

# **FlexNet Inventory Agent and Managed Devices**

# Contents

<b>Preface: FlexNet Inventory Agent and Managed Devices.....</b>	<b>vi</b>
<b>Chapter 1: Overview.....</b>	<b>7</b>
Introducing the FlexNet inventory agent.....	8
What Is Special on UNIX-Like Systems?.....	9
<b>Chapter 2: Deploying the FlexNet Inventory Agent.....</b>	<b>15</b>
Automated Adoption Summary.....	16
Self-Managed Deployment: Collecting the Software.....	17
Configuring Installation on Microsoft Windows.....	19
Edit the Configuration File for Microsoft Windows.....	19
Customizing the Windows Installer Command Line.....	21
Deployment Notes for Windows Platforms.....	22
Uninstalling from Windows Platforms.....	23
Configuring Installations on UNIX-like Platforms.....	23
Configure the Bootstrap File for UNIX.....	23
Installing FlexNet inventory agent on UNIX.....	26
Updating config.ini on a UNIX Device.....	29
Uninstalling FlexNet inventory agent from UNIX.....	30
Protecting Your Customizations.....	31
Checking the Installed Version.....	32
<b>Chapter 3: Command Lines.....</b>	<b>34</b>
mgspolicy Command Line.....	35
ndlaunch Command Line.....	37
ndschedag Command Line.....	43
ndtrack Command Line.....	44
ndupload Command Line.....	47
<b>Chapter 4: Preferences.....</b>	<b>50</b>
[Registry] Explained.....	54
AllowedPkgTypes.....	54
AutoPriority.....	55
Catchup.....	56
CheckCertificateRevocation.....	57
CheckServerCertificate.....	58
CommonAppDataFolder.....	59
DateTimeFormat.....	60
DefaultSchedulePath.....	61
Directory.....	61
Disabled (application usage agent).....	63
Disabled (schedule agent).....	63
DisablePeriod.....	65
Domain.....	65

DownloadSettings.....	66
ExcludeDirectory.....	68
ExcludeExtension.....	69
ExcludeFile.....	70
ExcludeMD5.....	71
GenerateMD5.....	72
Hardware.....	73
Host.....	73
http_proxy.....	75
https_proxy.....	75
IncludeDirectory.....	76
IncludeExecutables.....	77
IncludeExtension.....	78
IncludeFile.....	79
IncludeMachineInventory.....	80
IncludeMD5.....	81
IncludeRegistryKey.....	82
IncludeUserInventory.....	83
InstallDefaultSchedule.....	84
InstallProfile.....	85
InventoryFile.....	86
InventoryScriptsDir.....	86
InventoryType.....	87
LauncherCommandLine.....	88
LogFile (installation agent).....	89
LogFile (inventory agent).....	90
LogFile (policy agent).....	90
LogFileOld (installation agent).....	91
LogFileOld (policy agent).....	92
LogFileSize (installation agent).....	93
LogFileSize (policy agent).....	94
LogLevel (inventory agent).....	94
LogLevel (policy agent).....	96
LogModules (inventory agent).....	97
LogModules (policy agent).....	98
LowProfile (inventory agent).....	99
MachineID.....	99
MachineInventoryDirectory.....	101
MachineName.....	102
MachinePolicy.....	102
MachinePolicyDirectory.....	103
MachinePolicyPackageDirectory.....	104
MachineScheduleDirectory.....	105
MachineZeroTouchDirectory.....	106
MSI.....	107
Name.....	107
ndsensNetType.....	109
no_proxy.....	110
OnConnect.....	110
OracleInventoryUser.....	111
Password.....	112
PerformOracleInventory.....	113
PolicyServerURL.....	113
Port.....	114
Priority.....	115
ProgramFiles, ProgramFilesX86Folder, ProgramFilesX64Folder.....	116
Protocol.....	117

Proxy.....	118
Recurse.....	119
RunInventoryScripts.....	120
SelectorAlgorithm.....	121
ScheduleType.....	122
ShowIcon (installation agent).....	123
ShowIcon (inventory agent).....	124
SourceFile.....	125
SourceRemove.....	126
Startup.....	127
StrictInstall.....	128
SysDirectory.....	129
UIMode.....	129
Upload.....	130
UploadLocation.....	131
UploadPassword.....	132
UploadProxy.....	133
UploadRule.....	134
UploadSettings.....	135
UploadType.....	136
UploadUser.....	137
User.....	138
UserHardware.....	139
UserInteractionLevel (installation agent).....	140
UserInteractionLevel (inventory agent).....	141
UserInventoryDirectory.....	142
UserScheduleDirectory.....	143
UserZeroTouchDirectory.....	144
VersionInfo.....	145
WinDirectory.....	146
WMI.....	147
WMIconfigFile.....	147

# Legal Information

**Document Name:** FlexNet Inventory Agent and Managed Devices

**Part Number:** FMS-11.5.0-FA01

**Product Release Date:** 18 May 2016

## Copyright Notice

Copyright © 2016 Flexera Software LLC. All Rights Reserved.

This publication contains proprietary and confidential technology, information and creative works owned by Flexera Software LLC and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera Software LLC is strictly prohibited. Except where expressly provided by Flexera Software LLC in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera Software LLC intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software LLC, must display this notice of copyright and ownership in full.

FlexNet Manager Suite incorporates software developed by others and redistributed according to license agreements. Copyright notices and licenses for this externally-developed software are provided in the link below.

## Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see <http://www.flexerasoftware.com/intellectual-property>. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

## Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.

---

## Preface

---

# FlexNet Inventory Agent and Managed Devices

A managed device is a computer device that has the FlexNet inventory agent installed on it. This name, FlexNet inventory agent, refers in fact to a collection of code objects that can gather software and hardware inventory from the managed device, and optionally also collect data about application usage on the same device. This collected data is uploaded through an inventory beacon to the central application server, where it is imported to the operations databases and finally appears in the web interface of FlexNet Manager Suite.

One of the design criteria for the system is that it should be as automated as possible. For that reason, the management of inventory gathering is (in general) through discovery and inventory rules defined centrally in the web interface. These rules allow (amongst other things) for the automatic installation of the FlexNet inventory agent on targeted computer devices. The processes for working within this framework are documented in the online help.

However, in some environments, administrators prefer not to use this automation. Sometimes this is a desire for greater customization than is possible through the discovery and inventory rules in the web interface. At other times, it is a desire to continue to use existing infrastructure for software deployment.

This document provides sufficient information for experienced administrators to configure deployment packages for use in existing, third-party deployment tools. It also provides some insight into customization methods, and some of the preferences available for controlling the agents installed on the managed device.

---

# 1

---

## Overview

### Topics:

- *Introducing the FlexNet inventory agent*
- *What Is Special on UNIX-Like Systems?*

The FlexNet inventory agent is available on Windows and non-Windows (UNIX-like) operating systems. This chapter provides a quick introduction to both formats, taking the Windows version as the foundation, and then adding additional insights about the UNIX-like version.

# Introducing the FlexNet inventory agent

As you work to prepare your own deployment of the FlexNet inventory agent, it is helpful to have a clear understanding of what you are working with. This section starts with the Windows version, covering functionality that is common to all versions.

## What is the FlexNet inventory agent?

This is a collection of executable files or 'agents', typically installed with some additional configuration files and libraries. Its purposes are

- To take inventory of both the hardware and software on a computing device, and return an XML document (.ndi) describing this inventory
- To optionally track usage of applications on the same device, based on watching specified files that form part of the application.

The agents are implemented in C++ across multiple platforms, for all of which they are built from a common code base. Therefore, most features found on one platform are commonly available on all platforms (where applicable and possible).

## What is installed as part of the FlexNet inventory agent?

The installed FlexNet inventory agent includes a number of components that allow it certain levels of autonomy, self-management, self-upgrade, and software usage tracking. Examples include:

- The core agent for inventory collection, `ndtrack`
- The upload agent (`ndupload`) to transfer the inventory results to an inventory beacon
- The schedule agent (`ndschedag`) to coordinate execution of the other agents
- The policy agent (`mgspolicy`), responsible for managing the various rules that control the overall FlexNet inventory agent
- The installation agent (`ndlaunch`) which downloads policy, schedule, self-update and other packages required for operation.

## Is there an alternative to installing FlexNet inventory agent on a computing device?

Given that you need the functionality available through FlexNet inventory, the options are:

- Installing the FlexNet inventory agent locally on a computing device, as discussed in this document.
- Not installing locally on the computing device, but using remote execution, also known as zero-touch inventory (see below). Remote execution is normally managed automatically through the inventory beacons, or you can also manage it independently if need be. This uses the same core executable, `ndtrack`, with a reduced set of agents and libraries (for example, it does not automatically upload inventory results, and does not track application usage). In this format, it is called the FlexNet Inventory Scanner. For more information about this format, see the chapter *FlexNet Inventory Scanner* in the *FlexNet Manager Suite System Reference* PDF file.



Zero-touch inventory means that:

- In the case of Microsoft Windows, a copy of the FlexNet Inventory Scanner is run by a target computing device from a remote network share, normally `\MgsRET$\Inventory` on an inventory beacon. (The default location for this network share on the inventory beacon is `C:\Program Files (x86)\Flexera Software\Inventory Beacon\RemoteExecution\Public` on 64-bit systems, and `C:\Program Files\Flexera Software\Inventory Beacon\RemoteExecution\Public` on 32-bit systems.) The complete agent consists of:
  - `ndtrack.exe`
  - `mgscmn.dll` and `mgscmn64.dll`
  - `getSystemId.exe`
  - `uploader.dll`
  - `wmitrack.ini`.
- In the case of non-Windows platforms, the FlexNet Inventory Scanner is copied to the target device, executed, and then removed from the target device so that there is no remaining installation footprint. For UNIX, the zero-touch agent consists of:
  - `ndtrack.sh` – A universal executable for all UNIX platforms
  - `ndtrack.ini`.

The bulk of this document deals with the full FlexNet inventory agent, for local installation on managed devices.

## What Is Special on UNIX-Like Systems?

A general introduction to the FlexNet inventory agent is available in *Introducing the FlexNet inventory agent* on page 8. These questions focuses on what is distinctive about the agent implementations on supported UNIX-like operating systems.

### What are the disk space requirements?

After installation, the FlexNet inventory agent uses the `/opt` path for installed binary files and `/var/opt` for working data. The disk space requirements for `/opt` are fairly static while for `/var/opt`, the footprint depends on the installer package size required for self-update and how large the logs grow. The following table gives a round-figures guide disk space requirements, where **Package** is the size of the installer package, **/opt** is the space used under the `/opt` path and **/var/opt** is a rough guide of the space required under the `/var/opt` path:

Platform	Package	/opt	/var/opt
AIX	20 Mbyte	25 Mbyte	100 Mbyte
HP-UX	55 Mbyte	55 Mbyte	150 Mbyte
Linux i386	10 Mbyte	20 Mbyte	100 Mbyte
Linux x86_64	10 Mbyte	25 Mbyte	100 Mbyte

Platform	Package	/opt	/var/opt
Mac OS X	15 Mbyte	35 Mbyte	100 Mbyte
Solaris SPARC	20 Mbyte	20 Mbyte	100 Mbyte
Solaris x86	20 Mbyte	20 Mbyte	100 Mbyte

### What packaging technology is used to install the FlexNet inventory agent on UNIX platforms?

The most common/oldest packaging technology supported on each operating system:

- AIX — LPP
- HP-UX — SD-UX
- Linux — RPM (Redhat, Oracle, CentOS, Fedora, SuSE)
- Mac OS X — Mac Package Bundle
- Solaris — Sys V Package (pkg)

### Can the FlexNet inventory agent be installed on UNIX as a non-root/non-admin user?

No. The FlexNet inventory agent is installed on UNIX platforms using an operating-system-level packaging technology. This technology requires root or administrative rights to install the package. The agent also requires root privileges to register and run its services.

### Can `ndtrack` run as a non-root/non-admin user when taking zero touch inventory?

Partially. In the lightweight FlexNet Inventory Scanner used for zero touch inventory, the `ndtrack` executable will run and produce an `ndi` document, but will generally be unable to obtain all inventory details from the target platform. For example, information that cannot be obtained without root or administrative rights includes:

Platform	Missing inventory details (non-root users)
Linux	<ul style="list-style-type: none"> <li>• BIOS details (dmidecode): serial number, UUID, manufacturer, model, chassis type</li> <li>• All hard disk information (from device files).</li> </ul>
Solaris	<ul style="list-style-type: none"> <li>• MAC addresses of network adapters</li> <li>• x86 BIOS details (dmidecode): model, manufacturer</li> <li>• SPARC model using OpenPROM interface. It fails over to using the <code>sysinfo SI_PLATFORM</code> value which can be different.</li> </ul>
HP-UX	<ul style="list-style-type: none"> <li>• SD-UX installation evidence from <code>swlist</code> if access has been locked down with <code>swreg</code> or <code>swacl</code></li> </ul>

Platform	Missing inventory details (non-root users)
	<ul style="list-style-type: none"> <li>vPar evidence including VMType, VMName and vPar capacity (vparstatus requires root)</li> <li>Hard disk drive properties including capacity.</li> </ul>
Mac OS X	Mac OS X package bundle paths under <code>/Applications</code> or <code>/System/Library</code> that are not accessible by the executing user.
All platforms	<ul style="list-style-type: none"> <li>File evidence from any file system path not accessible by the executing user</li> <li>InstallAnywhere, InstallShield Multiplatform, Oracle Universal Installer evidence under paths not accessible by the executing user.</li> </ul>

### What are the directory security settings for the FlexNet inventory agent installed on UNIX?

The security settings for subdirectories of `/opt/managesoft`:

- The `lib` and `libexec` folders are completely locked down to root only.
- The `bin` folder is open to all, to allow easy access to the path of the executables in the folder when using privilege escalation tools like `sudo`.
- The executables in the `bin` folder are locked down to root only.
- `documentation` and `software tag` are readable by all.

The `/var/opt/managesoft` directory is only accessible by root.

### What services does the FlexNet inventory agent install on UNIX systems, and how are these managed?

The inventory agent installs two services which are automatically started after a successful installation and on every system boot. These services are:

- `ndtask` — The agents task launcher. This runs `ndschedag` which executes further command lines (for example, for `ndtrack`, `ndupload`, `ndlaunch`) to perform the required actions.
- `mgsusageag` — The usage tracking agent.

The services are managed using the operating-system-specific service management tools. For example, to start `ndtask`:

Platform	Command
AIX (see note)	<code>/usr/bin/startsrc -s ndtask</code>
HP-UX	<code>/sbin/init.d/ndtask start</code>
Linux	<code>/etc/init.d/ndtask start</code>

Platform	Command
Mac OS X	<code>/sbin/launchctl start com.flexerasoftware.ndtask</code>
Solaris	<code>/etc/init.d/ndtask start</code>

Similarly, to stop the `ndtask` service:

Platform	Command
AIX	<code>/usr/bin/stopsrc -s ndtask</code>
HP-UX	<code>/sbin/init.d/ndtask stop</code>
Linux	<code>/etc/init.d/ndtask stop</code>
Mac OS X	<code>/sbin/launchctl stop com.flexerasoftware.ndtask</code>
Solaris	<code>/etc/init.d/ndtask stop</code>

The `mgsusageag` service can be started and stopped in exactly the same way.



**Note** • For AIX, the agents are in a group called *managesoft* and can both be started using `startsrc -g managesoft` and stopped using `stopsrc -g managesoft`.

## What packaging technology can the FlexNet inventory agent inventory?

The FlexNet inventory agent can collect inventory from most package technologies supported by the operating systems, and some additional third-party packaging technologies:

Platform	Supported package technologies
All platforms	InstallAnywhere (IA), InstallShield Multiplatform (ISMP), BEA/Oracle Installer (BEA), Oracle Universal Installer (OUI), IBM Installation Manager (IIM)
AIX	LPP, RPM
HP-UX	Software Distributor SD-UX Package
Linux	RPM (Red Hat, CentOS, Oracle, SuSE, Fedora, etc), DPKG (Debian, Ubuntu).
Mac OS X	Mac Application Bundle, Mac Package Bundle
Solaris	Sys V Package (pkg), IPS
Windows	MSI, Add/Remove Programs Registry Key

However, the FlexNet inventory agent cannot collect inventory from some of the less common or newer operating system technologies and many third-party technologies. Some known examples include:

- All platforms — IBM InstallStream, IBM Tivoli Netcool Installer
- Mac OS X — Mac flat package.

## Can I change the executing priority of the FlexNet inventory agent?

For both UNIX and Windows, the default is a low priority, set with the option

```
-o LowProfile=true
```

For UNIX, this sets the nice level of the process to 10. To set the nice level to something higher requires a command line like the following example, which sets the nice level to 19:

```
* nice -n 19 /bin/sh ndtrack.sh -o LowProfile=false
```

## Where does FlexNet inventory agent save log files on UNIX?

The locations vary, depending on whether the FlexNet inventory agent is installed locally, or is running zero-touch inventory (see *Introducing the FlexNet inventory agent* on page 8):

- For the installed agent, the log files are found in `/var/opt/managesoft/log`.
- For zero-touch inventory, the location further depends on the account running the FlexNet inventory agent:
  - For the root user running zero-touch inventory, the log files are found in `/var/tmp/flexera/log`.
  - For a non-root user, represented as `userName`, the log files are found in `/var/tmp/flexera.userName`.

The following log files are available:

- `installation.log` — Log from `ndlaunch`, responsible for downloading and install all packages required for FlexNet inventory agent
- `policy.log` — Generated by the policy download agent, `mgspolicy`
- `schedule.log` — Log from the `ndschedag` schedule agent
- `tracker.log` — Generated by the inventory agent, `ndtrack`
- `uploader.log` — Log from the `ndupload` file upload agent
- `usageagent.log` — Generated by the `mgsusageag` usage tracking service.

## How can I manually [re-]configure the Unix inventory agent after installation?

From a console or remote terminal connection, run the command:

```
/opt/managesoft/bin/managesoft-configure
```

This will prompt for configuration items.

## Reading /etc/passwd

If you run a system trace on the UNIX version of `ndtrack`, you will see that it reads `/etc/passwd`. The UNIX `ndtrack` uses the `setpwent()` and `getpwent()` library calls to obtain the `pw_name`, `pw_dir` and `pw_shell` properties for each user. In particular, `ntrack` uses the `pw_dir` property (each user's home directory) to find user-based installation evidence for BEA and InstallAnywhere installation technologies.

---

# 2

---

## Deploying the FlexNet Inventory Agent

### Topics:

- *Automated Adoption Summary*
- *Self-Managed Deployment: Collecting the Software*
- *Configuring Installation on Microsoft Windows*
- *Configuring Installations on UNIX-like Platforms*
- *Protecting Your Customizations*
- *Checking the Installed Version*

It is not mandatory to deploy the FlexNet inventory agent onto devices from which you want to collect inventory: inventory can be remotely gathered for a computing device, without the footprint resulting from installation of any agent. However, you may decide on the local installation of the FlexNet inventory agent for a number of reasons:

- Better scheduling of activities in machine up-time
- Gathering usage data
- Avoiding the need to meet connectivity and access requirements to use remote gathering functionality
- Specialized inventory gathering not available through other methods.

Once you have decided on local installation of the FlexNet inventory agent on the device, you must decide on the deployment method:

- In a simple, well-connected network environment with straightforward options to manage passwords and remote access to computers, the simplest process is the automation provided within FlexNet Manager Suite. This process is called "adoption", and is summarized in *Automated Adoption Summary* on page 16. More details about adoption are available through the online help.
- You can decide to manually configure and deploy the FlexNet inventory agent to devices using your preferred deployment technology. The bulk of this chapter covers this option.

# Automated Adoption Summary

FlexNet Manager Suite can adopt computing devices automatically, based on rules that you set.

This approach bypasses the manual preparation of installation packages, and their deployment using the third-party deployment technology you may already use for distribution and installation of those packages. In this process, the configuration and deployment of FlexNet inventory agent is fully automated, and you do not need to understand the various code components involved, nor their many settings, nor do any custom configuration (and in fact, you cannot do any custom configuration or individual scheduling). On the other hand, there may be some additional preparation (depending on your environment) to allow this automation to work. Comparing the automated process with your third-party deployment technology may help to make a balanced choice about which approach to use, or whether to mix and match these in different parts of your environment. In summary, the process, including some steps that are common both to automated adoption and to your own package preparation and deployment, is as follows:

1. Ensure that you have the appropriate inventory beacons fully operational.

To do this, navigate to **Discovery & Inventory > Beacons** (in the **Network** group), and click **Deploy a beacon**. Consult the online help for these pages for more information. (This step is required for both approaches.)

2. Ensure that at least one inventory beacon is configured to cover the subnet containing the target computer devices.

This setting is available through the web interface for FlexNet Manager Suite at **Discovery & Inventory > Beacons**. See the online help there for more information. (This step is required for both approaches.)

3. If yours is a highly secure, locked down environment, you may need to open network ports on the target computer devices to allow for remote execution.

Since the inventory beacons use standard ports to access target devices and remotely install the FlexNet inventory agent, the required ports are already available in many environments. (The ports are documented in the online help.) Notice that wherever you choose to take inventory by remote execution rather than adopting target devices by installing FlexNet inventory agent, you also need to enable the appropriate ports for remote execution. Therefore, depending on your approach, this step may be required for at least parts of your environment in either deployment approach.

4. Ensure adequate credentials are available for the automated installation process to run. There are two possible approaches:

- You can allow the appropriate inventory beacons to run using a domain administrator account that has installation privileges on the target computer devices.
- You can record appropriate credentials in the secure Password Store available on each inventory beacon (for details, see the online help for the inventory beacon).

While this can be quite manageable in some environments, in others the existing deployment solution may have already solved the issue of credentials for installation in a way that you prefer.

5. Navigate to **Discovery & Inventory > Discovery and Inventory Rules**, and create one or more rules to discover computing devices within your enterprise, and then to adopt selected devices as required.

Rules consist of

- Targets that identify the devices, and (for all the devices identified within a single target) specify how to connect, whether to adopt, and whether to track application usage



- Actions that declare what to do to the targeted devices
- A schedule for implementing the action on the targeted devices.

By specifying multiple targets, you can choose which computer devices are adopted, and which are inventoried by remote execution. For more information, see the online help for these pages.



**Tip** • Part of the art in this automated deployment may be in declaring a schedule that suits when the target devices are available (that is, running, connected to the network, and not too busy). Particularly with individual workstations and laptops, this may require some external process management. For example, you may communicate to users that they should leave target devices running overnight, or some similar arrangement. Since the discovery and adoption rules execute on a repeated schedule, there are many such 'windows' when devices can be automatically adopted. Nonetheless, the timing of installation may be an issue you have already resolved with your third-party deployment solution.

6. Wait for the execution of the rule(s), the installation of the FlexNet inventory agent, and the resultant data uploads.

To monitor progress and results, navigate to the system menu (⚙️ ▼ in the top right corner), **System Health** > **System Tasks**, and see the online help there for more information.

When execution of the rules is completed, your targeted devices are adopted and reporting inventory from the FlexNet inventory agent according to the global schedule specified in the **System Settings** page, on the **Inventory** tab, under **Managing the processing queue for import and reconciliation**. As well, the FlexNet inventory agent receives regular policy updates (including changes to the rules it should execute), and you can even control the self-update mechanism when new versions of the FlexNet inventory agent are ready to deploy.

The adopted device is also visible in the web interface of FlexNet Manager Suite. For example, it is visible in **Discovery & Inventory** > **All Discovered Devices** page, and the **Agent installed** column shows that the FlexNet inventory agent has been successfully deployed and is reporting.



**Tip** • This column is populated only for devices adopted through the automated processes, using the standard FlexNet deployment package. If you choose to use alternate deployment methods using your own package, this column does not change state to show when the FlexNet inventory agent is deployed.

For those target devices where you use automated adoption, your work is complete. The remainder of this chapter deals with managing your own deployment through third-party tools.

## Self-Managed Deployment: Collecting the Software

You have decided to deploy the FlexNet inventory agent with a third-party tool of your choice. Start with the appropriate version(s) of the FlexNet inventory agent.

The FlexNet inventory agent is supported on a variety of platforms. Collect the version(s) required as follows:

1. On the workstation where you plan to configure the package for installing FlexNet inventory agent, use a web browser to log in to the web interface for FlexNet Manager Suite.
2. Navigate to the system menu (☰ ▼ in the top right corner) > **Data Inputs**, and select the **Inventory** tab.
3. Click **Download inventory agent**.  
The **Download inventory agent** page appears.
4. Prepare for editing the bootstrapping file:
  - For target devices running any version of Microsoft Windows: click the hyperlink to **Download bootstrapping template file**, saving the file to a folder of your choice (such as C:\temp). This file must be customized for your deployment, minimally to identify the bootstrap inventory beacon with which FlexNet inventory agent will first communicate (details are forthcoming in *Edit the Configuration File for Microsoft Windows* on page 19).
  - For target devices running UNIX-like systems: there is no equivalent download of a sample file, but full details for constructing your own are included in *Configure the Bootstrap File for UNIX* on page 23. No immediate action on the bootstrapping file is needed now.

5. From the **Inventory agent** option list, select the (first) version of the FlexNet inventory agent you want to deploy.

Be sure to scroll down, as the most recent releases are at the bottom of the option list. You may choose whichever recent version is approved for use in your enterprise. In general, it is recommended to deploy the latest version.

The same software is known by different (historical) names on different platforms (a managed device is one which has the FlexNet inventory agent locally installed). In the current release, you can select from the following supported platforms:

- FlexNet Inventory Agent, which is for Windows platforms




---

**Tip** • *In earlier releases, the FlexNet inventory agent was known as ManageSoft for Managed Devices.*

- ManageSoft for AIX Managed Devices
  - ManageSoft for HP-UX Managed Devices
  - ManageSoft for Linux (i386) Managed Devices
  - ManageSoft for Linux (x86\_64) Managed Devices
  - ManageSoft for Mac OS X Managed Devices
  - ManageSoft for Solaris (sparc) Managed Devices
  - ManageSoft for Solaris (x86) Managed Devices.
6. Click **Download**, and save the archive to a folder of your choice.
  7. If necessary, repeat the download of any additional versions that you choose to configure and deploy.

# Configuring Installation on Microsoft Windows

Broadly, there are two ways to configure the installation of the FlexNet inventory agent for Microsoft Windows:

- You can edit the Windows Installer command line to add transforms or specify that particular components should or should not be installed. For more details, see *Customizing the Windows Installer Command Line* on page 21.
- You can set preferences in the bootstrap configuration file, as described in *Edit the Configuration File for Microsoft Windows* on page 19.

If you are using alternative deployment methods but then wanting the managed device to fit neatly into the standard operating procedure of FlexNet Manager Suite, there is nothing more to do after those steps. In contrast, if you are customizing the installation or behavior of the FlexNet inventory agent, you could also review *Protecting Your Customizations* on page 31 in case you want to prevent your customizations being over-written by the system-wide standard policy. When you are satisfied with both forms of configuration/customization and your protection settings, you can use your preferred deployment tool to distribute the completed packages for installation on target devices.

## Edit the Configuration File for Microsoft Windows

Initial implementation of the FlexNet inventory agent is controlled by its bootstrapping configuration file.

In *Self-Managed Deployment: Collecting the Software* on page 17, you downloaded the template for the bootstrap configuration file (`mgssetup.ini`) for use on Microsoft Windows platforms. This file can contain many legacy settings, especially for those interested in customizing the behavior of the FlexNet inventory agent. For example, the latter part of the file allows for individual preferences to be set that will automatically be written to the registry of the managed device during the installation process (some of these are documented in the chapter on *Preferences* on page 50 for the advanced administrator).

However, for a straightforward deployment and installation, there are only two significant specifications required:

- Identifying the inventory beacon that the managed devices should contact after installation, to collect their policy (which includes rules to apply, schedules, and the like) — this much is mandatory
- Configure whether or not these managed devices should monitor application usage (by default they do not).

Some sections of the bootstrap file are historical, and should not be modified (as noted below).

1. Open a copy of `mgssetup.ini` in a flat text editor of your choice.

As you may need several versions of your custom bootstrap files, it is best practice to save a backup copy of the original, unedited file. In deployment, the file name must not be changed. It is therefore good practice to save each customized copy in its own directory. Also beware of editors which may insert special (non-ASCII) characters in the file.




---

**Note** • Lines starting with a semi-colon character (`;`) are comments and have no effect. 'Uncommenting' a line means to remove the semi-colon from the start of the line.

- To specify the inventory beacon from which this managed device should collect its policy, locate the following, and uncomment and customize the second line, following the guidelines below:

```
[Custom Install Settings]
DEPLOYSERVERURL = http://beacon.server.com/ManageSoftDL
```



**Tip** • An inventory beacon is derived from an earlier form of deployment server, and still deploys policy and agent updates to its managed devices.

- The trailing `ManageSoftDL` is mandatory, and identifies the web share on the inventory beacon which must be accessible to managed devices (check permissions).
- The value must be a URL (a UNC value of the form `\\servername\ManageSoftDL$` will not work). The URL may have either of the following protocols:
  - `http://`
  - `https://`
- The URL must be no longer than 80 characters.



**Important** • If this value is not set correctly, devices will not be managed and will require a software re-install.

If the details are specified correctly but the policy was not initially available (perhaps because of network issues, or because the inventory beacon was unavailable when the FlexNet inventory agent attempted to connect, or because policy had not yet been prepared and distributed from the central application server), FlexNet inventory agent retries policy collection at each reboot of the managed device until successful. (This behavior is different than for the FlexNet inventory agent on UNIX-like systems, where it makes a daily attempt to download policy at a random time each day.)

- Usage tracking from file evidence is disabled by default. To enable (during the installation process) future application usage metering (by tracking the time during which installed files are opened on the targeted devices), locate and uncomment the following line:

```
[Usage Agent]
USAGEAGENT_DISABLE = False
```

For most situations, you have finished configuring this file. Skip the advanced steps, and see the notes at the end.

- ADVANCED USE ONLY:** Optionally, insert any preferences for the operation of the FlexNet inventory agent in the later sections of the `mgssetup.ini` file that allow storing the preferences in the Windows registry.

For example, if you wanted to modify the current default values (shown here as an example) for the fail-over algorithms that determine how the FlexNet inventory agent attempts to prioritize inventory beacons when choosing which one to access, edit the following section, changing to your own preferences:

```
; Registry settings to be created under
; HKLM\Software\ManageSoft Corp\ManageSoft\NetSelector
[NetSelector]
desc0 = SelectorAlgorithm
```

```
val0 = MgsRandom;MgsPing;MgsADSiteMatch
```

More more information about the available preferences, see *Preferences* on page 50.



**Note** • Other settings in the `mgssetup.ini` file should typically be left unchanged. They may be edited for advance, legacy, or other unusual configuration scenarios. If you need such changes, follow the guidance provided within the template file itself.

5. Save the customized copy of `mgssetup.ini` in the same folder as the FlexNet Inventory Agent zip archive downloaded in *Self-Managed Deployment: Collecting the Software* on page 17.  
(If you are preparing distinct customizations for different parts of your computing estate, use folder naming to separate and distinguish the different copies of `mgssetup.ini`. Do not rename the `mgssetup.ini` file.)



**Tip** • For another advanced preference, see *Uninstalling from Windows Platforms* on page 23.

The following legacy sections of the configuration file should not be edited, and should be left commented out:

- ;BOOTSTRAPPEDPOLICY
- ;WAITFORPOLICY
- ;IGNOREPOLICYFAILURE
- ;PEER2PEER\_DISABLE.

## Customizing the Windows Installer Command Line

You can customize the Windows Installer in the usual ways.

The two paths for configuring your installation of the FlexNet inventory agent are:

- Configuring the `mgssetup.ini` file (for which see *Edit the Configuration File for Microsoft Windows* on page 19)
- Modifying the MSI installation, covered in this topic. This entire process is optional, for advanced use by those who have transforms planned.

You downloaded the installer archive in *Self-Managed Deployment: Collecting the Software* on page 17, saving it to a convenient directory. Return to that location now.

1. Unzip the downloaded archive (named like `managesoft-MM.N.B.zip`, where the placeholders represent release numbering).

The MSI file for the FlexNet inventory agent is called `FlexNet Inventory Agent.msi`. You can customize the installation, for example, by installing or preventing installation of particular components, or by applying transforms. To do this, you edit the Windows Installer command line, stored in the `Setup.ini` file that is in the same folder as the MSI.

2. In the unzipped archive, open `Setup.ini` in the flat text editor of your choice.

3. Locate the following section, and amend the command line as required:

```
[Startup]
CmdLine=/l*v "%TEMP%\FlexNet Inventory Agent.msi.log"
```

For example:

- To apply a transform, append `TRANSFORMS="custom.mst"` to the end of the command line.

Use semicolons (;) to separate multiple transforms. For example, to apply `custom1.mst` and `custom2.mst` for both initial installs and upgrades, the addition would look like:

```
TRANSFORMS="custom1.mst;custom2.mst"
```

- Add or remove any other Windows Installer command line options.

For example, to prevent installation of the application usage agent on computers being brought under management, add `REMOVE=aua` to the end of the `CmdLine`:

```
[Startup]
CmdLine=/l*v "%TEMP%\FlexNet Inventory Agent.msi.log" REMOVE=aua
```

4. Save the modified `Setup.ini` file.

When you have completed modifications to the configuration file and to the command line of the installer, you can use your preferred deployment tool(s) to pack, validate, and deploy the installation package. See the following deployment notes.

## Deployment Notes for Windows Platforms

You have finished editing the bootstrap configuration file (`Mgssetup.ini`) and, if necessary customized the MSI command line stored in `Setup.ini`. These files may now be handed off, together with `FlexNet Inventory Agent.msi` and the associated `Data1.cab` file, for packaging and deployment. The following notes about the installation may be helpful to the packaging team.

1. The `mgssetup.ini` bootstrap configuration file must be placed in the same folder as the `FlexNet Inventory Agent.msi` file before executing the install command.
2. In this location, the `mgssetup.ini` file must be readable by both the SYSTEM and installing user at installation time. This means that the MSI package cannot be installed from a mapped drive (which would only be visible to the installing user), or a file share which is not visible to both user accounts.
3. As usual, the installing account needs administrator privileges to complete the installation.
4. For operation, the FlexNet inventory agent services must be configured to run as the Local System account.
5. Once the FlexNet inventory agent has been successfully deployed and is reporting inventory, it is visible in the **Discovery & Inventory > All Discovered Devices** page.



**Tip** • The **Agent installed** column on that page is populated only for devices adopted through the automated processes, using the standard FlexNet deployment package. If you choose to use alternate deployment

*methods using your own package, this column does not change state to show when the FlexNet inventory agent is deployed.*

## Uninstalling from Windows Platforms

If you decide to remove the FlexNet inventory agent from a system running Microsoft Windows, you have two basic approaches:

- Use the deployment tool with which you completed the original installation to now remove it.
- Manually navigate to Add/Remove Programs (or equivalent) in Microsoft Windows and uninstall FlexNet inventory agent.



**Tip** • This standard Windows facility means that, depending on privileges in your environment, users of managed devices can remove FlexNet inventory agent. If you wish to prevent users on these target devices from using Windows Add/Remove Programs to uninstall the FlexNet inventory agent, uncomment this line in `mgssetup.ini` before deployment:

```
[Custom Install Settings]
ALLOWUNINSTALL = 0
```

*Of course, this reduces your options for uninstallation, and you must then use your deployment tool to do any planned removal.*

Uninstallation removes the executables, but leaves behind the folders `%ProgramFiles%\ManageSoft\Launcher\Cache`, and its peer `PkgCache`, since these may contain content that a future re-install of the FlexNet inventory agent can use. (On 64-bit platforms, substitute `%ProgramFiles(x86)%`.)

## Configuring Installations on UNIX-like Platforms

This section includes an overview of some ways that non-Windows platforms differ from Windows platforms. There is also information on preparing and deploying your bootstrap configuration file, and installing the FlexNet inventory agent on UNIX-like platforms.

### Configure the Bootstrap File for UNIX

The initial configuration of the FlexNet inventory agent can also be set for UNIX-like platforms.

For UNIX and OS X, there is no sample bootstrap configuration file available through the central application server. Instead, you can prepare your customized bootstrap configuration file as follows:

1. Copy the sample text from *Sample UNIX Bootstrap Configuration File* on page 25 into your preferred flat-text editor.
2. Locate and edit the following line to identify the inventory beacon from which the new managed device should download its initial policy:

```
MGSFT_BOOTSTRAP_DOWNLOAD=http://beacon.mydomain.com:8080/ManageSoftDL/
```

- In the automated adoption process (where the inventory beacon installs FlexNet inventory agent by remote execution), it is mandatory to use the HTTP protocol. If you are independently managing your own deployment, it's also normal to use the HTTP protocol for bootstrapping, because it is simpler to set up and get operational. However, if you require the HTTPS protocol for your own deployment, insert it in this value.
  - Replace the placeholder *beacon.mydomain.com* with the fully qualified domain name of the inventory beacon. If required, and provided that you are using the HTTP protocol, you may instead use the server's IP address. (There are widely publicized issues around using an IP address with the HTTPS protocol.)
  - If you are using the default port (80 for HTTP, and 443 for HTTPS), you can omit the port number. For any custom port numbers, include the port number in the URL as shown (:8080).
  - The string literal *ManageSoftDL* is the folder name for the download location. This is its default value. If your inventory beacon has a different name for the download folder, use that name instead.
3. Following those same guidelines, edit the following value for the upload location on the same inventory beacon.

```
MGSFT_BOOTSTRAP_UPLOAD=http://beacon.mydomain.com:8080/ManageSoftURL/
```

To bootstrap the UNIX agents, both the download and upload locations must be specified. (This is not the case for the agents on Windows, where only the download location is required.)



**Tip** • *The section about proxies in the bootstrap file is only required in the unusual circumstances that you have a proxy between the managed device(s) and the inventory beacon(s) (in which case follow the guidance in the template). When this is not the case, leave these settings commented out.*

4. The local web server on inventory beacons can be configured to use either the HTTP or HTTPS protocol for communication with managed devices. The choice of protocol is downloaded to managed devices as part of their policy. If you wish to enable the HTTPS protocol for data transfer between the managed device(s) and their inventory beacon(s), but want to prevent checking of the security certificates authenticating the inventory beacon server(s):
  - a) Uncomment and edit the following line to prevent certificate checking:

```
MGSFT_HTTPS_CHECKSERVERCERTIFICATE=false
```

- b) Uncomment and edit the following line to prevent a check for revocation of certificates:

```
MGSFT_HTTPS_CHECKCERTIFICATEREVOCATION=false
```

By default (when unspecified), both these preferences are true. If (instead of disabling) you are leaving the default certificate check in place, you must deploy a copy of the certificates to the managed devices. This



is described in *Installing FlexNet inventory agent on UNIX* on page 26, and there is more information in *Enabling the HTTPS Protocol on UNIX Agents* on page 28 and *HTTPS CA Certificate File Format (UNIX)* on page 29.



**Tip** • These two settings in the bootstrap configuration file configure certificate checking, and revocation checking, for all component agents on the managed device. After installation on the computer device, these settings can be checked in the `/var/opt/managesoft/etc/config.ini` file, in the `[ManageSoft \Common]` section, as `CheckServerCertificate` and `CheckCertificateRevocation`. It is also possible to over-ride the global setting for individual component agents. For more details, see `CheckServerCertificate` on page 58 and `CheckCertificateRevocation` on page 57.

5. Save the file as `mgsft_rollout_response`.



**Tip** • Leave `MGSFT_RUNPOLICY=1` unchanged, so that downloaded policy is applied after installation. For as long as policy is not available for any reason, on UNIX and OS X the agents run a daily check for policy at a random time between 8am and 11pm (local time on the managed device) until policy is successfully downloaded. (This catch-up behavior is different than the Windows agents, which rely on a machine reboot to check again for missing policy.) Once policy (with schedule) is initially downloaded, it is updated daily on the downloaded schedule, refreshing client settings, inventory-gathering schedule, and the like.

6. Configure your preferred deployment technology to install a copy of this file as `/var/tmp/mgsft_rollout_response` on the target device(s).

The path and file name are mandatory. This file must be present before FlexNet inventory agent is installed. Post installation scripts in the installation package for FlexNet inventory agent use properties from this file to create the initial configuration.



**Tip** • In preparing the Windows bootstrap file (`mgssetup.ini`), you could turn application usage tracking on for the managed devices using the bootstrap file. This is not possible in the bootstrap file for UNIX-like systems. To turn on usage tracking, the simplest path is to set usage tracking as part of defining targets (in the web interface of FlexNet Manager Suite), so that managed devices receive this setting as part of their downloaded policy. Manually editing `config.ini` for UNIX-like platforms is also possible (see *Updating config.ini on a UNIX Device* on page 29), but this approach is not as easy to scale.

## Sample UNIX Bootstrap Configuration File

```
# The initial download location(s) for the installation.
# For example, http://myhost.mydomain.com/ManageSoftDL/
# Refer to the documentation for further details.
MGSFT_BOOTSTRAP_DOWNLOAD=http://beacon.mydomain.com:8080/ManageSoftDL/

# The initial reporting location(s) for the installation.
# For example, http://myhost.mydomain.com/ManageSoftRL/
# Refer to the documentation for further details.
MGSFT_BOOTSTRAP_UPLOAD=http://beacon.mydomain.com:8080/ManageSoftRL/
```

```
# The initial proxy configuration. Uncomment these to enable proxy configuration.
# Note that setting values of NONE disables this feature.
# MGSFT_HTTP_PROXY=http://webproxy.local:3128
# MGSFT_HTTPS_PROXY=https://webproxy.local:3129
# MGSFT_PROXY=socks:socks.socksproxy.local:19121,direct
# MGSFT_NO_PROXY=internal1.local,internal2.local

# Check the HTTPS server certificate's existence, name, validity period,
# and issuance by a trusted certificate authority (CA). This is enabled
# by default and can be disabled with false.
# MGSFT_HTTPS_CHECKSERVERCERTIFICATE=true

# Check that the HTTPS server certificate has not been revoked. This is
# enabled by default and can be disabled with false.
# MGSFT_HTTPS_CHECKCERTIFICATEREVOCATION=true

# The run policy flag determines if policy will run after installation.
# "1" or "Yes" will run policy after install
# "0" or "No" will not run policy
MGSFT_RUNPOLICY=1
```

## Installing FlexNet inventory agent on UNIX

You downloaded the installers of interest in *Self-Managed Deployment: Collecting the Software* on page 17. You have also customized your bootstrap configuration file (or potentially multiple variants for different platforms and contexts) in *Configure the Bootstrap File for UNIX* on page 23.

1. Configure your deployment/installation tool to deliver the bootstrap configuration file to `/var/tmp/mgsft_rollout_response`.  
This file must be in place on the device before you run the installer for FlexNet inventory agent.
2. If the target computer device is to use the HTTPS protocol to communicate with an inventory beacon, and you require certificate checking to validate that the device is talking to the correct inventory beacon (for details, see *Enabling the HTTPS Protocol on UNIX Agents* on page 28):
  - a) Prepare a summary HTTP CA certificate for the target device(s) (see notes in *HTTPS CA Certificate File Format (UNIX)* on page 29)

- b) Deploy the certificate file as `/var/tmp/mgsft_rollout_cert` on the target device.

This file must be in place on the device before you run the installer for FlexNet inventory agent. During installation, the `/var/tmp/mgsft_rollout_cert` file is copied to `/var/opt/managesoft/etc/ssl/cert.pem`.



**Tip** • If you do not complete this as part of the deployment and installation process, after installation you can simply copy the completed certificate to `/var/opt/managesoft/etc/ssl/cert.pem` on a device where FlexNet inventory agent is locally installed.


3. For Solaris platforms where you want a silent installation without user interaction, prepare a flat text `admin` file with the following content to include in your deployment package:

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
```

```
space=quit
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
```

4. Configure the installation tool appropriately for each target platform.

There are different command lines for different UNIX platforms. Configure your deployment and installation technology to execute the following commands on the appropriate platforms (substituting *[VERSION]* and *[PACKAGE PATH]* with appropriate values):

Platform	Installation command line
AIX	<code>installp -aY -d managesoft.[VERSION].bff managesoft.rte</code>
HP-UX	<code>swinstall -v -x mount_all_filesystems=false -x allow_downdate=true -s [PACKAGE_PATH]/managesoft-[VERSION].depot managesoft</code>
Debian Linux x86	<code>dpkg --install managesoft_[VERSION]_i386.deb</code>
Debian Linux x86_64	<code>dpkg --install managesoft_[VERSION]_amd64.deb</code>
Linux x86	<code>rpm --upgrade --oldpackage --verbose managesoft-[VERSION]-1.i386.rpm</code>
Linux x86_64	<code>rpm --upgrade --oldpackage --verbose managesoft-[VERSION]-1.x86_64.rpm</code>
Mac OS X	<code>installer -verbose -pkg ManageSoft.pkg -target</code>
Solaris SPARC	<pre>pkgadd -n -a admin -r /dev/null -d managesoft-[VERSION].sparc.pkg ManageSoft</pre>  <p><b>Tip</b> • If you saved your <i>admin</i> file under a different name, modify the command line appropriately.</p>
Solaris x86	<code>pkgadd -n -a admin -r /dev/null -d managesoft-[VERSION].x86.pkg ManageSoft</code>

5. You can now hand these materials off for packaging and deployment.

After installation (assume the standard setting of `MGSFT_RUNPOLICY=1`), FlexNet inventory agent attempts to connect with the inventory beacon. You may need to protect your customization, for which see the notes in *Protecting Your Customizations* on page 31.

## Enabling the HTTPS Protocol on UNIX Agents

The FlexNet inventory agent (or more precisely, its component agents) may make use of the HTTPS protocol for communications with inventory beacons. Whereas Windows systems can manage the security certificates for you, on UNIX and OS X some manual configuration is required.



**Note** • *The HTTPS protocol is only available to the installed agents, and is not available to the zero-touch FlexNet Inventory Scanner.*

There are three security levels which can be enabled for HTTPS, using two preference settings. From the highest level of security to the lowest, these are as follows.

### Checking certificate(s) and excluding revocation

Full checking of each HTTPS server certificate involves having a local copy on the managed device of all the public certificates from each certificate authority (CA) that are used to validate the HTTPS server certificates. These certificates must be available in the `/var/opt/managesoft/etc/ssl/cert.pem` file on the managed device (see *HTTPS CA Certificate File Format (UNIX)* on page 29 for more details). The device must also be able to download the certificate revocation list from an HTTP location. For this level of security, both the `CheckServerCertificate` and `CheckCertificateRevocation` settings should be set to `True` (these are the default settings).

### Checking certificate(s)

This mid-level security model provides an encrypted channel and validation of the HTTPS server, but does not provide a way to revoke the certificate used to validate the HTTPS server. This may be adequate where you are confident of the longevity of your certificates.

Checking of the server certificate still requires the CA certificate be installed on the managed device in the `/var/opt/managesoft/etc/ssl/cert.pem` file. As well, the `CheckServerCertificate` preference must preserve its default value of `True`. Ignoring the revocation list can be configured by disabling (setting to `False`) the `CheckCertificateRevocation` settings for all component agents on the managed device.



**Tip** • *It is also possible to generally disable for most agents, but create exceptions where a particular agent still checks for possible certificate revocation. For details, see [CheckCertificateRevocation](#) on page 57.*

### Relying on encryption

If you are confident of the security of your infrastructure, it is possible to ignore the server certificates entirely. This provides an encrypted channel of communication, but does not provide validation that the device is actually talking to the correct HTTPS server.

Disabling checking of the server certificate can be achieved by disabling (setting to `False`) both the `CheckServerCertificate` and `CheckCertificateRevocation` settings for all component agents on the managed device. In this mode of operation, the CA certificate is not required to be installed on the managed device.

## HTTPS CA Certificate File Format (UNIX)

By default, if the FlexNet inventory agent is (or in more detail, its component agents are) configured to use the HTTPS protocol to communicate with a inventory beacon, the certificate(s) for the HTTPS are checked. This ensures that the agents are communicating with the correct inventory beacon server. By default, the certificates are also checked for currency: that is, a check is made that the certificate(s) have not been revoked. For the preference settings controlling these behaviors, see *Enabling the HTTPS Protocol on UNIX Agents* on page 28.

Certificate checking requires that a copy of the certificate(s) to be validated has been installed on the managed device.

The format of the HTTPS CA certificate file for the FlexNet inventory agent on UNIX systems is a copy of all the root Certificate Authority (CA) certificates that are used by the HTTPS web servers supplying content to the managed device or receiving content from the managed device. For most organisations, this will be a single certificate. The collection of one or more root CA certificates should be concatenated together into a single file. This certificate file should be saved using the PEM format. Each PEM-format certificate should be base-64 encoded plain text surrounded by a `BEGIN CERTIFICATE` header and an `END CERTIFICATE` footer. That is:

```
-----BEGIN CERTIFICATE-----
MIIDiTCcAnGgAwIBAgIQWO/IibrLpZ5Hts3u3xH7TzANBgkqhkiG9w0BAQUFADAR
MQ8wDQYDVQQDEwZ0ZncyazMwHhcNMTAxMTI1MDEyMDM4WhcNMTUxMTI1MDEyODAl
.....
wXvMSERKsNsJ6FwwXFGA3HBrRLTHzqzsfU1UAbV+SBm/FSFkuWsy4QWAuJCbnCnv
c3ClFHxqwaIq9UWvO5FR5kD4gK9LZOuy4B7tLTQmpJScFSiPZrIBalcQ5uWl
-----END CERTIFICATE-----
```

Ensure that each CA certificate (within a certificate chain up to the root CA) includes a reference to a downloadable Certificate Revocation List (CRL). The reference to the CRL must be described using the X509v3 extensions "X509v3 CRL Distribution Points". The CRL must be downloadable using the HTTP protocol and must be in DER format (a binary file).

To deploy the resulting certificate during deployment of FlexNet inventory agent to managed devices, see *Installing FlexNet inventory agent on UNIX* on page 26.

## Updating config.ini on a UNIX Device

For OS X and UNIX managed devices, many preferences are stored in the `config.ini` file stored in `/var/opt/managesoft/etc`. This file acts as a "virtual registry": that is, a repository on non-Windows platforms for settings that, on Windows platforms, are stored in the registry.

Initially, this file will contain a few factory default settings, plus the values imported from the `mgsft_rollout_response` file described in *Configure the Bootstrap File for UNIX* on page 23. Some of the installed agents (such as the schedule agent and the usage agent) read their preferences directly from this file.

If you wish to update settings in `config.ini`, you can use the following approach, which can be completed manually or can be scripted with an appropriate deployment tool. In this description, the example used is to disable network sense in an installed FlexNet inventory agent.

1. Create a temporary file following the format of `config.ini`, but including only the section names containing changes, and the changed values.

Example:

```
[ManageSoft\Launcher\CurrentVersion]
NetworkSense=False
[ManageSoft\NetSelector\CurrentVersion]
SelectorAlgorithm=MgsSubnetMatch(,false)
```

2. Save (or deploy) this patch as a temporary file on the target device (where FlexNet inventory agent is locally installed), giving it a distinct name.

For example, deploy (or save) the patch file to `/var/tmp/tempconfig.ini` on the target device(s).

3. Execute the `mgsconfig` tool with the `-i` option identifying your temporary patch file.

Example:

```
/opt/managesoft/bin/mgsconfig -i /var/tmp/tempconfig.ini
```

4. Remove the temporary file.

Inspecting the `/var/opt/managesoft/etc/config.ini` file shows that the settings in your temporary file have been applied to the `config.ini`.



**Tip** • The `config.ini` file is used only for the FlexNet inventory agent installed locally on the target device. In the case of zero touch inventory (where, on UNIX platforms, the FlexNet inventory agent is temporarily installed, executed, and removed again), the corresponding file is called `ndtrack.ini`. It is not mandatory to supply this file, since `ndtrack.sh` has built-in default settings that provide all necessary values that were not supplied in the command line. However, if you wish to override any of these default values on a more frequent basis, without needing to include the changed value in every command line, you can provide a customized `ndtrack.ini` file that must be installed in the same folder as `ndtrack.sh`. The schema of the `config.ini` and `ndtrack.ini` files is identical. However, where `config.ini` may contain preferences for all agents, `ndtrack.ini` may contain only preferences for `ndtrack.sh`, and only in zero touch mode. For example, there are no fail-over settings included in `ndtrack.ini`, because in zero-touch mode, the inventory beacon has just installed `ndtrack.sh` with instructions to upload the inventory results to only that same inventory beacon. Fail-over is not an option.

## Uninstalling FlexNet inventory agent from UNIX

If you need to manually uninstall FlexNet inventory agent from a managed device, use the following command lines.



**Note** • Uninstallation leaves behind the folder `/var/opt/managesoft` since it may contain a package cache that a future re-install of the FlexNet inventory agent can use.

Platform	Uninstallation command line
AIX	<code>installp -u managesoft</code>
Debian Linux x86 and x86_64	<code>dpkg --purge managesoft</code>
HP-UX	<code>swremove managesoft</code>
Linux x86 and x86_64	<code>rpm --erase --verbose managesoft</code>
Mac OS X	<code>/opt/managesoft/bin/uninstall-managesoft.command -force</code>
Solaris x86 and SPARC	<code>echo action=nocheck &gt; admin.mgs pkgrm -n -a admin.mgs managesoft rm -f admin.mgs</code>

## Protecting Your Customizations

As soon as its installation is complete, the FlexNet inventory agent attempts to contact the inventory beacon identified in the bootstrap configuration file, and if successful, requests its policy (including the schedule on which it should perform the specified actions).

You might desire either of two distinct outcomes here:

- Most likely, you may want this computing device to 'join the family', sharing the same schedule, policy, and controlled self-updating as all other instances of the FlexNet inventory agent on devices that were adopted automatically. If you take no special action, this is the default behavior.
- You may want to protect the unique settings on this instance of FlexNet inventory agent, such as a distinct set of failover inventory beacons. In this case, continue below.

To prevent the custom settings on this device being over-written by the standard policy, you must do three things:

1. Prevent the distribution of policy to those devices that are outside all known targets:
  - a) In the web interface for FlexNet Manager Suite, navigate to **Discovery & Inventory > Settings**.
  - b) In the **Beacon settings** group, select the check box **Migration mode: Restrict inventory settings to targeted devices**.
  - c) Click **Save**.

This setting must remain permanently on for as long as you want to keep some of your devices (with FlexNet inventory agent installed) outside the reach of standard policies and settings (including the agent inventory schedule specified a little higher on the same product page).

2. Ensure that the protected devices are never included in any target for discovery and inventory rules.

These rules are specified in **Discovery & Inventory > Discovery and Inventory Rules**. There is no programmatic way to prevent these protected devices from being accidentally included in a target; it must simply be a matter of corporate guidelines that (for example) the specified subnet or the particular machine(s) must never be included in any target.

3. Plan to arrange your own updates to the FlexNet inventory agent on all protected devices, probably using the same configuration and deployment methods you have used for initial installation.

## Checking the Installed Version

You may need to verify the installed version of FlexNet inventory agent on a managed device, either for manual checking or as part of a scripted solution. You can inspect the version in the web interface for FlexNet Manager Suite; or you can dig deeper (including programmatically). As expected, the methods for digging deeper depend on the operating system on the managed device.

### In the web interface

If the managed device has already reported inventory, you can inspect the results as follows:

1. Navigate to **Discovery & Inventory > All Inventory** (in the **Inventory** group).
2. On the **All Inventory** page, use searching or filters to find the managed device of interest, and click its **Name** to open its properties.
3. Select the **Applications** tab, and filter for `Inventory Manager Agent`. The installed version appears in the **Version** column.




---

**Tip** • Ignore the `FlexNet Agent` listing, which is for a separate entity.

The version shown is the friendly name (such as "2016 R2"). If you need to know major/minor version numbering, click through the **Product** value to reach the application properties, select the **Evidence** tab (with its default presentation for **Installer** evidence), and inspect the **Version** values for the evidence. Although typically wildcarded to match multiple builds, these show the major and minor numbering of the installer evidence that can produce the application record.

### For managed devices on Microsoft Windows

Microsoft Windows Installer does not offer a simple command to list the version of an installed package.

However, you can check the registry. The installed version of the FlexNet inventory agent is stored in

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ManageSoft\ETCPVersion
```



(on 64-bit machines) or

```
HKEY_LOCAL_MACHINE\SOFTWARE\ManageSoft\ETCPVersion
```

(on 32-bit machines). This contains a string version showing major, minor, and build numbers (such as 11.01.19).

## For managed devices on UNIX and OS X systems

While these platforms do not have a registry, you can still inspect registry preferences in `/var/opt/managesoft/etc/config.ini`. (This is populated from the bootstrap file `/etc/managesoft.ini` during installation.) Look for the following section in `config.ini`:

```
[ManageSoft]
InstallDir=/opt/managesoft
ETCPInstallDir=/opt/managesoft
ETCPVersion=11.0.1
```



**Note** • The format of the version number is different on UNIX-like systems than on Windows. Versions internally are numbered with four integers, *a*, *b*, *c*, and *d*. On Microsoft Windows, these are presented as *a.b.c.d* (*b* and *c* are not separated) but on UNIX-like systems, they are presented as *a.b.c* (with the final integer unavailable). This variation means that the example given here for UNIX-like systems is the same release number as shown above for Microsoft Windows.

This `config.ini` file remains available even after the FlexNet inventory agent is uninstalled (as noted in *Uninstalling FlexNet inventory agent from UNIX* on page 30).

In addition, on UNIX-like systems, there are command lines available for checking the installed version of a package. You can use the following command lines on each of the platforms:

Platform	Version inspection command line
AIX	<code>lsllpp -l managesoft.rte</code>
Debian Linux x86 and x86_64	<code>dpkg-query -W managesoft</code>
HP-UX	<code>swlist managesoft</code>
Mac OS X	<code>grep "ManageSoft " /Library/Receipts/ManageSoft.pkg/Contents/Info.plist</code>
RPM Linux x86 and x86_64	<code>rpm --query managesoft</code>
Solaris x86 and SPARC	<code>pkginfo -l ManageSoft</code>

---

# 3

---

## Command Lines

### Topics:

- *mgspolicy Command Line*
- *ndlaunch Command Line*
- *ndschedag Command Line*
- *ndtrack Command Line*
- *ndupload Command Line*

To assist with the preparation of custom solutions, this section summarizes the command lines for the various code entities that together function as the FlexNet inventory agent. The mapping of descriptive titles to executable names is:

- FlexNet inventory agent is `ndtrack`
- Installation agent is `ndlaunch`
- Policy agent is `mgspolicy`
- Schedule agent is `ndschedag`
- Upload agent (or uploader) is `ndupload`.

The application usage agent does not have any command line available (on Windows, it runs as a plug-in, and on UNIX-like systems, it runs as a service or daemon). However, there are still a few preferences relevant to the application usage agent. Since there is no command line, these can only be adjusted by directly editing the registry (or the `config.ini` file on UNIX-like platforms). Relevant items are listed in the chapter *Preferences* on page 50.

# mgspolicy Command Line

Command line reference for the policy agent.

The policy agent is responsible for checking and applying policy on the managed device.

Because server-side policy merging is run on the inventory beacons, the policy agent simply invokes the installation agent. The installation agent retrieves the resulting policy files from the appropriate inventory beacon, and installs the required packages. For command line details for the installation agent, see *ndlaunch Command Line* on page 37.

During adoption, the policy agent attempts to collect policy and its operating schedule from the bootstrapping inventory beacon (which has collected these from the central application server). If for any reason this collection does not succeed, then until the FlexNet inventory agent receives its operating schedule, the policy agent runs:

- On Windows platforms, each time the managed device reboots
- On UNIX-like platforms, once each day at a random time between 0800 and 2300 (local time on the managed device).


Once the operating schedule is received from the bootstrapping inventory beacon, this behavior is discontinued, and the policy agent is triggered regularly to check for changes, and apply the policy when amended. The policy agent uses preference settings on the managed device to determine the appropriate command line parameters.

## Synopses

`mgspolicy [options ...]`

Options:

`-o tag=value`  
`-s source`  
`-t User | Machine`

<code>-o</code>	<code>tag=value</code>	<p>Each instance over-rides the specified preference for the policy agent. Each parameter set at the command line must be accompanied by its own <code>-o</code> flag. Do not repeat any individual tag within the command line. Possible tags are listed below.</p>  <p><b>Note</b> • <i>In addition to the tags listed below, you may also set any preferences for the inventory agent (<code>ndtrack</code>) on the command line. When the policy agent launches <code>ndtrack</code>, these preferences will be passed to the <code>ndtrack</code> command line. When you set policy agent logging preferences on the command line, these are also passed to <code>ndtrack</code>, and the log file becomes a single combined file for both <code>mgspolicy</code> and <code>ndtrack</code> activities. For details, see <i>ndtrack Command Line</i> on page 44.</i></p>
-----------------	------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>-s</b>	<i>source</i>	Identifies the source location of server-side merged policy (.npl) files on the inventory beacon. If you do not specify a source, the policy agent uses the last known location (set in the <code>PolicyServerURL</code> preference). Example:  <pre>-s "http://beacon.mydomain.com/ManageSoftDL/Policies/Merged/Machine/deviceHostname.npl"</pre>
<b>-t</b>	User   Machine	Identifies the type of policy to be merged. This can be user or machine policy.

## Options

Possible tags for use in a command line with the `-o` options are shown below.



**Note** • Most preferences for the policy agent are not passed through to the installation agent when it is called. Either list the preferences in the `Common` registry key, or use separate preferences for the installation agent.

- *LogFile* (policy agent) on page 90
- *LogFileOld* (policy agent) on page 92
- *LogFileSize* (policy agent) on page 94
- *LogLevel* (policy agent) on page 96
- *LogModules* (policy agent) on page 98
- *MachinePolicy* on page 102
- *MachinePolicyDirectory* on page 103
- *MachinePolicyPackageDirectory* on page 104
- *PolicyServerURL* on page 113.

In addition, the following preferences set in the registry on the managed device influence the behavior of the policy agent:

- *DefaultSchedulePath* on page 61
- *InstallDefaultSchedule* on page 84
- *LauncherCommandLine* on page 88
- *StrictInstall* on page 128.

The following preferences available in earlier releases are now deprecated, and should not be used:

- `AllowedPkgSubtypes`
- `AutoDetectDC`
- `MinimumDCSpeed`
- `PolicySource`

- UserPolicy
- UserPolicyDirectory
- UserPolicyPackageDirectory.

## ndlaunch Command Line

Command line reference for the installation agent.

The installation agent on managed devices is now limited to installing various packages required by the FlexNet inventory agent, including updating the FlexNet inventory agent itself.


### Synopses

```
ndlaunch [-f preferenceFile | -o tag=value] [catalogFile]
```

```
ndlaunch -a PackageName [-o tag=value ...]
```

```
ndlaunch -d PackageName
```

```
ndlaunch -r urlOfCatalogFile [-o tag=value ...]
```

<b>-a</b>	<i>PackageName</i>	<p>Updates the named package from the last location from which it was downloaded and installed. If no update is necessary or available, runs the currently installed version (if execution is required). This parameter is equivalent to using both <code>-p</code> and <code>-r</code>. This is the mechanism by which the FlexNet inventory agent checks for any changes to its policy. Examples:</p> <pre>%BASE\ndlaunch -a mypolicy</pre> <p> <b>Tip</b> • Use the <code>%BASE</code> path substitution variable to include the full path to the FlexNet inventory agent installation.</p>
	<i>catalogFile</i>	<p>The full pathname to a locally available FlexNet catalog. If specified, the <i>catalogFile</i> filename must appear last on the command line. FlexNet catalogs have the file extension <code>.osd</code>.</p>
<b>-d</b>	<i>PackageName</i>	<p>Delete the application identified by the package name. You can identify the package name of any installed application by looking in the installation agent's application cache; each installed application has its own folder (named for the associated package) in this location.</p> <p><b>Warning</b> • Do not use this parameter to delete standard packages distributed by FlexNet Manager Suite through the inventory beacons.</p>
<b>-f</b>	<i>preferenceFile</i>	<p>Advanced parameter, recommended for use by experienced administrators only.</p> <p>Sets the value of a symbol used in a catalog by passing it the contents of a preference file using the installation agent's command line. The installation</p>

		<p>agent reads the list of symbol values specified in <i>preferenceFile</i>, and substitutes them for the symbols used in the catalog.</p> <p><i>preferenceFile</i> uses the standard <code>.ini</code> file format. For example:</p> <pre>[Symbols] symbolName="value" symbolName2="value2" symbolName3="value3"</pre> <p>where:</p> <ul style="list-style-type: none"> <li>• <code>symbolName</code> is the name of a symbol for which you wish to define a value</li> <li>• <code>value</code> is the value assigned to the symbol.</li> </ul>
<code>-o</code>	<code>tag=value</code>	<p>Each instance over-rides the specified preference for the installation agent. Each parameter set at the command line must be accompanied by its own <code>-o</code> flag. Do not repeat any individual tag within the command line. Possible tags are listed below. If your command line will exceed the length limit of your operating system's command line, use the <code>-f</code> parameter instead.</p>
<code>-r</code>	<code>urlOfCatalogFile</code>	<p>Check to see if an updated catalog file exists at the specified URL. If so, download and install it. Example:</p> <pre>%BASE\ndlaunch -r http://www.mysite.org/myapp.osd</pre>

## Options

Possible tags for use with the `-o` options are:

- *AllowedPkgTypes* on page 54
- *http\_proxy* on page 75
- *InstallProfile* on page 85
- *LogFile (installation agent)* on page 89
- *LogFileOld (installation agent)* on page 91
- *LogFileSize (installation agent)* on page 93
- *no\_proxy* on page 110
- *StrictInstall* on page 128
- *UserInteractionLevel (installation agent)* on page 140

In previous releases, `ndlaunch` had greater functionality and many more preferences. The following are now deprecated:

- `AddRemove`
- `AlertExecute`
- `AllowByteLevel`

- AllowedPkgSubtypes
- AllowPeerToPeer
- AllowRebootIfLocked
- AllowRebootIfServer
- AllowTimeoutIfLocked
- AlwaysDisplayReboot
- ApplyPolicy
- AskAboutDependencies
- AutoAlertExecute
- AutoDetectDC
- AutoPromptOnInstallCompletion
- AutoPromptOnUnInstallCompletion
- AutoRedundancy
- BrandARP
- CacheDir
- CacheDirectory
- CachedVersion
- CheckCatalogDigest
- CheckCertificateRevocation
- CheckFileDigest
- CheckRegistry
- CmdLineOverrides
- CompressMSA
- ConfirmSharedFileRemoval
- ConnectionAttempts
- DetectApplicationVersionConflicts
- DisplayAllAuthcode
- DownloadOnly
- DownloadPolicy
- EventLogs (installation agent)
- ForceCOMRegistration

- ForceReboot
- ForceRebootIfLocked
- ForceValidSignature
- GlobalConfigSource
- IgnoreConnectionWindows
- InstallationStatusRefreshPeriod
- InstallerARPMModify
- InstallerARPRemove
- LogInstallCheck
- LogInstallFail
- LogInstallPass
- LogLevel (installation agent)
- LogModules (installation agent)
- LogNotRequiredCheck
- LogUninstallFail
- LogUninstallPass
- LowProfile (installation agent)
- MaximumNetworkRetryPeriod
- MgsMsiArgs
- MinimumDCSpeed
- MsiAllPkgBaseURL
- MsiBaseURL
- MsiBaseURLList
- MsiCustomBaseURL
- MsiInstallArgs
- MsiReinstallFeatures
- MsiReinstallModeLevel
- MsiRepair
- MsiRepairLevel
- MsiSourceListUpdate
- MsiSourceLocation



- MsiUILevel
- MsiUninstallArgs
- MsiUserDomain
- NetworkHighSpeed (installation agent)
- NetworkHighSpeed (installation agent)
- NetworkHighUsageLowerLimit
- NetworkHighUsageUpperLimit
- NetworkLowUsage
- NetworkLowUsageLowerLimit
- NetworkLowUsageUpperLimit
- NetworkMaxByteLevelSpeed
- NetworkMaxRate (installation agent)
- NetworkMinSpeed (installation agent)
- NetworkRetries
- NetworkRetryPeriodIncrement
- NetworkSense (installation agent)
- NetworkTimeout (installation agent)
- NoStage
- PkgCacheDirectory
- PlatformSpecificPackages
- PolicyPackageRefreshPeriod
- PolicyRefreshPeriod
- PolicyServerPriority
- PostponeByDefault
- PostponementQueryBefore
- PostponeUserInteractionLevel
- PrivateAppAccess
- PromptOnCOMRegFailures
- PromptOnInstallCompletion
- PromptOnUnInstallCompletion
- PropagatePkgChanged

- PublicAppAccess
- QuietUntilUpdate
- RebootCmdLine
- RebootIfRequired
- RebootContinueAfterCmdFailure
- RebootPostCommand
- RebootPreCommand
- RebootPromptCycles
- RebootPromptWait
- ReInstallRequiresVersionChange
- RenotifyTimeout
- RotateConnectionsOnFailedConnectionAttempt
- RunAllUsersInstallAsUser
- SaveAllUserSymbols
- SecurityAnalysisFile
- ServiceConnectTimeout
- StagedInstall
- StagedOnly
- StageInactivePackages
- SupplyWorstCaseReturnValue
- UITimeoutWait
- UploadEventLogs
- UseLastUpdateLocation
- UseTrustDatabase
- VerifyCatalogSigned
- VerifyDownload
- VerifyFilesSigned
- VerifyTrustOrSign
- VirusScan
- VirusScanCommand

# ndschedag Command Line

Command line reference for the schedule agent

Schedules are created on the central application server, and distributed to inventory beacons from which they are retrieved by managed devices. Managed devices then install the schedules, which are later read by the managed device's scheduling agent.



**Tip** • Listings of scheduled tasks are contextual to the privileges of the credentials running the schedule query. To see a full list of scheduled activities, be certain to choose the **Run as administrator** option when starting the Windows command window. Even if you are already logged in using an account that is an administrator on the computer, use this option, as Microsoft Windows by default runs some commands with lower privilege levels.

## Synopsis

ndschedag [options...]

Options:

- A
- e (non-Windows devices only)
- o tag = value
- e eventId (non-Windows devices only)

<b>-A</b>		Run all events of type <code>Schedule Update</code> in the installed schedule.
<b>-e</b>		Non-Windows devices only. Display a list of all scheduled events and their next run time.
<b>-o</b>	tag=value	Advanced parameter, recommended for use by experienced administrators only. Each instance over-rides the specified preference for the schedule agent. Each parameter set at the command line must be accompanied by its own -o flag. Do not repeat any individual tag within the command line. Possible tags are listed below.
<b>-x</b>	eventId	Available on non-Windows devices only. Run the specified event.  To identify the <code>eventId</code> , open the file <code>/var/opt/managesoft/scheduler/schedules/sched.nds</code> with a text editor. Find the line that defines the relevant event in the file, and copy the <code>eventId</code> associated with the event. Paste the <code>eventId</code> on the command line.

### Command line example

The following command runs the event with the ID {25ec6994-8f40-4985-bf4c-d57566c707c3}:

```
ndschedag -x "{25ec6994-8f40-4985-bf4c-d57566c707c3}"
```

## Options

Possible tags for use with the `-o` options are:

- *Catchup* on page 56
- *DisablePeriod* on page 65 (Windows devices only)
- *MachineScheduleDirectory* on page 105
- *ndsensNetType* on page 109 (Windows devices only)
- *OnConnect* on page 110 (Windows devices only)
- *ScheduleType* on page 122 (Windows devices only)
- *Startup* on page 127
- *UIMode* on page 129 (Windows devices only)
- *UserScheduleDirectory* on page 143 (Windows devices only)

# ndtrack Command Line

Command line reference for the FlexNet inventory agent.

## Synopsis

```
ndtrack.exe [options...]
```

Options:

```
-o tag = value
-t User | Machine
```

These parameters override the default FlexNet inventory agent properties.

Do not repeat any individual tag within the command line. Possible tags are listed below (and see also the notes). Enclose values in double quotation marks if the values include spaces; otherwise, double quotation marks are optional. Special characters (double quotation marks, backslash) must be escaped with a backslash.

If neither the `-t` option nor the `-o InventoryType` option is specified, the type of inventory collected depends on the account running the FlexNet inventory agent:

- For the **LocalSystem** account, a computer (machine) inventory is collected
- For all other accounts, a user inventory is collected.




---

**Tip** • The `-t` option available for the installed FlexNet inventory agent is not supported for the lightweight FlexNet Inventory Scanner.

## Return codes

FlexNet inventory agent returns a zero on success. If you receive a non-zero return code, check the log file. Details of the log file may also be configured with command-line options, as described in the section on *Preferences* on page 50.

### Command line examples

This example collects a computer inventory and stores it locally (on the computer device where the FlexNet inventory agent is executing) for upload by a separate system:

```
ndtrack.exe
-t Machine
-o MachineZeroTouchDirectory="local-folder"
-o Upload=False
```

This example collects user-based inventory and uploads it to an inventory beacon:

```
ndtrack.exe
-t User
-o UploadLocation="http://InventoryBeacon/ManageSoftRL"
```

## Notes

1. Default values only apply if the parameter is not specified.
2. If no drive is indicated when specifying directory paths, the FlexNet inventory agent applies the path to every fixed drive of the local computer.
3. The FlexNet inventory agent supports all name/value combinations, although a warning is logged if a preference is used that does not appear in the list above.
4. A value can symbolically refer to another preference by enclosing its name thus:  $\$(preferenceName)$ . References can contain further references.

Example: The command

```
ndtrack.exe -o IncludeDirectory=$(WinDirectory)
```

includes the Windows directory in the scan. References are resolved after all preferences are loaded so there are no ordering issues.

5. Semicolon or comma-separated values are the only method for defining multiple values in the FlexNet inventory agent. (Do not repeat any individual tag within the command line.)

Example: The command

```
ndtrack.exe -o IncludeDirectory=C:\\;D:\\;E:\\
```

will scan the computer's C:, D:, and E: drives if they are fixed (hard) disks, but not if they are CD-ROM drives or logical drives (mapped to network locations).

6. To activate all logging, use the default preferences as follows:

```
Logging=true
LogLevel=A-Z
LogFile=<file>
```

```
LogModules=default
```

## Options

Possible tags for use with the `-o` options are:

- *DateTimeFormat* on page 60
- *Domain* on page 65
- *ExcludeDirectory* on page 68
- *ExcludeExtension* on page 69
- *ExcludeFile* on page 70
- *ExcludeMD5* on page 71
- *GenerateMD5* on page 72
- *Hardware* on page 73
- *IncludeDirectory* on page 76
- *IncludeExecutables* on page 77
- *IncludeExtension* on page 78
- *IncludeFile* on page 79
- *IncludeMachineInventory* on page 80
- *IncludeMD5* on page 81
- *IncludeRegistryKey* on page 82
- *IncludeUserInventory* on page 83
- *InventoryFile* on page 86
- *InventoryScriptsDir* on page 86
- *InventoryType* on page 87
- *LogFile (inventory agent)* on page 90
- *LogLevel (inventory agent)* on page 94
- *LogModules (inventory agent)* on page 97
- *LowProfile (inventory agent)* on page 99
- *MachineInventoryDirectory* on page 101
- *MachineName* on page 102
- *MachineZeroTouchDirectory* on page 106
- *MSI* on page 107

- *OracleInventoryUser* on page 111 (UNIX platforms only)
- *PerformOracleInventory* on page 113
- *ProgramFiles*, *ProgramFilesX86Folder*, *ProgramFilesX64Folder* on page 116
- *Recurse* on page 119
- *RunInventoryScripts* on page 120
- *ShowIcon* (*inventory agent*) on page 124
- *SysDirectory* on page 129
- *Upload* on page 130
- *UploadLocation* on page 131
- *UserHardware* on page 139
- *UserInteractionLevel* (*inventory agent*) on page 141
- *UserInventoryDirectory* on page 142
- *UserZeroTouchDirectory* on page 144
- *VersionInfo* on page 145
- *WinDirectory* on page 146
- *WMI* on page 147
- *WMIConfigFile* on page 147.

## ndupload Command Line

Command line reference for the uploader.

The uploader runs on managed devices and inventory beacons. It allows you to transfer event logs, inventories and other files from managed devices to FTP or local file servers.



---

**Tip** • Only HTTP and HTTPS protocols are supported by inventory beacons. File and FTP protocols are available for alternative upload arrangements.

The uploader can transfer any file to a specified URL. If an FTP URL is supplied, then a username and password must also be given.

The uploader deletes files from managed devices after they have been successfully uploaded. Files are not removed if the upload fails.

The file path supplied to the uploader can contain wildcards, so that multiple files of a similar type can be uploaded with a single command.

## Synopsis

ndupload.exe [options...]

Options:

-a  
 -f *path\and\filename.ext*  
 -o *tag = value*

-a		This is the same as -o UploadRule and means "upload all file types". If you run ndupload with no command line parameters, this is the default behavior.
-f	<i>path\and\filename</i>	Identifies the file to upload. This is the same as -o SourceFile.
-o	<i>tag=value</i>	Each instance over-rides the specified preference for the uploader. Each parameter set at the command line must be accompanied by its own -o flag. Do not repeat any individual tag within the command line. Possible tags are listed below.

## Return codes

The uploader returns a zero on success. If you receive a non-zero return code, check the log file. Details of the log file may also be configured with command-line options, as described in the section on *Preferences* on page 50.

### Command line examples

The following command uploads the file `file.txt` to the FTP location `ftp://server/dir1/dir2` using the login name `user1` and the password `abc123`:

```
ndupload -f file.txt -o UploadLocation=ftp://server/dir1/dir2
-o UploadUser=user1 -o UploadPassword=abc123
```

The following command uploads the file `file.txt` to the mapped drive `f:/dir1/dir2`:

```
ndupload -f file.txt -o UploadLocation=file:///f:/dir1/dir2
```

This example uploads the inventory file `myInventory.ndi` to an inventory beacon:

```
ndupload -f myInventory.ndi -o UploadLocation="http://InventoryBeacon/
ManageSoftRL"
```

## Options

Possible tags for use with the -o options are:

- *SourceFile* on page 125
- *SourceRemove* on page 126
- *TenantUID*, to override the value set in the registry (multi-tenant mode only)
- *UploadLocation* on page 131
- *UploadPassword* on page 132



- *UploadProxy* on page 133
- *UploadRule* on page 134
- *UploadType* on page 136
- *UploadUser* on page 137.

---

# 4

---

## Preferences

### Topics:

- *[Registry] Explained*
- *AllowedPkgTypes*
- *AutoPriority*
- *Catchup*
- *CheckCertificateRevocation*
- *CheckServerCertificate*
- *CommonAppDataFolder*
- *DateTimeFormat*
- *DefaultSchedulePath*
- *Directory*
- *Disabled (application usage agent)*
- *Disabled (schedule agent)*
- *DisablePeriod*
- *Domain*
- *DownloadSettings*
- *ExcludeDirectory*
- *ExcludeExtension*
- *ExcludeFile*
- *ExcludeMD5*
- *GenerateMD5*
- *Hardware*

This section contains an alphabetic listing of some useful preferences you can declare on agent command lines (for example, during testing or in scheduled tasks) or store in the registry for run-time assessment.

- *Host*
- *http\_proxy*
- *https\_proxy*
- *IncludeDirectory*
- *IncludeExecutables*
- *IncludeExtension*
- *IncludeFile*
- *IncludeMachineInventory*
- *IncludeMD5*
- *IncludeRegistryKey*
- *IncludeUserInventory*
- *InstallDefaultSchedule*
- *InstallProfile*
- *InventoryFile*
- *InventoryScriptsDir*
- *InventoryType*
- *LauncherCommandLine*
- *LogFile (installation agent)*
- *LogFile (inventory agent)*
- *LogFile (policy agent)*
- *LogFileOld (installation agent)*
- *LogFileOld (policy agent)*
- *LogFileSize (installation agent)*
- *LogFileSize (policy agent)*
- *LogLevel (inventory agent)*
- *LogLevel (policy agent)*
- *LogModules (inventory agent)*
- *LogModules (policy agent)*
- *LowProfile (inventory agent)*
- *MachineID*
- *MachineInventoryDirectory*

- *MachineName*
- *MachinePolicy*
- *MachinePolicyDirectory*
- *MachinePolicyPackageDirectory*
- *MachineScheduleDirectory*
- *MachineZeroTouchDirectory*
- *MSI*
- *Name*
- *ndsensNetType*
- *no\_proxy*
- *OnConnect*
- *OracleInventoryUser*
- *Password*
- *PerformOracleInventory*
- *PolicyServerURL*
- *Port*
- *Priority*
- *ProgramFiles,*  
*ProgramFilesX86Folder,*  
*ProgramFilesX64Folder*
- *Protocol*
- *Proxy*
- *Recurse*
- *RunInventoryScripts*
- *SelectorAlgorithm*
- *ScheduleType*
- *ShowIcon (installation agent)*
- *ShowIcon (inventory agent)*
- *SourceFile*
- *SourceRemove*
- *Startup*
- *StrictInstall*

- *SysDirectory*
- *UIMode*
- *Upload*
- *UploadLocation*
- *UploadPassword*
- *UploadProxy*
- *UploadRule*
- *UploadSettings*
- *UploadType*
- *UploadUser*
- *User*
- *UserHardware*
- *UserInteractionLevel (installation agent)*
- *UserInteractionLevel (inventory agent)*
- *UserInventoryDirectory*
- *UserScheduleDirectory*
- *UserZeroTouchDirectory*
- *VersionInfo*
- *WinDirectory*
- *WMI*
- *WMIconfigFile*

## [Registry] Explained

The following definitions of computer preferences use the placeholder `[Registry]`. This text represents the location of all relevant registry entries in the registry:

- On Windows servers, inventory beacons, and managed devices, agent registry entries on 32-bit operating systems are usually stored under the key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ManageSoft Corp\
```

For 64-bit operating systems, the normal key is:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ManageSoft Corp\
```

For these platforms, the `[Registry]` placeholder should be interpreted as the appropriate one of these keys.



**Tip** • Equivalent keys in the `HKEY_CURRENT_USER` hive show the 32-bit format for simplicity. On 64-bit devices, please modify the path appropriately.

- On non-Windows managed devices, registry entries are stored in the `/var/opt/ managesoft/etc/ config.ini` file. Within this file, registry keys are shown in square brackets. The lines below each key show the registry entries set under that key. For example, to set the type of inventory to be collected as a 'registry' preference on UNIX-like managed devices:

```
[ManageSoft\Tracker\CurrentVersion]
InventoryType=User
```

In other words, on UNIX-like devices, the documentation placeholder `[Registry]\` should be taken to mean "look in the `/var/opt/ managesoft/etc/config.ini` file for the following key".

### Absent keys

Not all registry keys cited in the following preferences exist by default. These are usually marked "Code internals, or manual configuration". If the registry key does not exist, or if it exists but has no value, the default value is provided by the FlexNet inventory agent (without modifying the registry). However, if you wish to override the default value for the preference, you should manually create the registry key shown, and populate it with your alternate value.

## AllowedPkgTypes

Command line | Registry

`AllowedPkgTypes` is a semi-colon (;) separated list of installable package types. Whereas the policy for the FlexNet inventory agent references all the packages (of all kinds) that are available for this managed device, the `AllowedPkgTypes` setting acts as a filter, so that only the listed types are included in policy updates. An empty value allows installation of all package types. Clearly this preference is intended for internal use, but may on rare occasions be useful for testing.

The package types available are:

- `Package` (normal FlexNet packages)
- `ClientSettings` (only packages containing fail-over settings)
- `ClientConfiguration` (only packages containing managed device settings)
- `Schedule` (only packages containing details of scheduled events).

## Values

<b>Values / range</b>	<code>Package, ClientSettings, ClientConfiguration, Schedule</code>
<b>Default value</b>	(No default.)
<b>Example values</b>	<code>ClientSettings;Schedule</code>

## Command line

<b>Tool</b>	Installation agent ( <code>ndlaunch</code> ), policy agent ( <code>mgspolicy</code> )
<b>Example</b>	<code>-o AllowedPkgTypes="ClientConfiguration"</code>

## Registry

<b>Installed by</b>	Code internals
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Common</code>

# AutoPriority

## Registry

`AutoPriority` determines whether an inventory beacon has a priority calculated at the time of download or upload (`True`), or whether it is fixed at the value declared in the `Priority` registry key (`False`).

Priorities are used to determine the order in which connections to reporting locations or distribution locations are attempted. If `AutoPriority` does not exist under a download or upload location's registry settings, the appropriate agent assumes the value `True`. If you want to use fixed priorities, you must create the `AutoPriority` registry key if it does not exist, and assign to it the value `False`.

## Values

<b>Values / range</b>	Boolean (True or False)
<b>Default value</b>	True

## Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
<b>User preference</b>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>
<b>Computer preference</b>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>

# Catchup

Command line | Registry

Catchup controls the behavior of events occurring on the managed device. It determines when the FlexNet inventory agent will catch up missed scheduled events. Possible actions include:

- **Always:** the agent always attempts to catch up on missed events
- **Never:** the agent never attempts to catch up on missed events
- **Query:** the agent queries the user regarding catch up on missed events (available for Windows devices only).

## Values

<b>Values / range</b>	Always, Never, Query
<b>Default value</b>	No default in registry; default behavior is <code>Never</code>
<b>Example values</b>	Always



## Command line

<b>Tool</b>	Scheduling agent (ndschedag)
<b>Example</b>	<code>-o Catchup=Query</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Schedule Agent\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Schedule Agent\CurrentVersion

# CheckCertificateRevocation

Command line | Registry

When transferring data to or from an inventory beacon using the HTTPS protocol, a web server certificate is applied to the data being transferred.

When receiving web server certificates from servers, the appropriate agent checks the CA (certification authority) server to ensure that the certificates are not on the CRL (certificate revocation list). If an agent cannot check the CRL (for example, the CA server is firewalled and cannot be contacted), the system can stall. To avoid this, you can use the `CheckCertificateRevocation` preference to prevent agents from performing the CRL check.

You can set this as a common registry entry, so that the same behavior occurs across all agents, and you can override the common behavior by setting an overriding registry entry for any individual agent if required. By default, this preference is set so that all agents check the CRL.

## Values

<b>Values / range</b>	Boolean (True or False)
<b>Default value</b>	True
<b>Example values</b>	False

## Command line

<b>Tool</b>	All agents
<b>Example</b>	<code>-o CheckCertificateRevocation="False"</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common or HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft \<Agent>\CurrentVersion where <Agent> is the registry key for an individual agent
<b>Computer preference</b>	[Registry]\ManageSoft\Common or [Registry]\ManageSoft \<Agent>\CurrentVersion where <Agent> is the registry key for an individual agent

## CheckServerCertificate

Command line | Registry

When transferring data to or from an inventory beacon using the HTTPS protocol, a web server certificate is applied to the data being transferred. All component agents can (and by default do) validate the public certificates received from the inventory beacon against their local copy (on Windows, in the certificate store; and on UNIX in the PEM file).

If you wish, you can use the `CheckServerCertificate` preference to prevent agents from performing the certificate check. (Without this check, the certificate is ignored, and the HTTPS protocol provides only encryption as security on the transfer, without validating that the agent is contacting the correct inventory beacon server.)

You can set this as a common registry entry, so that the same behavior occurs across all agents; and you can override the common behavior by setting an overriding registry entry for any individual agent if required. By default, this preference is set so that all agents check the downloaded certification against the CA certificate.

### Values

<b>Values / range</b>	Boolean (True or False)
<b>Default value</b>	True
<b>Example values</b>	False

### Command line

<b>Tool</b>	All agents
<b>Example</b>	-o CheckServerCertificate="False"

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common or HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft \<Agent>\CurrentVersion where <Agent> is the registry key for an individual agent
<b>Computer preference</b>	[Registry]\ManageSoft\Common or [Registry]\ManageSoft \<Agent>\CurrentVersion where <Agent> is the registry key for an individual agent

## CommonAppDataFolder

Command line | Registry

CommonAppDataFolder provides the path to the folder in which application details are located. On Windows managed devices, these are application details for [ALL USERS]. This is a system variable which you can override in the registry or on the command line.

### Values

<b>Values / range</b>	Local directory path. Read-only preference.
<b>Default value</b>	<p>The default installation of Windows uses:</p> <pre>%ALLUSERSPROFILE%\Application Data</pre> <p>where %ALLUSERSPROFILE% defaults to:</p> <ul style="list-style-type: none"> <li>• On Windows 2000/XP: C:\Documents and Settings\All Users</li> <li>• On Windows Vista: C:\ProgramData</li> <li>• On Windows 7 and higher: C:\Users\Public</li> </ul> <p>For Macintosh and UNIX devices, the value is:</p> <pre>/var/opt/managesoft</pre>
<b>Example values</b>	<p>Windows:</p> <pre>C:\Users\Public\Application Data</pre> <p>UNIX-like systems:</p> <pre>/var/opt/bin</pre>

## Variable

<b>Defined:</b>	Predefined within Windows.
<b>Reference as:</b>	<code>\$(CommonAppDataFolder)</code>


# DateTimeFormat

Command line | Registry

`DateTimeFormat` sets the date/time format for all inventory agent activity.

**Warning** • *Internal use only: do not edit.*

## Values

<b>Values / range</b>	Date/time format definition string, based on the ANSI C function <code>strftime()</code> .
<b>Default value</b>	<code>%Y%m%dT%H%M%S</code>
<b>Example value</b>	<p><code>%I.%M.%S %p - %A, %d %B %Y</code></p> <p>This would result in a date format such as 10.30.05 am - Monday, 26 February 2010</p>  <p><b>Note</b> • <i>Since the <code>DateTimeFormat</code> string is also used as a file name, you may use only characters that are valid in file names across all platforms (in particular, you may not use a colon).</i></p>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o DateTimeFormat="%I.%M.%S %p - %A, %d %B %Y"</code>

## Registry

<b>Installed by</b>	Agent code, or manual configuration
---------------------	-------------------------------------

<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## DefaultSchedulePath

Command line | Registry

DefaultSchedulePath gives the URL or path to the default machine schedule that is installed by the policy agent before applying policy if InstallDefaultSchedule is True.

Usually this preference is used to point to a schedule, but it can be configured to point to any .osd file that the policy agent wants the installation agent to process before applying policy.

### Values

<b>Values / range</b>	Any valid directory path and filename
<b>Default value</b>	\$(DownloadRootURL)/Schedules/Default Machine Schedule/Default Machine Schedule.osd
<b>Example values</b>	C:\temp\debugSchedule.osd

### Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	-o DefaultSchedulePath="C:\Schedules\Default Machine Schedule\Default Machine Schedule.osd"

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Policy Client\CurrentVersion

## Directory

Registry

Directory represents one of the following:

- For uploads, the folder representing the reporting location's upload folder
- For downloads, the location of the distribution location, within the specified host
- For trusted and excluded locations, the location of the distribution location, within the specified host.

## Values

<b>Values / range</b>	Folder location
<b>Default value</b>	(No default.)
<b>Example values</b>	/ManageSoftDL

## Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
<b>User preference</b>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\TrustedLocations\&lt;serverkey&gt;</pre> <p>For excluded locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\ExcludedLocations\&lt;serverkey&gt;</pre>
<b>Computer preference</b>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>[Registry]\ManageSoft\Launcher\CurrentVersion\TrustedLocations \&lt;serverkey&gt;</pre>

	For excluded locations:
	<code>[Registry]\ManageSoft\Launcher\CurrentVersion\ExcludedLocations \&lt;serverkey&gt;</code>

## Disabled (application usage agent)

Command line | Registry

`Disabled` specifies whether the application usage agent is inactive on this managed device. When set to `True`, the FlexNet inventory agent does not record application usage data. When set to `False`, the FlexNet inventory agent records application usage data.



**Note** • The schedule agent has a preference of the same name, used for a slightly different purpose.

### Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> )
<b>Default value</b>	<code>True</code>

### Command line

<b>Tool</b>	Application usage agent
<b>Example</b>	<code>-o Disabled=False</code>

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	Not available
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Usage Agent\CurrentVersion</code>

## Disabled (schedule agent)

Command line | Registry

`Disabled` determines whether the schedule agent is disabled on the managed device. By default, this value is set to `False`, enabling the schedule agent.

If a date and time in the future is specified, the schedule agent is disabled, and remains disabled until this date and time. Settings are not cleared as a given date and time passes, so that a date and time that has passed is also a possible value. Such past values are ignored, and have the same effect as the value `False` (that is, when the date is in the past, the schedule agent is enabled).

On Windows devices, management of this preference is different for user schedules and machine schedules:

- For user schedules, the user can run the schedule agent on the managed device. If users open a command line window, they can navigate to `$(ProgramFiles)\ManageSoft\Schedule Agent`, and run `ndschedag.exe`, without parameters. This presents a user interface where they can set the **Disabled** check box. When the schedule agent window closes (with the check box set), the schedule agent adds the number of seconds in the `DisablePeriod` preference to the current date and time, and writes the result to this `Disabled (schedule agent)` setting.
- Machine schedules cannot be disabled through any user interface, nor through command line interaction. If you need to temporarily disable a machine schedule, enter an appropriate date and time string in ISO format in the Computer preference registry setting listed in the **Registry** table below.

## Values

<b>Values / range</b>	Either <code>False</code> , or a date/time value in the ISO standard format <code>yyyymmddThhmmss</code> . If this date and time is in the future, it is the date and time at which the agent schedules will be re-enabled on this managed device.
<b>Default value</b>	<code>False</code>
<b>Example values</b>	<code>20160312T101520</code>

## Command line

<b>Tool</b>	Schedule agent ( <code>ndschedag</code> )
<b>Example</b>	<code>-o Disabled=20150823T143014</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Schedule Agent\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Schedule Agent\CurrentVersion</code>



# DisablePeriod

Command line | Registry

`DisablePeriod` is only applicable when `Disabled (schedule agent)` is set to `True`. It sets the number of seconds for which agent schedules are disabled when the user selects "Disabled" from the scheduling agent user interface on the managed device. (The scheduling agent user interface is only available on Windows devices.)

The default value is 3600 seconds (one hour). When set to 3600, agent schedules are automatically re-enabled after one hour.

## Values

<b>Values / range</b>	0 - 2147483647
<b>Default value</b>	3600
<b>Example values</b>	1800

## Command line

<b>Tool</b>	Scheduling agent (ndschedag)
<b>Example</b>	<code>-o DisablePeriod=600</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Schedule Agent\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Schedule Agent\CurrentVersion

# Domain

Command line | Registry

`Domain` contains the name of the domain of the local computing device, from which inventory is being gathered.

## Values

<b>Values / range</b>	Valid domain name. If no value is set, the FlexNet inventory agent uses the current local domain.
<b>Default value</b>	No default.
<b>Example value</b>	<code>MyDomain.com</code>

## Command line

<b>Tool</b>	Inventory agent
<b>Example</b>	<code>-o Domain=MyDomain.com</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Tracker\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>

# DownloadSettings

## Registry

`DownloadSettings` is a registry key (container) for several preferences that can control the download of data by the FlexNet inventory agent. These registry values are in sets that apply to a particular download location, for which reason the registry key must be completed with an identifier for the download location. The completed path leads to the relevant set of registry values, as shown below.

When configured by the failover list generated by an inventory beacon, the placeholder `<download_location>` takes the form of a GUID that identifies the download location on the particular inventory beacon (for example, `{EEFC121D-2BB3-4318-8014-8DDC339C7553}`).

For manual configuration, four name/value pairs must be specified, and others are optional. To omit an optional value, you may include the name and leave the value blank (as shown in the example below), or omit the name/value pair entirely. The values that may be set are:

- `Protocol` on page 117 – Mandatory. For download from an inventory beacon, this must be either `http` or `https`.
- `Name` on page 107

- *Directory* on page 61 – Mandatory. By default, the download location is called `ManageSoftDL`, but as this may have been renamed during installation, you need to check details of your implementation.
- *Host* on page 73 – Mandatory.
- *Port* on page 114 – Mandatory. As this has no default value, you must specify this setting to suit your environment (typically port 80 for HTTP and port 443 for HTTPS).
- *User* on page 138 – If omitted (or left with a blank value), anonymous authentication is used for downloads from this download location.
- *Password* on page 112 – Stores an encrypted copy of the password needed when Windows authentication is specified for the download.
- *Proxy* on page 118
- *Priority* on page 115
- *AutoPriority* on page 55.

## Values

<b>Values / range</b>	Requires a subkey that uniquely identifies the download location on an individual inventory beacon.
<b>Default value</b>	None.
<b>Example values</b>	<pre>[Registry]\ManageSoft\Common\DownloadSettings \{EEFC121D-2BB3-4318-8014-8DDC339C7553}   Protocol=http   Name=ss-server1 Download Location   Directory=ManageSoftDL   Host=ss-server1   Port=80   User=   Password=   Proxy=   Priority=10   AutoPriority=True</pre>

## Registry

<b>Installed by</b>	Download of failover settings, or manual configuration
<b>User preference</b>	<pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\DownloadSettings\ &lt;download_location&gt;</pre>
<b>Computer preference</b>	<pre>[Registry]\ManageSoft\Common\DownloadSettings \&lt;download_location&gt;</pre>

# ExcludeDirectory

Command line | Registry

`ExcludeDirectory` excludes a specified folder from inventory. If `Recurse` is `True`, then all subfolders are also excluded. This preference can accept multiple values in a list separated by either commas or semi-colons.

If a folder is identified in both the `ExcludeDirectory` and `IncludeDirectory` preferences, it is excluded. Exclusions (of the same thing) always override inclusions.

The value can symbolically refer to another preference by enclosing its name thus: `$(preferenceName)`. References can contain further references.



**Note** • Inclusions and exclusions can cover folders (and optionally their sub-folders), file name extensions, specific file names, and specific MD5 digest values. To resolve conflicting specifications, the specifications of folders provides a data set to which the following specifications are applied as filters, prioritized from lowest to highest as:

- File extension
- File name
- MD5 value.

For example, if file extension `exe` is included, filename `xcopy.exe` excluded, and MD5 value `123456...` (the MD5 for `xcopy.exe`) is included, then the inventory agent includes all files with extension `exe` except for all versions of `xcopy.exe` that do not have an MD5 value `123456...`

## Values

<b>Values / range</b>	Valid folders
<b>Default value</b>	(No default.)
<b>Example value</b>	<code>\$(WinDirectory)</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o ExcludeDirectory=C:\Temp</code>

## Registry

<b>Installed by</b>	Manual configuration
---------------------	----------------------

<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Tracker\ \CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## ExcludeExtension

Command line | Registry

`ExcludeExtension` excludes files with the specified extension from inventory, or excludes all files if set to the value `*` (asterisk). This filter is applied to the files within a folder included in inventory. This preference can accept multiple values, separated by commas or semicolons.

You can specify any valid file extensions (no leading dot required). Be aware that including `exe` in this list prevents tracking of executable files on Windows platforms.



**Note** • Inclusions and exclusions can cover folders (and optionally their sub-folders), file name extensions, specific file names, and specific MD5 digest values. To resolve conflicting specifications, the specifications of folders provides a data set to which the following specifications are applied as filters, prioritized from lowest to highest as:

- File extension
- File name
- MD5 value.

For example, if file extension `exe` is included, filename `xcopy.exe` excluded, and MD5 value `123456...` (the MD5 for `xcopy.exe`) is included, then the inventory agent includes all files with extension `exe` except for all versions of `xcopy.exe` that do not have an MD5 value `123456...`

### Values

<b>Values / range</b>	File extensions (no period required)
<b>Default value</b>	(No default.)
<b>Example value</b>	DLL

### Command line

<b>Tool</b>	Inventory agent
<b>Example</b>	<code>-o ExcludeExtension=dll</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## ExcludeFile

Command line | Registry

**ExcludeFile** excludes a specific file from inventory collection. This filter is applied to files within a folder included in inventory. This preference can accept multiple values, separated by commas or semi-colons.



**Note** • Inclusions and exclusions can cover folders (and optionally their sub-folders), file name extensions, specific file names, and specific MD5 digest values. To resolve conflicting specifications, the specifications of folders provides a data set to which the following specifications are applied as filters, prioritized from lowest to highest as:

- File extension
- File name
- MD5 value.

For example, if file extension *exe* is included, filename *xcopy.exe* excluded, and MD5 value *123456...* (the MD5 for *xcopy.exe*) is included, then the inventory agent includes all files with extension *exe* except for all versions of *xcopy.exe* that do not have an MD5 value *123456...*

## Values

<b>Values / range</b>	Valid file names.
<b>Default value</b>	(No default.)
<b>Example value</b>	myfile.txt

## Command line

<b>Tool</b>	Inventory agent
<b>Example</b>	-o ExcludeFile=myfile.txt

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## ExcludeMD5

Command line | Registry

For files within a folder included in inventory, the FlexNet inventory agent performs an MD5 checksum, and excludes any files from the inventory that have an MD5 value equal to any value stored in `ExcludeMD5`. This preference can accept multiple values, separated by commas or semicolons.



**Note** • Inclusions and exclusions can cover folders (and optionally their sub-folders), file name extensions, specific file names, and specific MD5 digest values. To resolve conflicting specifications, the specifications of folders provides a data set to which the following specifications are applied as filters, prioritized from lowest to highest as:

- File extension
- File name
- MD5 value.

For example, if file extension `exe` is included, filename `xcopy.exe` excluded, and MD5 value `123456...` (the MD5 for `xcopy.exe`) is included, then the inventory agent includes all files with extension `exe` except for all versions of `xcopy.exe` that do not have an MD5 value `123456...`

## Values

<b>Values / range</b>	Valid MD5 values
<b>Default value</b>	(No default.)
<b>Example value</b>	7d9d2440656fdb3645f6734465678c60

## Command line

<b>Tool</b>	Inventory agent
<b>Example</b>	<code>-o ExcludeMD5=7d9d2440656fdb3645f6734465678c60</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Tracker\ \CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# GenerateMD5

Command line | Registry

`GenerateMD5` specifies whether or not to calculate the MD5 digest of any file being tracked by the FlexNet inventory agent and include it with stored inventory data. MD5 digests, if included in inventory, are used as a key to uniquely identify files. If an MD5 is not present, files are identified by date and file size. MD5 digests are more reliable for this purpose, but be aware that calculating MD5 digests will degrade performance where many files are being tracked.

## Values

<b>Values / range</b>	Boolean (True or False).
<b>Default value</b>	False
<b>Example value</b>	True

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o GenerateMD5=True</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
---------------------	-----------------------------------------



<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## Hardware

Command line | Registry

**Hardware** allows you to track hardware either using Windows Management Instrumentation (WMI) or native APIs. If WMI is available, it is used for tracking. This preference is only effective when running in the machine context. (To track hardware in the user context, use `UserHardware`.) When set to `True`, allows the tracking of hardware inventory. When set to `False`, does not track hardware inventory.

### Values

<b>Values / range</b>	Boolean (true or false)
<b>Default value</b>	True
<b>Example value</b>	False

### Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o Hardware=False

### Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	N/A—use <code>UserHardware</code>
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## Host

Registry

**Host** identifies the inventory beacon on which the reporting location (uploads) or distribution location (downloads) is hosted.

## Values

<b>Values / range</b>	Valid host name
<b>Default value</b>	(No default.)
<b>Example values</b>	server.domain.com

## Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
<b>User preference</b>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\TrustedLocations\&lt;serverkey&gt;</pre> <p>For excluded locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\ExcludedLocations\&lt;serverkey&gt;</pre>
<b>Computer preference</b>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>[Registry]\ManageSoft\Launcher\CurrentVersion\TrustedLocations \&lt;serverkey&gt;</pre> <p>For excluded locations:</p> <pre>[Registry]\ManageSoft\Launcher\CurrentVersion\ExcludedLocations \&lt;serverkey&gt;</pre>

## http\_proxy

Command line | Registry

`http_proxy` gives the proxy settings for the installation agent (`ndlaunch`) when using the HTTP protocol. See also `no_proxy` on page 110.

### Values

<b>Values / range</b>	Any valid URL
<b>Default value</b>	Not to use a proxy.
<b>Example values</b>	<code>tmnis.com;tmnis.com.de</code>

### Command line

<b>Tool</b>	Installation agent ( <code>ndlaunch</code> )
<b>Example</b>	<code>-o http_proxy=tmnis.com;tmnis.com.de</code>

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent on a managed device (computer preference)
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Launcher\CurrentVersion</code>

## https\_proxy

Command line | Registry

`https_proxy` gives the proxy settings for the installation agent (`ndlaunch`) when using the HTTPS protocol. See also `no_proxy` on page 110.

### Values

<b>Values / range</b>	Any valid URL
-----------------------	---------------

<b>Default value</b>	Not to use a proxy.
<b>Example values</b>	tmnis.com;tmnis.com.de

### Command line

<b>Tool</b>	Installation agent (ndlaunch)
<b>Example</b>	-o http_proxy=tmnis.com;tmnis.com.de

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent on a managed device (computer preference)
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Launcher\CurrentVersion

## IncludeDirectory

Command line | Registry

`IncludeDirectory` includes a specified folder in inventory. If `Recurse` is `True`, then all subfolders are also included. When the value of this entry is set to "\", it means "include all folders". This preference can accept multiple values, separated by commas or semi-colons. If a folder is identified in both the `ExcludeDirectory` and `IncludeDirectory` preferences, it is excluded. Exclusions (of the same thing) always override inclusions.



**Note** • Inclusions and exclusions can cover folders (and optionally their sub-folders), file name extensions, specific file names, and specific MD5 digest values. To resolve conflicting specifications, the specifications of folders provides a data set to which the following specifications are applied as filters, prioritized from lowest to highest as:

- File extension
- File name
- MD5 value.

For example, if file extension `exe` is included, filename `xcopy.exe` excluded, and MD5 value `123456...` (the MD5 for `xcopy.exe`) is included, then the inventory agent includes all files with extension `exe` except for all versions of `xcopy.exe` that do not have an MD5 value `123456...`

## Values

<b>Values / range</b>	Valid folder
<b>Default value</b>	(No default.)
<b>Example value</b>	C:\Program Files

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o IncludeDirectory=C:\Temp

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent on a managed device (computer preference)
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# IncludeExecutables

Command line | Registry

`IncludeExecutables` extends the meaning of 'executable files' across platforms. If `IncludeDirectory` is blank (its default), this setting has no effect. However, when `IncludeDirectory` has any value:

- A setting of `False` means that only files with a file extension of `.exe` are included in executable files inventory (normally, this means only Windows executables)
- A setting of `True` means that (for files not already matched by other settings) the FlexNet inventory agent will report:
  - On Windows, files with an `exe` filename extension
  - On UNIX-like platforms, files with no filename extension and an execute bit set (in any of local, group, or universal scope in the file permission bits).

## Values

<b>Values / range</b>	Boolean (True or False)
<b>Default value</b>	True
<b>Example value</b>	False

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o IncludeExecutables=True

## Registry

<b>Installed by</b>	Installation (computer preference)
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# IncludeExtension

Command line | Registry

`IncludeExtension` includes files with the specified extension, or includes all files if this preference is set to the value `*`. This filter is applied within a folder included in inventory. This preference can accept multiple values, separated by commas or semi-colons. The dot separator for file extensions is not required.



**Note** • Inclusions and exclusions can cover folders (and optionally their sub-folders), file name extensions, specific file names, and specific MD5 digest values. To resolve conflicting specifications, the specifications of folders provides a data set to which the following specifications are applied as filters, prioritized from lowest to highest as:

- File extension
- File name
- MD5 value.

For example, if file extension `exe` is included, filename `xcopy.exe` excluded, and MD5 value `123456...` (the MD5 for `xcopy.exe`) is included, then the inventory agent includes all files with extension `exe` except for all versions of `xcopy.exe` that do not have an MD5 value `123456...`

## Values

<b>Values / range</b>	Any file extension
<b>Default value</b>	<code>sys;sys2;swtag;cmptag;lax;exe;</code>
<b>Example value</b>	<code>bat</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o IncludeExtension=bat</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>

# IncludeFile

Command line | Registry

`IncludeFile` searches for the specific file to report in inventory; but keep in mind that the search is constrained to folders that are specified as included in inventory. This preference can accept multiple values, separated with commas or semi-colons.



**Note** • Inclusions and exclusions can cover folders (and optionally their sub-folders), file name extensions, specific file names, and specific MD5 digest values. To resolve conflicting specifications, the specifications of folders provides a data set to which the following specifications are applied as filters, prioritized from lowest to highest as:

- File extension
- File name

- *MD5 value.*

For example, if file extension *exe* is included, filename *xcopy.exe* excluded, and MD5 value *123456...* (the MD5 for *xcopy.exe*) is included, then the inventory agent includes all files with extension *exe* except for all versions of *xcopy.exe* that do not have an MD5 value *123456...*

## Values

<b>Values / range</b>	Valid file
<b>Default value</b>	(No default.)
<b>Example value</b>	<code>xcopy.exe</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o IncludeFile=myfile.txt</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# IncludeMachineInventory

Command line | Registry

`IncludeMachineInventory` If `True`, FlexNet inventory agent performs a computer inventory including hardware and all user packages.

## Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> ).
-----------------------	------------------------------------------------------



<b>Default value</b>	True if running as <code>LocalSystem</code> or running a machine inventory on the command line.
<b>Example value</b>	<code>False</code>

### Command line

<b>Tool</b>	Inventory agent ( <code>ndtrack</code> )
<b>Example</b>	<code>-o IncludeMachineInventory=False</code>

### Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>

## IncludeMD5

Command line | Registry

`IncludeMD5` includes in the inventory report any files having the specific MD5 digest. This filter is applied to files within a folder included in inventory. This preference can accept multiple values, separated with commas or semicolons.



**Note** • Inclusions and exclusions can cover folders (and optionally their sub-folders), file name extensions, specific file names, and specific MD5 digest values. To resolve conflicting specifications, the specifications of folders provides a data set to which the following specifications are applied as filters, prioritized from lowest to highest as:

- File extension
- File name
- MD5 value.

For example, if file extension `exe` is included, filename `xcopy.exe` excluded, and MD5 value `123456...` (the MD5 for `xcopy.exe`) is included, then the inventory agent includes all files with extension `exe` except for all versions of `xcopy.exe` that do not have an MD5 value `123456...`

## Values

<b>Values / range</b>	
<b>Default value</b>	(No default.)
<b>Example value</b>	7d9d2440656fdb3645f6734465678c60

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o IncludeMD5=7d9d2440656fdb3645f6734465678c60

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# IncludeRegistryKey

Command line | Registry

Set `IncludeRegistryKey` to instruct the FlexNet inventory agent to return the contents of the specified registry keys or values.

In order to collect *all* values under a specified key, the key path specified must end with a trailing backslash. If the path specified corresponds to a key (rather than a registry value) but does not end with a trailing backslash, only the (Default) value (if it is set) for the specified key will be collected.



**Note** • *Default values in the registry are typically not set.*

For example:

- `HKLM\SOFTWARE\ManageSoft Corp\ManageSoft\` will track all values under the specified key (because it has a trailing backslash)
- `HKLM\SOFTWARE\ManageSoft Corp\ManageSoft` will only track the (Default) values under the specified key (where they exist).

When setting this preference, you can use:

- The \* wildcard to replace a key or value (including multiple times for different key elements in a single path)
- The abbreviations HKLM, HKCU, HKCR, HKU, HKCC. These will be automatically expanded to appropriate values.

## Values

<b>Values / range</b>	Valid registry key or value
<b>Default value</b>	If no value is specified, the FlexNet inventory agent uses  <pre>HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows \CurrentVersion \Uninstall\</pre>
<b>Example values</b>	Track all registry keys and values under HKEY_LOCAL_MACHINE\SOFTWARE:  <pre>HKLM\SOFTWARE\*\</pre> Track all values (only) under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft:  <pre>HKLM\SOFTWARE\Microsoft\*</pre> Use multiple wildcards:  <pre>HKLM\SOFTWARE\*\CurrentVersion\*\*</pre>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<pre>-o IncludeRegistryKey="HKEY_LOCAL_MACHINE\ SOFTWARE\Microsoft\Windows\CurrentVersion\ App Paths\"</pre>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# IncludeUserInventory

Command line | Registry

IncludeUserInventory, ff True, causes collection of user inventory.

## Values

<b>Values / range</b>	Boolean (True or False).
<b>Default value</b>	TRUE if running as user or running a user inventory (-t User on the command line)
<b>Example values</b>	False

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o IncludeUserInventory=False

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# InstallDefaultSchedule

Command line | Registry

When `InstallDefaultSchedule` is `True`, the policy agent invokes the installation agent to install the package specified by `DefaultSchedulePath` before applying policy. Used by automated processes for adoption of the managed device.

## Values

<b>Values / range</b>	Boolean (True or False)
<b>Default value</b>	False
<b>Example values</b>	True

## Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	<code>-o InstallDefaultSchedule=True</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Policy Client\CurrentVersion

# InstallProfile

Command line | Registry

`InstallProfile` is set to `Public` if the package is to be installed for All Users, or `Private` if it's to be installed for an individual user.

## Values

<b>Values / range</b>	Public, Private
<b>Default value</b>	(No default.)
<b>Example values</b>	Public

## Command line

<b>Tool</b>	Installation agent (ndlaunch)
<b>Example</b>	<code>-o InstallProfile=Public</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion

<b>Computer preference</b>	[Registry]\ManageSoft\Launcher\CurrentVersion
----------------------------	-----------------------------------------------

## InventoryFile

Command line | Registry

`InventoryFile` identifies the name of a local copy of the inventory file.

The name may consist of Windows properties that can be expanded to identify a value. For example, the default value `$(UserName) on $(MachineId).ndi` expands so that the name contains the account and machine ID related to the inventory run.

### Values

<b>Values / range</b>	*.ndi
<b>Default value</b>	<code>(UserName) on \$(MachineId).ndi</code>
<b>Example values</b>	<code>myComputer.ndi</code>

### Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o InventoryFile=myfile.ndi</code>

### Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## InventoryScriptsDir

Command line | Registry

`InventoryScriptsDir` gives the location of scripts to be run immediately before inventory data is uploaded through the inventory beacon. All scripts that exist in this location are run.

## Values

<b>Values / range</b>	String identifying an existing directory
<b>Default value</b>	<code>\$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft</code>
<b>Example values</b>	<code>C:\IT-Admin\</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o InventoryScriptDir=C:\data</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	Not available
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>

# InventoryType

Command line | Registry

`InventoryType` identifies the inventory type, either machine-based or user-based.

## Values

<b>Values / range</b>	String literals <code>User</code> or <code>Machine</code> .
<b>Default value</b>	<code>User</code> unless the account executing <code>ndtrack</code> is <code>LocalSystem</code> , in which case the default switches to <code>Machine</code> .
<b>Example values</b>	<code>Machine</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o InventoryType=User</code>

## Registry


<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# LauncherCommandLine

Command line | Registry

LauncherCommandLine specifies `ndlaunch` parameters to pass to the installation agent when applying policy information.

## Values

<b>Values / range</b>	Valid installation agent command line parameters (see <i>ndlaunch Command Line</i> on page 37).
<b>Default value</b>	(No default.)
<b>Example values</b>	<pre>-o UserInteractionLevel=Quiet</pre>  <p><b>Note</b> • The <code>-o</code> is required as part of the value to be appended to the command line of <code>ndlaunch</code>.</p>

## Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	<code>-o LauncherCommandLine="-o UserInteractionLevel=Quiet"</code>



## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Policy Client\CurrentVersion

# LogFile (installation agent)

Command line | Registry

LogFile gives the location of the log file (on the computer device where the FlexNet inventory agent is executing) when logging is enabled for `ndlaunch` (installation agent). All actions performed by `ndlaunch` are logged to this file. If you wish to use a log file located in a folder other than the default installation log folder, specify a full pathname.

## Values

<b>Values / range</b>	Valid log file name, including local and UNC network files.
<b>Default value</b>	<code>\$(TempDirectory)\ManageSoft\installation.log</code>
<b>Example values</b>	<code>C:\temp\MyInstallerLog.log</code>

## Command line

<b>Tool</b>	Policy agent ( <code>mgspolicy</code> )
<b>Example</b>	<code>-o Logfile=C:\temp\mylauncherlogfile.log</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Launcher\CurrentVersion

## LogFile (inventory agent)

Command line | Registry

`LogFile` gives the location of the log file (on the computer device where the FlexNet inventory agent is executing) when logging is enabled for `ndtrack`.

### Values

<b>Values / range</b>	Valid log file name.
<b>Default value</b>	<code>\$(TempDirectory)\ManageSoft\tracker.log</code>
<b>Example values</b>	<code>C:\temp\inventory.log</code>

### Command line

<b>Tool</b>	Inventory agent ( <code>ndtrack</code> )
<b>Example</b>	<code>-o LogFile=Inventory.log</code>

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>

## LogFile (policy agent)

Command line | Registry

`LogFile` gives the location of the log file (on the computer device where the FlexNet inventory agent is executing) when logging is enabled for `mgspolicy` (policy agent).

## Values

<b>Values / range</b>	Valid log file name, including local and UNC network files.
<b>Default value</b>	<code>\$(TempDirectory)\ManageSoft\policy.log</code>
<b>Example values</b>	<code>C:\temp\policy.log</code>

## Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	<code>-o LogFile=policy.log</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Policy Client\CurrentVersion</code>

# LogFileOld (installation agent)

Command line | Registry

When the main installation agent log file (defined in `LogFile (installation agent)`) reaches its maximum size (defined in `LogFileSize (installation agent)`), the file is renamed to the value in `LogFileOld`. This overwrites any existing `LogFileOld` file. A new log file is created, using the name defined in `LogFile (installation agent)`. This allows you to retain additional log information.

## Values

<b>Values / range</b>	Valid file name for local or UNC network files
<b>Default value</b>	<code>\$(TempDirectory)\ManageSoft\launcher.old.log</code>
<b>Example values</b>	<code>\$(TempDirectory)\installationold.log</code>

## Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	<code>-o LogFileOld=\$(TempDirectory)\installationold.log</code>

## Registry

<b>Installed by</b>	Installation of inventory beacon (computer preference)
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Launcher\CurrentVersion

# LogFileOld (policy agent)

Command line | Registry

When the main policy merging log file (defined in `LogFile (policy agent)`) reaches a particular size (defined in `LogFileSize (policy agent)`), the file is renamed to the value in `LogFileOld`. This overwrites any existing `LogFileOld` file. A new log file is created. This allows you to retain additional log information.

## Values

<b>Values / range</b>	Valid file name for local or UNC network files
<b>Default value</b>	<code>\$(TempDirectory)\ManageSoft\policy.old.log</code>
<b>Example values</b>	<code>\$(TempDirectory)\mgspolicyold.log</code>

## Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	<code>-o LogFileOld=\$(TempDirectory)\mgspolicyold.log</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
---------------------	-----------------------------------------

<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Policy Client\CurrentVersion

## LogFileSize (installation agent)

Command line | Registry

When the main installation agent log file (defined in `LogFile (installation agent)`) reaches the maximum size defined here in `LogFileSize (installation agent)`, the file is renamed (to the value in `LogFileOld (installation agent)`). This overwrites any existing `LogFileOld (installation agent)` file. A new log file is created, with the name defined in `LogFile (installation agent)`. This allows you to retain additional log information.

The size must be expressed as the number of bytes of the maximum allowed log size. If this entry is empty or set to zero, there is no log size limit and the size of the log file continues to grow.

### Values

<b>Values / range</b>	Number (number of bytes)
<b>Default value</b>	4000000
<b>Example values</b>	3126000 (3 Mb)

### Command line

<b>Tool</b>	Installation agent (ndlaunch)
<b>Example</b>	-o LogFileSize=1024000

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (adoption) sets the computer preference.
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Launcher\CurrentVersion

## LogFileSize (policy agent)

Command line | Registry

When the main policy merging log file (defined in `LogFile (policy agent)`) reaches the size defined in `LogFileSize`, the file is renamed (to the value in `LogFileOld`). This overwrites any existing `LogFileOld (policy agent)` file. A new log file is created. This allows you to retain additional log information.

The size must be expressed as the number of bytes of the maximum allowed log size. If this entry is empty or set to zero, there is no log size limit and the size of the log file continues to grow.

### Values

<b>Values / range</b>	Number (number of bytes)
<b>Default value</b>	4000000
<b>Example values</b>	3126000 (3 Mb)

### Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	-o LogFileSize=3126000

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (adoption)
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Policy Client\CurrentVersion

## LogLevel (inventory agent)

Command line | Registry

`LogLevel` sets the category of logging used by the FlexNet inventory agent. This logging is output to the file whose name is stored in the `LogFile` entry. Individual entries include the following:

- A - Schedule logging

- C - Callout logging
- G - General logging
- N - Network logging
- P - Preference logging
- S - Security logging
- U - User interface logging
- V - Verification logging

## Values

<b>Values / range</b>	A-z
<b>Default value</b>	A-z (this enables all logging)
<b>Example values</b>	G

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o LogLevel=G

## Registry

<b>Installed by</b>	Installation of the FlexNet inventory agent
<b>User preference</b>	In order of precedence: <ul style="list-style-type: none"> <li>• HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</li> <li>• HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common</li> </ul>
<b>Computer preference</b>	In order of precedence: <ul style="list-style-type: none"> <li>• [Registry]\ManageSoft\Tracker\CurrentVersion</li> <li>• [Registry]\ManageSoft\Common</li> </ul>

# LogLevel (policy agent)

Command line | Registry

`LogLevel` sets the level of logging used by the policy agent. This logging is output to the file whose name is stored in the `LogFile` (policy agent) entry. Individual entries include the following:

- A - Schedule logging
- C - Callout logging
- G - General logging
- N - Network logging
- P - Preference logging
- S - Security logging
- U - User interface logging
- V - Verification logging

## Values

<b>Values / range</b>	A-z
<b>Default value</b>	A-z (this enables all logging)
<b>Example values</b>	G

## Command line

<b>Tool</b>	Policy agent ( <code>mgspolicy</code> )
<b>Example</b>	<code>-o LogLevel=G</code>

## Registry

<b>Installed by</b>	Installation of the FlexNet inventory agent
<b>User preference</b>	In order of precedence: <ul style="list-style-type: none"> <li>• HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion</li> </ul>



	<ul style="list-style-type: none"> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common</li> </ul>
<b>Computer preference</b>	<p>In order of precedence:</p> <ul style="list-style-type: none"> <li>[Registry]\ManageSoft\Policy Client\CurrentVersion</li> <li>[Registry]\ManageSoft\Common</li> </ul>

## LogModules (inventory agent)

Command line | Registry

LogModules specifies the modules used to log events for FlexNet inventory agent. When unspecified, the default inventory agent log modules are used. To use custom logging, include the location of your custom logging DLL.

### Values

<b>Values / range</b>	default; <path to modules>
<b>Default value</b>	<p>On Microsoft Windows platforms:</p> <pre>default;C:\Program Files (x86)\ManageSoft\Common\mgssyslg.dll; C:\Program Files (x86)\ManageSoft\Common\mgsantlg.dll</pre> <p>On UNIX-like platforms:</p> <pre>default;\$(InstallDir)/lib/levtlog.so</pre>
<b>Example values</b>	C:\temp\myModule.dll

### Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o LogModules=C:\temp\myModule.dll

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

	[Registry]\ManageSoft\Common
--	------------------------------

## LogModules (policy agent)

Command line | Registry

`LogModules` specifies the modules used to log events for the policy agent. When unspecified, the default inventory agent log modules are used. To use custom logging, include the location of your custom logging DLL.

### Values

<b>Values / range</b>	<code>default; &lt;path to modules&gt;</code>
<b>Default value</b>	<p>On Microsoft Windows platforms:</p> <pre>default;C:\Program Files (x86)\ManageSoft\Common\mgssyslg.dll; C:\Program Files (x86)\ManageSoft\Common\mgsamtlg.dll</pre> <p>On UNIX-like platforms:</p> <pre>default;\$(InstallDir)/lib/levtlog.so</pre>
<b>Example values</b>	<code>C:\temp\myModule.dll</code>

### Command line

<b>Tool</b>	Policy agent ( <code>mgspolicy</code> )
<b>Example</b>	<code>-o LogModules=C:\temp\myModule.dll</code>

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Policy Client\CurrentVersion [Registry]\ManageSoft\Common

# LowProfile (inventory agent)

Command line | Registry

`LowProfile` determines the CPU priority of FlexNet inventory agent on the computer device where it is executing.

- When set to `True`, FlexNet inventory agent processes run with low priority. For UNIX-like systems, this sets the nice level of the process to 10 (see *What Is Special on UNIX-Like Systems?* on page 9 for other settings). On recent Windows platforms, it uses background processing mode (`PROCESS_MODE_BACKGROUND_BEGIN`). On legacy Windows platforms where this is not supported (such as Windows XP and earlier), it uses a priority of idle (`IDLE_PRIORITY_CLASS`).
- When set to `False`, the same processes run with normal priority.

## Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> ).
<b>Default value</b>	No default in registry; default behavior is <code>True</code> .
<b>Example values</b>	<code>False</code>

## Command line

<b>Tool</b>	Inventory agent ( <code>ndtrack</code> )
<b>Example</b>	<code>-o LowProfile=True</code>

## Registry

<b>Installed by</b>	Installation of the FlexNet inventory agent
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>

# MachineID

Command line | Registry

`MachineID` stores the computer name of the target device, as returned from the operating system (typically shifted to all upper case).

This value is derived from the operating system, and on recent versions of Windows, is the same (possibly apart from case) as the computer name visible in the **Computer > Properties** dialog, displayed as **Computer name**.

By default, this preference is not set manually, but is referenced as `$(MachineName)`. This reference causes the FlexNet inventory agent to query the operating system for the name of the machine.

However, it is possible to override the value with a manual setting. This may be useful, for example, when taking inventory of UNIX-like machines, where you may prefer a particular machine name to appear in inventory. This value may be set:

- In the installation bootstrap file `msgft_rollback_response`, used for custom installations on UNIX-like platforms (for more details, see *Configure the Bootstrap File for UNIX* in the *FlexNet Inventory Agent and Managed Devices* PDF file, available through the title page of online help.
- In the command line (on all platforms), as described below.
- In the registry (or, on UNIX-like platforms, for the full FlexNet inventory agent locally installed on the computer, the `config.ini` file; or when using `ndtrack.sh` alone as a lightweight inventory scanner, the `ndtrack.ini` file).

If this preference is customized to a non-default value, it should typically be configured under `[Registry]\ManageSoft\Common` (only) to ensure that all agents use the same value of `MachineId`. Note that a setting for any individual agent overrides the setting in `Common`.

## Values

<b>Values / range</b>	Alphanumeric (best restricted to ASCII characters).
<b>Default value</b>	<code>\$(MachineName)</code> This variable is expanded by FlexNet inventory agent by querying the operating system.
<b>Example values</b>	<code>MyMachine</code>

## Command line

<b>Tool</b>	Installation agent ( <code>ndlaunch</code> ), policy agent ( <code>mgspolicy</code> ), scheduling agent ( <code>ndschedag</code> ), usage agent, inventory agent ( <code>ndtrack</code> )
<b>Example</b>	<code>-o MachineId=MyMachine</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent on the target device (by adoption).
<b>User preference</b>	Not available.

<b>Computer preference</b>	<pre>[Registry]\ManageSoft\Launcher\CurrentVersion [Registry]\ManageSoft\Policy Client\CurrentVersion [Registry]\ManageSoft\Schedule Agent\CurrentVersion [Registry]\ManageSoft\Usage Agent\CurrentVersion [Registry]\ManageSoft\Common</pre> <p>(If set in both the Common hive and another hive, the value in the Common hive is not used for that agent.)</p>
----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## MachineInventoryDirectory

Command line | Registry

MachineInventoryDirectory defines the location in which to store machine inventories.



**Note** • The FlexNet inventory agent uses this option only for machine inventory when it is executing on a computer device in local mode. Local mode is set automatically when the base directory for the executable matches the value stored in the registry key `HKLM\Software\ManageSoft Corp\ManageSoft\Etc\InstallDir`. This is the normal state after installation of the agent on a computer device. (This means that this folder is not used for zero-touch inventory, whether collected by the FlexNet inventory agent or the light-weight FlexNet Inventory Scanner. For zero-touch inventory collection, see *MachineZeroTouchDirectory* on page 106.)

### Values

<b>Values / range</b>	Any valid file path.
<b>Default value</b>	<code>\$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft\Tracker\Inventories</code>
<b>Example values</b>	<code>C:\temp</code>

### Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o MachineInventoryDirectory=C:\ManageSoft Corp\ManageSoft\Tracker\Inventories</code>

### Registry

<b>Installed by</b>	Installation of the FlexNet inventory agent (computer preference)
---------------------	-------------------------------------------------------------------

<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## MachineName

Command line | Registry

`MachineName` contains the name of the local machine. Unlike `MachineId`, this preference should not be changed.

### Values

<b>Values / range</b>	Any valid machine name.
<b>Default value</b>	(No default.) If no value is set, FlexNet Inventory Scanner uses the current machine name.
<b>Example values</b>	MyMachine

### Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o MachineName=MyMachine

### Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## MachinePolicy

Command line | Registry

`MachinePolicy` gives the full directory path and file name of the merged policy file for machine policy on this managed device. This file is prepared on the inventory beacon, and is the summary of all applicable machine

policies. The setting can be overridden on the command line for testing purposes, but this setting should generally be read-only.

## Values

<b>Values / range</b>	Any valid directory path and file name of a well-formed merged policy file
<b>Default value</b>	<code>\$(MachinePolicyDirectory)\\$(MachineId).npl</code>
<b>Example values</b>	<code>C:\Document and Settings\All Users\Application Data\Flexera Software\FlexNet Inventory\Policy Client\Policies\Merged\Machine\WORKSTATION_34.npl</code>

## Command line

<b>Tool</b>	Policy agent ( <code>mgspolicy</code> )
<b>Example</b>	<code>-o MachinePolicy="C:\temp\testpolicy.npl"</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	Not available
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Selector\CurrentVersion</code>

# MachinePolicyDirectory

Command line | Registry

`MachinePolicyDirectory` gives the location to store the current machine policy on the local managed device.

## Values

<b>Values / range</b>	Valid folder and path.
<b>Default value</b>	<code>\$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft\Policy Client\Policies\Merged\Machine</code>
<b>Example values</b>	<code>C:\Temp\MachinePolicies</code>

## Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	<code>-o C:\Temp\MachinePolicies</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Policy Client\CurrentVersion

# MachinePolicyPackageDirectory

Command line | Registry

MachinePolicyPackageDirectory gives the location where package information associated with machine policy is cached.

## Values

<b>Values / range</b>	Valid folder and path.
<b>Default value</b>	<code>\$(CommonAppDataFolder)\ManageSoft Corp\ManageSoft\Policy Client\Packages</code>
<b>Example values</b>	<code>C:\Temp\MyMachinePolicy\PackageInfo</code>

## Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	<code>-o C:\Temp\MyMachinePolicy\PackageInfo</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
---------------------	---------------------------------------------------------------



<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Policy Client\CurrentVersion

## MachineScheduleDirectory

Command line | Registry

MachineScheduleDirectory gives the folder in which the machine schedules are stored.

A machine schedule is run for the computer on which it is installed, regardless of any users that may or may not have accounts on that machine.

---

**Warning** • *Altering this value is not recommended. Additional actions need to be taken when redirecting this to another folder. Contact your Flexera Software professional services consultant for further information.*

### Values

Values / range	Valid folder path
<b>Default value</b>	Windows devices: <pre>\$ (CommonAppDataFolder)\ManageSoft Corp\ ManageSoft\Schedule Agent\Schedules</pre> Non-Windows devices: <pre>\$(CommonAppDataFolder)/scheduler/schedules</pre>
<b>Example values</b>	<pre>C:\Program Files\Flexera Software\Schedule Agent\ MachineSchedules</pre>

### Command line

<b>Tool</b>	Scheduling agent (ndschedag)
<b>Example</b>	<pre>-o MachineScheduleDirectory="C:\Program Files\Flexera Software\schedule agent\ MachineSchedules"</pre>

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
---------------------	---------------------------------------------------------------

<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Schedule Agent\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Schedule Agent\CurrentVersion

## MachineZeroTouchDirectory

Command line | Registry

MachineZeroTouchDirectory identifies the directory where machine inventory files are written (temporarily, pending upload) during any inventory gathering by FlexNet inventory agent. If the upload (called as part of the inventory scanning process) proceeds normally, each temporary file is cleaned up after upload.



**Note** • The FlexNet inventory agent references this setting for any machine inventory involving remote execution (zero touch inventory gathering). Remote mode is set automatically when the registry key `HKLM\Software\ManageSoft Corp\ManageSoft\EtcInstallDir` does not exist or does not match the base directory for the executable. The registry key is typically missing during zero-touch inventory gathering, because the inventory agent has not been permanently installed on the managed device. (This means that, when the FlexNet inventory agent is locally installed on the managed device, this folder is not used. Instead, in this case see *MachineInventoryDirectory* on page 101.)

### Values

<b>Values / range</b>	Any valid directory.
<b>Default value</b>	<code>%temp%\FlexeraSoftware\</code> This is the temporary directory for the account running the inventory scan.
<b>Example values</b>	<code>C:\temp</code>

### Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o MachineZeroTouchDirectory=C:\temp</code>

### Registry

<b>Installed by</b>	Manual configuration
---------------------	----------------------

<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## MSI

Command line | Registry

MSI applies to inventory for Microsoft Windows devices:

- When set to `True`, Microsoft Installer (MSI) package information is added to the inventories.
- When set to `False`, FlexNet inventory agent does not include MSI package information in inventories.

### Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> ).
<b>Default value</b>	<code>True</code>
<b>Example values</b>	<code>False</code>

### Command line

<b>Tool</b>	Inventory agent ( <code>ndtrack</code> )
<b>Example</b>	<code>-o MSI=False</code>

### Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

## Name

Registry

Name is a human-friendly name for this upload or download record. For example, if the host name is beacon1.tmnis.org, you might choose to name this entry beacon1.tmnis.org Upload Location or beacon1.tmnis.org Download Location respectively.

### Values

<b>Values / range</b>	Any name
<b>Default value</b>	(No default.)
<b>Example values</b>	beacon1.tmnis.org Download Location

### Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
<b>User preference</b>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\TrustedLocations\&lt;serverkey&gt;</pre> <p>For excluded locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\ExcludedLocations\&lt;serverkey&gt;</pre>
<b>Computer preference</b>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>[Registry]\ManageSoft\Launcher\CurrentVersion\TrustedLocations \&lt;serverkey&gt;</pre>

	<p>For excluded locations:</p> <pre>[Registry]\ManageSoft\Launcher\CurrentVersion\ExcludedLocations &lt;serverkey&gt;</pre>
--	-----------------------------------------------------------------------------------------------------------------------------

## ndsensNetType

Command line | Registry

`ndsensNetType` applies only on Windows devices. This value determines when a `When connected to network trigger` is deemed to have occurred, causing the command given by `ndsensNetUp` to be executed. It will only trigger if the network is of a certain type. There are three possible values:

- 1 Local area network (LAN)
- 2 Wide area network (WAN)
- 3 Either LAN or WAN.

The scheduling agent monitors the specified network type(s). For example, if `ndsensNetType=2`, the agent only monitors for connections to WANs.

### Values

<b>Values / range</b>	1, 2, 3
<b>Default value</b>	3
<b>Example values</b>	1

### Command line

<b>Tool</b>	Scheduling agent ( <code>ndschedag</code> )
<b>Example</b>	<code>-o ndsensNetType=2</code>

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Schedule Agent\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Schedule Agent\CurrentVersion</code>

## no\_proxy

Command line | Registry

`no_proxy` lists the addresses for which the installation agent should ignore the proxy settings entered in the `http_proxy` registry entry.

### Values

<b>Values / range</b>	Any valid URL.
<b>Default value</b>	(No default.)
<b>Example values</b>	<code>tmnis.com;tmnis.com.de</code>

### Command line

<b>Tool</b>	Installation agent (ndlaunch)
<b>Example</b>	<code>-o no_proxy=tmnis.com;tmnis.com.de</code>

### Registry

<b>Installed by</b>	Installation of FlexNet inventory agent on a managed device (computer preference)
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Launcher\CurrentVersion

## OnConnect

Command line only

`OnConnect` applies only on Windows devices. Determines scheduling tasks that are initiated when an `OnConnect` trigger occurs. There are two options:

- `True`: Run missed tasks (automatically setting the preference `Catchup=Always`) and run all events with `OnConnect` triggers
- `False`: Do nothing.

## Values

<b>Values / range</b>	Boolean (True or False)
<b>Default value</b>	False
<b>Example values</b>	True

## Command line

<b>Tool</b>	Scheduling agent (ndschedag)
<b>Example</b>	-o OnConnect=True

# OracleInventoryUser

Command line | Registry

`OracleInventoryUser` specifies the member of the `dba` database group which the inventory agent impersonates for Oracle introspection (collecting inventory data from an Oracle database) on a UNIX host. (This preference applies only to UNIX platforms. On Microsoft Windows, `ndtrack` executes as `SYSTEM`, which must be a member of the `ora-dba` database group on the Windows Server. For manual execution on the command line in Microsoft Windows, you can alternatively specify another account that has administrator privileges and is a member of `ora-dba`.)

## Values

<b>Values / range</b>	An exact match for any Oracle user name that is a current member of the <code>dba</code> database group on the UNIX host server.
<b>Default value</b>	None.
<b>Example value</b>	<code>dbauser</code>

## Command line

<b>Tool</b>	Inventory agent on UNIX only ( <code>ndtrack.sh</code> )
<b>Example</b>	-o OracleInventoryUser= <code>dbauser</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>Computer preference</b>	<p>[Registry]\ManageSoft\Tracker\CurrentVersion in the config.ini or ndtrack.ini file:</p> <pre>[ManageSoft\Tracker\CurrentVersion] OracleInventoryUser=dbauser</pre>

# Password

## Registry

`Password` stores the encrypted password required for authentication during upload of files from the managed device to the inventory beacon.

## Values

<b>Values / range</b>	This can be any characters. The characters are encrypted.
<b>Default value</b>	<p>For FT protocol only:</p> <pre>Anonymous</pre> <p>Otherwise, there is no default.</p>

## Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
<b>User preference</b>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>
<b>Computer preference</b>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>



# PerformOracleInventory

## Registry

The FlexNet inventory agent collects Oracle inventory each time the inventory collection job is scheduled. You can use the `PerformOracleInventory` option to prevent `ndtrack` from collecting Oracle inventory. The agent stops collecting Oracle inventory until this option is reset to `True`.

## Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> )
<b>Default value</b>	<code>True</code>
<b>Example values</b>	<code>False</code>

## Command line

<b>Tool</b>	<code>ndtrack</code>
<b>Example</b>	<code>-o PerformOracleInventory=False</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Tracker\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\Managesoft\Tracker\CurrentVersion</code>

# PolicyServerURL

## Command line | Registry

`PolicyServerURL` gives the location of policy for server-side merging. The policy agent passes this path through to the installation agent.

---

**Warning** • *Internal use only: do not edit.*

## Values

<b>Values / range</b>	Valid URL.
<b>Default value</b>	<code>http://beacon.domain.com/ManageSoftDL/Policies/Merged/Machine/deviceHostname.npl</code>
<b>Example values</b>	<code>http://InvBeaconB/ManageSoftDL/Policies/Merged/Machine/MgdDev2.npl</code>

## Command line

<b>Tool</b>	Policy agent (mgspolicy)
<b>Example</b>	<code>-o PolicyServerURL=http://InvBeaconB/ManageSoftDL/Policies/Merged/Machine/MgdDev2.npl</code>

## Registry

<b>Installed by</b>	Code internals
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Policy Client\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Policy Client\CurrentVersion

# Port

Registry

Port is the port number for data transfer.

## Values

<b>Values / range</b>	Integers
<b>Default value</b>	(No default.)
<b>Example values</b>	1099

## Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
---------------------	--------------------------------------------------------------

<p><b>User preference</b></p>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\TrustedLocations\&lt;serverkey&gt;</pre> <p>For excluded locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\ExcludedLocations\&lt;serverkey&gt;</pre>
<p><b>Computer preference</b></p>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>[Registry]\ManageSoft\Launcher\CurrentVersion\TrustedLocations \&lt;serverkey&gt;</pre> <p>For excluded locations:</p> <pre>[Registry]\ManageSoft\Launcher\CurrentVersion\ExcludedLocations \&lt;serverkey&gt;</pre>

## Priority

### Registry

Priority determines the order in which upload or download connections to inventory beacons should be attempted if `AutoPriority` is False. (If `AutoPriority` is True, the `Priority` registry key is set dynamically for each download and upload activity.)

To define a location as primary server for download and/or upload, set `AutoPriority` to False, and set `Priority` to 1.

To define a location as one that should not be used for download and/or upload, set `AutoPriority` to False, and set `Priority` to 100.

## Values

<b>Values / range</b>	Recommended range of 0 - 100
<b>Default value</b>	50
<b>Example values</b>	75

## Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
<b>User preference</b>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>
<b>Computer preference</b>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>

# ProgramFiles, ProgramFilesX86Folder, ProgramFilesX64Folder

Command line | Registry

`ProgramFiles`, `ProgramFilesX86Folder`, `ProgramFilesX64Folder` are a set of options that point to the Windows program files folder (across various versions of the operating system) on the target device where inventory is being gathered. `Program Files` exists for backwards compatibility; current distributions operate using `ProgramFilesX86Folder`.



**Note** • `ndtrack` (the executable underlying the FlexNet inventory agent) is a 32-bit executable.

## Values

<b>Values / range</b>	The valid folder name where applications are installed.
<b>Default values</b>	<p>Default values respectively on a 32-bit platform are:</p> <ul style="list-style-type: none"> <li>• Program Files</li> <li>• Program Files</li> <li>• "" (empty value).</li> </ul> <p>Default values on a 64-bit platform are:</p> <ul style="list-style-type: none"> <li>• Program Files</li> <li>• Program Files (x86)</li> <li>• Program Files</li> </ul>
<b>Example values</b>	<code>-o ProgramFiles="D:\Program Files"</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>option</code>

## Registry


<b>Installed by</b>	Predefined by Microsoft Windows.
<b>Reference as</b>	<code>\$(ProgramFiles)</code>

# Protocol

## Registry

`Protocol` identifies the uploads (or download) protocol for transferring files from (or to) the managed device.

## Values

<b>Values / range</b>	<p>http, https, file, ftp</p> 
-----------------------	-------------------------------------------------------------------------------------------------------------------

	<b>Note</b> • Only <i>http</i> and <i>https</i> are supported by inventory beacons. The other values are available for testing or custom solutions.
<b>Default value</b>	(No default.)

## Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
<b>User preference</b>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\TrustedLocations\&lt;serverkey&gt;</pre> <p>For excluded locations:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion\ExcludedLocations\&lt;serverkey&gt;</pre>
<b>Computer preference</b>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre> <p>For trusted locations:</p> <pre>[Registry]\ManageSoft\Launcher\CurrentVersion\TrustedLocations \&lt;serverkey&gt;</pre> <p>For excluded locations:</p> <pre>[Registry]\ManageSoft\Launcher\CurrentVersion\ExcludedLocations \&lt;serverkey&gt;</pre>

# Proxy

## Registry

`Proxy` describes the server name and port number of the SOCKS proxy (available for both uploads and downloads).

## Values

<b>Values / range</b>	<ul style="list-style-type: none"> <li>Valid SOCKS proxy server name and port number in the form of <code>socks:hostname:portNumber</code></li> <li><code>direct</code>, which tells the installation agent to try to directly access the URL if connection attempts through the proxy server fail.</li> </ul>
<b>Default value</b>	(No default.)
<b>Example values</b>	<code>socks:myServer:1080, direct</code>

## Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
<b>User preference</b>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>
<b>Computer preference</b>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>

# Recurse

Command line | Registry

`Recurse` controls whether FlexNet inventory agent drills down for inventory collection:

- When set to `True`, FlexNet inventory agent includes folders beneath the top-level folder(s) specified by `IncludeDirectory`.
- When set to `False`, FlexNet Inventory Scanner does not recurse folders beneath the top level folder(s). It only tracks files immediately within the folder(s) specified by `IncludeDirectory`.

## Values

<b>Values / range</b>	Boolean (True or False).
<b>Default value</b>	True
<b>Example values</b>	False

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o Recurse=False

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# RunInventoryScripts

Command line | Registry

When `RunInventoryScripts` is `True`, this preference specifies that inventory scripts should be run after managed devices have been inventoried. All scripts located in the location specified by `InventoryScriptsDir` are then executed immediately after inventory data collection is complete.

## Values

<b>Values / range</b>	Boolean (True or False).
<b>Default value</b>	False
<b>Example values</b>	True



## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o RunInventoryScripts=True</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	Not available
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# SelectorAlgorithm

## Registry

`SelectorAlgorithm` specifies the algorithm(s) used to assign values to the `Priority` registry key for download and upload locations.

After application of the nominated algorithm(s), the managed device will attempt to collect packages from the highest priority server. In the event of connection failure, the managed device uses the other prioritized servers remaining in the list as failover servers.

The NetSelector includes the following algorithms:

- `MgsADSiteMatch`: Moves all servers in the current managed device's site to the front of the priority list
- `MgsBandwidth`: Priorities are based on end-to-end bandwidth availability to the server
- `MgsDHCP`: Priorities are based on lists of servers specified in DHCP
- `MgsDomainMatch`: Priorities are determined by closest match in domain name
- `MgsIPMatch`: Priorities are determined by closest IP address match
- `MgsNameMatch`: Matches prefixes in computer names
- `MgsPing`: Priorities are determined by fastest ping response time
- `MgsRandom`: Random priorities are assigned
- `MgsServersFromAD`: (Windows devices only) Priorities are determined according to lists of servers specified in Active Directory
- `MgsSubnetMatch`: Moves all servers in the current subnet to the front of the priority list, but retaining the relative order of existing priorities.

Each algorithm may be given an integer parameter that determines the number of servers to which priorities will be assigned. Some algorithms may also be given an additional boolean attribute that can cause unmatched

servers to be discarded from the list (priority set to the string literal `invalid`). Some algorithms also accept other parameters.

## Values

<b>Values / range</b>	<code>MgsADSiteMatch, MgsBandwidth, MgsDHCP, MgsDomainMatch, MgsIPMatch, MgsNameMatch, MgsPing, MgsRandom, MgsServersFromAD, MgsSubnetMatch</code>
<b>Default value</b>	<code>MgsRandom;MgsPing;MgsADSiteMatch</code>
<b>Example values</b>	<p><code>MgsRandom(3)</code></p> <p>This means that ManageSoft should randomly assign the top three servers (based on the priorities currently assigned).</p> <p><code>MgsADSiteMatch(, True);MgsSubnetMatch</code></p> <p>This means that the NetSelector lists servers outside the current managed device's site as "invalid". (<code>MgsSubnetMatch</code> will only prioritize valid servers set by <code>MgsADSiteMatch</code>.)</p>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\NetSelector\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\NetSelector\CurrentVersion</code>

# ScheduleType

Command line only

`ScheduleType` applies only for Windows managed devices. Determines the type of schedule that is run when the scheduling agent is run via the command line. There are two options:

- `Machine`—Runs the schedule agent in "Machine" mode. You must have administrator privileges, or run under the Local System account. Any schedules installed on the same command line are installed as Machine schedules. This option can also be used to view scheduled tasks from the currently installed Machine schedule. (A Machine schedule is run for the machine on which it is installed, regardless of any users that may or may not have accounts on that machine.)
- `User`—Runs the scheduling agent in "User" mode. Will not work if the scheduling agent is run as the Local System account. Any schedules installed on the same command line are installed as User schedules. This option can also be used to view scheduled tasks from the currently installed User schedule. (A User schedule is run for the specified user account on the machine where the schedule resides.)

## Values

<b>Values / range</b>	Machine, User
<b>Default value</b>	<p>If running as the Local System account:</p> <p>Machine</p> <p>If running as any User account (including Administrator):</p> <p>User</p>

## Command line

<b>Tool</b>	Scheduling agent (ndschedag)
<b>Example</b>	<code>-o ScheduleType=User</code>

# ShowIcon (installation agent)

Command line | Registry

ShowIcon influences visibility to the user on the computer device being inventoried:

- When set to `True`, `ndlaunch` displays an icon in the system tray when it is installing a package. This icon displays, regardless of the value of the `UserInteractionLevel` (installation agent) preference. If this icon is double-clicked and `UserInteractionLevel` (installation agent) is set to `Status` or `Auto`, the progress display toggles from being hidden to being visible.
- When set to `False`, no icon will display.



**Tip** • On modern versions of Windows, this preference affects FlexNet inventory agent when it is being executed by a particular user account. When (as is normal) the inventory is being run by the local `SYSTEM` account, no icon is visible in the system tray.

## Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> ).
<b>Default value</b>	No registry default. If no value is supplied, the behavior depends on <code>UserInteractionLevel</code> . Where this is <code>Full</code> , the behavior is equivalent to <code>ShowIcon=False</code> ; and otherwise the behavior is as for <code>ShowIcon=True</code> .
<b>Example values</b>	<code>False</code>

## Command line

<b>Tool</b>	Installation agent (ndlaunch)
<b>Example</b>	<code>-o ShowIcon=False</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	In order of precedence: <ul style="list-style-type: none"> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion</li> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common</li> </ul>
<b>Computer preference</b>	In order of precedence: <ul style="list-style-type: none"> <li>[Registry]\ManageSoft\Launcher\CurrentVersion</li> <li>[Registry]\ManageSoft\Common</li> </ul>

# ShowIcon (inventory agent)

Command line | Registry

ShowIcon influences visibility to the user on the computer device being inventoried:

- When set to `True`, FlexNet inventory agent displays an icon in the system tray when it is collecting inventory. This icon displays, regardless of the value of the `UserInteractionLevel` preference. If this icon is double-clicked and `UserInteractionLevel` is set to `Status` or `Auto`, the progress display toggles from being hidden to being visible.
- When set to `False`, no icon will display.



**Tip** • On modern versions of Windows, this preference affects FlexNet inventory agent when it is being executed by a particular user account. When (as is normal) the inventory is being run by the local `SYSTEM` account, no icon is visible in the system tray.

## Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> ).
-----------------------	------------------------------------------------------

<b>Default value</b>	No registry default. If no value is supplied, the behavior depends on <code>UserInteractionLevel</code> . Where this is <code>Full</code> , the behavior is equivalent to <code>ShowIcon=False</code> ; and otherwise the behavior is as for <code>ShowIcon=True</code> .
<b>Example values</b>	<code>False</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o ShowIcon=False</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent
<b>User preference</b>	In order of precedence: <ul style="list-style-type: none"> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</li> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common</li> </ul>
<b>Computer preference</b>	In order of precedence: <ul style="list-style-type: none"> <li>[Registry]\ManageSoft\Tracker\CurrentVersion</li> <li>[Registry]\ManageSoft\Common</li> </ul>

# SourceFile

Command line | Registry

`SourceFile` identifies the file or files to be uploaded by the upload agent.

## Values

<b>Values / range</b>	Either a UNC ( <code>\\MYCOMPUTER\...</code> ) or a drive ( <code>C:\</code> ) path to the required file or files. Wildcard characters can be used in the filename component.
<b>Default value</b>	(No default.)
<b>Example values</b>	<code>C:\Temp\*.log</code>

## Command line

<b>Tool</b>	Upload agent (ndupload)
<b>Example</b>	<code>-o SourceFile=c:\temp\mylogfile.log</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Uploader\ \CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Uploader\CurrentVersion

# SourceRemove

Command line | Registry

`SourceRemove` determines whether the upload agent removes the uploaded file(s) from the source location after a successful upload. If `True`, the files are removed from the source location.

## Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> )
<b>Default value</b>	<code>True</code>
<b>Example values</b>	<code>False</code>

## Command line

<b>Tool</b>	Upload agent (ndupload)
<b>Example</b>	<code>-o SourceRemove</code> <code>-o SourceRemove=False</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
---------------------	-----------------------------------------

<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Uploader\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Uploader\CurrentVersion

## Startup

Command line | Registry

**Warning** • *Internal use only: do not edit.*

The scheduling agent checks the `Startup` flag to determine if it has been called as part of system startup or while an end-user is logging on:

- For system startup, the scheduling agent is run by the scheduled event `MGSStartup`.
- For logon processing (Windows only), the scheduling agent is run as a result of the value for the registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\SchedulingAgent_nDG`. The scheduling agent cannot be called during logon processing for non-Windows devices.

The command line parameter `-o Startup=True` is passed in by default. Setting this preference elsewhere in the registry will have no effect.



**Notice** • *If the command line is changed to `-o Startup=False`, the scheduling agent will still trigger missed events if configured.*

### Values

<b>Values / range</b>	Boolean (True or False)
<b>Default value</b>	False

### Command line

<b>Tool</b>	Scheduling agent (ndschedag)
<b>Example</b>	<code>-o Startup=True</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Schedule Agent\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Schedule Agent\CurrentVersion

## StrictInstall

Command line | Registry

If `StrictInstall` is set to `True`, the policy agent returns a non-zero exit code if any package in policy fails to install. If set to `False`, the policy agent may return a zero exit code even if packages failed to install. Do not use the policy agent's return code to test for success unless this preference is set to `True`.

### Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> )
<b>Default value</b>	No registry default; default behavior <code>False</code>
<b>Example values</b>	<code>True</code>

### Command line

<b>Tool</b>	Policy agent ( <code>mgspolicy</code> ), installation agent ( <code>ndlaunch</code> )
<b>Example</b>	<code>-o StrictInstall=True</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	In order of precedence: <ul style="list-style-type: none"> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion</li> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common</li> </ul>
<b>Computer preference</b>	In order of precedence:



- |                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• [Registry]\ManageSoft\Launcher\CurrentVersion</li> <li>• [Registry]\ManageSoft\Common</li> </ul> |
|-------------------------------------------------------------------------------------------------------------------------------------------|

## SysDirectory

Command line | Registry

`SysDirectory` is a FlexNet pre-defined variable for the path to the Windows `System` folder. Intended for use as a reference, as `$(SysDirectory)`, but the value can be over-written.

### Values

<b>Values / range</b>	Valid folder name.
<b>Default value</b>	The path to the system folder on the Windows operating system.
<b>Example values</b>	<code>C:\Winnt\System32</code>

### Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o SysDirectory=C:\Windows\System</code>

### Registry

<b>Installed by</b>	Predefined within FlexNet inventory agent / Windows.
<b>Reference as</b>	<code>\$(SysDirectory)</code>

## UIMode

Command line only

`UIMode` only applies for Windows managed devices.

Determines the user interface when the scheduling agent is run from the command line. There are two options:

- `LogUI`—Display the event log user interface.
- `EventUI`—Display currently scheduled events.

## Values

<b>Values / range</b>	LogUI, EventUI
<b>Default value</b>	EventUI

## Command line


<b>Tool</b>	Scheduling agent (ndshedag)
<b>Example</b>	-o UIMode=LogUI

# Upload

Command line | Registry

When `Upload` is set to `True`, the inventory files generated by FlexNet inventory agent are immediately uploaded to the reporting location on the inventory beacon. When set to `False`, the inventory files are not uploaded (for example, in case you wish to inspect or validate their contents).

## Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> ).
<b>Default value</b>	<p>True</p>  <p><b>Tip</b> • The default value is the inverse of the default for the lightweight FlexNet Inventory Scanner.</p>
<b>Example values</b>	False

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o Upload=False

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# UploadLocation

Command line | Registry

Unless the `Upload` parameter is `False`, an upload location must be specified, and `UploadLocation` is a convenient method. The value must be a URI of the form:

```
protocol separator server-name/location
```

where

- `protocol` and `separator` comprise one of these combinations (note three slashes for the file protocol):

- `http://`
- `https://`
- `ftp://`
- `file:///`



**Tip** • If you are uploading to an inventory beacon, use one of the HTTP or HTTPS protocols. The FlexNet Beacon software does not provide native support for FTP or file shares. (Those protocols are available only for configuration of your own uploads, such as for disconnected inventory beacons.)

- `server-name` is either the IP address or a server name (where the system running the FlexNet inventory agent has DNS access to resolve server names) of the destination for the upload (such as a parent inventory beacon, or a central application server).



**Tip** • If the target server is not listening on the default port number for the selected protocol, add the port number to the server name, separated by a colon. Example:

```
https://myServer.example.com:886/ManageSoftRL
```

- `location` is the reporting location that receives uploads on the destination server. In a default installation, this may be called `ManageSoftRL`, but may have been given any name during installation. This is a web service that receives the uploaded inventory, and stores it (briefly) in `%CommonAppData%\Flexera Software`

\Incoming\Inventories on the inventory beacon. The scheduled task Upload FlexNet logs and inventories, which by default runs every minute, then transfers the file to the central server.



**Note** • If `Upload` is `True` and `UploadLocation` is not supplied, the uploader looks for upload information in `[Registry]\ManageSoft\Common\UploadSettings` (and similarly in `HKEY_CURRENT_USER`), where the upload information inside each key will be used in sequence until the file(s) are successfully uploaded. (The uploader ignores servers listed in `UploadSettings` whose priority values are "invalid" or non-numeric.)

## Values

<b>Values / range</b>	Any valid URL.
<b>Default value</b>	(No default.)
<b>Example values</b>	<code>ftp://server/dir1/dir2</code>

## Command line

<b>Tool</b>	Upload agent ( <code>ndupload</code> ) or inventory agent ( <code>ndtrack</code> )
<b>Example</b>	<code>-o UploadLocation=http://server/dir1/dir2/</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Uploader\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Uploader\CurrentVersion</code> <code>[Registry]\ManageSoft\Tracker\CurrentVersion</code> <code>[Registry]\ManageSoft\Common</code>

# UploadPassword

Command line | Registry

`UploadPassword` provides the password of the specified `UploadUser`. If the option is not used, the upload agent attempts to use the sequence of upload passwords from the registry.

## Values

<b>Values / range</b>	A valid password.
<b>Default value</b>	(No default.)
<b>Example values</b>	<code>mypassword</code>

## Command line

<b>Tool</b>	Upload agent (ndupload)
<b>Example</b>	<code>-o UploadPassword=mypassword</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Uploader\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Uploader\CurrentVersion

# UploadProxy

Command line | Registry

UploadProxy provides the name of the proxy server required for UploadLocation. Specifying `direct` tells the uploader to try directly accessing the URL, if connection attempts through the proxy server fail.

If the option is not used, the uploader refers to the sequence of any upload proxies identified in the failover list of inventory beacons. (For uploads, this list is saved in the registry under `[Registry]\ManageSoft\Common\UploadSettings\.`) Normally this means that if this preference is not set, proxies are not used for uploads from the managed device to the inventory beacon.

## Values

<b>Values / range</b>	The name and port number of a proxy server, in the format: <code>Socks:serverName:portNumber,direct</code>
<b>Default value</b>	(No default.)
<b>Example values</b>	<code>Socks:lyonesse:1080,direct</code>

## Command line

<b>Tool</b>	Upload agent (ndupload)
<b>Example</b>	<code>-o UploadProxy=Socks:lyonesse:1080,direct</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Uploader\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Uploader\CurrentVersion

# UploadRule

Command line | Registry

`UploadRule` identifies the upload rule governing this command. Each rule covers a specific file type, mapping it to an upload destination set aside for receiving the matching file type. The mappings can be found in `[Registry]\ManageSoft\Common\Rules`. Setting this option also tells the uploader to transfer files of the specified type.



**Note** • If this option is set, any value of `UploadLocation` is ignored.

## Values

<b>Values / range</b>	One of Inventory, Log, PolicyComplianceLog, SMSStatusMessage
<b>Default value</b>	(No default.)
<b>Example values</b>	Log

## Command line

<b>Tool</b>	Upload agent (ndupload)
<b>Example</b>	<code>-o UploadRule=Log</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Uploader\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Uploader\CurrentVersion

# UploadSettings

## Registry

`UploadSettings` is a registry key (container) for several preferences that can control the upload of data by the FlexNet inventory agent (or the lightweight inventory scanner configurable for UNIX-like platforms, where the values can be stored in the `ndtrack.ini` configuration file). These registry values are in sets that apply to a particular reporting location, for which reason the registry key must be completed with an identifier for the reporting location. The completed path leads to the relevant set of registry values, as shown below.

When configured by the failover list generated by an inventory beacon, the placeholder `<reporting_location>` takes the form of a GUID that identifies the reporting location on the particular inventory beacon (for example, `{8909c9ba-8492-420e-b6e0-100ecf115b0a}`). In contrast, if you are manually configuring `UploadSettings` in an `ndtrack.ini` file for the lightweight inventory scanner on UNIX-like platforms, you may use any string of ASCII characters (excluding white space) that is unique within the context of the `ndtrack.ini` file for these locations.

Four name/value pairs must be specified, and others are optional. To omit an optional value, you may include the name and leave the value blank (as shown in the example below), or omit the name/value pair entirely. The values that may be set are:

- *Protocol* on page 117 – Mandatory. For upload to an inventory beacon, this must be either `http` or `https`.
- *Name* on page 107
- *Directory* on page 61 – Mandatory. By default, the reporting location is called `ManageSoftRL`, but as this may have been renamed during installation, you need to check details of your implementation.
- *Host* on page 73 – Mandatory.
- *Port* on page 114 – Mandatory. As this has no default value, you must specify this setting to suit your environment (typically port 80 for HTTP and port 443 for HTTPS).
- *User* on page 138 – If omitted (or left with a blank value), anonymous authentication is used for uploads to this reporting location.
- *Password* on page 112 – Stores an encrypted copy of the password needed when Windows authentication is specified for the upload.



**Tip** • Since this value is encrypted, it cannot be used when manually editing an `ndtrack.ini` configuration file. Use of this setting in a manually-edited file dictates anonymous authentication.

- `Proxy` on page 118
- `Priority` on page 115
- `AutoPriority` on page 55.

For an alternative more suited to command lines, see `UploadLocation`.

## Values

<b>Values / range</b>	Requires a subkey that uniquely identifies the reporting location on an individual inventory beacon.
<b>Default value</b>	None.
<b>Example values</b>	<pre>[Registry]\ManageSoft\Common\UploadSettings\{8909c9ba-8492-420e- b6e0-100ecf115b0a}   Protocol=http   Name=Server Room Reporting Location   Directory=ManageSoftRL   Host=WIN-KCPMHPHA0GR   Port=80   User=   Password=   Proxy=   Priority=100   AutoPriority=True</pre>

## Registry

<b>Installed by</b>	Download of failover settings, or manual configuration
<b>User preference</b>	<pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre>
<b>Computer preference</b>	<pre>[Registry]\ManageSoft\Common\UploadSettings \<reporting_location&gt;< pre=""> </reporting_location&gt;<></pre>

# UploadType

Command line | Registry

`UploadType` determines whether the upload agent uploads machine generated files or user generated files.



## Values

<b>Values / range</b>	Machine, User
<b>Default value</b>	<p>If running as the SYSTEM user:</p> <p>Machine</p> <p>If runner as any other user:</p> <p>User</p>
<b>Example values</b>	Machine

## Command line

<b>Tool</b>	Upload agent (ndupload)
<b>Example</b>	-o UploadType=Machine

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Uploader\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Uploader\CurrentVersion

# UploadUser

Command line | Registry

UploadUser provides the username of the account required to access the location to which files are to be uploaded.

It is rare to specify this preference. In general, the credentials for bootstrapping are embedded in the upload URLs; and for normal operation, the inventory beacon downloads a failover list of all available inventory beacons (including encrypted credentials). The FlexNet inventory agent saves the contents of this list into the registry on the managed device, under the DownloadSettings and UploadSettings keys. There may be several keys thereunder, one for each download (or upload) location included in the failover list.

Therefore when (as is normal) UploadUser is not defined, the upload agent attempts to use credentials saved in the [Registry]\ManageSoft\Beacon\UploadSettings group of keys.

## Values

<b>Values / range</b>	Valid user name
<b>Default value</b>	(No default.)
<b>Example values</b>	JoSmith

## Command line

<b>Tool</b>	Upload agent (ndupload)
<b>Example</b>	-o UploadUser=JoSmith

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Uploader\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Uploader\CurrentVersion

# User

## Registry

User stores the account name required for authentication during transfer of files between the managed device and the inventory beacon.

When configured by the failover list generated by an inventory beacon, the placeholders *<reporting\_location>* and *<distribution\_location>* take the form of GUIDs that identify the relevant location on the particular inventory beacon (for example, {8909c9ba-8492-420e-b6e0-100ecf115b0a}). In contrast, if you are manually configuring UploadSettings in an ndtrack.ini file for the lightweight inventory scanner on UNIX-like platforms, you may use any string of ASCII characters (excluding white space) that is unique within the context of the ndtrack.ini file for these locations.

## Values

<b>Values / range</b>	Valid user name.
<b>Default value</b>	For FTP protocol only: Anonymous

	Otherwise, there is no default.
<b>Example value</b>	NewUser

## Registry

<b>Installed by</b>	Failover list for inventory beacons, or manual configuration
<b>User preference</b>	<p>For uploads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ ManageSoft\Common\UploadSettings\ &lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>
<b>Computer preference</b>	<p>For uploads:</p> <pre>[Registry]\ManageSoft\Common\UploadSettings\&lt;reporting_location&gt;</pre> <p>For downloads:</p> <pre>[Registry]\ManageSoft\Common\DownloadSettings\&lt;distribution_location&gt;</pre>

# UserHardware

Command line | Registry

`UserHardware` allows you to track hardware (in user-based inventories), either using Windows Management Instrumentation (WMI) or native APIs. If WMI is available, it is used for tracking.

This preference is only effective when running in the user context. To track hardware in the machine context, use `Hardware`.

When set to `True`, allows the tracking of hardware inventory. When set to `False`, does not track hardware inventory.

## Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> ).
<b>Default value</b>	<code>False</code>
<b>Example values</b>	<code>True</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o UserHardware=True</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	N/A—use Hardware

# UserInteractionLevel (installation agent)

Command line | Registry

`UserInteractionLevel` defines the degree of user interaction with the installation agent (this is controlled separately from the inventory agent). The preference setting for this agent also controls behavior during device reboot.) Possible values are:

- **Full:** Installation activities operate in full interactive mode. The user has full control over a package's installation options, and will see all dialogs during the download, installation and uninstall phases. This is generally not encouraged for packages used with the FlexNet inventory agent.
- **Auto:** Installation activities are fully displayed, but no user interaction is required unless an error occurs. Installation proceeds automatically, using the default install values. Again, this is not generally recommended for packages used with the FlexNet inventory agent.
- **Quiet:** No user interface is displayed during operation, and no user feedback or interaction is available.
- **Status:** Only status dialogs are displayed (for example, progress dialogs).

## Values

<b>Values / range</b>	Full, Auto, Quiet, Status
<b>Default value</b>	Full  (This default was appropriate to previous scenarios for the installation agent, and should probably be changed now.)
<b>Example values</b>	Quiet

## Command line

<b>Tool</b>	Installation agent (ndlaunch)
<b>Example</b>	<code>-o UserInteractionLevel=Quiet</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	<p>In order of precedence:</p> <ul style="list-style-type: none"> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Launcher\CurrentVersion</li> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common</li> </ul>
<b>Computer preference</b>	<p>In order of precedence:</p> <ul style="list-style-type: none"> <li>[Registry]\ManageSoft\Launcher\CurrentVersion</li> <li>[Registry]\ManageSoft\Common</li> </ul>

# UserInteractionLevel (inventory agent)

Command line | Registry

`UserInteractionLevel` defines the user interaction method of the `ndtrack` agent. Possible values are:

- **Full:** The `ndtrack` agent operates in full interactive mode.
- **Auto:** When `ShowIcon` is `True`, the `ndtrack` agent icon displays during inventory activities. The user is able to double-click the icon to access the `ndtrack` agent user interface. When `ShowIcon` is `False`, a progress bar displays during inventory activities.
- **Quiet:** The `ndtrack` agent is not displayed during operations, and no user feedback or interaction is available.
- **Status:** Only status dialogs are displayed (for example, progress dialogs).

## Values

<b>Values / range</b>	Full, Auto, Quiet, Status
<b>Default value</b>	Status
<b>Example values</b>	Quiet

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	-o UserInteractionLevel=Quiet

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	<p>In order of precedence:</p> <ul style="list-style-type: none"> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</li> <li>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Common</li> </ul>
<b>Computer preference</b>	<p>In order of precedence:</p> <ul style="list-style-type: none"> <li>[Registry]\ManageSoft\Tracker\CurrentVersion</li> <li>[Registry]\ManageSoft\Common</li> </ul>

# UserInventoryDirectory

Command line | Registry

UserInventoryDirectory defines the location for the user inventories on the computer device.



**Note** • The FlexNet inventory agent uses this option only for user-based inventory when it is executing on a computer device in local mode. Local mode is set automatically when the base directory for the executable matches the value stored in the registry key `HKLM\Software\ManageSoft Corp\ManageSoft\EtcpInstallDir`. (This means that this folder is not used for zero-touch inventory, whether

collected by the FlexNet inventory agent or the light-weight FlexNet Inventory Scanner. For zero-touch inventory collection, see *UserZeroTouchDirectory* on page 144.)

## Values

<b>Values / range</b>	Valid location.
<b>Default value</b>	<code>\$(AppDataFolder)\ManageSoft Corp\ ManageSoft\Tracker\Inventories</code>
<b>Example values</b>	<code>C:\temp</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o UserInventoryDirectory=C:\ManageSoft Corp\ManageSoft\Tracker \Inventories</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker \CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>

# UserScheduleDirectory

Command line | Registry

`UserScheduleDirectory` applies only on Windows devices.

Determines the folder where the user schedules are stored. A user schedule is run for the specified user account on the machine where the schedule resides.



**Note** • Changing this preference is not recommended.

## Values

<b>Values / range</b>	Valid folder and path.
<b>Default value</b>	<code>\$(CommonAppDataFolder)\ManageSoft Corp\ ManageSoft\Schedule Agent\Schedules</code>
<b>Example values</b>	<code>C:\Program Files\Flexera Software\schedule agent\ UserSchedules</code>

## Command line

<b>Tool</b>	Scheduling agent (ndschedag)
<b>Example</b>	<code>-o UserScheduleDirectory="C:\Program Files\Flexera Software\schedule agent\ UserSchedules"</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Schedule Agent\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Schedule Agent\CurrentVersion</code>

# UserZeroTouchDirectory

Command line | Registry

`UserZeroTouchDirectory` specifies the directory where user inventory files are written (temporarily, pending upload) during a remote ("zero touch") inventory gathering process. If the upload (called as part of the inventory scanning process) proceeds normally, each temporary file is cleaned up after upload.



**Note** • The FlexNet inventory agent references this setting for any user-based inventory involving remote execution (zero touch inventory gathering). Remote mode is set automatically when the registry key `HKLM\Software\ManageSoft Corp\ManageSoft\EtcpInstallDir` does not exist or does not match the base directory for the executable. The registry key is typically missing during zero-touch inventory gathering, because the FlexNet inventory agent has not been permanently installed on the managed device. (This means that, when the FlexNet inventory agent is locally installed on the managed device, this folder is not used. Instead, in this case see [UserInventoryDirectory](#) on page 142.) Also note that the default for the FlexNet inventory agent is different than the default for the lightweight FlexNet Inventory Scanner.



## Values

<b>Values / range</b>	Any valid folder
<b>Default value</b>	<code>%temp%\FlexeraSoftware\</code> This is the temporary directory for the account running the inventory scan. (For the installed FlexNet inventory agent, this is the same default value as for MachineZeroTouchDirectory.)
<b>Example values</b>	<code>C:\temp</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o UserZeroTouchDirectory=C:\temp</code>

## Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>

# VersionInfo

Command line | Registry

When `VersionInfo` is set to `True`, FlexNet inventory agent includes file version header information in the inventory.

When set to `False`, FlexNet inventory agent does not include file version header information in the inventory.

## Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> ).
<b>Default value</b>	<code>True</code>
<b>Example values</b>	<code>False</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o VersionInfo=False</code>

## Registry

<b>Installed by</b>	Code internals, or manual configuration
<b>User preference</b>	HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion
<b>Computer preference</b>	[Registry]\ManageSoft\Tracker\CurrentVersion

# WinDirectory

Command line | Registry

`WinDirectory` references the path to the Windows folder. You can also use `WindowsFolder`: this is a Windows pre-defined variable for the same folder.

## Values

<b>Values / range</b>	
<b>Default value</b>	<code>C:\Windows</code>
<b>Example values</b>	<code>C:\Windows</code>

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o WinDirectory=C:\Winnt</code>

## Registry

<b>Installed by</b>	Predefined within FlexNet inventory agent / Microsoft Windows
<b>Reference as:</b>	<code>\$(WinDirectory)</code>

## WMI

Command line | Registry

When `WMI` is set to `True`, the Windows Management Instrumentation (WMI) tracking is specified as the preferred option for tracking hardware. In this case, if WMI is not available (and the `Hardware` preference is set to `True`), FlexNet inventory agent attempts to track hardware using a native API.

When set to `False`, FlexNet Inventory Scanner uses another tracking mechanism instead of WMI.

### Values

<b>Values / range</b>	Boolean ( <code>True</code> or <code>False</code> ).
<b>Default value</b>	When taking machine-based inventory: <code>True</code> When taking user-based inventory: <code>False</code>
<b>Example values</b>	<code>False</code>

### Command line

<b>Tool</b>	Inventory agent ( <code>ndtrack</code> )
<b>Example</b>	<code>-o WMI=True</code>

### Registry

<b>Installed by</b>	Manual configuration
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>

## WMIConfigFile

Command line | Registry

`WMIConfigFile` defines the location of the Windows Management Instrumentation (WMI) configuration file, used to inform the FlexNet inventory agent what hardware components it should track. This is only used if `WMI` is `True`.

## Values

<b>Values / range</b>	Valid location.
<b>Default value</b>	<code>\$(ProgramPath)\wmitrack.ini</code>
<b>Example values</b>	The value should point to a valid <code>.ini</code> file that contains WMI configuration.

## Command line

<b>Tool</b>	Inventory agent (ndtrack)
<b>Example</b>	<code>-o WMIConfigFile=C:\Program Files\ManageSoft\Tracker\wmitrack.ini</code>

## Registry

<b>Installed by</b>	Installation of FlexNet inventory agent (computer preference)
<b>User preference</b>	<code>HKEY_CURRENT_USER\Software\ManageSoft Corp\ManageSoft\Tracker\CurrentVersion</code>
<b>Computer preference</b>	<code>[Registry]\ManageSoft\Tracker\CurrentVersion</code>