Manual

SMOscan in 15 minutes

Version 5.92 and higher

Holger Schmeken
Developer of Scanning Tools

Support:
support@softwaremanagement.org

Softwaremanagement.org ITS, Inc.
55 Madison Avenue, Suite 400
Morristown, 07960 New Jersey
USA

Softwaremanagement.org ITS
Mendelstr. 11
48149 Münster
Germany

Table of Contents

1 Motivation behind providing specialized scan tools 4

1.1 Continuity 4

1.2 Specialization 4

2 Simple scenario with one company 5

3 Windows environment 6

3.1 Windows Workstations 6

3.1.1 Logon script – traditional form 6
3.1.2 Logon script – Group Policy (GPO) 6
3.1.3 KiXtart script 7
3.1.4 Warning message at logon 7

3.2 Windows Terminal Sessions 8

3.2.1 Logon script 8
3.2.2 Citrix servers 8

3.3 Windows Servers 9

3.3.1 Simple scheduler service 9
3.3.2 Extended scheduler service 9
3.3.3 Manual creation of scan jobs 10
3.3.4 Automated creation of scan jobs 11

3.4 Windows Virtualization 11

3.4.1 Microsoft App-V 11
3.4.2 Microsoft Hyper-V and Microsoft Virtual Machines 12
3.4.3 VMware ThinApp 12
3.4.4 VMware XenApp and VMware Virtual Machines 12

4 Apple environment 13

4.1 Apple Workstation or Server 13

4.1.1 Login script to execute scan 13
4.1.2 Automated start of login script at login 14
4.1.3 Refinement for centralized management 14

5 Linux environment 15

5.1 Linux Workstation or Server 15

5.1.1 Auto mounting SMOshare$ 15
5.1.2 Login script to execute scan 16
5.1.3 Automated start of login script every week 16
5.1.4 Refinement for centralized management 16

6 Collecting results over different channels 17

6.1 Network shares 17

6.1.1 Configuration of SMOscan.ini 17

6.2 HTTP/HTTPS 17

6.2.1 Windows 2003: Installation of IIS 18
6.2.2 Windows 2008/2012: Installation of IIS 19
6.2.3 Create new website (Port 80 if possible) 19
6.2.4 Copy modul SMOcollectIIS.isa to website 19
6.2.5 Windows 2008/2012: additional step to copy module 20
6.2.6 Prepare the user and password 20
6.2.7 Configuration of website 21
6.2.8 Windows 2003: activate module SMOcollectIIS.isa 21
6.2.9 Windows 2008/2012: activate module SMOcollectIIS.isa 21
6.2.10 Verify configuration 21
6.2.11 Configuration of SMOscan 22

7 Cost centers 23

7.1 Motivation for cost centers 23

7.2 Automation via SMOscan 23

8 Complex scenario with subsidiaries 25

8.1 Pulling results from Hamburg to London 25

8.2 Pushing results from Hamburg to London 26

8.3 Updating different sites 27

9 Appendix 28

9.1 Naming convention of scan results 28

9.2 Calling parameters of SMOscan 28

9.3 Configuration parameters of SMOscan.ini 28

9.3.1 Performance 28
9.3.2 Lifetime 29
9.3.3 Behavior 29
9.3.4 Ignore directories, users or computers 30
9.3.5 Terminal session 32

9.4 Checklists 34

9.4.1 Installation 34

1 Motivation behind providing specialized scan tools

Softwaremanagement.org's SMOscan is designed to identify software and hardware assets in different IT environments. This chapter provides background information about fundamental reasons for using specialized Software Asset Management (SAM) scan tools in contrast to more general tools.

1.1 Continuity

The data SMOscan collects is matched against a product database. This process results in fingerprint-like software identification. For this purpose the scan engine must by all means provide the same data even after it had major release changes. Quality processes at Softwaremanagement.org ensure this behavior.

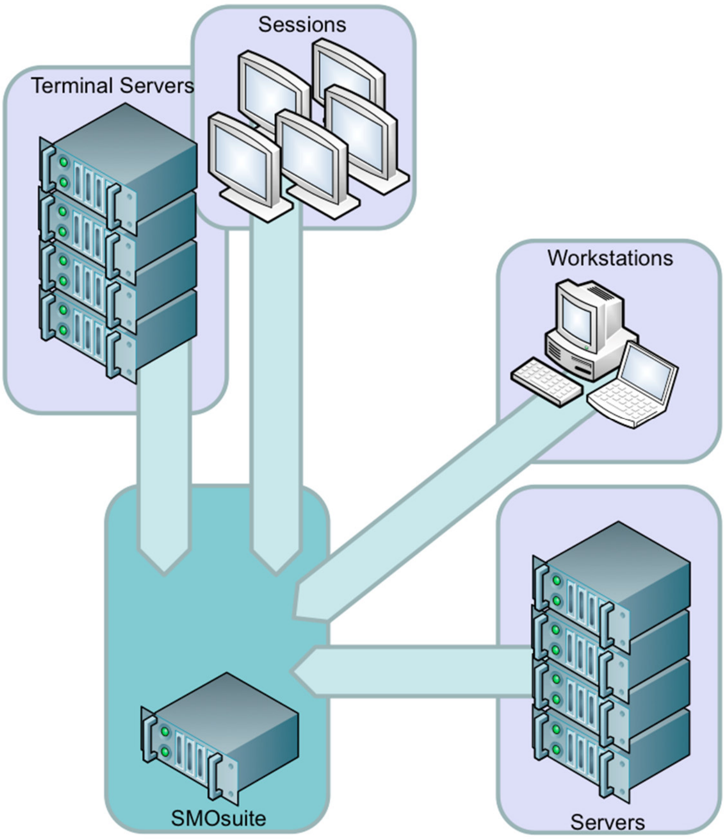
Most SAM tools today use a catalog of software products for identification. The question is if they meet continuity guarantees. Many SAM tools do not provide integrated scan engines. They use products from publishers like Microsoft, Symantec, LANDesk and others. Those publishers will most probably not ensure compatibility with derived software catalogs. In contrast SMOscan is designed and developed under the requirement of full compatibility with existing and future releases of the software catalog.

1.2 Specialization

The pure collection of software assets has its limitations. E.g. the file "msaccess.exe" is exactly the same executable with no options to tell apart the full from the runtime product. But from the standpoint of licensing is it very important to distinguish between them. This is also true for many other products like Microsoft SQL Server Editions Datacenter, Standard and Express. They all use the very same executable but licensing is completely different. Not to forget the huge price tag between the free Express and the high-priced Datacenter edition. Thus SMOscan is enriched with specialized algorithms to exactly identify certain products. These products are ranging from Microsoft Exchange, SQL Server, Office, Access and many others to Autodesk and their specialized licensing.

2 Simple scenario with one company

The following scenario is easy to handle from the perspective of the SMOscan. One company has multiple servers, a terminal server farm with multiple sessions and multiple workstations.



The workstations and terminal sessions will be scanned automatically with a login script for the users. The servers and terminal servers will be scanned with automated scheduler jobs.

Of course the servers need to be prepared first. Very likely directories need to be excluded from the servers and terminal servers. The mechanisms involved will be described in detail in the following chapters.

3 Windows environment

In the next paragraphs you will learn how to use the scan in more complex scenarios and infrastructures. Please keep in mind that the scan itself is just a simple tool to gather asset information. In the easiest case you just create a new scan media with the SMOmedia Manager. In contrast to the following scenarios this is simply a manual approach. You just visit the workplace, start the scan and a new result will be written to the scan media. If all the workplaces in this category have been inventoried you can import all the gathered results.

3.1 Windows Workstations

The SMOscan should be invoked whenever a person is starting to use a computer system. This way the scan can obtain valuable information about the users involved. This is important to gather information about the computers a person is using for his work. This can help to determine the best workplace to apply user bound software licenses.

The following chapters will show different ways to integrate the scan in the technical environment.

3.1.1 Logon script – traditional form

The logon script of the users is a very convenient way to integrate the scan. The script will run at every logon of a user.

```
start /b \\yourserver\smoshare$\smoscan.exe
```

The start command will make sure that the scan is running in parallel to the login script. This way the script will start the scan and it will not wait for the end of the scan operation.

3.1.2 Logon script – Group Policy (GPO)

Modern networks rely on Group Policies to trigger specific behaviour at logon time. In these scenarios the scan is started in the Group Policy for logon scripts. For the general task Microsoft has created different Group Policies with different purposes:

Computer Configuration\Administrative Templates\System\Logon

This section is executed before the user can use his desktop. All operations executed here are expected to be finished before the desktop is available. Please do not start child processes like the scan here. The scan is an operation that is expected to run in the background while the user is working with his computer. If a child process started in this section is still running when the logon procedure has finished these processes will be killed by the operating system! In case you are wondering why the scan is not executed after login this might be the best explanation. We recommend the next section for best practice.

User Configuration\Administrative Templates\System\Logon

The programs or scripts defined in “Run this program at user login” are executed after the desktop is available. Here is the place to embedd the script to start the scan via GPO. The scan will then be executed in the context of the current user.

3.1.3 KiXtart script

In addition to the logon script the KiXtart processor provides an extended language for easy scripting. The KiX language does not support a command for the parallel start of a process. To prevent that the user have to wait for the scan to finish its execution the solution is to start a batch file. Within this batch file the approach of the normal logon script is used. As a result the scan is started with the start command as described in the previous paragraph 3.1.1 Logon script.

3.1.4 Warning message at logon

Executables can be started from untrusted servers. However this might lead to warning messages at logon In this case the settings of the Internet Explorer must be adjusted. The Windows Explorer will change its behaviour according to these settings. Open the settings of the Internet Explorer and select security. In Intranet security cou can add the server hosting the SMOshare$ to the trusted sites. You can make further adjustments in the Intranet section.

3.2 Windows Terminal Sessions

The SMOscan needs to be started in every Terminal Session. This way the use of applications and most importantly the device from where this use has been initiated can be monitored (important for licensing). The scan will recognize the special environment of the session. It will start the monitoring application SMOmonitorApplication or the 64-bit counterpart called SMOmonitorApplication64. This tool is just analyzing the list of processes in the task manager. The analysis is recurring every x milliseconds and this is determined by the parameter TERMINALSESSIONCYCLE=x.

3.2.1 Logon script

Again the logon script of the users is the easiest solution to handle workstations and terminal sessions. Please read the paragraph 3.1.1 Logon script.

3.2.2 Citrix servers

3.2.2.1 Special logon script

If there is no ordinary logon script available we can use the Citrix logon script on Citrix Terminal Servers. This script is called usrlogon.cmd and is located on the server itself. The commands in this script will be executed when a user is starting a new terminal session on the citrix server. In general the logon script is the preferred solution because the Citrix script needs to be modified for every single terminal server.

3.2.2.2 Preparation

Every terminal server in a Citrix farm should be prepared before the SMOmonitorApplication is used. Every published application is treated with special respect to the user experience. As a consequence applications will be stopped gracefully if a published application has been terminated. But this is not desirable for the SMOmonitorApplication. This application should be cleaned up as soon as possible to preserve memory space on the terminal servers. For this purpose Citrix has implemented a specialized registry hive:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Citrix\wfshell\TWI
```

In this hive a string names LogoffCheckSysModules can be found. It should be extended with ',SMOmonitorApplication.exe,SMOmonitorApplication64.exe'. This will guarantee that Citrix will not wait for the application. More details on this mechanism can be found on http://support.citrix.com/article/CTX891671.

3.3 Windows Servers

Usually server systems are running for longer periods of time without users logging on these systems. Thus the use of login scripts will not be sufficient to get results in a determined time frame. For this purpose the tool SMOat will find and scan all servers that can be identified in the network of the company. It needs to be started with administrative rights to be allowed to create jobs on remote systems.

3.3.1 Simple scheduler service

Since Windows NT the simple scheduler API is supported. This API allows you to create a job on a remote system that will run in the context of LOCALSYSTEM. This approach has its problems because LOCALSYSTEM is no ordinary user. In fact it is not designed to have access to network shares over the network (it is a null session). The share must be prepared to allow this kind of access. In the setup process the share called SMOshare$ is modified for this purpose. This means that access from an external LOCALSYSTEM account is considered as a user acting in the user group everyone. Just to point out the consequences this also means that a foreign system plugged into the network can access the share and act as everyone. The NTFS settings will make sure that this will not cause to much harm but still it is more access than shares usually will allow.

```
\\yourserver\smoshare$\smoat.exe
```

3.3.2 Extended scheduler service

Since Windows 2000 a new scheduler API is supported. Here the jobs can be running in the context of a user account. To use this functionality the "SMOat.ini" needs to be modified first. Just exchange the AT command and its parameters with the SCHTASKS command and its parameters:

```
JOBCREATE=schtasks /create /tn SMOscan /tr
"\\YOURSERVER\SMOshare$\SMOscan.exe" /sc weekly /d SU /st
18:00:00 /ru %USERNAME% /rp %USERPASSWORD% /s %s
```

```
JOBDELETE=schtasks /delete /tn SMOscan /f /s %s
```

Make sure that YOURSERVER is replaced by the name of the server system hosting the shared directory SMOshare$.

The use of SCHTASKS will automatically make sure that the scan is running in a well-defined context and that the access to the network share SMOshare$ is done in a well-known context. The user and its password are passed as parameters while calling the tool SMOat:

```
\\yourserver\smoshare$\smoat –username:"x" –userpassword:"y"
```

Once you have configured the scheduler service to be used in your environment it is time to scan the servers.

3.3.3 Manual creation of scan jobs

Start SMOat directly or pass the user name and password as parameters. Then create a list of servers found in the network. This will send a broadcast message to all computers in the network. The computers with an active browser service will reply to the broadcast message. These systems will be written to the file called "SMOatComputers.ini". In addition to that you can edit the file to add computers manually.

To determine the servers the systems is querying the network in two passes. In the first pass all servers and all computers with active shares are found – these can be workstations or servers. In the second pass all real workstations will be queried. If a system has been found in the first pass as a server the second pass will correct the information to workstation. This process will make sure that servers and workstations can be treated separately.

Jobs to remotely scan the computers can now be created based on the list "SMOat-Computer.ini". This can be done by blindly going through the list or by scanning just those systems that are missing in the folder results.

3.3.4 Automated creation of scan jobs

All of the manual operations can be automated in a batch file. This batch can be used to automatically track and scan new server systems. For this the batch just needs to be started frequently in its own job created for this purpose:

```
SMOat.exe /computers:"servers" /operation:"updatelist"
SMOat.exe /computers:"servers" /operation:"jobmissing"
/username:"x" /userpassword:"y"
```

The supported operations are: "updatelist" to find new systems, "joball" to scan all systems in the list and "jobmissing" to scan all systems missing in the results folder.

3.4 Windows Virtualization

In the Windows environment different products for virtualization can be used. The following paragraphs will point out how these products are handled.

3.4.1 Microsoft App-V

In the product Microsoft App-V the applications are distributed in packages. Users are assigned rights to use these packages. At logon the client for App-V will update the start menu to show the users those applications that have been assigned to the user. If the user is clicking on the icon of the application for the first time or if the package is set to automatic downstreaming the package will be streamed to the client. After the downstreaming has been finished the application is installed on the current computer or session. Thus the scan will claim licenses for those applications that have been installed locally in the App-V client.

The scan will automatically identify that App-V is installed. It will adjust its behaviour accordingly and it will scan the App-V bubble for every installed package. The procedure for the scan of workstations and servers applies here (see 3.1 and 3.3).

On Terminal Servers the use of the applications will be monitored by SMOmonitorApplication that is started by SMOscan. The monitor will adjust its behaviour if an application running in an App-V bubble is detected. In this case the whole bubble for this application will be scanned and reported. As a result not only the used application is reported but all applications in the used package are reported. This is a different behaviour to the pure monitoring of used applications (see 3.2). On the positive side this approach needs less resources of the Terminal Server than monitoring the use of single applications in every App-V bubble. Due to the limited resources of Terminal Servers we are currently prefering this behaviour.

3.4.2 Microsoft Hyper-V and Microsoft Virtual Machines

The product Hyper-V is running virtual machines on Windows Servers. Every virtual machines needs to be scanned that is needed for productive use. The procedure for the scan of servers applies here (see 3.3).

3.4.3 VMware ThinApp

The product ThinApp has similar features than the product App-V. The behaviour to scan ThinApp is not implemented yet.

3.4.4 VMware XenApp and VMware Virtual Machines

The product XenApp makes the installation and distribution of applications easier. Different workstations and servers can be installed with a defined set of applications. These applications will be installed locally. As a result the servers, terminal servers and workstations can be scanned as usual (see 3.1, 3.2 and 3.3).

4 Apple environment

4.1 Apple Workstation or Server

The client for Apple Macintosh OS X can be started from the Windows share "SMOshare$". The executable is called "SMOscan.Macintosh.exe" and will work on all 64 bit capable Macbooks. It will use the same "SMOscan.ini" that the Windows scan is using. Windows and Macintosh use different naming schemes for network resources. Thus it is recommended to use MODE=HTTP for scanning both environments with the same settings in "SMOscan.ini".

All files described in the following paragraphs can be found on the Windows share in "SMOshare$\Clients\SMOscan Macintosh.zip".

4.1.1 Login script to execute scan

Create a script called "org.softwaremanagement.loginscript.sh". This script has to exist on every Workstation or Server. It will be executed at login and mounts the central "SMOshare$" in the home directory of the current user. Finally it will execute the scan "SMOscan.Macintosh.exe". Please replace SMOSERVER with the DNS name or IP address of the Windows server providing the "SMOshare$". The guest account should be activated on the Windows server to allow access:

```
#!/bin/bash
mkdir ~/mount
mkdir ~/mount/SMOshare$
/sbin/mount -t smbfs smb://guest@SMOSERVER/SMOshare$ ~/mount/SMOshare$
~/mount/SMOshare$/SMOscan.Macintosh.exe
```

Copy "org.softwaremanagement.loginscript.sh" to the folder "/Library/Scripts" of the workstation. Set its ownership and execution flags:

```
sudo cp org.softwaremanagement.loginscript.sh /Library/Scripts
sudo chown root /Library/Scripts/org.softwaremanagement.loginscript.sh
sudo chmod 755 /Library/Scripts/org.softwaremanagement.loginscript.sh
```

4.1.2 Automated start of login script at login

Create a file called "org.softwaremanagement.loginscript.plist". It has to exist on every Workstation or Server. This file will instruct Mac OS X to execute "org.softwaremanagement.loginscript.sh" at every login:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" \
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<dict>
         <key>Label</key>
         <string>org.softwaremanagement.loginscript</string>
         <key>ProgramArguments</key>
         <array>
                <string>/Library/Scripts/org.softwaremanagement.loginscript.sh</string>
         </array>
         <key>RunAtLoad</key>
         <true/>
</dict>
</plist>
```

Copy "org.softwaremanagement.loginscript.plist" to "/Library/LaunchAgents" with root rights. Set ownership and access flags:

```
sudo cp org.softwaremanagement.loginscript.plist /Library/LaunchAgents
sudo chown root /Library/LaunchAgents/org.softwaremanagement.loginscript.plist
sudo chmod 644 /Library/LaunchAgents/org.softwaremanagement.loginscript.plist
```

4.1.3 Refinement for centralized management

In a more refined approach the LoginScript.sh should not call SMOscan.Macintosh.exe directly. Instead it should call a script that is located in SMOshare$:

```
#!/bin/bash
mkdir /tmp/SMOshare$
/sbin/mount -t smbfs smb://guest@SMOSERVER/SMOshare$ /tmp/SMOshare$
source /tmp/SMOshare$/LoginScript.Macintosh.sh
```

This centralized LoginScript called "LoginScript.Macintosh.sh" gives you more flexibility to adjust the execution of "SMOscan.Macintosh.exe" and other executables you might find important to execute:

```
#!/bin/bash
/tmp/SMOshare$/SMOscan.Macintosh.exe
```

5 Linux environment

5.1 Linux Workstation or Server

Similar to the Macintosh all Linux systems are capable to execute applications from Windows shares. The executable is called "SMOscan.Linux.exe" and will work on most 64 bit capable Linux systems following the LSB standard. The scan will use the same "SMOscan.ini" that the Windows scan is using. Windows and Linux use different naming schemes for network resources. Thus it is recommended to use MODE=HTTP for scanning both environments with the same settings in "SMOscan.ini".

The Linux scan will read the installed packages from RPM or DPM. The hardware part is using the package LSHW for analysis. This package should be installed prior to the execution of SMOscan to get reliable hardware information. Search for LSHW (Hardware Lister) in the package manager of your Linux system.

All files described in the following paragraphs can be found on the Windows share in "SMOshare$\Clients\SMOscan Linux.zip".

5.1.1 Auto mounting SMOshare$

Linux like the Macintosh allows individual mounting of the central "SMOshare$". But very likely most Linux systems are running in typical server configuration. Thus it is not useful to wait for the logon of individuals to execute the scan. Instead a job should execute the scan periodically. For this scenario it would be more appropriate to mount the "SMOshare$" permanently. The guest account should be activated on the Windows server to allow access. This solution can be applied to the Macintosh too.

Usually a directory in the local file system acts as a mount point to the shared folder. Open the command shell and create a directory in /media called SMOshare$ with root access:

```
sudo mkdir /media/SMOshare$
```

Open the file "/etc/fstab" with root access:

```
sudo gedit /etc/fstab
```

Add the following line to "/etc/fstab". It will instruct Linux to mount the share at logon automatically. Please remove line breaks to add just one line to fstab. Please replace "SMOSERVER" with the DNS name or IP address of your Windows server:

```
//SMOSERVER/SMOshare$   /media/SMOshare$   cifs
guest,_netdev,soft,nosuid,uid=1000,gid=1000,vers=2.1   0   0
```

The Linux system can now execute SMOscan.Linux.exe via /media/SMOshare$.

5.1.2 Login script to execute scan

Create a script called "org.softwaremanagement.loginscript.sh". This script has to exist on every Workstation or Server. It will execute the scan "SMOscan.Linux.exe":

```
#!/bin/bash
/media/SMOshare$/SMOscan.Linux.exe
```

5.1.3 Automated start of login script every week

Copy "org.softwaremanagement.loginscript.sh" to the folder "/etc/cron.weekly". This will instruct the cron daemon to execute the script every week. Make adjustments to the execution rights as necessary:

```
sudo cp org.softwaremanagement.loginscript.sh /etc/cron.weekly
sudo chown root /etc/cron.weekly/org.softwaremanagement.loginscript.sh
sudo chmod 755 /etc/cron.weekly/org.softwaremanagement.loginscript.sh
sudo touch /etc/cron.weekly/org.softwaremanagement.loginscript.sh
sudo anacron -f -d
```

The last instruction should test the execution of the cron jobs. According to the cron documentation you have other options as well like executing daily and monthly. This is very much open to your management environment and skills.

5.1.4 Refinement for centralized management

Similar to the Macintosh environment the script to execute the scan can be located on the share itself. The local script "org.softwaremanagement.loginscript.sh" will just trigger the central script called "LoginScript.Linux.sh":

```
#!/bin/bash
source /media/SMOshare$/LoginScript.Linux.sh
```

The "LoginScript.Linux.sh" will be similar to the script in chapter 5.1.2.

6 Collecting results over different channels

The standard installation creates the central share called 'SMOshare$'. This share acts as the central place where scan results will be created in the subdirectory 'Results'.

6.1 Network shares

The SMOscan will create a ZIP file according to the name and domain of the scanned windows system. This ZIP will then be transferred to the destination. For this purpose the SMOscan running in the context of the current user needs to have full access to the subdirectory 'Results' via the rights of the used file system.

6.1.1 Configuration of SMOscan.ini

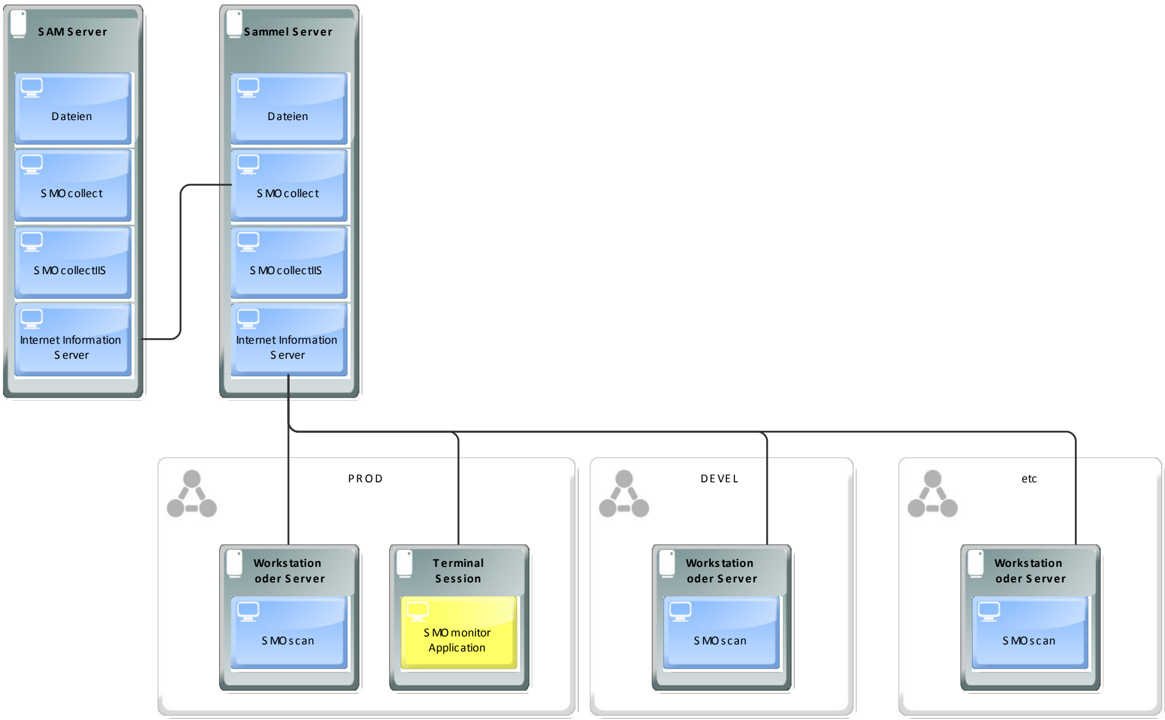
```
[Options]
MODE=FILE
DESTINATION=\\yourserver\smoshare$\results
```

Please be aware that operating systems might show different behaviors. Windows is not case sensitive but OS X and Linux are. Windows can directly address the shares of other systems with the UNC naming convention. OS X and Linux need to mount the shared folder of remote systems first. Then this local folder is used to address the remote server indirectly.

6.2 HTTP/HTTPS

The file mode has the undesirable side effect that all users have full write and read access on 'SMOshare$\Results'. This will also allow the uncontrolled exchange of files. This is not tolerable in environments with a high security profile. As an alternate the SMOscan can deliver its results via HTTP/HTTPS to a specially prepared Internet Information Server. This in comparison is a very restricted approach and at the same time it even allows the exchange of data over the boundary of domains. Furthermore the use of this channel does simplify the configuration of the Firewalls because the communication is reduced to the port 80 or 443 respectively.

The use of web technologies offers a path to scalability too. The following picture shows a high scalability approach. One collect server is getting the results from workstations, terminal sessions and servers in the different environments PROD, DEVEL and so forth. The collection server could actually be a farm of Internet Information Server fed with web requests by a centralized load balancer. On the IIS servers our module 'SMOcollectIIS.isa' will be installed. This module is transforming the communication stream back to an ordinary scan result in the form of a file.



6.2.1 Windows 2003: Installation of IIS

System panel > Add/Remove Software > Add Windows Components > Internet Information Service (IIS) > Details

[x] Common files

[x] Snap-In

[x] World Wide Web Service

6.2.2 Windows 2008/2012: Installation of IIS

Administrative Tools > Server Manager: add the role Web Server (IIS) with the features Application Development > ISAPI Extensions, Management > IIS Scripts, Management > IIS 6 compatibility > IIS 6 Scripting Tools.

Alternativ über folgendes Script:

```
pkgmgr /iu:IIS-WebServerRole;IIS-WebServer;IIS-CommonHttpFeatures;IIS-StaticContent;IIS-DefaultDocument;IIS-DirectoryBrowsing;IIS-HttpErrors;IIS-HttpRedirect;IIS-ApplicationDevelopment;IIS-ASPNET;IIS-NetFxExtensibility;IIS-ASP;IIS-CGI;IIS-ISAPIExtensions;IIS-ISAPIFilter;IIS-HealthAndDiagnostics;IIS-HttpLogging;IIS-LoggingLibraries;IIS-RequestMonitor;IIS-HttpTracing;IIS-Security;IIS-BasicAuthentication;IIS-URLAuthorization;IIS-IPSecurity;IIS-RequestFiltering;IIS-Performance;IIS-HttpCompressionStatic;IIS-HttpCompressionDynamic;IIS-WebServerManagementTools;IIS-ManagementConsole;IIS-ManagementScriptingTools;IIS-ManagementService;WAS-WindowsActivationService;WAS-ProcessModel;WAS-NetFxEnvironment;WAS-ConfigurationAPI;IIS-IIS6ManagementCompatibility;IIS-Metabase;IIS-WMICompatibility;IIS-LegacyScripts;IIS-LegacySnapIn
```

6.2.3 Create new website (Port 80 if possible)

In the standard configuration the standard site is 'c:\inetpub\wwwroot'. New sites can be created with System panel > Management > Internet Information Service. If possible the new site should be a subdirectory of wwwroot to inherit the appropriate NTFS rights necessary for operation.

6.2.4 Copy modul SMOcollectIIS.isa to website

It is possible to run the Internet Information Server in 32- or 64-bit mode. The following command will activate the mode via the command shell:

32 bit:

```
cscript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET W3SVC/AppPools/Enable32bitAppOnWin64 1
```

64 bit:

```
cscript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/Enable32bitAppOnWin64 0
```

Now the module that fits to selected mode needs to be copied to the created website. Choose SMOcollectIIS.isa for 32-bit and SMOcollectIIS64.isa for 64-bit. The 64 bit module needs to be renamed to 'SMOcollectIIS.isa' after it has been copied to the website.

6.2.5 Windows 2008/2012: additional step to copy module

Please copy 'SMOcollectIIS.isa' to the file 'SMOcollectIIS.dll'. As a result you will have the two files 'SMOcollectIIS.isa' and 'SMOcollectIIS.dll' in the created website. This step is necessary to compensate flaws in the Windows Server 2008 and 2012.

6.2.6 Prepare the user and password

IIS servers in the Internet need to make sure that the results they are receiving are really meant for the company. Thus the SMOscan will send a user and password that is verified before the result is accepted. This user is not a real windows account. Instead it is created in the registry of the IIS server.

This is done with the file 'SMOshare$\SMOcollectIIS.reg'. Open the file with an editor and replace the User with your own choice. In Password and Destination you can insert you own password and target directory to save the scan results to.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Softwaremanagement.org\SMOcollectIIS\User]
"Password"="xxx"
"Destination"="xxx"
```

Attention: use double slash characters in Destination – like the following example

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Softwaremanagement.org\SMOcollectIIS\Test]
"Password"="Secret"
"Destination"="c:\\Programme\\Softwaremanagement.org\\SMOshare$"
```

The selected directory needs to grant full access to the windows users IUSR_COMPUTERNAME and IWAM_COMPUTERNAME and IIS_WPG (if available). This is necessary to allow the module running in the IIS in the context of one of these users to write the data stream to a result file.

Save the registry file after these modifications. Then import the file into the registry of the web server running the IIS module.

6.2.7 Configuration of website

System control > Management > Internet Information Service > Tab "Root Directory"

[x] Read
[x] Log visits

Execution rights: Scripts and executable files

6.2.8 Windows 2003: activate module SMOcollectIIS.isa

Open the IIS management. In the configuration you can modify the registered Web Service Extension. Add a new extension called "SMOcollectIIS.isa" and assign it to "SMOcollectIIS.isa" in the directory of the website. This will give the IIS the permission to load and use the module.

6.2.9 Windows 2008/2012: activate module SMOcollectIIS.isa

Open the IIS management. In the configuration of the website you will find the handler assignment on the top right. Create a new handler assignment and in the following dialog you will have to set the following information:

Query path: SMOcollectIIS.isa
Module: IsapiModul
Executable file: PATH OF THE WEB SITE\SMOcollectIIS.dll
Name: SMOcollectIIS

6.2.10 Verify configuration

Start the browser and open the following website to verify your configuration:
http://127.0.0.1/SMOcollectIIS.isa?check&username=Test&userpassword=Secret

The module SMOcollectIIS will reply with the following page:

```
SMOcollectIIS - checking setup...
OK - username 'Test' exists in registry
OK - password 'Secret' is correct
OK - created file 'c:\Programme\Softwaremanagement.org\SMOshare$\SMOcollectIIS-SUCCESS.txt'
Copyright © 2000-2014 Softwaremanagement.org
```

6.2.11 Configuration of SMOscan

The user and password configured in the registry of the IIS server must be set in the "SMOscan.ini". This way the SMOscan can send these credentials to be allowed to upload the result of the scan to the web server:

```
[Options]
USERNAME=Test
USERPASSWORD=Secret
MODE=HTTP
DESTINATION=195.202.37.66 (IP or DNS name without http:)
PORT=80
PORTSSL=off
PROXYSERVERNAME=195.202.37.3:8080 (IP or DNS:Port without http:)
```

If you do not use a proxy server then remove the parameter from the configuration file. The IIS should now accept the results of the SMOscan. Furthermore it will accept results send from SMOcollect services via the relay functionality.

7 Cost centers

In the context of the Software Management Suite the "cost center" is the smallest object for organization. So the "cost center" can adjust to your level of thinking. It does not have to represent what is known as "cost center" in the business sense.

It is the smallest atom for organization and Licenses and workplaces can be bound to these "cost centers". Finally in the reporting the "License Balance Report" can be created for selected "cost centers".

7.1 Motivation for cost centers

The cost center mechanism can be used to created reports for specific organizational units. For example the three cost centers "Hamburg", "London" and "Vienna" could be created. The licenses and workplaces are assigned to these cost centers. Thus the "License Balance Report" can be created for "Hamburg" or "Vienna" alone. Of course the reporting will also offer the eagle perspective by including all of the cost centers.

The assignment of the license to the cost center is a manual process. But the assignment of workplaces to cost centers can be automated with the help of SMOscan.

7.2 Automation via SMOscan

The standard configuration does collect all of the scan results in the directory "SMOshare$\Results". The service SMOcollect will import the results into the database and the cost center will be "none".

This behavior changes when a subdirectory is created in the folder "Results". For example we create the directories "Hamburg" and "London".

Now we just have to make sure that the SMOscan is writing the result to the correct directory. Instinctively but not recommendable this can be done with different SMOscan configuration files like "SMOscanHamburg.ini" and "SMOscanLondon.ini". Then the SMOscan at the specific location will be called with the configuration file appended like SMOscan SMOscanHamburg.ini.

This will work but quite obviously this has disadvantages. Instead of having one centralized configuration file we will have multiple.

Instead it is recommended to tell the SMOscan which cost center to use. This can be done with the calling parameter /cc: like in the following example. The configuration file just points to the directory "Results":

SMOscan /cc:"Hamburg"

This way the scan will append the subdirectory "Hamburg" to the directory "Results". Then the result will be created there. The service SMOcollect will import the result and the assigned cost center will be "Hamburg". Right now the /cc: parameter is supported for Windows and OS X.

If you collect results at different sites it is also recommended to create a subdirectory in the local result directory:

Hamburg:	SMOshare$\Results\Hamburg SMOscan /cc:"Hamburg"
London:	SMOshare$\Results\London SMOscan /cc:"London"

The use of dedicated directories will allow the SMOcollect service in the headquarter London to collect the results from the different locations (like a spider in the net). It will be configured to collect from "\\Hamburg\SMOshare$\Results" at a specific time window. In the process the containing subdirectories will be collected too. Thus the directory "Hamburg" its contents is transferred as well. With the help of specified directories the different locations can be kept separately. Their results will be assigned to cost centers with the same name as the directory.

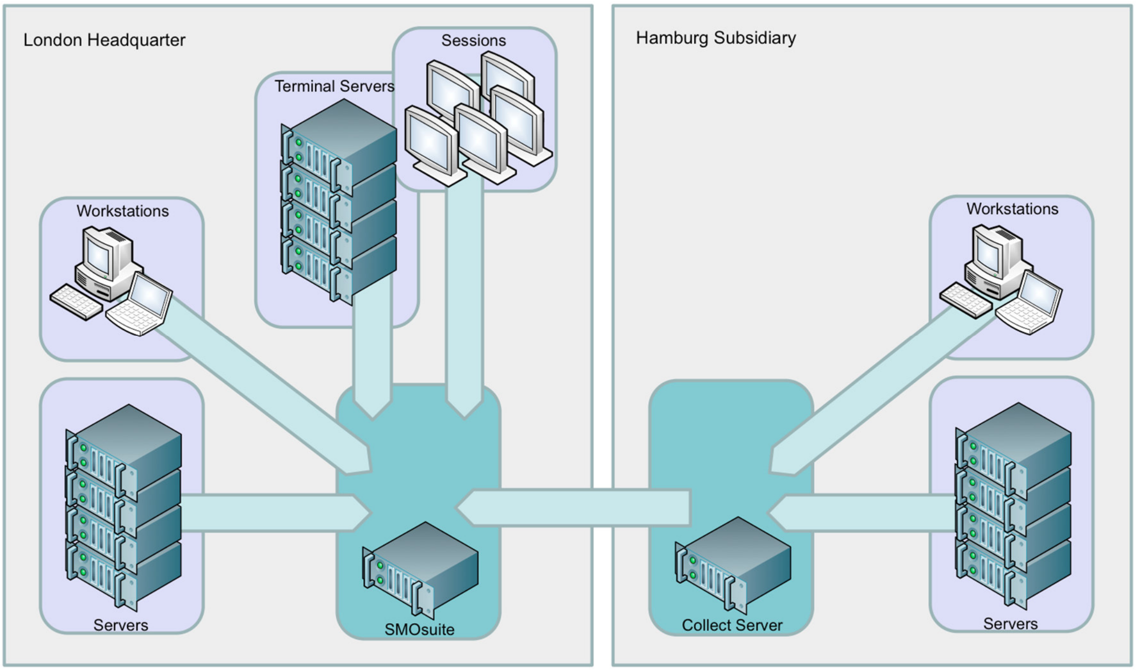
The collection from different sites is described in more detail in the next chapter.

8 Complex scenario with subsidiaries

Many companies have multiple subsidiaries and one headquarter. Due to limitations in bandwidth it makes sense to install the scan on a server of the subsidiary. This way the scan does not consume valuable bandwidth for every logon of a user.

As described in the previous chapter it does make sense to create a subdirectory Results/London in the headquarter and Results/Hamburg in the Hamburg subsidiary. The results are two independent sites collecting scan results.

Now the results of the Hamburg subsidiary just need to be transferred to the London headquarter for analysis. For this the service SMOcollect offers two solutions:



8.1 Pulling results from Hamburg to London

Very likely the London headquarter has direct file access to the collect server of Hamburg. Thus the central service SMOcollect can act like a spider in the net. It will be configured to pull the scan results from all of the subsidiaries at specific times.

For this the subsidiary needs to be defined as a source and the time window needs to be defined. All this can be done in the main configuration file "SMOcollectSrv.ini" of the service "SMOcollect".

The following setting defines the SubsidiaryHamburg as a source. This source will be pulled at the days and times specified in RELAYn. While pulling all the subdirectories in \\HHSMO\SMOshare$\Results will be pulled. This makes sure that the subdirectory Hamburg with all of its results gets pulled too. After the first pull the central results directory will contain the directories London and Hamburg – a clean separation of the different sites that can be used for automated assignment of cost centers. The SubsidiaryVienna is only intended to point out that multiple sources can be named here.

```
[Options]
...
RELAY1=MON,00:00-05:59,SubsidiaryHamburg
RELAY2=TUE,00:00-05:59,SubsidiaryHamburg
RELAY3=WED,00:00-05:59,SubsidiaryHamburg
RELAY4=THU,00:00-05:59,SubsidiaryHamburg
RELAY5=FRI,00:00-05:59,SubsidiaryHamburg,SubsidiaryVienna
RELAY6=SAT,00:00-05:59,SubsidiaryHamburg,SubsidiaryVienna
RELAY7=SUN,00:00-05:59,SubsidiaryHamburg,SubsidiaryVienna

[SubsidiaryHamburg]
TYPE=SOURCE
MODE=FILE
DIRECTORY=\\HHSMO\SMOshare$\Results

[SubsidiaryVienna]
TYPE=SOURCE
MODE=FILE
DIRECTORY=\\VNSMO\SMOshare$\Results
```

In this scenario the subsidiary Hamburg is passive. Still the local service SMOcollect should not be deactivated because it is needed for updates (see last chapter). It makes sense to deactivate the import functionality in Hamburg. It is not needed because the database is just maintained in London. This can easily be done in the SMOcollectSrv.ini of Hamburg:

```
[Options]
IMPORT=off
```

8.2 Pushing results from Hamburg to London

Sometimes the subsidiary is more loosely coupled with the headquarter. In this case a direct file access from headquarter to the collect server is not possible. On the other hand headquarter and subsidiary will have very likely access to the Internet. So the headquarter could use an Internet Information Server to receive the results via HTTP or HTTPS as mentioned in the previous chapters.

For this the Hamburg subsidiary needs to actively push the results to the public IIS of the headquarter. So instead of modifying the SMOcollectSrv in London it is now the file in Hamburg that needs to be adjusted.

Again the push will include all subdirectories under directory given in the parameter DESTINATION. Thus the subdirectory Hamburg with all of its results will be transferred to the London headquarter too.

```
[Options]
...
RELAY1=MON,03:00-05:59,HeadquarterLondon
RELAY2=WED,03:00-05:59,HeadquarterLondon
RELAY3=FRI,03:00-05:59,HeadquarterLondon

[HeadquarterLondon]
TYPE=DESTINATION
MODE=HTTP
PORT=443
PORTSSL=on
DIRECTORY=192.168.178.1
USERNAME=Test
USERPASSWORD=Secret
```

8.3 Updating different sites

Obviously the maintenance of multiple sites is more challenging for the administration. Therefore the service SMOcollect is watching the subdirectory SMOshare$\Update for changes. All updates for the SMOscan will be released as "SMOscanUpdate.zip" and "SMOmonitorUpdate.zip". Just copy the file to the subfolder Update and the service SMOcollect will automatically finish the update process. The outcome of the update will be documented in the log file "SMOcollectSrv.log".

If multiple subsidiaries are involved we recommend creating a central batch file. This batch file will just copy the "SMOscanUpdate.zip" to all of the subsidiaries.

9 Appendix

9.1 Naming convention of scan results

Windows workstation or server	pcNAMEinDOMAIN.zip
Windows terminal server MONITOR	taUSERinUSDOMAINonTSNAMEinTSDOMAINfromDEVICE.zip
Windows terminal server NTFS	tuUSERinUSDOMAINonTSNAMEinTSDOMAINfromDEVICE.zip
Linux workstation or server	lxNAME.zip
Mac OS X workstation or server	mcNAME.zip

9.2 Calling parameters of SMOscan

SMOscan x /cc:"y"

The calling parameter x is the name of the configuration file to use. If x is not provided then the standard configuration file called SMOscan.ini is used.

The calling parameter /cc:"y" will provide the cost center to use. The DESTINATION parameter of the configuration file will be combined with the cost center y to determine the destination folder of the result.

9.3 Configuration parameters of SMOscan.ini

9.3.1 Performance

[Options] **STARTDELAY**=x|RANDOM(x,y)

After the start of SMOscan the application will sleep for x milliseconds before the scan continues. This parameter can also be randomized by drawing the number randomly between x and y. The parameter will allow the startup of the computer to complete before the scan starts to analyze the hard drives.

[Options] **DELAY**=x|RANDOM(x,y)
[Options] **DELAYSERVER**=x|RANDOM(x,y)

The process SMOscan will sleep for x milliseconds or a randomized number after every file that has been analyzed. This will put less strain on the operating queue of the hard drive reserving more disk performance for the current user or system. The parameters for workstation (DELAY) and servers (DELAYSERVER) can be set separately.

[Options] **PRIORITY**=LOWEST|IDLE

The process SMOscan will be running either with LOWEST or IDLE priority. This will guarantee a sufficient number of free processor cycles to the user and other services.

9.3.2 Lifetime

[Options] **LIFETIME**=x|RANDOM(x,y)
[Options] **LIFETIMESERVER**=x|RANDOM(x,y)

9.3.3 Behavior

The following parameters are more or less exclusive for the windows platform.

LINKS=on|off
LINKSSERVER=on|off

Links to applications loaded from UNC network shares will be followed and the application will be analyzed and integrated into the scan result. For server systems this often does not make sense. Thus it is recommended to set LINKSSERVER=off.

LINKSWEB=on|off

Rarely the links to internal websites are important for licensing. In this case set this parameter to on to analyze these links. In the tool SMOadmin you will have to assign the link to a product of the software catalog to allocate licenses on all workplaces with this specific link. In most cases products with web-based tools will manage their licensing autonomously - via the users allowed for login for example.

WINDOW=on|off

Only the scan for windows offers a graphical user interface to provide some feedback about the scan process. This is useful for scanning standalone systems or on demand scanning. For the use in login scripts it is recommended to suppress the window.

ERRORS=on|off

With the GUI enabled it might be useful to provide dialogs to the user in case of an error. This is useful for testing the scan and for scanning with the window enabled. In any other case it is recommended to turn the dialogs for error messages off.

OFFICE=KEY|GUID

For several years the recommended way to identify the Microsoft Office edition was the GUID of the installer. Microsoft also adopted the ISO19770 standard for software identification tags. The scan will query these tags to identify the edition.

Unfortunately the newer Office versions do not support these standards correctly.With keys every possible edition can be unlocked. The problem is that the GUID and Software Tag will not reflect the unlocked edition. Instead the previous information from the installation is preserved.

To solve these issues the combined mode KEY|GUID will identify the edition according to the used key. As a fallback it will use the GUID.

OWNERSHIP=on|off

The setting “off” will supress the information about the owner of the executable file that is available in NTFS file systems.

ANONYMOUS=on|off

The setting “on” will remove all private information about users: name and user groups of current user, ownership information of files.

9.3.4 Ignore directories, users or computers

[**DirectoriesIgnored**] NOn=y

Sometimes directories need to be ignored on purpose. Especially on server systems you will likely find directories for the sole purpose of providing software packages for installation. If the package contains the executable files in an uncompressed form then the SMOscan will likely find them. Thus licenses are allocated for software that is only provided for others not installed locally. To circumvent the problem a number of entries counting from 1 to 100 can be provided to describe the ignored directories. For example NO1=\Installation\ will ignore ALL directories containing the text "\Installation\". For OX X this was improved with a more mature approach where * or complex regex expressions can be used for filtering. The general problem with these directory filters is that they are applied to all directories without exception. But in some cases you would like to filter on servers only for example.

Thus we have provided a second mechanism to ignore directory that works for servers only. Just create a file called "**SMOscan.off**" in a directory on a server system. This file will make sure that the directory is ignored. Since Administrators of these systems can only create files the approach is fairly safe from being misused. Of course ordinary users cannot exclude any directories on their workstations by creating this file. The file "SMOscan.off" is also useful for terminal servers to exclude software hosted for terminal sessions only. This way the results of the session monitoring will contain the software but not the scan of the terminal server itself. Please consult your product use rights to make sure that the applied filters are permitted.

[**ComputersIgnored**] NOn=y

Due to technical or jurisdictional restrictions some computers might be excluded from the scan. To make sure that the scan is not executed it can be defined in this section. For convenience * and more complex regular expressions can be used. For example NO1=REAL* will exclude all computers named with the text "REAL" at the beginning.

[**UsersIgnored**] NOn=y

The same filter can be applied to users. Just define the user filter with * or as a regular expression. For example NO1=REGEX:CEO[NE].* will exclude all users named with CEON or CEOE at the beginning - but not CEOS. The term REGEX: makes sure that the filter is used as a regular expression.

9.3.5 Terminal session

[Options] **TERMINALSESSIONMODE**=MONITOR|NTFS

Set to MONITOR the scan will monitor the used applications in the terminal session by starting "SMOmonitorApplication.exe". The result is a file named "ta...zip". Set to NTFS the scan will include all applications in local drives of the Terminal Server that the current user has execute rights on (according to the NTFS rights and the groups of the user). The result is a file called "tu...zip". Set to MONITOR|NTFS the scan will combine both modes and create two results.

[Options] **DELAYSESSION**=x|RANDOM(x,y)

Similar to DELAY and DELAYSERVER the parameter DELAYSESSION will put the scan to sleep. This happens between every NTFS file or process that has been analyzed.

[Options] **MONITORAPPLICATIONCYCLE**=x

The monitoring of the terminal session will repeatedly query the list of active processes. The time to wait between these queries is defined in x milliseconds.

[Options] **INSTANCES**=x

If TERMINALSESSIONMODE uses NTFS then the scan will analyze the files on the hard drives of the terminal server. If this happens in all sessions in parallel then the workload on the hard drive could have a negative impact on the productive work in the sessions. Therefore a maximum of 10 parallel instances of the scan is allowed. The standard limit is 4.

[**ComputersRename**] NOn=x,y

As mentioned in the naming convention for results the name for sessions does contain the name of the terminal server. Most often the terminal servers use one central load balancer. This will distribute the users according to the workload of the terminal servers. As a result you will have multiple results of the scan with all the different terminal servers involved. Thus it is recommended to aggregate all the equally installed terminal servers under one name for the terminal farm. The section ComputersRename will allow to rename the server x to the name of the farm y. The following example will rename all terminal servers beginning with TSA to be renamed to FARM1 and all beginning with TSB to be renamed to FARM2.

```
NO1=TSA*,FARM1
NO2=TSB*,FARM2
```

More complex rules can be implemented with regular expressions:

```
NO1=REGEX:TS[AC].*,FARM1
NO2= REGEX:TSB[24].*,FARM2
```

This is very important to reduce the amount of data that is processed in the databases of the Software Management Suite. Thus the mechanism to aggregate servers to farm should be used from the start. To switch to farm names at a later stage will lead so a significant amount of work to remove all the already collected data!

This section is also useful for the scan of Exchange clusters. The scan will be executed on every server that is part of the cluster. The logical name used for the cluster is needed to identify the Exchange software that is running on the cluster:

```
NO1=EXCHG1,EXCHA
NO2=EXCHG2,EXCHA
```

As a result the scan can query the logical cluster EXCHA to determine the Exchange software running on server EXCHG1.

[**SessionsIgnored**] NOn=x

Similar to users or computers ignored specific sessions can be ignored too. You can define an ignore filter for the file name of the session. Here that can use * or more complex regular expressions. For example NO1=*fromVPN* will filter all results that came from a device name "VPN" at the beginning.

9.4 Checklists

9.4.1 Installation

□ Windows Server: Features general

```
DISM /enable-feature /online /featureName:NetFx4 /featureName:NetFx4ServerFeatures /featureName:NetFx4Extended-ASPNET45 /featureName:IIS-WebServerRole /featureName:IIS-WebServer /featureName:IIS-CommonHttpFeatures /featureName:IIS-StaticContent /featureName:IIS-DefaultDocument /featureName:IIS-DirectoryBrowsing /featureName:IIS-HttpErrors /featureName:IIS-HttpRedirect /featureName:IIS-ApplicationDevelopment /featureName:IIS-ASPNET45 /featureName:IIS-NetFxExtensibility45 /featureName:IIS-ISAPIExtensions /featureName:IIS-ISAPIFilter /featureName:IIS-HealthAndDiagnostics /featureName:IIS-HttpLogging /featureName:IIS-LoggingLibraries /featureName:IIS-Security /featureName:IIS-RequestFiltering /featureName:IIS-Performance /featureName:IIS-HttpCompressionStatic /featureName:IIS-WebServerManagementTools /featureName:IIS-ManagementScriptingTools /featureName:IIS-Metabase /featureName:IIS-WMICompatibility /featureName:NetFx4Extended-ASPNET45 /featureName:IIS-ApplicationInit /featureName:IIS-ManagementConsole /featureName:IIS-LegacySnapIn /featureName:WAS-ProcessModel /featureName:WAS-WindowsActivationService /featureName:WAS-ConfigurationAPI /featureName:WCF-Services45 /featureName:WCF-HTTP-Activation45 /featureName:IIS-IIS6ManagementCompatibility
```

□ Windows Server: Features .Net 2.0 to 3.5

```
DISM /Online /Enable-Feature /FeatureName:NetFx3 /All /Source:d:\sources\sxs
```

□ SMOmanagerWeb:

127.0.0.1, Advanced Settings: 32 Bit for Application Pool, WCF45

□ SMOcollectIIS:

Handler Mapping: ISAPI-dll, Edit Feature Permissions: Execute, Request Restrictions: Execute

Handler Mapping: SMOcollectIIS.isa, Edit Feature Permissions: Execute, Request Restrictions: Execute

http://127.0.0.1/SMOcollectIIS.isa?check&username=test&userpassword=test

□ SQL Server: Setup

Collation: Latin1_general_Ci_As, Database and Reporting Services

Move Temp Database if necessary:

```
USE master

ALTER DATABASE tempdb MODIFY FILE (NAME = tempdev, FILENAME =
'T:\SQL\MSSQL10_50.MSSQLSERVER\MSSQL\Data\tempdb.mdf');

ALTER DATABASE tempdb MODIFY FILE (NAME = templog, FILENAME =
'T:\SQL\MSSQL10_50.MSSQLSERVER\MSSQL\Data\templog.ldf');
```

□ SMOdatabase: Setup

Manual Separation of Data and Log Partitions:

```
DECLARE @source varchar(500)

DECLARE @destination_data varchar(500)

DECLARE @destination_log varchar(500)

DECLARE @sql varchar(5000)


SET @source = 'C:\Install\SMOsuite\SMOdatabase\Data'

SET @destination_data = 'D:\SQL\MSSQL10_50.MSSQLSERVER\MSSQL\Data\SMO'

SET @destination_log = 'L:\SQL\MSSQL10_50.MSSQLSERVER\MSSQL\Data\SMO '


SET @sql = 'RESTORE DATABASE SvProDaten FROM DISK = ''' + @source +
'\SvProDaten'' WITH MOVE ''SvProDaten_data'' TO ''' + @destination_data +
'\SvProDaten_data.mdf'', MOVE ''SvProDaten_log'' TO ''' + @destination_log +
'\SvProDaten_log.ldf'', replace'

EXEC (@sql)

SET @sql = 'RESTORE DATABASE SMOdatabase FROM DISK = ''' + @source +
'\SMOdatabase'' WITH MOVE ''SMOdatabase_data'' TO ''' + @destination_data +
'\SMOdatabase_data.mdf'', MOVE ''SMOdatabase_log'' TO ''' + @destination_log
+ '\SMOdatabase_log.ldf'', replace'

EXEC (@sql)

SET @sql = 'RESTORE DATABASE SMOconnect FROM DISK = ''' + @source +
'\SMOconnect'' WITH MOVE ''SMOconnect_data'' TO ''' + @destination_data +
'\SMOconnect_data.mdf'', MOVE ''SMOconnect_log'' TO ''' + @destination_log +
'\SMOconnect_log.ldf'', replace'

EXEC (@sql)

go

EXEC SvProDaten.dbo.p_RestoreUsers
```

□ SMOscan: Setup

SMOscan.ini: For MODE=HTTP:

USERNAME=

USERPASSWORD=

Safety Advisory: in HTTP mode you should copy SMOshare$\Results to Softwaremanagement\Results. This way the results are not accessible via the share.

Check OFFICE mode: OFFICE=KEY|GUID

Locations of Scan Server Installations:

RELAY1=

RELAY2=

□ Activation of User Information:

Download SMOfeatureSoftwaremanagement.zip and Copy to SMOshare$

Start SMOdatabaseMgr and select Activate and then Start

□ SMOmessageSrv: blat configuration:

blat.exe -install MAILSERVER 127.0.0.1 3 25 SMOmessageSrv USERNAME USERPASSWORD

blat SMOmessageSrv.ini -p SMOmessageSrv -to "RECIPIENT@YOURCOMPANY.COM" -f "SENDER@YOURCOMPANY.COM" -replyto "SENDER@YOURCOMPANY.COM" -subject "blat.exe: test successfull" -debug -hdrencb

□ SMOadInterface: Setup

Assignment of logins to AD users via SMOcollectSrv.ini:

ASSIGNEMPLOYEE=on

CREATEEMPLOYEE=off

□ SMOvCenter: Setup

SMOvCenter.ini: For MODE=HTTP create additional IIS user to save to SMOshare$\Exchange\XML:

USERNAME=

USERPASSWORD=

VMware PowerCLI 5.5 Download

https://my.vmware.com/web/vmware/details?downloadGroup=PCLI550&productId=352

Call in Batch: SMOvCenter /VSERVER=Vcenter.contoso.com /VMUSER=SMOIMPORT /VMUSERPASSWORD=SMOIMPORT

□ Preparation of Servers

Installation Directories: SMOscan.ini, SMOscan.off

Citrix Terminal Servers

Farms of Terminal Servers or Exchange Servers