

HOW TO COMPILE AND INSTALL OCS INVENTORY NG AGENT UNDER SOLARIS

I.	Introduction	3
A.	Purpose	3
B.	Philosophy	3
C.	A word of caution	3
II.	Prerequisites	4
III.	Downloading the software	5
A.	Prerequisite software	5
B.	Perl modules	6
C.	Main software	6
IV.	Installing prerequisite software	7
A.	Package installation	7
1.	Sneep	7
2.	Sunfreeware packages	7
B.	Post-installation configuration	7
V.	Compiling and installing the required Perl modules	8
A.	Preliminary configuration	8
B.	Generic steps	8
C.	Special cases	9
VI.	Compiling and installing the agent	11
VII.	Compiling and installing the standalone binary	12
A.	Purpose	12
B.	Building	12
C.	Installation	12
VIII.	Post-installation	13
A.	Configuration	13
1.	Sneep configuration	13
2.	OCS agent configuration file	13
3.	Specific configuration	14
B.	First time launch: standard agent	14
C.	First time launch: standalone agent	14
D.	Scheduling	15
E.	Cleaning up	15
IX.	Deployment	16
A.	Prerequisite software	16
B.	Standalone OCS agent	16
C.	Serial number registration	16
D.	Manual inventory or scheduling	16

I. Introduction

A. Purpose

This Howto describes the steps to get the OCS Inventory NG agent up and running on a Solaris 10 box. It also gives a technique to deploy the agent to other Solaris hosts.

It has been tested on Sparc hardware running the 05/09 release of Solaris 10, with OCS Inventory NG Agent version 1.02 (internal version 1.0.1). However, it should work on i86 and also with different Solaris 10 and OCS agent releases.

Compilation and installation of generic software under Solaris is generally much more difficult than on the mainstream platforms: Windows and Linux, and also much less documented. This is why platform-specific documents such as this one are needed.

B. Philosophy

Several assumptions are made about the directories and configuration options to use. This aims at simplifying the document. Experienced administrators may make different choices.

The focus here is on fine-grained, step-by-step control of the installation. For instance, the CPAN tool for automated downloading and dependency checking of Perl modules is not used.

C. A word of caution

Installation has been tested on machines which had extra software installed, such as found on the Solaris "software companion" DVD or popular free software found on the Internet. So there might be more software to install from a fresh Solaris 10 installation than is described in this document.

II. Prerequisites

Before starting the installation, please make sure that you have the following:

- An operative OCS Inventory NG server.
- A Solaris 10 box (obviously !). It is assumed that the entire Solaris distribution is installed. If not, some packages may be missing for proper operation of the agent.
- Super-user ("root") access to the target Solaris machine.

All the commands in this procedure must be executed as super-user.

- Access to the Internet.

Direct access from the target Solaris 10 machine is not required.

III. Downloading the software

The software listed below must be downloaded from the Internet using the Web sites in the “source” column. It must then be copied to the target Solaris machine under the following directory, which must first be created:

/var/tmp/ocs

A. Prerequisite software

Software name	Version	Packages	Source
Sneep	2.7	SUNWsneep	sunsolve.sun.com
OpenSSL	0.9.8k	SMCssl	www.sunfreeware.com

The Sneep tool is used to store the machine serial number so it can be sent with every inventory. While not strictly required, it is still quite useful.

Downloading software from Sun requires a Sun account.

- In a web browser, open the Sun “Service Tools Bundle” URL:

<http://www.sun.com/service/stb/index.jsp>

- Select:

Download Now

If you do not have a Sun account

- To create a Sun account, select:

Register Now

- Fill in the form and select:

Submit.

- Once done with registration, go back to the STB download page.

End If

- Select or enter the following information:

Platform: Multi-platform.

Tick the “I agree to [...]” box.

Enter your Sun login and password.

- Select:

Log In and Continue

- Under "Required files", select:

install_stb.sh.tar.gz

The file is then downloaded.

B. Perl modules

Module name	Version	Package file name	Source
XML::NamespaceSupport*	1.09	XML-NamespaceSupport-1.09.tar.gz	search.cpan.org
XML::SAX *	0.96	XML-SAX-0.96.tar.gz	search.cpan.org
IO::Compress::Base *	2.015	IO-Compress-Base-2.015.tar.gz	search.cpan.org
Compress::Raw::Zlib *	2.015	Compress-Raw-Zlib-2.015.tar.gz	search.cpan.org
IO::Compress::Zlib *	2.015	IO-Compress-Zlib-2.015.tar.gz	search.cpan.org
Compress::Zlib *	2.015	Compress-Zlib-2.015.tar.gz	search.cpan.org
URI *	1.37	URI-1.37.tar.gz	search.cpan.org
HTML::Tagset *	3.20	HTML-Tagset-3.20.tar.gz	search.cpan.org
HTML::Parser *	3.60	HTML-Parser-3.60.tar.gz	search.cpan.org
PAR::Dist *	0.44	PAR-Dist-0.44.tar.gz	search.cpan.org
Archive::Zip *	1.26	Archive-Zip-1.26.tar.gz	search.cpan.org
GetOpt::ArgvFile *	1.11	Getopt-ArgvFile-1.11.tar.gz	search.cpan.org
AutoLoader *	5.68	AutoLoader-5.68.tar.gz	search.cpan.org
PAR *	0.988	PAR-0.988.tar.gz	search.cpan.org
Module::Build * ¹	0.2808	Module-Build-0.2808.tar.gz	search.cpan.org
version * ²	0.76	version-0.76.tar.gz	search.cpan.org
Module::ScanDeps *	0.89	Module-ScanDeps-0.89.tar.gz	search.cpan.org
XML::Parser *	2.36	XML-Parser-2.36.tar.gz	search.cpan.org
XML::Simple	2.18	XML-Simple-2.18.tar.gz	search.cpan.org
Net::IP	1.25	Net-IP-1.25.tar.gz	search.cpan.org
LWP	5.825	libwww-perl-5.825.tar.gz	search.cpan.org
Proc::Daemon	0.03	Proc-Daemon-0.03.tar.gz	search.cpan.org
Proc::PID::File	1.24	Proc-PID-File-1.24.tar.gz	search.cpan.org
PAR::Packer	0.982	PAR-Packer-0.982.tar.gz	search.cpan.org
Crypt::SSLeay	0.57	Crypt-SSLeay-0.57.tar.gz	search.cpan.org
Net::SSLeay	1.35	Net-SSLeay-1.35.tar.gz	search.cpan.org
XML::SAX::Expat	0.40	XML-SAX-Expat-0.40.tar.gz	search.cpan.org

Note: "*" after the name of a module denotes a prerequisite for another module.

C. Main software

Software name	Version	Package file name	Source
OCS Inventory NG agent	1.0.1	OCSNG_UNIX_AGENT-1.02.tar.gz	www.ocsinventory-ng.org

¹ More recent versions require a version of Perl more recent than the 5.8.4 which is bundled with Solaris 10.

² Undocumented dependency of the Module::ScanDeps module.

IV. Installing prerequisite software

A. Package installation

See III.A for the list of software to install.

1. Sneep

Sneep is part of the Service Tools Bundle so the STB installation script must be used:

- Unpack the STB installation script:

```
cd /var/tmp/ocs
gzip -dc install_stb.sh.tar.gz | tar xf -
rm install_stb.sh.tar.gz
chmod 755 install_stb.sh
```

- Launch the installer:

```
./install_stb.sh
```

- Select "m" to modify the list of packages.

- Answer "y" for Sneep, "n" for the other packages, except if you want to install other packages as well (e.g. Sun Explorer).

- Back to the menu, press Enter to install the package(s).

2. Sunfreeware packages

Binary packages from sunfreeware.com install in /usr/local.

Installation is straightforward:

```
cd /var/tmp/ocs
gunzip <package file>.gz
pkgadd -d <package file> all
```

Then answer "y" to all the questions, if any.

B. Post-installation configuration

The dynamic linker ld.so.1(1) must be informed of the new directories in which to look for dynamic libraries.

Here is what is needed here:

```
crle -u -l /usr/sfw/lib -l /usr/local/ssl/lib
crle -u -64 -l /usr/sfw/lib -l /usr/local/ssl/lib
```

Other library paths may be added for other software, e.g.:

/opt/sfw/lib and /usr/local/lib

V. Compiling and installing the required Perl modules

A. Preliminary configuration

The version of Perl which is bundled with Solaris 10 was compiled using Sun "cc" compiler, which is not available by default. Traditionally, the "gcc" compiler is used for compiling free software under Solaris.

Perl configuration needs to be modified so that Perl module compilation uses gcc with the proper command-line options.

For this, edit Perl main configuration file:

```
vi /usr/perl5/5.8.4/lib/sun4-solaris-64int/Config.pm
```

Locate the following lines:

```
[...]
cc='cc'
ccflags='-D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64 -xarch=v8 -D_TS_ERRNO'
[...]
cccdlflags='-KPIC'
[...]
ccflags_uselargefiles='-D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64'
ccname='cc'
[...]
ld='cc'
[...]
optimize='-xO3 -xspace -xildoff'
[...]
'cc' => 'cc',
'ccflags' => '-D_LARGEFILE_SOURCE -D_FILE_OFFSET_BITS=64 -xarch=v8 -
D_TS_ERRNO',
[...]
```

Replace their contents with the lines below:

```
cc='gcc'
ccflags=''
cccdlflags=''
ccflags_uselargefiles=''
ccname='gcc'
ld='gcc'
optimize=''

'cc' => 'gcc',
'ccflags' => '',
```

B. Generic steps

Apply the steps below to every Perl module listed in III.B, in the same order as they are listed in the software table:

- Unpack the module:

```
cd /var/tmp/ocs
```

```
gzip -dc <module>.tar.gz | tar xf -  
cd <module>
```

- See below: "Special cases" for possible specific steps to be applied to the current module.

- Build the Makefile then the software:

```
perl Makefile.PL  
make
```

- Test the module:

```
make test
```

- Install the module:

```
make install
```

- Remove the source files:

```
cd /var/tmp/ocs  
rm -rf <module directory>
```

C. Special cases

The special cases below apply during Perl module compilation and installation described above:

- Ignore the following messages while generating the Makefile with: "perl Makefile.PL":

```
WARNING: LICENSE is not a known parameter.  
[...]  
'LICENSE' is not a known MakeMaker parameter name.
```

- For XML::Parser:

When generating the Makefile, specify where Expat libraries and header files can be found:

```
perl Makefile.PL EXPATLIBPATH=/usr/sfw/lib EXPATINCPATH=/usr/sfw/include
```

- For Proc::PID::File:

The "verified: real" test failure is expected. Indeed, the "File.pm" module file specifies:

```
Please note that verification will only work for the operating systems
listed below and that the os will be auto-sensed.  See also DEPENDENCIES
section below.
```

```
Supported platforms: Linux, FreeBSD
```

- For Crypt::SSLeay and Net::SSLeay:

During the Makefile generation, a few questions are asked. For each question, press Enter so the default answer is used.

- For Crypt::SSLeay:

After installing the module, check that the Net::SSL sub-package was installed:

```
ls -l /usr/perl5/site_perl/5.8.4/sun4-solaris-64int/Net/SSL.pm
```

If not, recompile and reinstall the Crypt::SSLeay module.

VI. Compiling and installing the agent

- Unpack the OCS Inventory NG Agent archive:

```
cd /var/tmp/ocs
gzip -dc OCSNG_UNIX_AGENT-1.02.tar.gz | gtar xf -
cd ocsinventory-Agent-1.0.1
```

Note: as this archive contains specific GNU tar extensions, the "gtar" command must be used to unarchive it.

- Generate the Makefile and compile the agent:

```
perl Makefile.PL
make
```

- Install the agent:

```
make install
```

Several questions are asked before the installation. Use the table below to answer them.

Question	Answer
Do you want to configure the agent?	Enter (= accept).
Where do you want to write the configuration file?	1 (= /usr/local/etc/ocsinventory).
Do you want to create the directory /usr/local/etc/ocsinventory?	Enter (= accept).
What is the address of your ocs server?	Name or IP address of the OCS server.
Do you need credential for the server?	Enter (= no), except if login/password required by the OCS server.
Do you want to apply an administrative tag on this machine?	n (= decline).
Do you want to install the cron task in current user crontab?	n (= decline): it does not work.
Where do you want the agent to store its files?	Enter (= /var/opt/ocsinventory-agent).
Do you want to create the /var/opt/ocsinventory-agent directory?	Enter (= accept).
Should I remove the old linux_agent?	Enter (= decline): it does not work.
Do you want to use OCS-Inventory software deployment feature?	Enter (= accept).
Do you want to send an inventory of this machine?	n (= decline). Will be tested later.

VII. Compiling and installing the standalone binary

A. Purpose

Performing the whole OCS inventory NG Agent compilation and installation on each Solaris machine to inventory would be a long process. This is why a self-sufficient executable containing the OCS agent and all required Perl modules can be generated and then deployed to all the Solaris machines.

B. Building

Here is how to generate the OCS agent standalone binary on the compilation machine.

- Launch the generation script:

```
cd /var/tmp/ocs/Ocsinventory-Agent-1.0.1/tools  
./standalone.sh
```

Note: the operation takes up to several minutes because all the required Perl modules must be included in the binary.

The standalone binary file is:

```
/var/tmp/ocs/Ocsinventory-Agent-1.0.1/ocsinventory-agent.bin
```

C. Installation

- Create the target directory if it does not exist:

```
mkdir -p /usr/local/bin
```

- Copy the standalone agent:

```
cp -p /var/tmp/ocs/Ocsinventory-Agent-1.0.1/ocsinventory-agent.bin  
/usr/local/bin
```

VIII. Post-installation

A. Configuration

1. Sneep configuration

- Enter the machine serial number in Sneep:

```
/opt/SUNWsneep/bin/sneep -s <machine serial number>
```

Note:

- Modern hardware (e.g. Sparc hardware from 2007 on) allows Sneep to automatically retrieve the serial number from the machine.
- Once the serial number has been registered, it is stored in NVRAM so it persists across Solaris re-installations, except if the NVRAM is cleared.

So if the hardware supports serial number retrieval or if the serial number has already been entered into the NVRAM with Sneep, the above command is not required.

2. OCS agent configuration file

- Edit OCS Inventory NG agent configuration file:

```
vi /usr/local/etc/ocsinventory/ocsinventory-agent.cfg
```

- If required, change or add parameters in this file.

Common modifications include:

1. OCS server name modification

```
server=http://<new OCS server>/ocsinventory
```

2. Using the HTTPS protocol for secure communication

```
server=https://<OCS server>/ocsinventory
```

3. Specifying a tag for the client machine

```
tag="<client tag>"
```

Tags are user-specified values used to organize machines into custom categories.

Etc.

3. Specific configuration

If using HTTPS, extra steps must be taken, such as installing SSL/TLS certificates. Also, credentials may be required by the OCS server. This is beyond the scope of this document.

B. First time launch: standard agent

- Launch the OCS Inventory NG agent:

```
/usr/per15/bin/ocsinventory-agent --force
```

Ignore any errors about undetected hardware, e.g. inability to detect the memory configuration. This is frequent on Solaris because there is not enough input to include all possible hardware.

If other errors are displayed:

- Re-launch the command with the extra "--debug" switch and check the output.
- Often, the error is in the data transmission to the OCS server.

In this case, the "--debug" switch does not help. However, there is generally an HTTP error code which can help in troubleshooting. Also, check the server logs and configuration.

If necessary, use the Solaris snoop(1M) command to capture the network traffic between the client and the server.

If the agent issues an HTTP "403 Forbidden" error, chances are the server does not accept unsolicited inventories. Check the server configuration, in particular the "INVENTORY_SESSION_ONLY" and "FREQUENCY" parameters.

Also, the server may accept only HTTPS and not HTTP.

C. First time launch: standalone agent

The standalone agent should be tested on the compilation machine before deploying it to the other Solaris machines.

As this agent does not take the configuration file into account, all the options must be specified on the command line.

- Launch the inventory using the standalone agent:

```
/usr/local/bin/ocsinventory-agent.bin --force --server http://<OCS server>/ocsinventory [--tag="<tag>"] [other options]...
```

with: <OCS server> the name or IP address of the OCS Inventory NG server.

If using HTTPS, replace:

```
http://<OCS server>/ocsinventory
```

with:

```
https://<OCS server>/ocsinventory
```

The following error message is displayed:

```
could not find ParserDetails.ini in /var/tmp/par-root/cache-651e7db1fc5d028bf5449be601a1d499/SAX
```

It is due to the way the PAR::Packer Perl module packages the XML::SAX module. This is a known bug tracked here:

<http://rt.cpan.org/Public/Bug/Display.html?id=38067>

As of this writing, the bug is not corrected. It does not seem to affect the agent operation, though, so please ignore the error message.

D. Scheduling

Here is how to perform and send the inventory automatically on a regular basis.

- Edit the super-user crontab:

```
crontab -e
```

- At the end of the file, add lines to schedule the OCS agent launch:

```
# Perform and send OCS inventory.  
38 4 * * 1-5 /usr/perl5/bin/ocsinventory-agent --lazy > /dev/null 2>&1
```

In this example, the inventory is performed every workday at 4:38 a.m.

The "--lazy" switch tells the agent to send the inventory only if none was sent recently, typically on the same day.

- Save the crontab file.

E. Cleaning up

- Keep a copy of all the prerequisite software listed in III.A. This software will be used for deployment to other machines.

- Remove the directory used for package storage and compilation:

```
rm -rf /var/tmp/ocs
```

IX. Deployment

Once the OCS Inventory NG agent has been compiled and installed on a first Solaris machine, only a subset needs to be copied to the other Solaris machines on the network.

A. Prerequisite software

Ideally, all the prerequisite software listed in III.A should be installed on all the machines where the OCS agent is to be deployed.

However:

- OpenSSL is only required if HTTPS is used to communicate with the OCS server.
- If Sneep is not installed, only the serial number will be missing from the inventory.

Based on these considerations, install all the desired prerequisite software to all the Solaris machines which are to be inventoried.

If HTTPS is used, also copy the necessary SSL/TLS certificates.

B. Standalone OCS agent

The most straightforward way to deploy the standalone agent is by direct copy from the compilation machine.

For example, from the compilation machine, copy the standalone agent with:

```
scp -p /usr/local/bin/ocsinventory-agent.bin <target machine>:/usr/local/bin
```

This example assumes that the target machine allows ssh access and has a "/usr/local/bin" directory.

C. Serial number registration

On each Solaris machines to be inventoried, if Sneep was installed, record the serial number in Sneep as in VIII.A.1.

D. Manual inventory or scheduling

If no inventory scheduling is desired, perform a manual inventory as in VIII.C.

If, on the contrary, periodic inventories are desired, add the following lines to the super-user crontab:

```
# Perform and send OCS inventory.
38 4 * * 1-5 /usr/local/bin/ocsinventory-agent.bin --lazy --server
http://<OCS server>/ocsinventory [--tag="<tag>"] [other options]... >
/dev/null 2>&1
```

As seen in VIII.C, replace "http" with "https" if HTTPS is required by the server. Also customize the crontab entry if necessary.