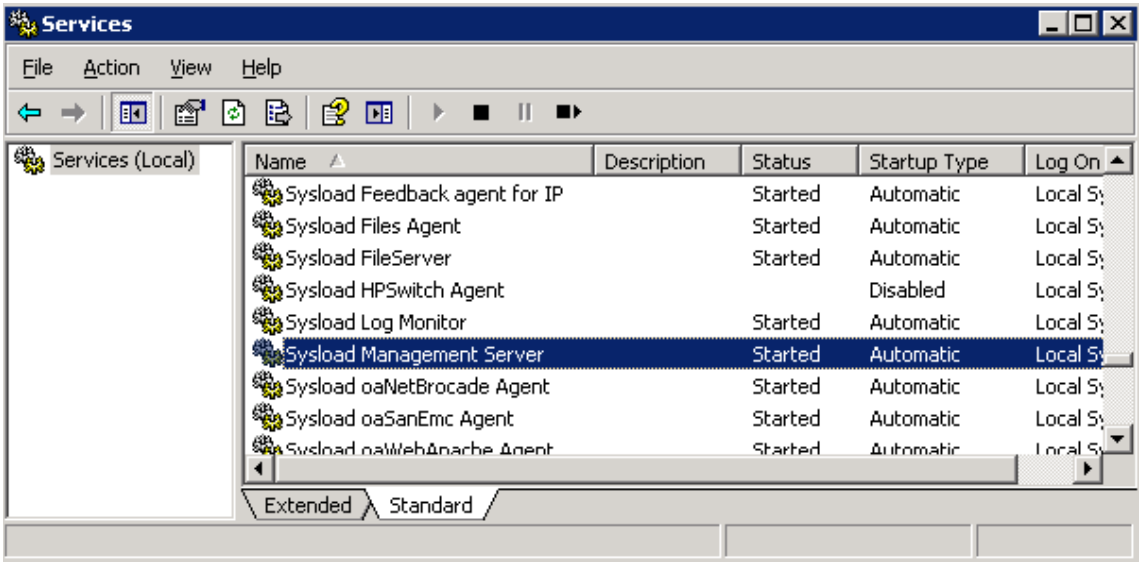
Example : C:\Program Files\Sysload\sldmgts\resources\agenttypes\agenttypes.txt



See Chapter Agent Properties

```
<type><name>oaScSGM     </name>     <id>1107</id></type>
```

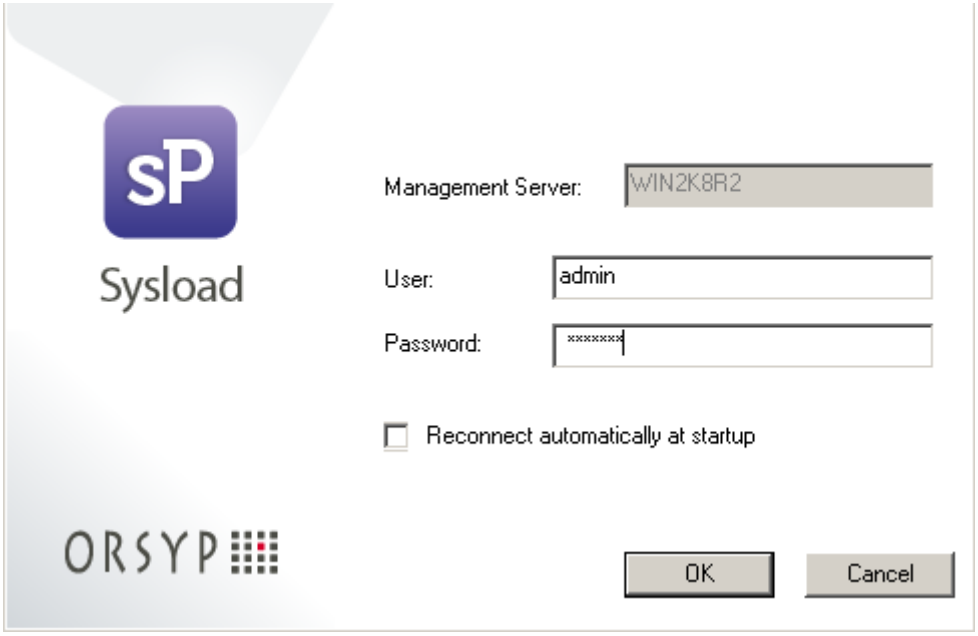
5. Restart the service « Sysload Management Server »

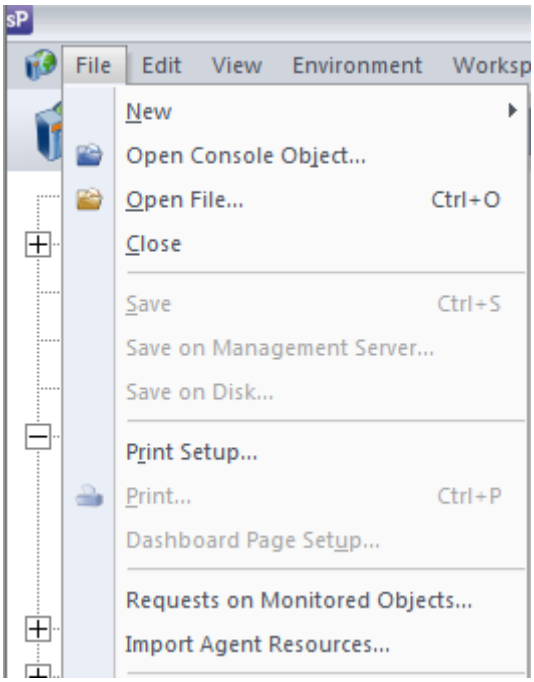


Automic
Professional Services

Update Sysload Analyst Console Ressources Files

Minimum Sysload Console v5.60



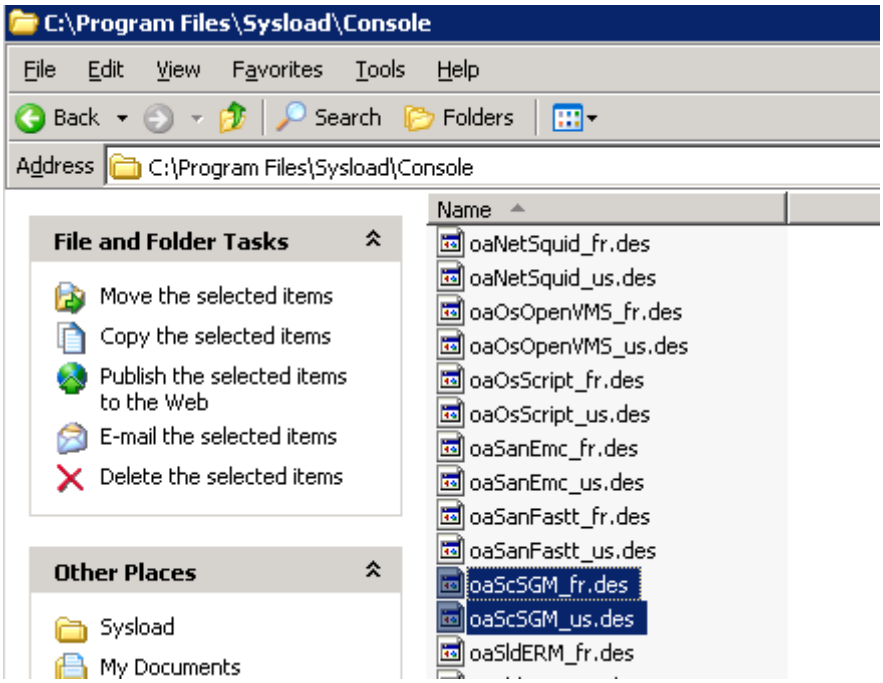


Else manual process to import ressource file :

1. Unzip the package ***oaScSGM-v5.80_build_8-addons-console.zip***

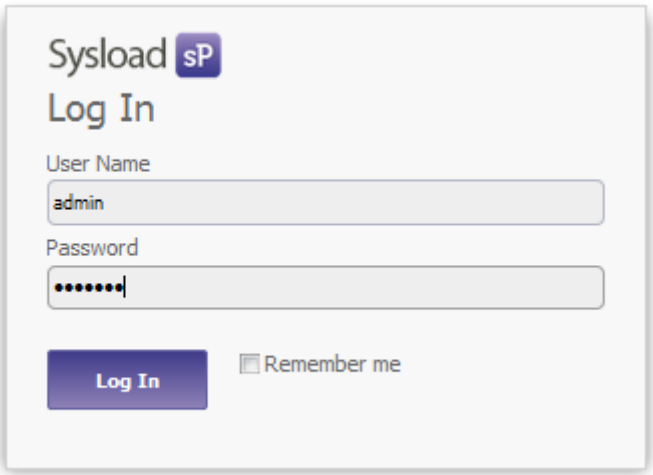
2. Copy files in the Sysload Analyst Console ressources agents directory

Example : C:\Program Files\Sysload\Console

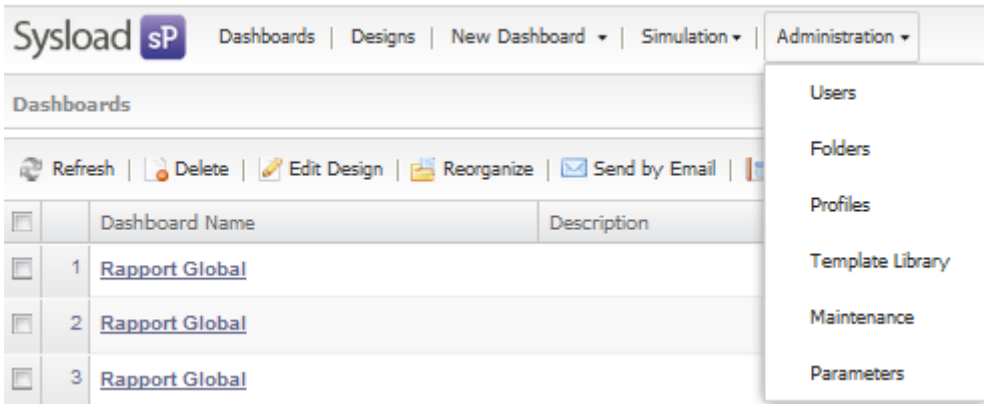


Update Sysload Portal ressources files

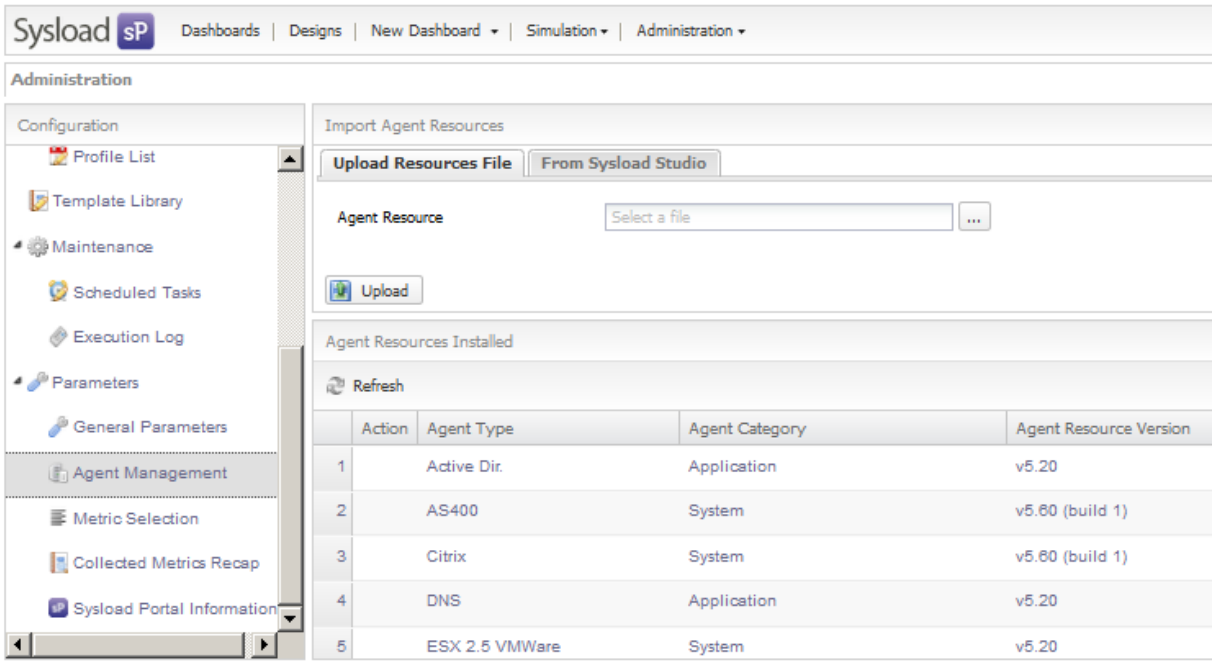
1. Connect to Sysload Portal (http:// <SppHost>:<SppPort>)



2. Click on ***Administration > Parameters***



3. Click on ***Agent Management***



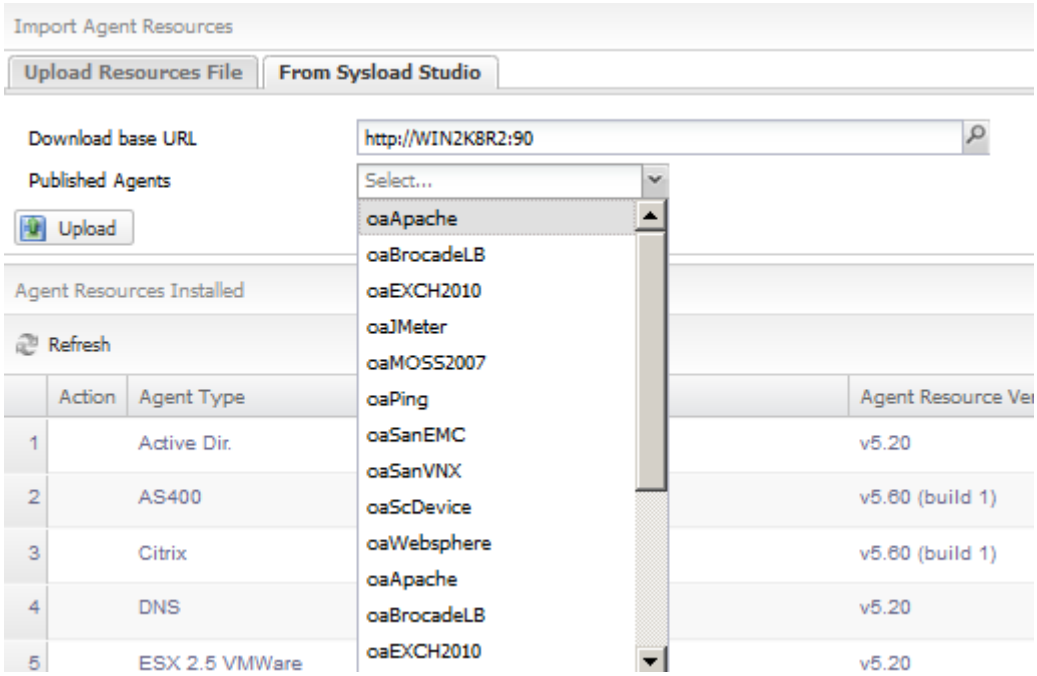
4. Go to ***From Sysload Studio***

   a. *Set the Sysload Studio Server URL* ***: http://<SpsHost>>:<SpsPort>***

   b. *Click on* 

   c. *Select the published Monitored object* ***: oaScSGM***

   d. *Click on* ***Upload***

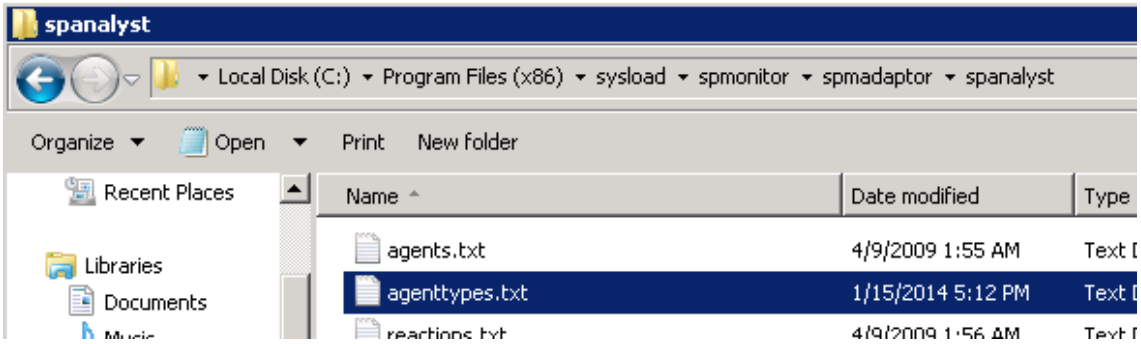


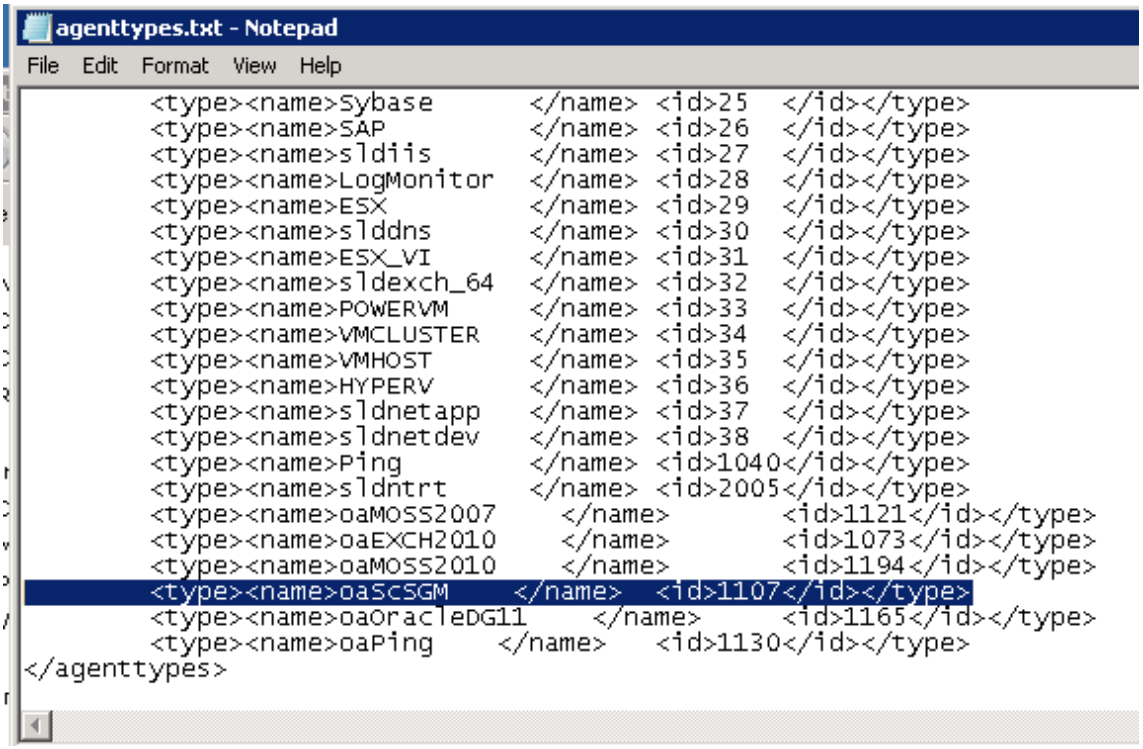
5. The following message is displayed. Click on OK to finish the process

Update Sysload Monitor configuration - Adaptator SPA Event

1. Modify the Adaptor config files

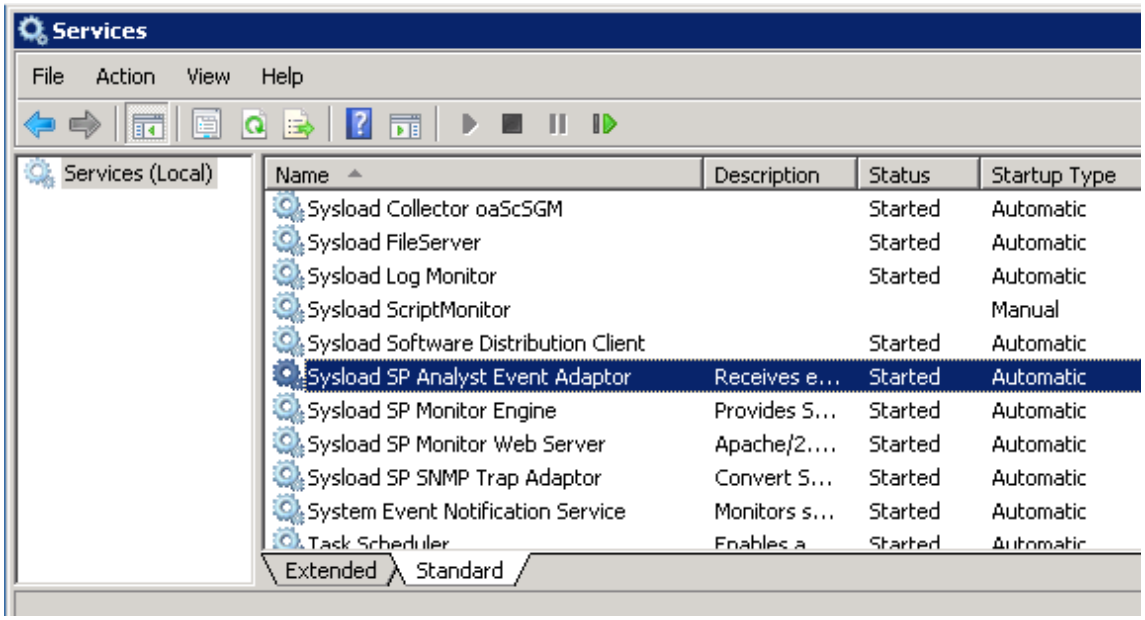
Example : C:\Program Files (x86)\sysload\spmonitor\spmadaptor\spanalyst\agenttypes.txt





See Chapter Agent Properties

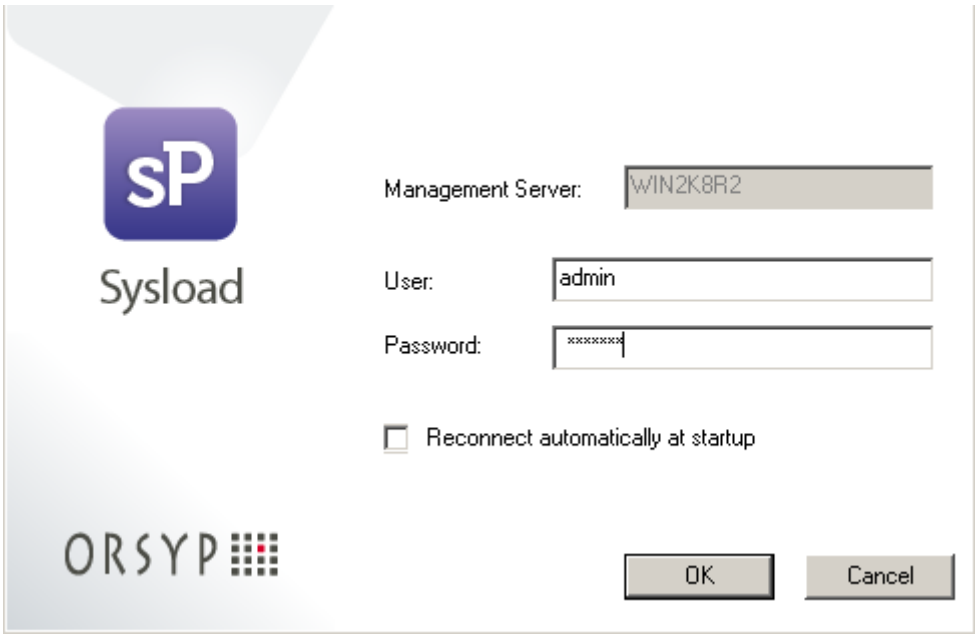
```
<type><name>oaScSGM     </name>     <id>1107</id></type>
```



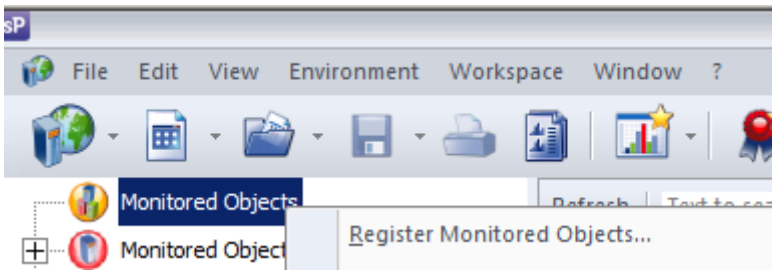
6. Add to the Management Server

Add New Monitored object

1. Verify your firewall policies
2. Close Sysload Analyst Console, if it is opened
3. Open Sysload Analyst Console, (to reload new ressource agent file)



4. In “Monitored Objects” , Click right « Register Monitored Objects …»



5. Enter information, and click on « Apply » :

Parameter	Value

Agent type	oaScSGM
Stations	<AgentHost>
Sysload Portal	Check the case or not

Remark: The number of agent added must match the number of monitored objects

7. Functionnal Parameters Sysload Analyst

7.1 Event models

Example event

Model EVT NAME	EVT name	Priority	Duration (Seconds)	Condition [Domain -- Activity][Instance][Operator][Threshold]	Unit	Message	Type Agent	Agents list

8. Update (New Build)

1. Get the new setup package
2. Unzip the package
3. Run command Prompts as **Administrator** and go to the temporary directory:

```
cd /d c:\Sources\spa-ag_oaScSGM-v5.80_build_8-win-x64-install_cmd
```

4. Run setup command with **patch** parameter

```
.\setup.exe –task patch
```

```
SP Analyst Agent oaScSGM 5.80 (monitoring core version) :Installation
(patch)
 ...

    . Checking homedir:'C:\Program Files\Sysload'...

    . Checking user...

    . Stopping Sysload services...

    . Installing/Updating Sysload files...

    . Starting Sysload services...


SP Analyst Agent oaScSGM 5.80 (monitoring core version) :Installation
(patch) Done
```

5. If new metric : Update Management Server ressources files (see chapter Setup)
6. If new metric : Update Sysload Console ressources files (see chapter Setup)
7. Check the new build in the agent webpage (see chapter Check)

9. Uninstall

1. Run command Prompts as **Administrator** and go to the temporary directory:

```
cd /d c:\Sources\spa-ag_oaScSGM-v5.80_build_8-win-x64-install_cmd
```

2. Run setup command with **uninstall** parameter

```
.\setup.exe -task uninstall
```

Follow the instruction : <CR> equivalent to carriage return

```
SP Analyst Agent oaScSGM 5.80 (monitoring core version) :Uninstallation ...

    . Checking homedir:'C:\Program Files\Sysload'...

    . Stopping Sysload services...

    . Removing Sysload files...

SP Analyst Agent oaScSGM 5.80 (monitoring core version) :Uninstallation done
```

10. ANNEXE

10.1 [Conf] Add new Monitored Object

1. Check your token server, if a SA – OPENAGENT token is available
2. Open the Services Manager
3. Stop the Collector agent service
    - Sysload Collector oaScSGM
4. Go to the agent install directory, create a backup and edit the agent configuration file ***oaScSGM.ini***
5. Modify the following parameters and save the file :

- List Monitored Object : For example, add <NewMo>

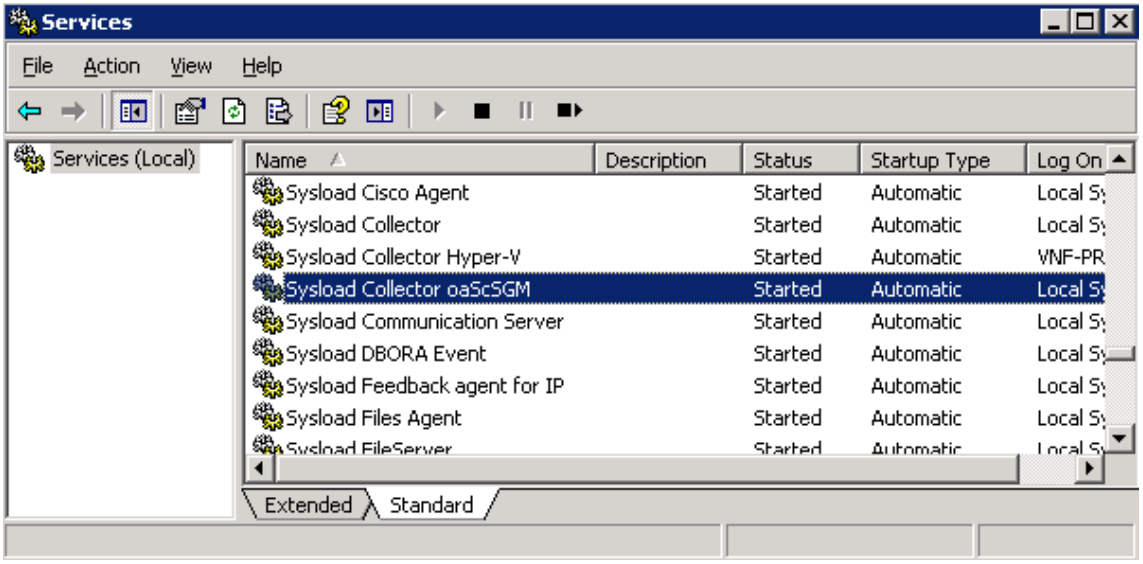
```
[General]
Agents=<List Actual MO>,<NewMO>
```

- Configure the New Monitored Object

```
[Agent_<NewMo>]
SGM_host=<SGM Host or IP>
DB_name=<MyDB_SGM>
DB_user=global
&DB_password&=8ef4a37b67
Site="NewSite"
SGM_port=9001
debug=0
EventIniFile=C:\Program Files\Sysload\oaScSGM\event_<NewMo>.ini
```

6. Start the Collector agent service

- Sysload Collector oaScSGM



7. Check the agent log

10.2 [Script] GetSCMetrics.pl (V1.50)

```
#!/usr/bin/perl
################
#
# Main script for StreamCore SGM Custom Agent
# Script Version 1.50 (September 2014)
#
################

################
# --- LIB
################

use strict;
use warnings;
use DateTime;
use Time::HiRes;
use Getopt::Long qw(:config no_ignore_case);
#use Pod::Usage;
use XML::LibXML;
use XML::LibXML::NodeList;

use charnames ':full';
binmode(STDOUT, ":utf8");

use open qw(:std :utf8);

use less 'please';
use less 'CPU';
use less 'memory';

sub __msg($$) {
  my ($fd, $msg) = @_;

  $fd->print($msg);
}


################
################
# --- ARGUMENTS
################

#msg::activeDebugGroup('main');

my $process_cmdline = $0 . (scalar @ARGV > 0 ? ' '.join(' ', @ARGV) : '');

my $host_SGM = undef;
my $port_SGM = '9001';
my $databases_name = undef;
my $user_db = 'global';
my $password_db = undef;
my $site = undef;
my $libSCdir = undef;
my $collect_period = 10;
```

```
my $exit_value = 0;
my $debug = 0;

GetOptions(     'hostSGM|h=s{,}'        => \$host_SGM,
                'portSGM|port=s{,}'     => \$port_SGM,
                'db=s{,}'       => \$databases_name,
                'userdb|u=s{,}'     => \$user_db,
                'passworddb|p=s{,}'     => \$password_db,
                'site|s=s{,}'       => \$site,
                'libdir|lib=s'      => \$libSCdir,
                'debug=s'       => \$debug,
                'period=s'      => \$collect_period,
        );



###############
###############
# --- SC LIB
###############

use lib '$libSCdir';
use msg;
use bool;
use cli_config;
use client;
use utils qw(timethis);
use result;


# Site name without space
# ~~~~~~~~~~~~~~~~~~~~~~~~~~
my $site_cleaned = $site;
# replace space
$site =~ s/ /_/g;

# INIT files / fichier
# ~~~~~~~~~~~~~~~~~~~~~~~~~~
my $resultfile = "GetScSGM_XML_".$databases_name."_".$site_cleaned.".txt";
my $rulefile = "GetScSGM_ListRules_".$databases_name."_".$site_cleaned.".txt";
my $log_output = "result_output_".$databases_name."_".$site_cleaned.".txt";
#my $sgfile = "GetScSGM_SGStats_".$databases_name."_".$site_cleaned.".txt";

# Debug File
# ~~~~~~~~~~~~~~~~~~~~~~~~~~
if ($debug ne 0) {
        my $log_debug = "oaScSGM_Script_debug_".$databases_name."_".$site_cleaned.".log";
        open LOG_debug, ">".$log_debug or die "could not open $log_debug";
        print LOG_debug "\n $process_cmdline starting ".scalar(localtime)."\n";
        print LOG_debug "OpenAgent\n\n";
}


###############
###############
# --- CONNEXION
###############


if ($debug ne 0) {
        #info("Connexion à la base [$databases_name] ...");
        print LOG_debug "Log in SGM Database [$databases_name] ...\n";
}

my $cli = new client(dbname  => $databases_name,
                user    => $user_db,
```

```
                passwd  => $password_db,
                                        host      => $host_SGM,
                                        port      => $port_SGM);
if ($cli->status ne $CLI_CONNECTION_OK) {
   s_error("Connexion failed on SGM Database '%s': %s", $databases_name, $cli->errorMessage);
   if ($debug ne 0) {
        #info("--> Connexion à la base [$databases_name] ... KO ");
        print LOG_debug "-->Connexion failed on SGM Database  [$databases_name] ... KO \n";
   }

   $exit_value = 1;
}
else
{

  if ($debug ne 0) {
        #info("--> Connexion à la base [$databases_name] ... OK ");
        print LOG_debug "Connexion successful on SGM Database  [$databases_name] ... OK \n";
   }

}




################
# --- REQUEST
################

my $result_Appli;
my $result_Rules;
my $result_Site_Id;
my $result_Site_Config;

#my $result_SG;
#my $result_Optimization;
#my $result_Voip;



if ($debug ne 0) {
        #print "my site is $site";
        print LOG_debug "my site is $site\n";

        timethis sub {
        #### --- Output XML Format
   $result_Appli = $cli->exec("services;site $site;stat --period $collect_period");

        #### --- Output RAWTEXT Format
        $result_Site_Config = $cli->exec("up;up;services;site $site;show");
        $result_Site_Id = $cli->exec("up;up;services;site $site;id");
        $result_Rules = $cli->exec("up;up;services;site $site;options output=rawtext;list --recursive");
        #$result_Optimization = $cli->exec("services;site $site;options output=rawtext;statOptimization --period 10");
        #$result_Voip = $cli->exec("services;site $site;options output=rawtext;voipStat");
        #my $sg=$site;
        #$sg=~ s/ /-/g;
        #$result_SG = $cli->exec("up;up;infrastructure;sg $sg;options output=rawtext;poll");

        }, " ... effectuée en %s", true;

}
else
{
        #### --- Output XML Format
   $result_Appli = $cli->exec("services;site $site;stat --period $collect_period");
```

```
	#### --- Output RAWTEXT Format
	$result_Site_Config = $cli->exec("up;up;services;site $site;show");
	$result_Site_Id = $cli->exec("up;up;services;site $site;id");
	$result_Rules = $cli->exec("up;up;services;site $site;options output=rawtext;list --recursive");
	#$result_Optimization = $cli->exec("services;site $site;options output=rawtext;statOptimization --period 10");
	#$result_Voip = $cli->exec("services;site $site;options output=rawtext;voipStat");
	#my $sg=$site;
	#$sg=~ s/ /-/g;
	#$result_SG = $cli->exec("up;up;infrastructure;sg $sg;options output=rawtext;poll");
}
################
# --- CHECK
################

if ($result_Appli->status eq $CLI_RESULT_OK) {
	if ($debug ne 0) {
		print LOG_debug "[STATS_DATA] Command : $result_Appli->{cmd})\n";
		print LOG_debug "[STATS_DATA] Size du résultat : ".@{$result_Appli->getResult}."\n";
	}
}
else {
	s_error("[STATS_DATA] Request Rule Stat on Site '%s'-- FAILED ", $site);
	for my $error (@{$result_Appli->errors}) {
		s_error("[STATS_DATA] Error Message : ".$error->{raw});
	}
	$exit_value = 1;
}

if ($result_Site_Config->status eq $CLI_RESULT_OK) {
	if ($debug ne 0) {
		print LOG_debug "[SITE_CONFIG] Command : $result_Site_Config->{cmd})\n";
		print LOG_debug "[SITE_CONFIG] Size du résultat : ".@{$result_Site_Config->getResult}."\n";
	}
}
else {
	s_error("[SITE_CONFIG] Request Site Stat on Site '%s'-- FAILED ", $site);
	for my $error (@{$result_Site_Config->errors}) {
		s_error("[SITE_CONFIG] Error Message : ".$error->{raw});
	}
	$exit_value = 1;
}

if ($result_Site_Id->status eq $CLI_RESULT_OK) {
	if ($debug ne 0) {
		print LOG_debug "[SITE_ID] Command : $result_Site_Id->{cmd})\n";
		print LOG_debug "[SITE_ID] Size du résultat : ".@{$result_Site_Id->getResult}."\n";
	}
}
else {
	s_error("[STATS_DATA] Request Site ID on Site '%s'-- FAILED ", $site);
	for my $error (@{$result_Site_Id->errors}) {
		s_error("[SITE_ID] Error Message : ".$error->{raw});
	}
	$exit_value = 1;
}

if ($result_Rules->status eq $CLI_RESULT_OK) {
	if ($debug ne 0) {
		print LOG_debug "[RULES] Command : ($result_Rules->{cmd})\n";
		print LOG_debug "[RULES] Size du résultat : ".@{$result_Rules->getResult}."\n";
	}
}
else {
	s_error("[RULES] Request Service List Definition on Site '%s'-- FAILED ", $site);
	for my $error (@{$result_Rules->errors}) {
		s_error("[RULES] Error Message : ".$error->{raw});
```

```
        }
        $exit_value = 1;
}


#if ($result_SG->status eq $CLI_RESULT_OK) {
#       if ($debug ne 0) {
        #print LOG_debug "[SG] Command : ($result_SG->{cmd})\n";
        #print LOG_debug "[SG] Taille du résultat : ".length($result_SG->getResult)."\n";
#       }
#}
#else {
#       s_error("[SG] Request Groomer Stat on Site '%s'-- FAILED = %s",$site);
#       for my $error (@{$result_SG->errors} ) {
#    s_error($error->{raw});
#  }
#  $exit_value = 1;
#}

################
# --- OPENAGENT
################

        # fichier de resultat output
        # ~~~~~~~~~~~~~~~~~~~~~~~~~~

        if ($debug ne 0) {
                open LOG_output, ">".$log_output or die "could not open $log_output";
                print LOG_output "openagent";
                print LOG_output "\n\n";
        }
        print "openagent";
        print "\n";
        print "\n";

################


################
# --- Metric Site Config
################

if ($result_Site_Config->status eq $CLI_RESULT_OK) {

        if ($debug ne 0) {
                print "[CONFIG] Command : ($result_Site_Config->{cmd})\n";
                #print "[CONFIG] Taille du résultat : ".length($result_Site_Config->getResult)."\n";
                #print "[CONFIG] Size du résultat : ".@{$result_Site_Config->getResult}."\n";
        }
        my @site_config = @{$result_Site_Config->getResult};
        if ($debug ne 0) {
                print "[CONFIG] Size du résultat : ".@site_config."\n";
        }
        foreach my $item (@site_config) {

                if ( $item =~ /(.*lat.*|.*Rate.*|.*lng.*)(=)(.*)/) {
                #if ( $item =~ /(.*)(=)([0-9])/) {
                                if ($debug ne 0) {
                                        if ($3 eq "unlimited") {
                                                printf LOG_output "site_config_".$1."###99999999\n";
                                        }
                                        else {
                                                printf LOG_output "site_config_%s###%s\n", $1, $3;
                                        }
                                }
                                else
```

```
                                        {
                                                if ($3 eq "unlimited") {
                                                        print "site_config_".$1."###99999999\n";
                                                }
                                                else {
                                                        print "site_config_".$1."###".$3."\n";
                                                }
                                        }
                                }
                        }
}
else {
        for my $error (@{$result_Site_Config->errors}) {
                s_error($error->{raw});
   }
   $exit_value = 1;
}


################
# --- Metric Site Stat
################

if ($result_Appli->status eq $CLI_RESULT_OK) {

        if ($debug ne 0) {
                print "[STAT] Command : ($result_Appli->{cmd})\n";
                #print "[STAT] Taille du résultat : ".length($result_Appli->getResult)."\n";
                print "[STAT] Size du résultat : ".@{$result_Appli->getResult}."\n";
        }

        # ~~~~~~~~~~~~~~~~~~~~~~~~~~~
        # fichier de resultat XML de la request CLI
        # ~~~~~~~~~~~~~~~~~~~~~~~~~~~
        #

        open XML, ">".$resultfile or die "could not open $resultfile";
   print XML $result_Appli->getResult();
        XML->autoflush();
        close XML;

        # ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        # fichier de resultat Rules list
        # ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        open LIST_RULE, ">".$rulefile or die "could not open $rulefile";
        print LIST_RULE $result_Rules->getResult();
        LIST_RULE->autoflush();
        close LIST_RULE;


        # ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        # fichier de resultat SG list
        # ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        #open LIST_SG, ">".$sgfile or die "could not open $sgfile";
        #print LIST_SG $result_SG->getResult();
        #LIST_SG->autoflush();
        #close LIST_SG;


        # ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        # recupere Id du site
        # ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        my $sitelid = "381";
        my $siteid = "44047";
        my $checksiteid = $result_Site_Id->getResult()->[0];
        if ($debug ne 0) {
```

```
		print "Site ID => $checksiteid \n\n";
		print LOG_debug "Site ID => $checksiteid \n\n";
}
if ( $checksiteid =~ /(site )([\d]+)\s([\d]+)/) {
		$siteid = $2;
		$sitelid = $3;
		# replace space
		$siteid =~ s/ //g;
		$sitelid =~ s/ //g;
		if ($debug ne 0) {
			print "Site ID => ID : $siteid and LID $sitelid \n\n";
			print LOG_debug "Site ID => ID : $siteid and LID $sitelid \n\n";
		}
}
else {
		print "WARNING can not get the Site ID => $site \n\n";
		print LOG_debug "WARNING can not get the Site ID => $site \n\n";
		$exit_value = 1;
}



######################

# --- metric get in Rules file ---
my $qostype;
my $history;
#my $tplgrpid;
#my $tplid;
my $ruletype;
my $voipOptionStatic;
my $isterm;
my $netflow;

# --- metric get in Rules stats ---
my $rulename;
my $rulelid;
my $ruleid;
my $intermediate_rulename;
my $local_traffic_name_forFallback;
my $access_link_name_forFallback;
my $shapping_name_forFallback;
#my @rulenameForFallback;
#my $ruleFallbackName;

# --- metric For Application - Shapping Site : get in Rules stats ---
my $shapping_name_forOtherSite;
my $shapping_application_name;
my %hash_appl_nbcnx;

# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Parse resultat XML LibXML
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~

my $parser = XML::LibXML->new();
$parser->recover(1);
my $doc = $parser->parse_file($resultfile) or die "could not open $resultfile";
#my $root = $doc->documentElement;
#--print "\n root = $root \n";

#my $rootelement = $doc -> getDocumentElement();
#--$elementname = $rootelement -> nodeName();
#my $elementname = $rootelement -> getName();
#--print "Root name=$elementname\n";
```

```
	my $nodes = $doc->findnodes('//*');
	foreach my $node ($nodes->get_nodelist) {

		#my $nbchildren;
		#for ($node->findnodes('*')->size) {
		#	$nbchildren = "$_ Properties or Metrics";
		#}
		if ($node->nodeName() eq "rule" ) {

			$qostype=undef;
			$history=undef;
			#$tplgrpid=undef;
			#$tplid=undef;
			$ruletype=undef;
			$voipOptionStatic=undef;
			$isterm=undef;
			$netflow=undef;

			##### ------------------------------------------ #####
			##### ------------ RULE PROPERTIES -------------- #####
			##### ------------------------------------------ #####

			#printf "%s => %s : \t has %s\n", $node->localname, $node->{name}, $nbchildren;
			$rulename = $node->{name};
			$rulelid = $node->{lid};
			$ruleid = $node->{id};

			if ($debug ne 0) {
				#printf LOG_output "\n[%s - %s] ---- %s ---- \n", $rulelid, $ruleid, $rulename;
				#printf LOG_debug "Site ID => ID : $siteid and LID $sitelid \n\n";
				printf LOG_debug "\n[%s - %s] ---- %s ---- \n", $rulelid, $ruleid, $rulename;
			}

			open LIST_RULE, "<$rulefile" or die "Can't find file : $rulefile";
			while(my $line = <LIST_RULE>) {
				# Format
				if ( $line =~ /(rule)( $rulelid)( $ruleid)( qosType=)(.*)( history=)(.*)( tplGrpId=)(.*)( tplId=)(.*)(
ruleType=)(.*)( voipOptionStatic=)(.*)( isTerm=)(.*)( netflow=)(.*)/){
					$qostype=$5;
					$history=$7;
					#$tplgrpid=$9;
					#$tplid=$11;
					$ruletype=$13;
					$voipOptionStatic=$15;
					$isterm=$17;
					$netflow=$19;
					if ($isterm eq 0) {
						$intermediate_rulename = $rulename;
					}

					if ($debug ne 0) {
						printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "qosType",
$rulename, $qostype;
						printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "history",
$rulename, $history;
						#printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "tplGrpId",
$rulename, $tplgrpid;
						#printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "tplId",
$rulename, $tplid;
						printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "ruleType",
$rulename, $ruletype;
						printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid,
"voipOptionStatic", $rulename, $voipOptionStatic;
						printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "isTerm",
$rulename, $isterm;
```

```
                                        printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "netflow",
$rulename, $netflow;
                                }
                                print "qosType#".$rulename."##".$qostype."\n";
                                print "history#".$rulename."##".$history."\n";
                                #print "tplGrpId#".$rulename."##".$tplgrpid."\n";
                                #print "tplId#".$rulename."##".$tplid."\n";
                                print "ruleType#".$rulename."##".$ruletype."\n";
                                print "voipOptionStatic#".$rulename."##".$voipOptionStatic."\n";
                                print "isTerm#".$rulename."##".$isterm."\n";
                                print "netflow#".$rulename."##".$netflow."\n";
                        }
                        # Format
                        elsif ( $line =~ /(rule)( $rulelid)( $ruleid)( qosType=)(.*)( history=)(.*)( ruleType=)(.*)(
voipOptionStatic=)(.*)( isTerm=)(.*)( netflow=)(.*)/){
                                $qostype=$5;
                                $history=$7;
                                $ruletype=$9;
                                $voipOptionStatic=$11;
                                $isterm=$13;
                                $netflow=$15;
                                if ($isterm eq 0) {
                                        $intermediate_rulename=$rulename;
                                }
                                if ($debug ne 0) {
                                        printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "qosType",
$rulename, $qostype;
                                        printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "history",
$rulename, $history;
                                        printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "ruleType",
$rulename, $ruletype;
                                        printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid,
"voipOptionStatic", $rulename, $voipOptionStatic;
                                        printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "isTerm",
$rulename, $isterm;
                                        printf LOG_output "[%s - %s] %s#%s##%s\n", $rulelid, $ruleid, "netflow",
$rulename, $netflow;
                                }
                                print "qosType#".$rulename."##".$qostype."\n";
                                print "history#".$rulename."##".$history."\n";
                                print "ruleType#".$rulename."##".$ruletype."\n";
                                print "voipOptionStatic#".$rulename."##".$voipOptionStatic."\n";
                                print "isTerm#".$rulename."##".$isterm."\n";
                                print "netflow#".$rulename."##".$netflow."\n";
                        }
                }
                close LIST_RULE;

        } elsif ($node->nodeName() ne "stat" ) {

                ##### ------------------------------- #####
                ##### ------------ STAT ------------- #####
                ##### ------------------------------- #####

                if($ruleid) {
                        my $metric = $node->localname;
                        # replace
                        $metric=~ s/10s//g;

                        $metric=~ s/60s//g;
                        $metric=~ s/1min//g;

                        $metric=~ s/600s//g;
                        $metric=~ s/10min//g;

                        if ($debug ne 0) {
```

```
                                        #printf LOG_output "%s -> %s and %s => %s\n", $rulelid, $sitelid, $ruleid, $siteid;
                                }
                                ########

                                if ($metric eq "ruleFallback") {
                                        #my $flagfallback=$node->textContent;

                                        ## if is not fallback rule and it is local or access link rule ==> the rule have a fallback
rule ==> save the rulename
                                        #if ( $flagfallback eq 0 && defined $ruletype && $ruletype eq 200) {
                                        #       push @rulenameForFallback, $rulename;
                                        #
                                        #
                                        #       if($debug ne 0) {
                                        #               printf LOG_debug "[%s - %s -- %s] ADD '%s' have a fallback \n",
$rulelid, $ruleid, $ruletype, $rulename ;
                                        #
                                        #       }
                                        #}
                                        ## if is fallback rule => get the rulename
                                        #if ( $flagfallback eq 1 && (!defined $ruletype || $ruletype eq 201 ) ) {
                                        #       $ruleFallbackName = pop @rulenameForFallback;
                                        #
                                        #       if($debug ne 0) {
                                        #               printf LOG_debug "[%s - %s -- %s] GET fallback for '%s' = '%s' \n",
$rulelid, $ruleid, $ruletype, $rulename, $ruleFallbackName ;
                                        #       }
                                        #}
                                }
                                else {
                                ########

                                        #if (($ruleid eq $siteid) && ($rulelid eq $sitelid)) {
                                        if (($ruleid eq $sitelid) && ($rulelid eq $siteid)) {
                                                ### ===== --------- ===== ###
                                                ### ===== Site Stat ===== ###
                                                ### ===== --------- ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_debug "[%s - %s -- %s] site_stat_%s#%s##%s\n", $rulelid,
$ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                                        printf LOG_output "site_stat_%s#%s##%s\n", $metric, $rulename,
$node->textContent;
                                                }

                                                print "site_stat_".$metric."#".$rulename."##".$node->textContent."\n";
                                        }
                                        ### ===== --------- ===== ###
                                        ### ===== Rule Stat ===== ###
                                        ### ===== --------- ===== ###

                                        ### === List Rule Type
                                                #['unknown', 0],
                                                #['localTraffic', 10],
                                                #['siteFallback', 11],
                                                #['tplFallback', 12],
                                                #['accessLink', 100],
                                                #['accessLinkFallback', 101],
                                                #['shaping', 200],
                                                #['grooming', 201],
                                                #['shapingOtherSites', 202],
                                                #['infrastructure', 300],
                                                #['applicative', 1000],
                                                #['VoIP', 1001],
                                                #['unknown-voip', 1011],
                                                #['applicativeGr', 1100],
                                                #['fallback', 1200]
```

```
                                        ### === ============================ ===

                                        elsif (defined $ruletype && $ruletype eq 0) {
                                                ### ===== Intermediate - Rule Stat ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_output "[%s - %s -- %s] intermediate_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                                }

                                                print "intermediate_".$metric."#".$rulename."##".$node->textContent."\n";
                                        }
                                        elsif (defined $ruletype && $ruletype eq 10) {
                                                ### ===== Local Traffic - Rule Stat ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_output "[%s - %s -- %s] local_traffic_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                                }
                                                $local_traffic_name_forFallback=$rulename;
                                                print "local_traffic_".$metric."#".$rulename."##".$node->textContent."\n";
                                        }
                                        elsif (defined $ruletype && $ruletype eq 11) {
                                                ### ===== Local Traffic - Fallback Rule Stat ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_output "[%s - %s -- %s] terminal_fallback_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $local_traffic_name_forFallback." - Fallback", $node->textContent;
                                                }

                                                print "terminal_fallback_".$metric."#".$local_traffic_name_forFallback." -
Fallback##".$node->textContent."\n";
                                        }
                                        elsif (defined $ruletype && $ruletype eq 12) {
                                                ### ===== Intermediate Rule - Fallback - Rule Stat ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_output "[%s - %s -- %s] terminal_fallback_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $intermediate_rulename." - Fallback", $node->textContent;
                                                }

                                                print "terminal_fallback_".$metric."#".$intermediate_rulename." -
Fallback##".$node->textContent."\n";
                                        }
                                        elsif (defined $ruletype && $ruletype eq 100) {
                                                ### ===== Access link - Rule Stat ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_output "[%s - %s -- %s] access_link_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                                }
                                                $access_link_name_forFallback=$rulename;

                                                print "access_link_".$metric."#".$rulename."##".$node->textContent."\n";
                                        }
                                        elsif (defined $ruletype && $ruletype eq 101) {
                                                ### ===== Access link - Fallback Rule Stat ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_output "[%s - %s -- %s] terminal_fallback_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $access_link_name_forFallback." - Fallback", $node->textContent;
                                                }

                                                print "terminal_fallback_".$metric."#".$access_link_name_forFallback." -
Fallback##".$node->textContent."\n";
                                        }
                                        elsif (defined $ruletype && $ruletype eq 200) {
                                                ### ===== Shaping - Rule Stat ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_output "[%s - %s -- %s] shaping_%s#%s##%s\n", $rulelid,
$ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                                }
```

```
                                        $shapping_name_forFallback=$rulename;
                                        $shapping_name_forOtherSite=$rulename;

                                        print "shaping_".$metric."#".$rulename."##".$node->textContent."\n";
                                }
                                elsif (defined $ruletype && $ruletype eq 201) {
                                        ### ===== Grooming - Rule Stat ===== ###
                                        if($debug ne 0) {
                                                printf LOG_output "[%s - %s -- %s] grooming_%s#%s##%s\n", $rulelid,
$ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                        }

                                        print "grooming_".$metric."#".$rulename."##".$node->textContent."\n";
                                }
                                elsif (defined $ruletype && $ruletype eq 202) {
                                        ### ===== Shaping Other Site- Rule Stat ===== ###
                                        if($debug ne 0) {
                                                printf LOG_output "[%s - %s -- %s] shaping_%s#%s##%s\n", $rulelid,
$ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                        }

                                        print "shaping_".$metric."#".$rulename."##".$node->textContent."\n";
                                }
                                elsif (defined $ruletype && $ruletype eq 300) {
                                        ### ===== infrastructure - Rule Stat ===== ###
                                        if($debug ne 0) {
                                                printf LOG_output "[%s - %s -- %s] infrastructure_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                        }

                                        # print "infrastructure_".$metric."#".$rulename."##".$node->textContent."\n";
                                }
                                elsif (defined $ruletype && $ruletype eq 1000) {
                                        ### ===== Terminal Data - Rule Stat ===== ###
                                        if($debug ne 0) {
                                                printf LOG_output "[%s - %s -- %s] terminal_data_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                        }

                                        print "terminal_data_".$metric."#".$rulename."##".$node->textContent."\n";
                                }
                                elsif (defined $ruletype && $ruletype eq 1001) {
                                        ### ===== Terminal Audio Video - Rule Stat ===== ###
                                        if($debug ne 0) {
                                                printf LOG_output "[%s - %s -- %s] terminal_av_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                        }

                                        print "terminal_av_".$metric."#".$rulename."##".$node->textContent."\n";
                                }
                                elsif (defined $ruletype && $ruletype eq 1011) {
                                        ### ===== Unknown Terminal Audio Video - Rule Stat ===== ###
                                        if($debug ne 0) {
                                                printf LOG_output "[%s - %s -- %s] terminal_av_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                        }

                                        print "terminal_av_".$metric."#".$rulename."##".$node->textContent."\n";
                                }
                                elsif (defined $ruletype && $ruletype eq 1100) {
                                        ### ===== ApplicativeGr - Rule Stat ===== ###
                                        if($debug ne 0) {
                                                printf LOG_output "[%s - %s -- %s] applicative_gr_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                        }
```

```
                                        # print "applicative_gr_".$metric."#".$rulename."##".$node->textContent."\n";
                                }
                                elsif (defined $ruletype && $ruletype eq 1200) {
                                        ### ===== Terminal Fallback - Rule Stat ===== ###
                                        if($debug ne 0) {
                                                printf LOG_output "[%s - %s -- %s] terminal_fallback_%s#%s##%s\n",
$rulelid, $ruleid, $ruletype, $metric, $rulename, $node->textContent;
                                        }

                                        print "terminal_fallback_".$metric."#".$rulename."##".$node->textContent."\n";
                                }

                                ### RULE ID and RULE LID is NOT referenced in LOCAL RULE LIST = SHAPPING
RULE OTHER SITE
                                else {
                                        if ($rulename =~/(Fallback)/) {
                                                ### ===== Shaping - Fallback Rule Stat ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_output "[%s - %s -- NA]
terminal_fallback_%s#%s##%s\n", $rulelid, $ruleid, $metric, $shapping_name_forFallback." - Fallback", $node->textContent;
                                                }

                                                print "terminal_fallback_".$metric."#".$shapping_name_forFallback." -
Fallback##".$node->textContent."\n";

                                        }
                                        else {
                                                ### ===== Shaping - Other site Rule Stat ===== ###
                                                if($debug ne 0) {
                                                        printf LOG_output "[%s - %s -- NA]
terminal_shaping_%s#%s##%s\n", $rulelid, $ruleid, $metric, $shapping_name_forOtherSite." - ".$rulename, $node-
>textContent;
                                                }

                                                print "terminal_shaping_".$metric."#".$shapping_name_forOtherSite." -
".$rulename."##".$node->textContent."\n";

                                                # ---- Counter NB Connexions for Application shapping
                                                $shapping_application_name=$rulename;
                                                if ($metric =~ /(NbCnx)/) {
                                                        my ($currentNbCnx,$NbCnx,$cnxvalue)=0;
                                                        $cnxvalue=$node->textContent;
                                                        my $key=$shapping_application_name;
                                                        chomp $key;
                                                        #$key =~ s/\-//g;
                                                        #$key =~ s/\[//g;
                                                        #$key =~ s/\]//g;
                                                        #$key =~ s/ /_/g;
                                                        #$key =~ s/_//g;
                                                        if ( exists( $hash_appl_nbcnx{$key} ) ) {
                                                                $currentNbCnx = $hash_appl_nbcnx{$key} ;
                                                                $NbCnx = $currentNbCnx + $cnxvalue;
                                                        } else {
                                                                $NbCnx = $cnxvalue;
                                                        }
                                                        $hash_appl_nbcnx{$key} = $NbCnx;
                                                        if($debug ne 0) {
                                                                printf LOG_debug "[COUNT %d + %d]
application_shapping_ruleNbCnx#%s##%d\n", $currentNbCnx, $cnxvalue, $key, $NbCnx;
                                                        }
                                                }
                                                #else {
                                                        #if($debug ne 0) {
                                                        #       printf LOG_debug "[COUNT App Shapping]
application_shapping_ruleNbCnx###\n";
                                                        #}
```

```
                                                #}
                                        }
                                }

                                        } # end else not metric name = ruleFallback
                                } # end if $ruleid is empty
                        } #end else not metric rule stat
                } #end foreach nodelist

                ### ===== Application - Shaping - Other site Rule Stat ===== ###
                ### ----
                ### ====> NB CONNEXION for an Application
                while( my ($appl,$value) = each(%hash_appl_nbcnx) ) {
                        if($debug ne 0) {
                                printf LOG_output "[NA - NA -- NA] application_shapping_ruleNbCnx#%s##%d\n", $appl, $value;
                        }
                        print "application_shapping_ruleNbCnx#".$appl."##".$value."\n";
                }



                LOG_output->autoflush();



}
else {
#         if ($result_Appli->status ne $CLI_RESULT_OK) {
  for my $error (@{$result_Appli->errors}) {
    s_error($error->{raw});
  }
  $exit_value = 1;
}



################
if($debug ne 0) {
  #info("\n\nDeConnexion à la base $databases_name...");
  printf LOG_debug "\n\nLog out SGM Database [$databases_name]..."

}

$cli->DESTROY;



################


exit $exit_value;
```