

General Utilities

Version 2.00

User Guide

**Nanometrics Inc.
Kanata, Ontario
Canada**

© 2000–2005 Nanometrics Inc. All Rights Reserved.

General Utilities Version 2.00 User Guide

The information in this document has been carefully reviewed and is believed to be reliable for Version 2.00.xx. Nanometrics, Inc. reserves the right to make changes at any time without notice to improve the reliability and function of the product.

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Nanometrics Inc.

Nanometrics, Inc.
250 Herzberg Road
Kanata, Ontario, Canada K2K 2A1
Tel (613)592-6776
Fax (613)592-5929
Email info@nanometrics.ca

Part number 14831R2

Release date 2005-01-06

Contents

Tables	iii
Chapter 1	
NmxSC v1.0.0	1
1.1 Description	1
1.1.1 Synopsis	1
1.2 Operation	1
1.2.1 Creating a service	1
1.2.2 Deleting a service	2
1.2.3 Managing a service	2
1.2.3.1 Starting a service manually	2
1.2.3.2 Stopping a service manually	2
1.2.3.3 Querying a service	3
1.2.3.4 Changing service options	3
1.3 Usage Summary	3
1.4 Files	3
1.5 Environment	3
Chapter 2	
NmxWatchdog v2.0.0	5
2.1 Description	5
2.2 Installation	5
2.3 Service configuration	6
2.4 Operation	6
2.5 Configuration	6
2.5.1 [Watchdog.ini]	7
2.5.2 [WatchTiming]	7
2.5.3 [WatchEntry <i>N</i>]	8
2.6 Example <code>watchdog.ini</code> file	9
2.7 Files	9
2.8 Environment	9

Tables

1-1	Service options.	1
2-1	[WatchTiming] section parameters.	7
2-2	[WatchEntry] section parameters.	8

1.1 Description

NmxSC service management software is a command-line utility that can create, delete, start, stop and query a Windows service.

A Windows service is a special program which runs in the background to provide some service or computing capability on a continuous basis. Service programs usually cannot run on their own, but must be started, stopped, and managed through the Windows Service Control Manager. Service programs must therefore respond appropriately to commands and queries from the Service Control Manager.

Services are normally managed using the Services Panel which is accessible through the Windows Control Panel. NmxSC allows you to create, delete, and manage services from the command line.

1.1.1 Synopsis

```
NmxSC create servicename binPath= path\service.exe [options]  
NmxSC <start | stop | delete | query> servicename
```

1.2 Operation

1.2.1 Creating a service

- ▶ To create or install a service, enter this command at a command prompt:

```
NmxSC create servicename binPath= path\service.exe [options]
```

where *path* is the full absolute path of the directory containing the service program executable. Table 1-1 lists options that you may specify to control the behaviour of the service. The default option in each case is **boldface**.

Table 1-1 Service options

Option	Description
type= < own share>	share means the service may be shared by other services; own means the service is used alone.
interact= < yes no>	Indicates whether or not the service may interact with the desktop (for example, via dialogs). This does not affect child programs started by the service.

Table 1-1 Service options (Continued)

Option	Description
<code>start= <auto demand></code>	Indicates whether the service should start automatically when the system boots (<code>auto</code>), or manually on demand.
<code>error= <normal severe critical ignore></code>	Specifies the type of error reported to the Windows event log if the service malfunctions.

1.2.2 Deleting a service

In general, you should not have to delete a service unless the name or location of the executable file changes or the service is no longer needed.

- ▶ If a new service executable is created:
 - a) Replace the old executable with the new one in the same location.
 - b) Stop and restart the service.
- ▶ If the executable changes name or location:
 - c) Stop the service.
 - d) Delete the service with this command:


```
NmxSC delete servicename
```
 - e) Recreate the service specifying the new binPath and options.

1.2.3 Managing a service

Once a service is created, the service may be started, stopped, deleted, or queried from the command prompt using NmxSC. Some of these operations (start, stop, query) may also be carried out using the Services Panel (accessible through the Windows Control Panel).

1.2.3.1 Starting a service manually

- ▶ To start a service, enter the following command:


```
NmxSC start servicename
```

If the service cannot be started, an error message will be written to the command window. Typical reasons that a service cannot be started include:

- ◆ The service is already running.
- ◆ The service is not installed.
- ◆ The Service Control Manager cannot find the specified executable file.
- ◆ A service-specific error has occurred (see log file for the specific service).

1.2.3.2 Stopping a service manually

- ▶ To stop a service, enter this command:


```
NmxSC stop servicename
```

If the service is already stopped or is not installed, an error message will be written to the screen.

1.2.3.3 Querying a service

- ▶ To determine the current status of a service, enter this command:
`NmxSC query servicename`

1.2.3.4 Changing service options

If you need to change the service properties (for example, change it from starting on demand to starting automatically on bootup), you can use either of these methods:

- ▶ Double-click on the service in the Services Panel to open a properties dialog.
- ▶ Delete and recreate the service with the desired options using NmxSC.

1.3 Usage Summary

NmxSC will create, delete, start, stop, and query a named Windows service.

- ▶ To create or install a new service, use this syntax:
`NmxSC create servicename binPath= path\service.exe [options]`
where *path* specifies the full absolute path of the executable, and options may include any of these flags:
`type= <own | share> (default = own)`
`interact= <yes | no> (default = yes)`
`start= <auto | demand> (default = demand)`
`error= <normal | severe | critical | ignore> (default = normal)`
- ▶ To start, stop, delete or query an existing service, use this syntax:
`NmxSC <start | stop | delete | query> servicename`

1.4 Files

`NmxSC.exe`

1.5 Environment

NmxSC runs on Windows NT, Windows 2000, and Windows XP Professional.

NmxWatchdog

v2.0.0

2.1 Description

NmxWatchdog is a Windows Service that starts and monitors tasks. It starts up one or more tasks specified in its configuration file, then monitors each task and restarts any task that stops running (or that stops pinging the NmxWatchdog semaphore). This maximizes the availability of critical data acquisition and communication programs by ensuring that they are always restarted after stopping for any reason.

NmxWatchdog also monitors the disk space of the current drive. If the number of megabytes of free space falls below a user configured level, the watchdog will shut down and close all of its child processes. It is then up to the user to clear sufficient disk space and restart the watchdog. This is done to avoid corruption of the hard disk when the system runs out of disk space.

NmxWatchdog must always be installed and run as a Windows Service. Like any service, it may be started and stopped manually using the Windows Services Panel, or from a command line using the Nanometrics Service Control Manager, NmxSC. However, it is normally configured to start automatically when the system boots up. In this mode, it remains running when users log on or off, but shuts down and closes its child processes on system shutdown.

NmxWatchdog maintains a log of its activity in a date-stamped log file, `watchdog.yyyymmdd.log`, where `yyymmdd` is the current date. Any unexpected termination of watchdog (or failure to start) is also written to the Windows Event log.

2.2 Installation

- ▶ To install NmxWatchdog as a service which starts automatically on bootup, enter this command in a command window:

```
NmxSC create NmxWatchdog binPath= C:\nmx\bin\NmxWatchdog.exe start= auto
```

NmxSC will check for the specified executable, verify that the service does not already exist, then create the service. The service will then appear in the Windows Services Panel.

- ▶ If you have made an error installing the service, you can delete the service using the command
`NmxSC delete NmxWatchdog`
and then recreate it.

2.3 Service configuration

To configure the NmxWatchdog service options, open the Windows Services Panel (via the Control Panel), and double click on the NmxWatchdog icon to open its properties dialog. This will allow you to set the startup option (auto or manual) and other options.

Alternatively, you may simply delete the service using NmxSC, then recreate it (using NmxSC) with the desired options.

For maximum reliability, configure NmxWatchdog to start automatically on bootup.

2.4 Operation

NmxWatchdog runs as a Windows Service.

You may start, stop, or determine the status of the service using either of these methods:

- ▶ Use the Windows Services Panel.
- ▶ Use the command line using the following commands:
 - ▶ To start NmxWatchdog:
`NmxSC start NmxWatchdog`
 - ▶ To stop NmxWatchdog:
`NmxSC stop NmxWatchdog`
 - ▶ To determine whether NmxWatchdog is running:
`NmxSC query NmxWatchdog`

Should the service fail to start, an error code will be written to the command window:

- ▶ If the error code is 1001-1008, the error is a failure to read the `watchdog.ini` file (described in Section 2.5). See the watchdog log file in the `nmx\user` directory for an explanation of the problem.

Watchdog can be set to start and monitor third-party (non-Nanometrics) tasks:

- ▶ Set the *PingSemaphore* flag to `FALSE` for the appropriate WatchEntry (see also Section 2.5.3).

2.5 Configuration

NmxWatchdog is configured by setting options in a plain text configuration file, `c:\nmx\user\watchdog.ini`. This file has three types of sections:

- ◆ [Watchdog.ini]
- ◆ [WatchTiming]
- ◆ [WatchEntry N]

See Section 2.6 for an example `watchdog.ini` file.

- ▶ Each time the `watchdog.ini` file is edited, you should check to see that all watchdog tasks are started successfully. Any problems will be listed in the watchdog log file.

2.5.1 [Watchdog.ini]

The [Watchdog.ini] section must be the first section in the file. It has no variables.

2.5.2 [WatchTiming]

The [WatchTiming] section defines the start, restart, and ping timing for tasks. It must contain the parameters described in Table 2-1, in the order listed.

Table 2-1 [WatchTiming] section parameters

Parameter	Description
<i>LogFileDirectory</i>	The directory in which to store the watchdog log file. The directory name must be enclosed in double quotation marks. Watchdog will create the directory if it does not already exist.
<i>NumberEntries</i>	The number of tasks to start and monitor. You must have one WatchEntry section for each task that you want watchdog to monitor. If you have more WatchEntry sections than <i>NumberEntries</i> indicates, watchdog ignores the extra entries.
<i>Verbosity</i>	Parameter controlling how much log information is written to the log file. This should be one or more of the following choices, separated by the vertical bar " " symbol: <ul style="list-style-type: none"> • <code>Error</code> – messages indicating errors with the NmxWatchdog service. • <code>Warning</code> – messages indicating changes of state of monitored tasks, or possible minor problems with the NmxWatchdog service. • <code>Info</code> – messages indicating the normal progress of the service. • <code>Debug</code> – detailed messages (usually useful only to the developer).
<i>SemPingInterval</i>	How often, in seconds, watchdog expects task to ping it. If a task has not pinged within this time then watchdog considers it in error and kills and restarts it. Default is 30 seconds.
<i>SemPingStartDelay</i>	The number of seconds watchdog waits after starting a task before expecting the first ping. Default is 60 seconds.
<i>ErrorExitRestartDelay</i>	The number of seconds watchdog waits before restarting a task that has shutdown improperly (either due to a crash or watchdog killing it). Default is 120 seconds.
<i>NormalExitRestartDelay</i>	The number of seconds watchdog waits before restarting a task that has shutdown properly. If a task shut downs with a exit code of 0, watchdog assumes it shut down properly. Default is 60 seconds.

Table 2-1 [WatchTiming] section parameters (Continued)

Parameter	Description
<i>MinimumDiskSpace(Megs)</i>	The minimum number of megabytes of disk space the watchdog program will allow on the current drive. If the disk space falls below this point, the watchdog exits shutting down all its child processes.

2.5.3 [WatchEntry *N*]

Each WatchEntry section defines a single task to be started and monitored by Nmx-Watchdog.

- ▶ There must be a separate WatchEntry section for each task to be started:
 - ▶ If there are 2 or more WatchEntry sections then each section must be numbered, starting at 1. That is, [WatchEntry 1], [WatchEntry 2], ..., [WatchEntry *n*].
 - ▶ If there is only one WatchEntry section, then the section header must not have a number. That is, the section header is [WatchEntry].

Each WatchEntry section must have the parameters described in Table 2-2, in the order listed.

Table 2-2 [WatchEntry] section parameters

Parameter	Description
<i>ProgramTitle</i>	Name of the task to start. Watchdog will refer to the task in the log file using this name. The parameter value must be enclosed in double quotation marks.
<i>ProgramPathName</i>	Filename of the program to start, with mandatory extension and full pathname, including drive, and directory if not in the path. Parameters to the program may be specified on the line as well. The entire parameter value must be enclosed in double quotation marks.
<i>WorkingDirectory</i>	Working directory in which this program is to run. The parameter value must be enclosed in double quotation marks.
<i>ExitAction</i>	Indicates what watchdog should do if the task stops. Options are: <ul style="list-style-type: none"> • <code>Ignore</code> – do not restart the task. • <code>Restart</code> – restart the task.
<i>PingsSemaphore</i>	Whether this task will ping the watchdog semaphore to indicate it is alive (<code>TRUE</code> or <code>FALSE</code>). If this flag is <code>TRUE</code> then watchdog will consider the process dead if it does not receive the ping within the [WatchTiming] <i>SemPingInterval</i> . (Watchdog can also be set to start and monitor third-party (non-Nanometrics) tasks: Set the <i>PingSemaphore</i> flag to <code>FALSE</code> for the appropriate WatchEntry.)
<i>StartDelay</i>	The number of seconds to wait before starting this task.

2.6 Example watchdog.ini file



Note All java programs should be started with the `-Xrs` command switch. This will ensure that the programs remain running when each user logs out.

This example `watchdog.ini` file would configure NmxWatchdog to start two tasks:

```
[ watchdog.ini ]

[ WatchTiming ]
LogFileDirectory = "c:\nmx\user\WatchdogLogs"
NumberEntries = 2
Verbosity = Error|Warning
SemPingInterval = 30s
SemPingStartDelay = 1m
ErrorExitRestartDelay = 10s
NormalExitRestartDelay = 10s
MinimumDiskSpace(Megs) = 100

[ WatchEntry 1 ]
ProgramTitle = "NaqsServer"
ProgramPathname = "java -Xrs -ms5m -cp c:\nmx\bin\NaqsServer.jar NaqsServer"
WorkingDirectory = "c:\nmx\user"
ExitAction = Restart
PingsSemaphore = TRUE
StartDelay = 6s

[ WatchEntry 2 ]
ProgramTitle = "DataServer"
ProgramPathname = "java -Xrs -ms5m -cp c:\nmx\bin\DataServer.jar DataServer"
WorkingDirectory = "c:\nmx\user"
ExitAction = Restart
PingsSemaphore = TRUE
StartDelay = 6s
```

2.7 Files

`Watchdog.ini`

2.8 Environment

NmxWatchdog runs on Windows NT, Windows 2000, Windows XP Professional.

