USER GUIDE

Product	Snow Inventory Agent for Windows
Version	5.3
Release date	2018-04-05
Document date	2021-03-17

CONTENTS

1 INTRODUCTION

This document describes how to install and run the Snow Inventory Agent for Windows.

The Snow Inventory Agent for Windows is part of the Snow Inventory solution and is used for inventory of Windows computers. The agent scans the computers and saves the collected data to a compressed and encrypted file, which is sent to a Snow Inventory server (Master Server or Service Gateway).

The agent can be installed locally on each computer (recommended), or be used as an agentless/zero footprint solution with no local installation. In the agentless/zero footprint scenarios each computer is scanned either with the agent as a part of a logon script, or as a remotely executed command.

The benefit of having the agent installed locally is that it can meter usage of applications on the computer, as well. When an inventory scan of the computer is triggered remotely or via a logon script, no metering data is gathered.

For detailed information on the configuration of the agent, refer to the document *Snow Inventory Agents Configuration Guide*.

1.1 PREREQUISITES

1.1.1 DEPENDENCY ON OTHER SNOW PRODUCTS

This version of the Snow Inventory Agent can only be used in a Snow Inventory Server 5 environment.

1.1.2 SUPPORTED OPERATING SYSTEMS

For information on supported operating systems, see Snow System Requirements document, in the Snow Knowledge Base.

2 INSTALLATION

The Snow Inventory Agent for Windows can be deployed either using the Snow Inventory Admin Console or by using existing deployment infrastructure. For the latter case, the Snow Software support can deliver the agent as a pre-configured MSI package.

During the installation, an environment variable called **%SNOW_AGENT%** is created, pointing at the installation directory. This variable can be used to dynamically reference the agent installation folder from scripts, or when browsing for the folder in Windows Explorer.

Once installed on the computer the Snow Inventory Agent is started through a Windows service named **Snow Inventory Agent**. The service is set to start automatically when the computer is started.

The Snow Inventory Agent for Windows consists of one executable file called **snowagent.exe** and one configuration file called **snowagent.config**. These two files are by default located in the same directory. To see a command line summary of the executable, use the following command from a command prompt:

```
snowagent.exe -?
```

If the agent files need to be copied, they can be found on the Snow Inventory server or on a computer where the agent is already installed, see Location of files.

NOTE

- When a configuration file is copied, verify that information on site, agent configuration, and Inventory server is correct for the target computer.
- The only supported install location is **%ProgramFiles%**.

2.1 SNOW INVENTORY ADMIN CONSOLE

Installation via the Snow Inventory Admin Console requires network access to the target computers, and a user account with local administrator access on the target computers, typically the domain administrator account.

For detailed information on agent deployment from the Admin Console, refer to the *User Guide* of the *Snow Inventory Admin Console*.

2.2 PREPARED MSI PACKAGES

Installation can be performed using a pre-configured MSI package.

Installation packages are prepared by and ordered from Snow Support. The current configuration file needs to be provided before an MSI package can be prepared. If no configuration file exists, certain information is needed in order to create one.

Required information:

- address to the Snow Inventory Server, including port number

- site name

Optional information:

- name of the configuration file
- if the Snow Inventory Oracle Scanner should be included
- if previous versions of the Snow Inventory Client for Windows should be removed

Use one of the following methods for installation:

1. Deploy the MSI package using an existing deployment infrastructure.
2. Copy the MSI package to any folder on the target computer, and then run the installation package locally.
3. Deploy the MSI package using Active Directory GPO.
4. Deploy the MSI package using Windows logon scripts.

For more information on deployment using Active Directory GPO or Windows logon scripts, see the sections that follow.

2.2.1 ACTIVE DIRECTORY GPO

A pre-configured MSI package for the Snow Inventory Agent can be used to perform installation via Active Directory Group Policy Object (GPO). The following steps will create a Group Policy that installs the agent on all computers within the OU (Organizational Unit) where the policy is created.

To install using Active Directory GPO:

1. Put the MSI package on a network share.
2. Log on to a domain controller with a user account that has domain administrator rights.
3. Start **Group Policy Management from Start|ControlPanel|AdministrativeTools|Group Policy Management**.
   If the option **Group Policy Management** is not available, the **Group Policy Console** is missing. The installation file (gpmc.msi) can be downloaded from the Microsoft webpage using the following link: http://www.microsoft.com/downloads/details.aspx?familyid=0a6d4c24-8cbd-4b35-9272-dd3cbfc81887&displaylang=en
4. Open **Forest Name|Domains|DomainName|Group Policy Objects**.
5. Click on **Action** at the top of the page, and then click **New**.
   Pick a name for the policy, for example "Snow Inventory Agent Installation".
6. Right-click the created policy in the list, and then click **Edit**.
   Modify the script to match the path to the network share and the MSI package names.
7. Click **Computer Configuration|Software Settings**.
8. Right-click **Software Installation**, click **New** and then click **Package**.
   Select the MSI package to be installed.
   Press **OK** twice to apply the changes, and close the **Group Policy Object Editor**.
9. Right-click the OU that contains the target computers for the deployment, or **Domain Computers** for all computers. Select **Link existing GPO** and then pick the Group Policy that was created (in step 5-8)

2.2.2 LOGON SCRIPTS

A pre-configured MSI package for the Snow Inventory Agent can be used to perform installation via logon scripts.

To install using a logon script:

1. Put the MSI package on a network share.
2. Use the script below.
   Modify the script to match the path to the network share and the MSI package names.

```
@ECHO OFF
REM ### CHECK IF 32 OR 64 BIT
IF %PROCESSOR_ARCHITECTURE%==AMD64 GOTO SIXFOUR
IF %PROCESSOR_ARCHITECTURE%==x64 GOTO SIXFOUR

REM ### PERFORM 32-BIT INSTALL
msiexec /I \\SERVER\SHARE\snowagent-5.2.0.x86.msi /qn

GOTO END

:SIXFOUR

REM ### PERFORM 64-BIT INSTALL
msiexec /I \\SERVER\SHARE\snowagent-5.2.0.x64.msi /qn

:END
```

2.3 COPY OF AGENT FILES

The Snow Inventory Agent can be installed using a copy of the agent files. Installation can be performed via logon scripts, or manually using the command line.

The following files are required:

- snowagent.exe
- snowagent.config

NOTE
For manual installation, local administrator access is required on the target computer.

2.3.1 LOGON SCRIPTS

1. Copy the Snow Inventory Agent files to a network share.
2. Use the script below.
   Modify the **SNOWFOLDER** parameter to match the server share with the agent files.

```
@ECHO OFF
REM ### CONFIGURE LOCAL FOLDER FOR INSTALLATION
SET LOCALFOLDER=%ProgramFiles%\Snow Software\Inventory\Agent
REM ### CONFIGURE PATH TO INV AGENT SHARE HERE
SET SNOWFOLDER=\\SERVER\SHARE
```

```
REM ### CHECK IF 32 OR 64 BIT
IF %PROCESSOR_ARCHITECTURE%==AMD64 GOTO SIXFOUR
IF %PROCESSOR_ARCHITECTURE%==x64 GOTO SIXFOUR

REM ## 32 BIT

REM ### CHECK IF AGENT IS ALREADY INSTALLED
IF EXIST "%LOCALFOLDER%\snowagent.exe" GOTO END

REM ### CHECK THAT THE SERVER PATH EXISTS
IF EXIST "%SNOWFOLDER%\Agent\32bit\snowagent.exe" (

MKDIR "%LOCALFOLDER%"
MKDIR "%LOCALFOLDER%\data"
XCOPY /d /y "%SNOWFOLDER%\Agent\32bit\*.*" "%LOCALFOLDER%"
"%LOCALFOLDER%\snowagent.exe" install
)

GOTO END

:SIXFOUR

REM ## 64 BIT

REM ### CHECK IF CLIENT IS ALREADY INSTALLED
IF EXIST "%LOCALFOLDER%\snowagent.exe" GOTO END

REM ### CHECK THAT THE SERVER PATH EXISTS
IF EXIST "%SNOWFOLDER%\Agent\64bit\snowagent.exe" (

MKDIR "%LOCALFOLDER%"
MKDIR "%LOCALFOLDER%\data"
XCOPY /d /y "%SNOWFOLDER%\Agent\64bit\*.*" "%LOCALFOLDER%"
"%LOCALFOLDER%\snowagent.exe" install
)

GOTO END

:END
```

2.3.2 COMMAND LINE

1. Copy the Snow Inventory Agent files to the target computer.
2. In a command prompt on the target computer, navigate to the folder where the agent files are placed.
3. Run the following command to install the Snow Inventory Agent in the default location (C:\Program Files\Snow Software\Inventory\Agent):

   ```
   snowagent.exe install
   ```

4. Copy **snowagent.config** to the default location.
5. In a command prompt, navigate to the default location and start the Windows service:

   ```
   snowagent.exe start
   ```

6. The installation is completed.
   Since the files that were originally copied to the target computer are no longer needed, they can be removed.

2.4 REMOVAL OF THE AGENT

As a general recommendation, use the same method for removal of the agent as was used for the installation.

2.4.1 ACTIVE DIRECTORY GPO

Method for removal of software using a Group Policy Object depends on the deployment tool that has been implemented in the organization. For removal of the Snow Inventory Agent, refer to the organization's general process for software removal.

2.4.2 COMMAND LINE

The agent can be uninstalled from a computer by running a command from a command prompt. From Snow Inventory Agent for Windows version 5.2.4 this command will only remove the Windows agent service, all related files will remain on the system until manually removed.

To perform a complete uninstallation of the Windows service and related files, use the Windows Installer or the Active Directory GPO.

2.4.3 WINDOWS INSTALLER

The agent can be removed from a computer by using the Windows Installer. In this way, both the Windows service and all related files will be removed.

1. On the computer, open the **Control Panel** and select **Programs and Features**.
2. In the list of currently installed programs, right-click **Snow Inventory Agent** and select **Uninstall**.

2.5 LOCATION OF FILES

The table below shows the locations of the agent files and the log files.

File name	Location	Description
snowagent.exe	%ProgramFiles%\Snow Software\Inventory\Agent	Executable file
snowagent.config	%ProgramFiles%\Snow Software\Inventory\Agent	Configuration file
*.snowpack	%ProgramFiles%\Snow Software\Inventory\Agent\data	Inventory result file
snowagent.log	%ProgramFiles%\Snow Software\Inventory\Agent\data	Log file
sios.jar	%ProgramFiles%\Snow Software\Inventory\Agent	Binary file of the Snow Inventory Oracle Scanner
sios.xml		

3 UPGRADE FROM SNOW INVENTORY CLIENT

An upgrade from Snow Inventory Client for Windows 3.x can be deployed using the following methods:

- Upgrade using an MSI package, either via Active Directory GPO or an existing deployment infrastructure
- Upgrade via the Inventory 5 Admin Console and the Discovery functionality. For details, see the document *Admin Console User guide* for Snow Inventory Server 5.
- Manual copy of agent files
- Upgrade via the Inventory 3.2 Administration Console

3.1 UPDATE PACKAGES

Since the Snow Inventory Agent is built specifically for 32-bit or 64-bit Windows operating systems it is important that the correct binary is deployed onto each system. This is normally done by using an update package which contains both the 32-bit and the 64-bit binary.

1. Obtain a copy of the 32-bit agent, available from any of the official product release packages.
2. Use the 32-bit agent to install the agent binary from the update package. The 32-bit agent will recognize the target system and pick the correct binary from the update package.
   If the executable and the desired **snowagent.config** file are located in the same directory:

   ```
   snowagent.exe install -u update_package.snowpack
   ```

   If the executable and the desired **snowagent.config** file are located in different directories, specify the location of the configuration file using the **-c** option:

   ```
   snowagent.exe -c use_this_config.config install -u update_package.snowpack
   ```

It is possible to write scripts that deploy the agent without an update package, but in that case the correct binary must be used. The 32-bit **snowagent.exe** binary will not allow itself to be installed onto a 64-bit system.

3.2 UPGRADE USING SNOW INVENTORY 3.2 ADMIN CONSOLE

Customers that are using the upgrade functionality for the Inventory Client for Windows in Inventory Server 3.2, can use it for upgrade to Inventory Agent for Windows, as well.

Before any upgrade is started, these files must be copied to the **client** folder of the Snow Inventory 3.2 server running the Administration Console:

- snowagent.exe
- snowagent.config
- update.exe / update64.exe

Order the files from Snow Support.

To upgrade the Windows Client via the Inventory 3.2 Administration Console:

1. In the **Administration Console**, click **Update** and then click **Add**.
   The **Update Editor** dialog box appears.
2. Type a **Name** and a **Description** of the update.
3. Clear the **Configuration** check box.
4. Select the **Files** check box.
5. In the **File name** list, select the files:
   - snowagent.exe
   - snowagent.config
   - update.exe / update64.exe
6. In the **Recipients** section, select targets of the update.
7. In the **Target Operating System** section, select target OS of the update.
8. Click **OK** to save the update and close the **Update Editor** dialog box.
   The update is added to the **Client Updates** list.

The next time the Inventory Client communicates with the server, it will be notified of the update package and will download it.

Once the files **snowagent.exe** and **snowagent.config** are downloaded, the Snow Inventory Agent will be installed and the **update.exe**/**update64.exe** will start the agent.

When both the new Inventory Agent and the old Inventory Client are running, the **update.exe**/**update64.exe** will stop the client service. When the service has been successfully stopped, the agent will uninstall the service, and then remove all files and registry entries related to the Inventory Client.

NOTE
In case an automatic upgrade is run while a scan is already in progress by the Inventory Client for Windows (3.7), the first scheduled scan by the Inventory Agent for Windows (5.x) will take place according to the schedule set in the agent configuration file.

If an instant scan result is required: In the **Devices** view of the Snow Inventory 5 Admin console, use the **Initiate scan** command to perform a manual scan of any number of selected devices.

4 COMMAND LINE

The general command line format for the Snow Inventory Agent for Windows is:

```
snowagent.exe [options] [<command> [<arguments>]]
```

Global option	Description
-w <dir>	Override working directory otherwise inherited from shell/ environment
-v	Enable verbose mode, bump all logging to max level (enable all tags)
-c <dir> snowagent.config	Override the default location of the configuration file.
-data-dir <dir>	Override data directory (regardless of configuration)
-log-dir <dir>	Override log directory (regardless of configuration)
-? or **-h**	Print help text and exit

Command	Option	Description
copyright		Show copyright notice
install		Install agent on this device
legacycleanup		Uninstall any existing Snow Inventory Client for Windows 3.x
manifest		Print the manifest in json format and exit
scan		Scan the current device and create a snowpack file in the target destination (as set by configuration). This will start a scan regardless of any schedule or other configuration. It will not start a scan if there's a running scan in progress. After the scan has completed the agent will exit. The scan result is not automatically sent.
send		Send files from the data directory. Files that are successfully sent are deleted
start		Start (if not running) a service or daemon
stop		Stop any running service or daemon
test		Run a self-test. Check configuration and server connectivity, then exit with a 0 (if successful) otherwise with a non-zero exit code
uninstall		Remove the agent (Windows service) from this device.
update	-package <location>	Check for updates now, and install if any are found. Use an update package from a specific location

Command	Option	Description
version		Print the current snowagent version and exit

5 SCANNING

The Snow Inventory Agent for Windows scans the file system and the registry, and collects information from the WMI. A combination of data from these sources are then used by the Snow Recognition Service (SRS) to identify the details of the software, such as product name, version, and edition.

5.1 SCHEDULED SCAN

Snow Inventory Agent for Windows can be configured to run on a daily, weekly, or monthly basis or at system start-up. Configuration of the schedule is done via Snow Inventory Admin Console.

Randomization can be applied to the daily, weekly, and monthly occurrence types. It adds a random delay each time a scan is scheduled which will spread the start time among the agents and distribute the load when many agents are configured to scan at the same time.

Before a scheduled scan starts, the agent checks when the last scan was performed. If it was run the same day, no new scan will be performed. This means that a scheduled scan will never be performed more frequently than once a day. This check is not done when a scan is run from the command line.

When a scheduled scan is completed, the result will be sent immediately to an Inventory server.

5.2 MANUAL SCAN

When a change has been made or if a scan needs to be run for support reasons, a scan can be manually initiated outside of the regular schedule.

5.2.1 SCAN OF ONLINE COMPUTERS

Use the commands below to trigger the agent to start a manual scan of the current computer and send the scan result to a server endpoint as defined in the agent configuration file. A manual scan will start regardless of any schedule or other configuration. However, if there is a scan already running, a manual scan will not start.

Scan the current computer:

```
snowagent.exe scan
```

Send the scanning result:

```
snowagent.exe send
```

The `snowagent.exe scan` command runs in the user context—i.e. in the context of the user who manually executes the command—with implied limitations, if the user is not administrator of the computer.

One can trigger the Snow Inventory Agent Windows service to run a scan in the service context—i.e. in the context of the user account that runs the Snow Inventory Agent Windows service—by manually executing the `sc control SnowInventoryAgent5 128` command.

Start a scan that dumps only metering data from computer memory, aggregates, and sends it:

```
sc control SnowInventoryAgent5 132
```

NOTE
This command triggers the Snow Inventory Agent Windows service to execute in the service context—i.e. as the user account that is configured to run the Windows service—which means it does not require to be run in an administrator context.

This example will send metering data in a few seconds instead of performing a regular scan, which—depending on the Inventory Agent configuration and computer performance—can take minutes to finish.

5.2.2 SCAN OF OFFLINE COMPUTERS

In scenarios where a computer cannot communicate with the Snow Inventory Server, a “one time scan” can be performed.

1. Copy the Snow Inventory Agent files to a directory on a USB memory stick.
2. On the offline computer to be scanned:
    a. Insert the memory stick into the USB port.
    b. In a command prompt, navigate to the folder where the agent files are placed.
    c. Run the following command to make the agent perform an instant scan of the current computer:

       ```
       snowagent.exe scan
       ```

       The scan result is put on the USB memory stick in a subdirectory called **data**.
3. Move the USB stick to an online computer with access to Snow Inventory.
4. On the online computer, run the following command to make the agent send the scan result in the **data** subdirectory to a server endpoint, as defined in the agent configuration file:

   ```
   snowagent.exe send
   ```

5.3 AGENTLESS SCAN

A remote script or a logon script can trigger the Snow Inventory Agent to perform a scan on the computer without installing the agent locally. To achieve this the agent files need to be placed on a network share where the domain users have read and execute access. However, an agentless scan is limited to software and hardware inventory, no information on software metering will be gathered.

5.3.1 SCAN VIA REMOTE SCRIPTS

1. Copy the Snow Inventory Agent files to a network share.
2. Use the script below to create a batch file called **snowagent.bat**.

Modify the **SNOWFOLDER** parameter to match the server share with the agent files.

```
@ECHO OFF
REM ### CONFIGURE PATH TO INV AGENT SHARE HERE
SET SNOWFOLDER=\\SERVER\SHARE

REM ### CHECK IF 32 OR 64 BIT
IF %PROCESSOR_ARCHITECTURE%==AMD64 GOTO SIXFOUR
IF %PROCESSOR_ARCHITECTURE%==x64 GOTO SIXFOUR

REM ## 32 BIT

REM ### CHECK THAT THE SERVER PATH EXISTS
IF EXIST "%SNOWFOLDER%\Agent\32bit\snowagent.exe" (
"%SNOWFOLDER%\Agent\32bit\snowagent.exe" scan
"%SNOWFOLDER%\Agent\32bit\snowagent.exe" send
)

GOTO END

:SIXFOUR

REM ## 64 BIT

REM ### CHECK THAT THE SERVER PATH EXISTS
IF EXIST "%SNOWFOLDER%\Agent\64bit\snowagent.exe" (
"%SNOWFOLDER%\Agent\64bit\snowagent.exe" scan
"%SNOWFOLDER%\Agent\64bit\snowagent.exe" send
)

GOTO END

:END
```

3. Download **psexec** from the Microsoft TechNet web page, http://technet.microsoft.com/en-us/sysinternals/bb897553.aspx to any computer that can reach the target computers via the network.
4. Create a text file called **computers.txt** that contains the target computer names.

NOTE
The "DOMAIN\USER" needs to be a local administrator on the target computers.

5. Run the following command on the computer with **psexec**:

```
psexec @computers.txt –u DOMAIN\USER –p PASSWORD –c –h snowagent.bat
```

5.3.2 SCAN VIA LOGON SCRIPTS

1. Copy the Snow Inventory Agent files to a network share.
2. Use the script below.
   Modify the **SNOWFOLDER** parameter to match the server share with the agent files.

```
@ECHO OFF
REM ### CONFIGURE PATH TO INV AGENT SHARE HERE
SET SNOWFOLDER=\\SERVER\SHARE
```

```
REM ### CONFIGURE LOCAL FOLDER FOR LOG FILE HERE
REM ### THIS IS NEEDED TO SCAN ONLY ONCE A DAY
SET LOCALFOLDER=%USERPROFILE%

REM ### GET NORMALIZED DATE
SET CLEANDATE=%DATE:/=-%

REM ### IF ALREADY SCANNED TODAY - DO NOTHING
IF EXIST "%LOCALFOLDER%\SNOWSCAN_%CLEANDATE%.txt" GOTO END

REM ### CHECK IF 32 OR 64 BIT
IF %PROCESSOR_ARCHITECTURE%==AMD64 GOTO SIXFOUR
IF %PROCESSOR_ARCHITECTURE%==x64 GOTO SIXFOUR

REM ### 32 BIT
REM ### CHECK THAT THE SERVER PATH EXISTS
IF EXIST "%SNOWFOLDER%\Agent\32bit\snowagent.exe" (
"%SNOWFOLDER%\Agent\32bit\snowagent.exe" scan
"%SNOWFOLDER%\Agent\32bit\snowagent.exe" send

REM ### DELETE OLD LOG FILES
DEL /Q "%LOCALFOLDER%\SNOWSCAN*.txt" 1>nul 2>nul

REM ### WRITE LOG FILE FOR TODAY
ECHO SCAN > "%LOCALFOLDER%\SNOWSCAN_%CLEANDATE%.txt"
)

GOTO END

:SIXFOUR

REM ## 64 BIT

REM ### CHECK THAT THE SERVER PATH EXISTS
IF EXIST "%SNOWFOLDER%\Agent\64bit\snowagent.exe" (
"%SNOWFOLDER%\Agent\64bit\snowagent.exe" scan
"%SNOWFOLDER%\Agent\64bit\snowagent.exe" send

REM ### DELETE OLD LOG FILES
DEL /Q "%LOCALFOLDER%\SNOWSCAN*.txt" 1>nul 2>nul

REM ### WRITE LOG FILE FOR TODAY
ECHO SCAN > "%LOCALFOLDER%\SNOWSCAN_%CLEANDATE%.txt"
)

GOTO END

:END
```

6 METERING OF APPLICATION USAGE

The primary focus of metering is desktop applications. Metering of usage is performed whenever a Windows process is running. Processes are measured whether the application is open on the desktop or running in the background

The following meta data is uniquely gathered per process:

- file name
- if the process is local
- if the process is virtual
- virtual app package name
- username
- date
- file version info
- usage data type (web, RDP, or VDI).

Since the main purpose of metering is to measure the usage of desktop applications, processes are not regarded as running when the screen is locked. This means that scheduled tasks, anti-virus software, and other processes executing in the background will not be regarded as used while the screen is locked.

Application usage (the "run count") is calculated as follows:

- The agent will check for events every 15 seconds (default value).
  Use the **wmi.poll_interval** setting in the agent configuration file to change the interval; 5 seconds is minimum and 120 seconds is maximum.
- The agent increases the run count of an application only when it gets notified of a start event. If an application is started before the agent is running, that application will get a run count of "0".
- When a scan is performed and the inventory result file is created, the run count will be set to "1" if the run counts reported from the agent sum up to "0".
- When multiple instances of the same application are started, the agent counts all individual run starts during that session. However, the total duration will be counted from when the first instance is started until the last instance is closed.
- A scheduled scan will contain all metering information that has been gathered since the last scheduled scan. A scheduled scan will clear all temporary metering files stored on disk, and new ones will be created after the scan with a run count of "0".
- A manual scan will contain all metering information that has been created since the last scheduled scan. During the manual scan, the metering files stored on disk will be deleted, but their content is kept in memory. After the manual scan, the agent will recreate all metering files as they were before the scan.
- At midnight local time, the agent will finalize the metering files for the currently running processes with an end time of 23.59.59. New metering files will be created with a start time of 00.00.00 on the next day.

For an example of how application usage is calculated, see Metering example .

7 CLOUD APPLICATION DISCOVERY AND METERING

The Snow Inventory Agent can report usage of cloud applications, also known as Software as a Service (SaaS). It reports usage of cloud applications accessed via the internet browser, used for example for messaging, office software, invoicing and service desk services.

The feature is disabled by default, but can be enabled in the Snow Inventory Admin Console.

For more detailed information about how the feature works and the available browser extensions, refer to the document *Technical Description, Cloud Application Metering*.

8 HEARTBEAT

In addition to the scheduled scans, the Snow Inventory Agent for Windows sends heartbeats according to a configurable interval. The heartbeat is a simple communication between the agent and the Inventory Server where the agent asks the server for tasks. Examples of tasks include agent updates, configuration updates, addition/removal of support files, and initiation of scans outside of the configured schedule.

The minimum interval between heartbeats is 10 minutes.

For details on configuration of heartbeats, refer to the settings **http.poll_interval** and **http.poll_variance** in the *configuration-doc.html* file.

9 TROUBLESHOOTING

The following chapter contains tips related to troubleshooting both the installation of the Snow Inventory Agent and the communication between the agent and a Snow Inventory server.

9.1 INSTALLATION

Local administrator privileges are required on the computer where the agent is to be installed.

9.1.1 DEPLOYMENT

Deployment via SCCM and Active Directory GPOs using an MSI package could result in that the agent appears not to be installed as the application is not visible in Programs and Features. However, the service is installed and running, and the agent gathers inventory data. An attempt to manually uninstall the agent will generate a message saying that an uninstallation is possible only for installed products.

As a workaround, order a new MSI package from the Snow Software support. Install the new agent manually or deploy it by using a logon script, see Prepared MSI packages.

NOTE
This issue is currently being investigated in collaboration with Microsoft.

9.1.2 HOW TO REDEPLOY THE AGENT USING GPO AFTER MANUAL REMOVAL

When the agent is deployed using Active Directory GPOs, a key (GUID) is set in the registry of the local computer. The key is not related to the agent, but rather to the GPO itself and includes information on the deployment. As long as the key is there, no new GPO enforcement will be made, trying to run an installation of the agent MSI.

The key (GUID) representing the Inventory agent is stored in:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\AppMgmt
\REGISTRY ID
```

In case the agent is removed manually from **Windows Control Panel|Program and Features**, the registry key will not be removed. The computer will hereby enter a state where the Active Directory thinks the GPO is applied and that the agent MSI is installed, while the agent is in fact removed.

To solve this and have the agent reinstalled on the computer, select one of the following options:

- **Remove the registry key**
  Delete the key (GUID) representing the Inventory agent. The GPO will then be reapplied and run an installation of the agent MSI.
- **Make a change in the GPO settings**
  For example, when the MSI is removed or replaced, the GPO will be reapplied with a new "revision number".

9.2 REMOTE TROUBLESHOOTING

The **sc** command is the built-in command in Windows to communicate with the Service Control Manager and services. To use the **sc** command towards a remote computer, local administrator privileges are required on that computer.

The available **sc** commands are listed below.

- Scan the computer:

  ```
  sc \\MACHINENAME control SnowInventoryAgent5 128
  ```

- Send the inventory result (to the endpoint defined in the configuration file of the remote computer):

  ```
  sc \\MACHINENAME control SnowInventoryAgent5 129
  ```

- Force update routine to run (execute any already staged updates):

  ```
  sc \\MACHINENAME control SnowInventoryAgent5 130
  ```

- Run legacy cleanup routine (will remove traces of 3.x client after a 30 seconds delay):

  ```
  sc \\MACHINENAME control SnowInventoryAgent5 131
  ```

10 APPENDIX

10.1 EXTRACT COMPUTER LIST FROM ACTIVE DIRECTORY

To extract a list of all computers in an Active Directory the **dsquery** tool can be used. The tool is available on all domain controller servers.

Run the following command from a command prompt to extract a list of all computers in the Active Directory, and export the result to the file **C:\computers.txt**. The **limit** parameter may need to be increased to be able to get the complete list.

```
DSQUERY COMPUTER -o rdn -limit 1000 > c:\computers.txt
```

10.2 MAPPING OF CONFIGURATION FILE ELEMENTS

The table shows mappings for a number of settings in the configuration file of the Snow Inventory Client for Windows 3.7 and the configuration file of the Snow Inventory Agent for Windows, and what they are named in the user interface of the Snow Inventory 5 Admin Console.

Inventory Client 3.7 (client.ini)	Inventory Agent 5 (snowagent.config)	Inventory 5 Admin Console	Description
Site	SiteName	Site Name	Needs to be the same, or there will be duplicates
Name	ConfigName	Name	
DestinationHTTP	<Endpoint><Address>	Server	
HTTPProxy **HTTPProxyPassword** **HTTPProxyServer** **HTTPProxyUsername**	<Proxy>	Proxy settings	Only if proxy is used
IDXEndPoint	<SystemSettings> <Setting key=”idx.endpoint”	System settings Idx_endpoint	Only if web metering is used

10.3 PROCESSOR DETAILS

For CPU recognition, the Snow Inventory Agent for Windows uses the System Information Development Kit (SDK) from CPUID, which is included in the agent. The CPUID SDK is dynamically linked (dll) at runtime and provides detailed information about the processor of the computer.

NOTE
Snow recommends to disable CPUID, which is achieved by using the setting **hardware.scan.enable_cpuid**.

10.4 SCANNING THE REGISTRY

The Snow Inventory Agent can query the registry to find software details, and for that uses a pre-defined list of registry keys to be scanned. For information on what registry keys the agent queries by default, refer to the file **win_snowagent_defaultBehavior.config.xml**.

The agent will collect a maximum of 128 results from each of the registry queries.

10.5 POWERSHELL FUNCTIONALITY

The Snow Inventory Agent for Windows has support for running Windows PowerShell scripts as part of the inventory scanning process. The built-in functionality uses the output of the Windows PowerShell scripts to create software or custom registry keys within the inventory result that is sent from the agent to the Inventory Master Server. This will enable scanning of additional information from software products, but can also be used for custom tasks such as identifying which users are local administrators on each machine.

The Snow Inventory Agent will look for scripts with file names starting with "Scan-" and are located in any of the following folders:

- current working directory, e.g. C:\Program Files\Snow Software\Inventory\Agent
- the subfolder "data", e.g. C:\Program Files\Snow Software\Inventory\Agent\data
- the subfolder "psscripts", e.g. C:\Program Files\Snow Software\Inventory\Agent\psscripts

NOTE
PowerShell Script Execution Policy
PowerShell's default script execution policy does not apply to scripts written exclusively for Snow Inventory Agent for Windows.

10.6 METERING EXAMPLE

In this example, the application Microsoft Word is used for three consecutive days.

Day	Action	Comment
Monday, 3 PM	The user starts one instance of MS Word.	Since the agent is running, it gets notified on this start event and starts a new metering period with **Run count = 1**.

Day	Action	Comment
Monday, 6 PM	The user leaves for home; MS Word is left open and the computer is unlocked.	MS Word is considered to be "running" all night. At midnight, a new metering file will be created with Run count = 0.
Tuesday, 8 AM	The user continues to work with MS Word (one instance) all day.	At 09.00 a scheduled scan is performed, and all metering files are consumed and deleted. The total duration will be summed up, as will the Run counts: **1** (the metering file from Monday) + **0** (the metering file from Tuesday) = **1** After the scan, new metering files will be created for all currently running processes, with Run count = 0.
Tuesday	The user locks the computer several times during the day.	The Run count will not be affected by this, but the "duration" will be paused while the screen is locked.
Tuesday, 6 PM	The user leaves for home; MS Word is left open, but this time the computer is locked.	This will not affect the Run count. At midnight, nothing will happen since the computer is locked.
Wednesday, 8 AM	The user unlocks the computer.	When the computer is unlocked, the agent will notice that it is a new day, and create new metering files with Run count = 0. At 09.00 a scheduled scan is performed, and all metering files are consumed and deleted. The total duration will be summed up. Since the run count is "0" (no start events since the last scan) the scan result will be **Run count = 1**. After the scan, new metering files will be created for all currently running processes, with Run count = 0.
Wednesday, 9 AM-11 AM	The user starts three additional instances of MS Word.	These three start events will all be gathered by the agent setting the Run count = 3.
Wednesday, 11 AM	The user closes all instances of MS Word. The user does not start MS Word any more that day.	The metering period ends. It has a Run count = 3, since three start events was gathered by the agent.
Thursday	The user does not use MS Word at all.	At 09.00 a scheduled scan is performed including the last metering duration from Wednesday with **Run count = 3**.

In this example, the total **Run count** for the application Microsoft Word is reported as;

1 (scan on Tuesday) + **1** (scan on Wednesday) + **3** (scan on Thursday) = **5**.