

Server Monitoring

Contents

Ganglia	1
Nagios	3
Open Computer and Software Inventory Next Generation	8

Ganglia

Ganglia is a web based program developed by Berkeley University designed to monitor machine in a cluster or grid. It uses carefully engineered data structures and algorithms to achieve very low per-node overheads and high concurrency. The implementation is robust, has been ported to an extensive set of operating systems and processor architectures, and is currently in use on over 500 clusters around the world. It has been used to link clusters across university campuses and around the world and can scale to handle clusters with 2000 nodes.

The ganglia system contains:

Ganglia Monitoring Daemon (gmond)

Gmond is a multi-threaded daemon which runs on each cluster node you want to monitor.

Ganglia Meta Daemon (gmetad)

At each node in the tree, a Ganglia Meta Daemon (*gmetad*) periodically polls a collection of child data sources, parses the collected XML, saves all numeric, volatile metrics to round-robin databases and exports the aggregated XML over a TCP sockets to clients.

Ganglia PHP Web Frontend

The Ganglia web frontend provides a view of the gathered information via real-time dynamic web pages.

Installation on management node

Install the following packages on the management node

http://sourceforge.net/project/showfiles.php?group_id=43021&package_id=35280

```
rpm -ivh ganglia-gmond-3.0.3-1.fc4.i386
rpm -ivh ganglia-gmetad-3.0.3-1.fc4.i386
rpm -ivh ganglia-web-3.0.3-1.noarch
rpm -ivh ganglia-devel-3.0.3-1.fc4.i386
```

```
chkconfig --level 345 gmond on
chkconfig --level 345 gmetad on
```

Install graphical view

```
rpm -ivh rrdtool-1.0.41-1.8.0.i386
```

You may need some perl extentions with this program

```
rpm -ivh perl-Crypt-DES-2.03-2.1.fc3.rf.i386
rpm -ivh perl-Net-SNMP-4.1.2-0.1.fc3.rf.noarch
```

Installation on other nodes

for all other nodes

```
rpm -ivh ganglia-gmond-3.0.3-1.fc4.i386
```

Ganglia Configuration – for cluster or grid

<http://xcat.org/doc/ganglia-HOWTO.html#Ganglia%20Management%20Node%20Installation>

Now that Ganglia is setup on the management node add the rest of the nodes.

IANS, for installed nodes install the gmond RPM, copy the management node `/etc/gmond.conf` to the node `/etc`, configure gmond to start on boot, and restart the gmond service.

To setup for unattended installation for diskful nodes:

1. Create a gmond xCAT group: `addattr noderange gmond`
2. Create a gmond sync directory: `mkdir /install/post/sync/gmond`
3. Copy `/etc/gmond.conf` to the sync directory:

```
cd /install/post/sync/gmond
mkdir etc
chmod 755 etc
cp /etc/gmond.conf etc/
chmod 644 etc/gmond.conf
```

4. Copy the gmond RPM into the `otherrpms` directory:

```
mkdir -p /install/post/otherrpms/${osver}/${uname -m}/
cp /usr/src/*/RPMS/*/ganglia-gmond-*.rpm
/install/post/otherrpms/${osver}/${uname -m}/
```

To setup for wareCAT (stateless) nodes:

1. Export `WWIMAGE` (see wareCAT 2.4 HOWTO), e.g.:

```
export WWIMAGE=compute_x86_64
```

2. Install RPM:

```
rpm --root /vnfs/$WWIMAGE -ivh /usr/src/*/RPMS/*/ganglia-gmond-*.rpm
```

3. Setup service to start on boot:

```
chroot /vnfs/$WWIMAGE /sbin/chkconfig --level 345 gmond on
```

4. Update excludes to include support for Ganglia. Edit `/etc/warewulf/vnfs/excludes` and `/etc/warewulf/vnfs/excludes-aggressive`:

Insert above: `/var/*/*`

```
+ var/lib/ganglia
+ var/lib/ganglia/rrds
```

5. Rebuild VNFS. Read the wareCAT 2.4 HOWTO.
6. Reboot nodes.

Nagios

Nagios is a web based server and network monitoring program. It shows the server load balance, and network usage, it also shows details about network programs running on the server, such as SMTP, POP3 and HTTP. There a graphical charts and log reports which can be easily generated. It also has an effective notification system which can send system administrators logs via email or sms.

Using RPM

```
yum -y install nagios nagios-plugins-all nagios-nrpe
```

With the rpm install the nagios configuration files for services hosts and commands are all in one file. It would be a good idea to split these configurations into different cfg files.

Change directory to /etc/nagios

In nagios.cfg enable all the following

```
cfg_file=/etc/nagios/contactgroups.cfg
cfg_file=/etc/nagios/contacts.cfg
cfg_file=/etc/nagios/dependencies.cfg
cfg_file=/etc/nagios/escalations.cfg
cfg_file=/etc/nagios/hostgroups.cfg
cfg_file=/etc/nagios/hosts.cfg
cfg_file=/etc/nagios/services.cfg
cfg_file=/etc/nagios/timeperiods.cfg
```

Open localhost.cfg and move all configurations parts into the files above.

Start nagios: /etc/init.d/nagios start

Note: nagios may need to be added as a service

```
chkconfig --add nagios
chkconfig nagios on
```

Copy these configuration settings for Nagios httpd config file for accessing the system over http (located in: /etc/httpd/conf.d/nagios.conf)

```
ScriptAlias /nagios/cgi-bin/ /usr/lib/nagios/cgi-bin/
<Directory /usr/lib/nagios/cgi-bin/>
Options ExecCGI
order deny,allow
deny from all
allow from 127.0.0.1
allow from all
AuthType Basic
AuthName "nagios"
AuthUserFile /etc/nagios/passwd
require valid-user
</Directory>
```

```
Alias /nagios/ /usr/share/nagios/html/
<Directory /usr/share/nagios/html/>
Options None
order deny,allow
deny from all
allow from 127.0.0.1
allow from all
AuthType Basic
AuthUserFile /etc/nagios/passwd
AuthName "nagios"
```

```
require valid-user
</Directory>
```

Create a htpasswd file to restrict access to Nagios on http

```
htpasswd -cmd etc/nagios/passwd nagios
```

Access the web link to run Nagios

Review settings in /etc/Nagios/cgi.cfg

<http://localhost/nagios/>

NRPE on Clients

NRPE allows the Nagios master to monitor other machines on the network. This needs to be installed and configured on the Master too.

```
yum -y install nagios-nrpe
```

Modify the /etc/nagios/nrpe.cfg so it includes the following:

```
server_port=5666
allowed_hosts=127.0.0.1 #(for clients added the master nagios server ip)
dont_blame_nrpe=1

command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20

command[check_disk_boot]=/usr/lib/nagios/plugins/check_disk -w 20 -c 10 -p /boot
command[check_disk_root]=/usr/lib/nagios/plugins/check_disk -w 20 -c 10 -p /
command[check_disk_home]=/usr/lib/nagios/plugins/check_disk -w 20 -c 10 -p /home

command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c 200
```

Make the same changes when installing nrpe on client machines.

Add nrpe as service and start it.

```
chkconfig --add nrpe
chkconfig nrpe on
```

Allow the same setups on the each of the Linux client machines.

To check to see the results

```
check_nrpe -H nausicaa -c check_disk_usr
```

Modifying configuration files on Master to allow checks on remote machines

hosts.cfg – add new host

```
define host{
    use                linux-server
    host_name          nausicaa
    address            nausicaa.acorp.local
    check_command      check-host-alive
    # Your defined contact group name
}
```

Hostgroups.cfg – add host to a group

```
define hostgroup{
  hostgroup_name  acorp
  alias           Acorp Servers
  members        localhost,nausicaa,windowssys,nagiossrv
}
```

commands.cfg – add the new nrpe check

```
define command {
  command_name check_remoteusers
  command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c check_users
}

define command {
  command_name check_remoteprocs
  command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c check_total_procs
}

define command {
  command_name check_remotealive
  command_line $USER1$/check_ping -H $HOSTADDRESS$ -w 3000.0,80% -c
5000.0
,100% -p1
}

define command {
  command_name check_remotedisk_root
  command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c check_disk_root
}
```

services.cfg – add it as a service for new host

```
# Nausicaa
define service{
  use                local-service
  host_name          nausicaa
  service_description Current Load
  check_command      check_remoteload
}
```

Restart nagios.

NRPE on Windows machines

NRPE for windows can be obtained from: www.anandhacorp.co.uk/win-nagios.zip

Note: This version has been customised for the existing setup therefore no changes are required for the configuration file. (unless the server running the Nagios Master fails)

Unzip **win-nagios.zip** in the application directory of the windows machine (usually C: drive)

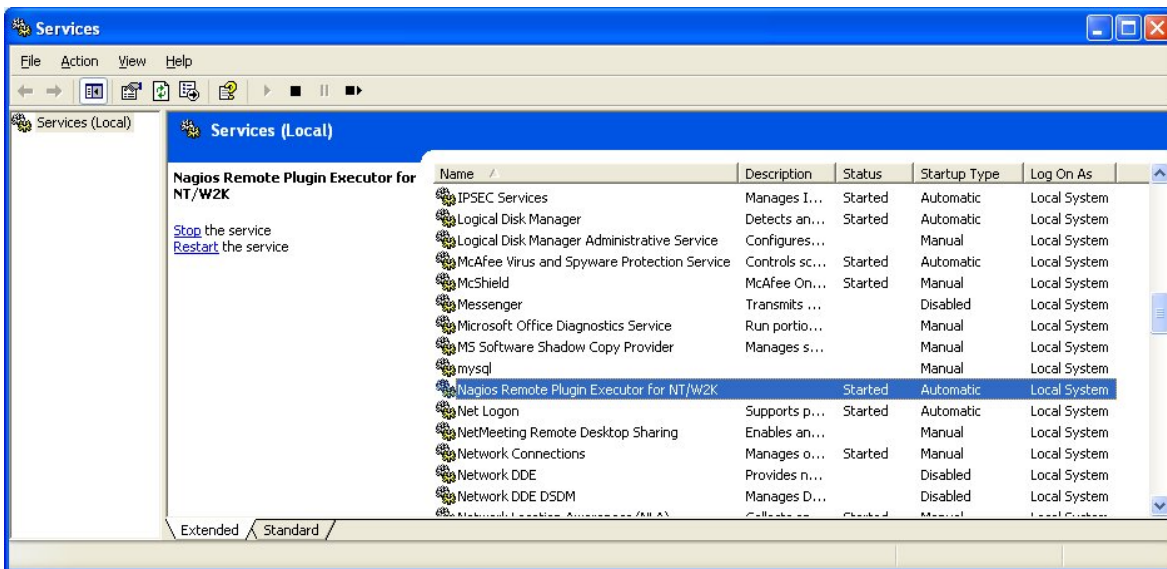
Open command line and navigate to this directory

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\aan>cd \
C:\>cd nagios
C:\nagios>_
```

cd into **bin** and run **nrpe_nt**
Install nrpe as a service by typing: `nrpe_nt -i`

This installs the SSL version which is required to communicate with the Nagios Master.



The plugins available in `c:/nagios/bin/` only work on the local system.
To test the plugins: copy the following into command line

```
C:\nagios\bin\diskspace_nrpe_nt.exe c: 70 90
```

This will return the current disk space on the local machine

For the nagios master to communicate with a windows client Windows Script Files are required. A Windows script (*.wsf) file is a text document containing Extensible Markup Language (XML) code. These scripts can be executed using `cscript.exe`.

```
c:\windows\system32\cscript.exe //NoLogo C:\nagios\bin\check_all_services.wsf
```

Add this as a service on the `nrpe.cfg` file on the local machine (located in `c:\nrpe\bin\nrpe.cfg`)

```
command[nt_servicecheck]=c:\windows\system32\cscript.exe //NoLogo  
C:\nagios\bin\check_all_services.wsf
```

Save nrpe.cfg and restart nrpe on the local machine

Log on to the Linux server and type the following command to see if the windows script works

```
/usr/lib/nagios/plugins/check_nrpe -H windowsys -c nt_servicecheck
```

It should display

```
SERVICES OK: 57 Automatic Services Running
```

NRPE on UNIX

Prerequisites:

- RS506A + OSS646C supplements
- Install Glib and Ncurses
- Install GWX Libraries
- Install GNU Development Tools

Download Unix version of NRPE from here: www.anandhacorp.co.uk/nagios-unix.zip

Unpack the content in /usr/local/nagios and create a directory and symbolic link for the check plugins

```
mkdir /usr/lib/nagios  
ln -s /usr/local/nagios/libexec /usr/lib/nagios/plugins
```

```
/etc/init.d/nrpe start
```

```
Run ps -ef | grep nrpe
```

There should be a process running.

Run a test from the Nagios server to ensure the server can be seen.

```
Check_nrpe -H UNIXSERVER -c check_load
```

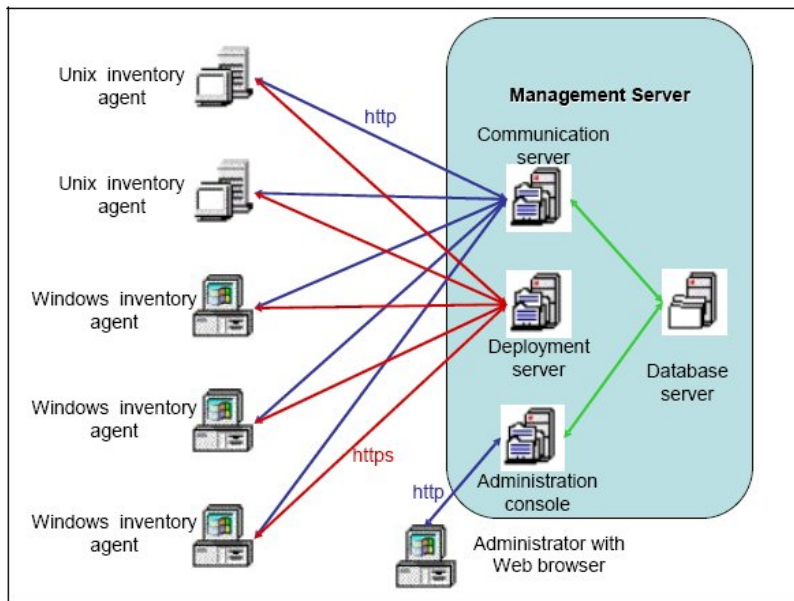
Open Computer and Software Inventory Next Generation

Open Computer and Software Inventory Next Generation is an application designed to help a network or system administrator keep track of the computer configuration and software installed on the network. It also allows deploying packages on Windows and Linux computers.

Management server contains 4 main components:

1. **Database server**, which store inventory information
2. **Communication server**, which will handle HTTP communications between database server and agents.
3. **Administration console**, which will allow administrators to query the database server through their favorite browser.
4. **Deployment server**, which store all package deployment configuration (require HTTPS!)

These 4 components can be hosted on a single computer or on different computers to allow load balancing. For more than 10000 inventoried computers, it is better to use at least 2 different servers, one for the database server + Communication server and the other for a database replica + Administration server + Deployment server.



Server Install Requirements

Apache version 1.3.33 or higher / Apache version 2.0.46 or higher.

Mod_perl version 1.29 or higher.

Mod_php version 4.3.2 or higher.

PHP 4.3.2 or higher, with ZIP and GD support enabled.

PERL 5.6 or higher.

Perl module XML::Simple version 2.12 or higher.

Perl module Compress::Zlib version 1.33 or higher.

Perl module DBI version 1.40 or higher.

Perl module DBD::Mysql version 2.9004 or higher.

Perl module Apache::DBI version 0.93 or higher.

Perl module Net::IP version 1.21 or higher.

Perl module SOAP::Lite version 0.66 or higher (not mandatory) MySQL version 4.1.0 or higher with InnoDB engine active.

Make utility like GNU make.

Server Installation

Use yum to install the perl packages.

yum install perl-XML-Simple

```
yum install perl-Compress-Zlib
yum install perl-DBI
yum install perl-DBD-MySQL
yum install perl-Apache-DBI
yum install perl-Net-IP
yum install perl-SOAP-Lite
```

Install addition php package for ZIP and GD support

```
yum install php-pecl-zip
yum install php-gd
```

Main Package

Download "OCSNG_LINUX_SERVER_1.01.tar.gz" from OCS Inventory Web Site.

Unpack it.

```
tar -xvzf OCSNG_LINUX_SERVER_1.01.tar.gz
cd OCSNG_LINUX_SERVER_1.01
```

Run "setup.sh" installer. During the installer, default choice is presented between []. For example, [y]/n means that "y" (yes) is the default choice, and "n" (no) is the other choice.

```
sh setup.sh
```

Run through the options, keeping everthing default

Now, you can restart Apache web server for changes to take effect.

```
/etc/init.d/httpd restart or /etc/init.d/apache restart
```

Open your favorite web browser and point it on URL

"http://administration_console/ocsreports" to connect the Administration server.

As database is not yet created, this will begin OCS Inventory setup process. Otherwise, you can rerun configuration process by browsing

http://administration_console/ocsreports/install.php URL (this must be used when upgrading OCS Inventory management server).

Fill in information to connect to MySQL database server with a user who has the ability to create database, tables, indexes, etc (usually root):

MySQL user name

MySQL user password

MySQL hostname

Agent Install

Download and unzip OCSNG_WIN32_AGENT_1.01.zip. This package contains 3 files:

OcsAgentSetup.exe, agent installer with Windows service included. We recommend using this package.

OcsAgent.exe, to install standalone agent on a non network connected computer to allow running the inventory manually with /LOCAL command line switch (or if you do not want to use service).

OcsLogon.exe, launcher of OCS Inventory NG agent to use when deploying agent through a login script or Active Directory GPO in the domain. If agent is already installed, it just runs the agent. Otherwise, it downloads agent's binaries from Communication server, setup it and launch it.

When OCS Inventory NG Agent "OCSInventory.exe" is launched, it contacts Communication server using HTTP protocol to ask what is has to do. Server can answer "nothing" (not time

for an inventory and no package to deploy), and so agent stops. When agent is launched, it will generate and send an inventory only. Otherwise, server may answer that Agent has to:

Send an inventory: Agent retrieve all computer properties and send them using HTTP protocol to server. Server answer this only if last inventory date in the database is older than general option "FREQUENCY", specified in days (see § 6.2 Managing OCS Inventory NG general options.)

Discover the network: Agent retrieve all computer properties, scan his sub network for active devices listening on the network, and send these informations using HTTP protocol to server. Server answer this only if computer is elected to run IPDISCOVERY (see § 7 Using IP discovery feature.)

Deploy a package: Agent contact deployment server using HTTPS protocol to get information file, download package fragments from repository, rebuild package and launch it.

Each time an inventory is done, Agent writes a configuration file "OCSInventory.dat" in his agent folder where it will put configuration options downloaded from the Communication server.

Deploying Agent using launcher OcsLogon.exe through Login Script

Launcher "OcsLogon.exe" is a little tool able to run inside a login script or an Active Directory GPO. His goal his to launch OCS Inventory NG Agent on computer, and if Agent is not installed, to setup Agent on computer. Launcher "OcsLogon.exe" will try to connect by default to the Communication Server using a DNS name "ocsinventory-ng", like if you open your favorite web browser and enter the URL: <http://ocsinventory-ng/ocsinventory>.

To use a different URL if you cannot add this DNS name, just rename "OcsLogon.exe" with the DNS name or IP address of the Communication Server (for example "ocsinventory.domain.tld.exe" if you've created for your server a DNS record "ocsinventory.domain.tld" or "192.168.1.2.exe" if your server has 192.168.1.2 as IP address). Launcher then will try to connect to the DNS name or IP address you've named it (<http://ocsinventory.domain.tld/ocsinventory> or <http://192.168.1.2/ocsinventory>).

Launcher will first check if OCS Inventory NG agent is installed, and if not, will contact Communication Server in HTTP to download latest agent binaries and setup locally on the computer:

Standalone Agent in the folder "C:\ocs-ng" by default or, if the locally connected user do not have permission to create folder in the root directory, in the folder "ocsng" in the user's temporary directory. If Standalone agent is already installed, launcher will just run the agent.

Service Agent in folder "C:\Program Files\OCS inventory Agent" by default.

Copy files "OcsLogon.exe" (or the renamed one) to a shared folder somewhere in your network. This folder must readable by all your users. Then add a call to "OcsLogon.exe" (or to the renamed one) in the login script of your users.

Here is the current login script:

```
@echo off
net use z: \\winsrv01\ocs /y
"z:\10.91.1.17.exe" /debug /np /install
"c:\Program Files\OCS Inventory Agent\OCSInventory.exe" /debug /server:10.91.1.17
net use z: /d /y
exit
```

here is a sample version:

```
@echo of
echo Running system inventory, please wait...
REM Call to OCS Inventory NG agent for deployment
REM Using shared folder MY_SHARE on server MY_SERVER
REM Connect to Communication server at address 192.168.1.2
REM Enable debug log with /DEBUG to create OcsLogon.log and computer_name.log
REM Force setup agent version 4030 if agent not up to date
REM Deploy service version of agent using /INSTALL
"\\MY_SERVER\MY_SHARE\192.168.1.2.exe" /DEBUG /NP /INSTALL /DEPLOY:4030
echo Done. Thanks a lot.
```

Put this script named "ocs.bat" for example on your Domain Controller in the folder "%WINDIR%\SYSVOL\Domain\Scripts", where "%WINDIR%" is generally "C:\WINNT" or "C:\Windows". Next, you have to link login script with every users registered in your Active Directory domain. You can do this using "Active Directory users and computers" tool

OCS Inventory NG is able to automatically install agent on computers when launcher "OcsLogon.exe" is used through login script or GPO. Agent's files are downloaded from the Communication server. You just have to upload the agent package into the Administration console and to activate the deployment feature by setting "DEPLOY" general option to ON (see § 6.2 Managing OCS Inventory NG general options.). Uploaded file must be one of the following:

"ocsagent.exe" file for Windows agent, to deploy Agent without Windows service. This file is included in package OCSNG_WIN32_AGENT_XX.zip.

"ocspackage.exe" file, created using OCS Inventory NG Packager, to deploy Windows service version of Agent, even if user connected does not have Administrator privileges.

Deploy Package

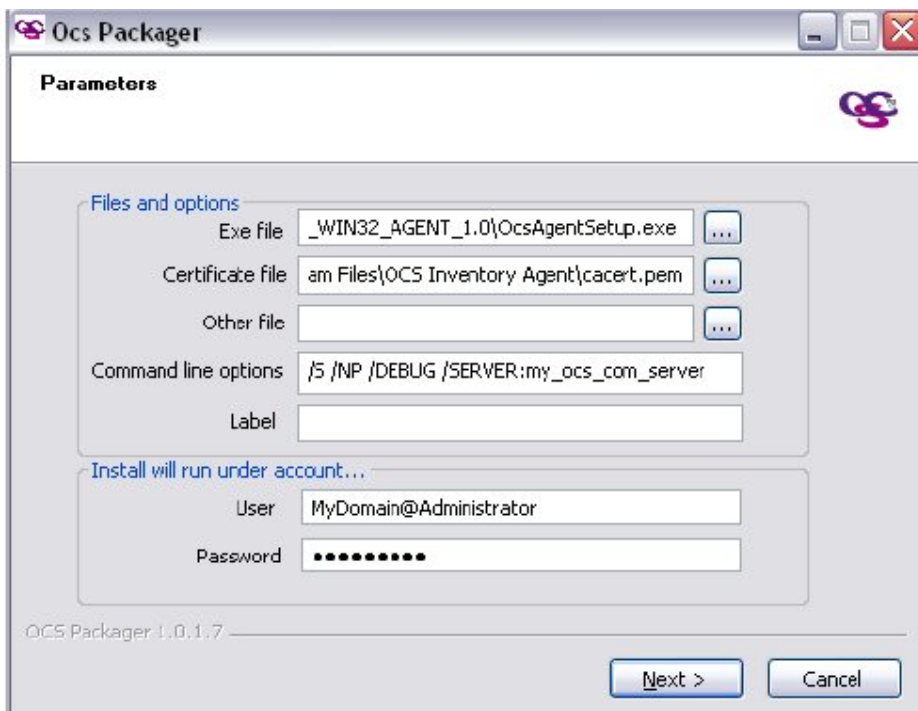
To create "ocspackage.exe" file, just run OCS Inventory NG Packager, and fill in following informations:

Path to file "OcsAgentSetup.exe", installer of OCS Inventory NG service Agent, included in package OCSNG_WIN32_AGENT_XX.zip.

Path to the Certificate file to use, for checking server certificate when using package deployment feature. Optionally, another file to include in setup.

Command line parameters for running "OcsAgentSetup.exe" service installer, at least "/S" to run installer in silent mode, and "/SERVER:my_ocs_com_server_address" to specify "my_ocs_com_server" as address of OCS Inventory NG Communication Server.

Username (domain@account for an NT or Active Directory account) and password of an Administrator account on client computers. "OcsAgentSetup.exe" will be run under this account on client computers, to allow installing service even if user connected does not have Administrator privileges.



Click “Agent” toolbar menu, browse your hard drive to select agent file and click “send” button.

