

# NSClientListener.dll

The NSClientListener module is written to allow backwards compatibility with the old NSClient and check\_nt. It has a listener (server) that accepts checks from the check\_nt command and responds accordingly. Due to the nature of the protocol and the limitation in the client NRPE is recommended but if you like this works just fine for "simple things".

The following check\_nt checks are supported.

- CLIENTVERSION
- CPULOAD
- UPTIME
- USEDDISKSPACE
- MEMUSE
- SERVICESTATE
- PROCSTATE
- COUNTER

## Configuration Sections

### NSClient Section

*This is a wrapper page the actual data is on the following page [NSClientListener/config](#)*

- 1.
- 1.
- 1.
1. Ovreview
  1. port
  2. obfuscated\_password
  3. password
  4. allowed\_hosts
  5. bind to address
  6. socket timeout
  7. socket back log
  8. version

### Ovreview

This is the [NSClientListener] module configuration options.

Option	Default value	Description
port	12489	The port to listen to
obfuscated_password		An obfuscated version of password.
password		The password that incoming client needs to authorize themselves by.
allowed_hosts		A list (coma separated) with hosts that are allowed to connect to NSClient++ via NSClient protocol.
socket_timeout	30	The timeout when reading packets on incoming sockets.

### Advanced options:

Option	Default value	Description
socket_back_log		Number of sockets to queue before starting to refuse new incoming connections. This can be used to tweak the amount of simultaneous sockets that the server accepts. This is an advanced option and should not be used.
bind_to_address		The address to bind to when listening to sockets, useful if you have more then one NIC/IP address and want the agent to answer on a specific one.
version	auto	The version number to return for the CLIENTVERSION check (useful to "simulate" an old/different version of the client, auto will be generated from the compiled version string inside NSClient++)

#### **port**

The port to listen to

#### **Default**

12489

#### **obfuscated\_password**

An obfuscated version of password. For more details refer to the password option below.

#### **Default**

Empty string whjich means we will use the value from password instead.

#### **password**

The password that incoming client needs to authorize themselves by. This option will replace the one found under Settings for NSClient. If this is blank the option found under Settings will be used. If both are blank everyone will be granted access.

#### **Default**

Empty string whjich means we will use the value from password in the [Settings] section instead.

#### **allowed\_hosts**

A list (coma separated) with hosts that are allowed to poll information from NSClient++. This will replace the one found under Setting for NSClient if present. If not present the same option found under Settings will be used. If both are blank all hosts will be allowed to access the system.

**BEWARE:** NSClient++ will not resolve the IP address of DNS entries if the service is set to startup automatically. Use an IP address instead or set cache\_allowed\_hosts=0 see above.

#### **Default**

Empty list (falls back to the one defined under [Settings])

#### **bind\_to\_address**

The address to bind to when listening to sockets. If not specified the "first" (all?) one will be used (often the correct one).

## Values

IP address of any interface of the server.

## Default

Empty (first (all?) interface will be used)

## socket\_timeout

The timeout when reading packets on incoming sockets. If the data has not arrived within this time we will bail out. and discard the connection.

## Default

30 seconds

## socket\_back\_log

Number of sockets to queue before starting to refuse new incoming connections. This can be used to tweak the amount of simultaneous sockets that the server accepts. This is an advanced option and should not be used.

## version

The version number to return for the CLIENTVERSION check (useful to "simulate" an old/different version of the client, auto will be generated from the compiled version string inside NSClient++)

## Values:

If given any string will be returned unless auto in which case the proper version will be returned

## Default

auto

# Examples

Check the size of C:\ and make sure it warns at 80% used and a critical warning at 95% used:

```
define command {
    command_name check_nt_disk
    command_line $USER1$/check_nt -s passphrase -H $HOSTADDRESS$ -p 12489
        -v USEDSPACE -l $ARG1$ -w $ARG2$ -c $ARG3$
}
check_command check_nt_disk!c!80!95
```

Check Perfmon value for File IO reads

```
define command {
    command_name check_io_read
    command_line /usr/local/nagios/libexec/check_nt -H $HOSTADDRESS$
        -v COUNTER -l "\\System\\File Read Bytes/sec"
}
check_command check_io_read
```