

Integrating AutoIT With Nagios XI 5

Purpose

This document describes how to automate and monitor Windows actions, tasks, and tests using AutoIT and Nagios XI. AutoIT is a powerful scripting tool that enables automation of Windows tasks, and when integrated with Nagios XI, it allows administrators to monitor automated processes efficiently.

If you are integrating AutoIT with Nagios XI 2024, see [Integrating AutoIT with Nagios XI 2024](#)

Target Audience

This guide is intended for Nagios XI Administrators who are interested in monitoring automated actions on Windows systems.

Prerequisites

Before proceeding, ensure that you have the following:

- NSClient++ installed on the Windows machine, configured to allow NRPE checks from the Nagios XI server. Reference the following documentation for setup:
 - [Installing The XI Windows Agent](#)
 - [Configuring The XI Windows Agent](#)
 - [Enabling The NRPE Listener In NSClient++ 0.4.x](#)
- Firefox Web Browser installed. It can be downloaded from [Mozilla](#).
- AutoIT and SciTE Script Editor installed on the Windows machine. They can be downloaded from:
 - **AutoIT:** [Download Here](#)
 - **SciTE Script Editor:** [Download Here](#)

For additional guides and tutorials on AutoIT scripting, visit the [AutoIT main site](#).

Integrating AutoIT With Nagios XI 5

Configure NSClient++ Service

AutoIT requires control of the Windows workspace, so enabling **Allow service to interact with desktop** for the NSClient++ service is recommended.

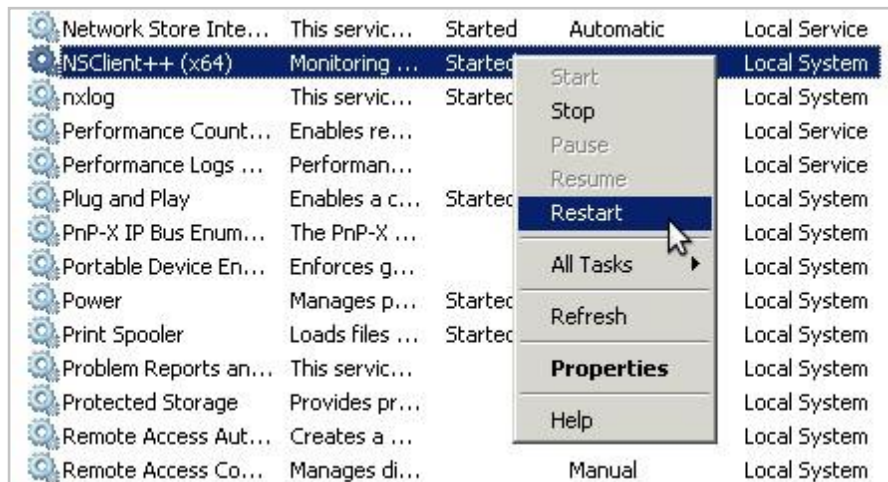
1. Open the **Services Console** (Run **services.msc** in Windows).
2. Locate **NSClient++** in the list.
3. Right-click **NSClient++** and select **Properties**.
4. Navigate to the **Log On** tab.
5. Check the box **Allow service to interact with desktop**.



6. Click **OK**.

Integrating AutoIT With Nagios XI 5

7. Restart NSClient++ by right-clicking the service and selecting **Restart**.



Creating a Script

This example demonstrates how to create a script that opens Firefox, navigates to a URL, measures page load time, and outputs the time to the console.

Open **SciTE Editor** and paste the following script:

```
Func _WinWaitActivate($title,$text,$timeout=0)
    WinWait($title,$text,$timeout)
    If Not WinActive($title,$text) Then WinActivate($title,$text)
    WinWaitActive($title,$text,$timeout)
EndFunc

$title_string = "Yahoo"
Local $begin = TimerInit()
Run("C:\\Program Files (x86)\\Mozilla Firefox\\firefox.exe www.yahoo.com")
_WinWaitActivate($title_string, "")
Local $dif = TimerDiff($begin)
$time_string = $dif
WinClose($title_string, "")
ConsoleWrite($time_string)
```

Integrating AutoIT With Nagios XI 5

Explanation:

- Once the page has loaded, the title of the web browser becomes Yahoo. The script is looking for the string **Yahoo**, see variable declaration on line 7 of the script `$title_string = "Yahoo"`.
- Once it detects the string, the time it took is calculated (line 11) and the time difference is stored in the variable `$time_string` (line 12). Finally, the script outputs the variable `$time_string` to the console.
- For reuse, save this script to the `C:\Program Files\NSClient++\scripts\` directory. This example saves the file as `loadtime.au3`. Saving the script is not a requirement, but it may be useful, later.
- The next step is to compile this script into an EXE file, this is necessary, so the output is correctly written to the console.
- It calculates and logs the time taken for the page to load.

Note: If the script fails, ensure the expected webpage title is "Yahoo", as it may vary by region.

Saving the Script

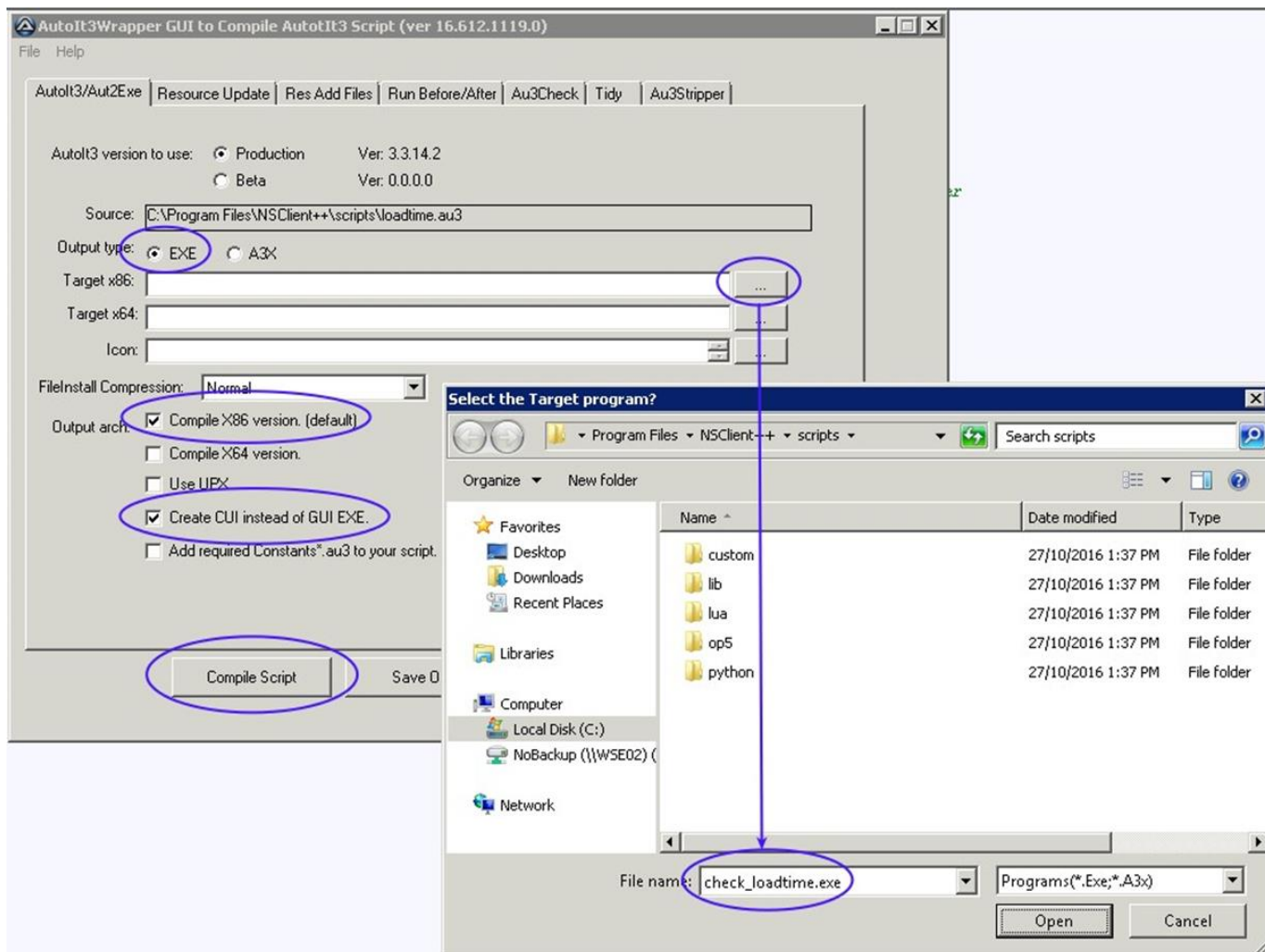
- Save the script as `loadtime.au3` in `C:\Program Files\NSClient++\scripts\`

Compiling the Script

Compiling the script into an executable ensures proper console output:

1. In SciTE Editor, click **Tools > Compile**.
2. Configure the following settings:
 - **Target x86:** Browse to `C:\Program Files\NSClient++\scripts\`
 - **Name:** `check_loadtime.exe`
 - **Compile X86 version:** Checked
 - **Create CUI instead of GUI.EXE:** Checked
3. Click **Compile Script**.
4. The compiled script `check_loadtime.exe` will be generated.

Integrating AutoIT With Nagios XI 5



To test the script, open a command prompt and run:

```
C:\Program Files\NSClient++\scripts\check_loadtime.exe
```

Do not interact with the system until the script completes execution.

```
C:\> Administrator: Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator.B0X293>"C:\Program Files\NSClient++\scripts\check_loadtime.exe"
3744.10895798209
```

Integrating AutoIT With Nagios XI 5

Configuring NSClient++ to Execute Script

1. Open `C:\Program Files\NSClient++\nsclient.ini` in Notepad.
2. Locate the **External Scripts** section.
3. Add the following entry:

```
[/settings/external scripts/scripts]  
check_loadtime = scripts\check_loadtime.exe
```

4. Save the file and restart NSClient++.

Integrating AutoIT Script with Nagios XI

A plugin called `check_autoit_timer.sh` is required to process script output and apply monitoring thresholds.

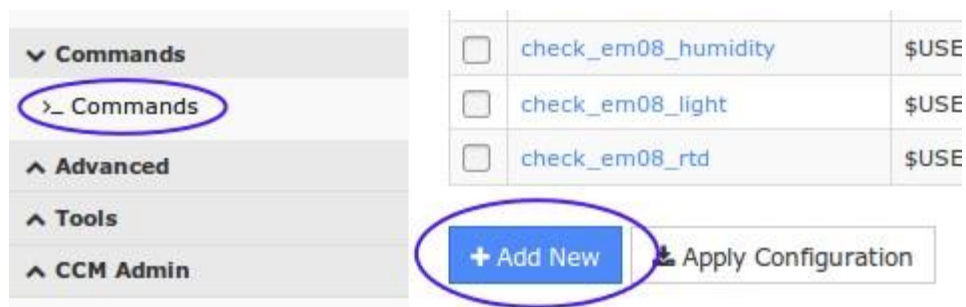
1. Download `check_autoit_timer.sh` from the [Nagios Exchange](#).
2. Upload the plugin to Nagios XI via **Admin > System Extensions > Manage Plugins**.

Configuring Nagios XI

Now the check must be configured in the Nagios XI Web Interface using Core Configuration Manager (CCM).

Creating a Check Command

1. Navigate to **Configure > Core Config Manager (CCM)**.
2. In the left pane, expand **Commands** and click **>_ Commands**.



Integrating AutoIT With Nagios XI 5

3. Click **Add New** and enter the following:

- **Command Name:** check_autoit_timer
- **Command Line:** \$USER1\$/check_autoit_timer.sh -H \$HOSTADDRESS\$ -p 5666 -c "\$ARG1\$" -w \$ARG2\$ -c \$ARG3\$
- **Command Type:** check command
- **Active:** Checked

4. Click **Save**.

Command Management

Command Name *

Example: check_example

Command Line *

Example: \$USER1\$/check_example -H \$HOSTADDRESS\$ -P \$ARG1\$ \$ARG2\$

Command Type:

Active ?

Available Plugins

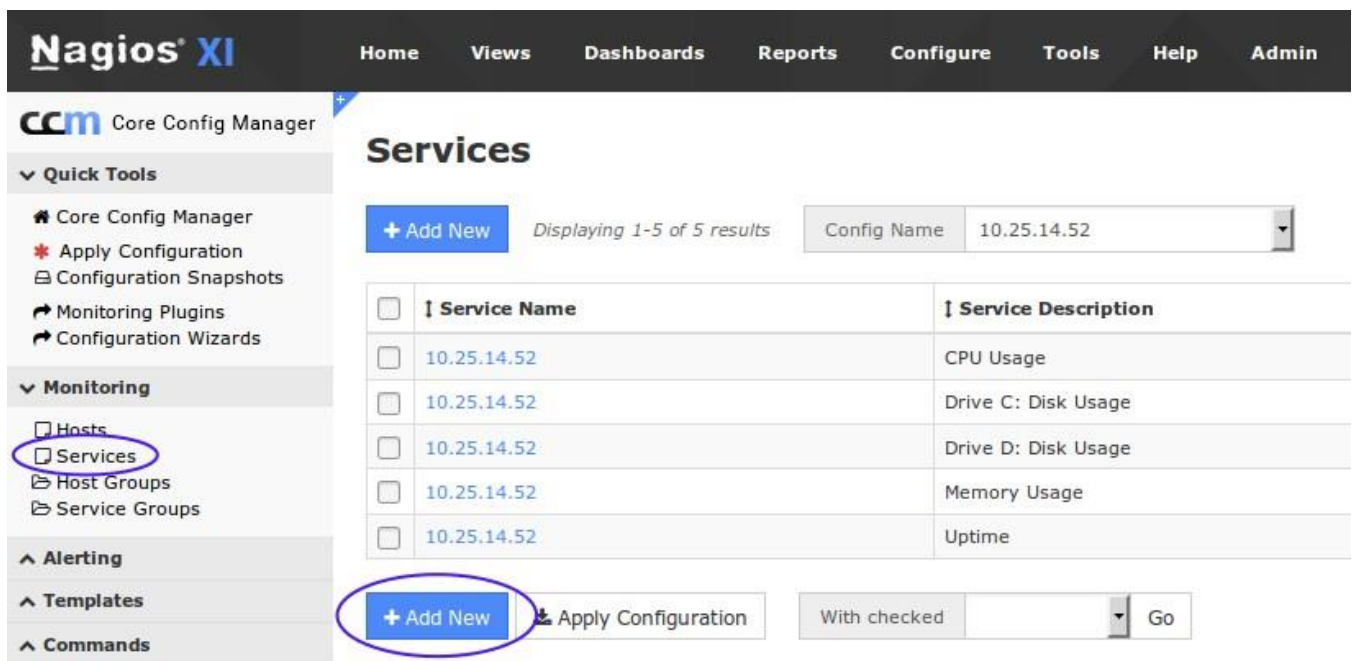
?

Integrating AutoIT With Nagios XI 5

Creating a Service for Autolt Monitoring

The final step is to create a new service definition that is associated with the remote windows host. This example assumes that this Windows host is already being monitored, so a HOST object already exists. If not, run the **Windows Server Configuration Wizard** and then return to this step. This guide uses the host **10.25.14.52** as an example:

1. Navigate to **Monitoring > Services** in CCM.
2. Click **Add New**.



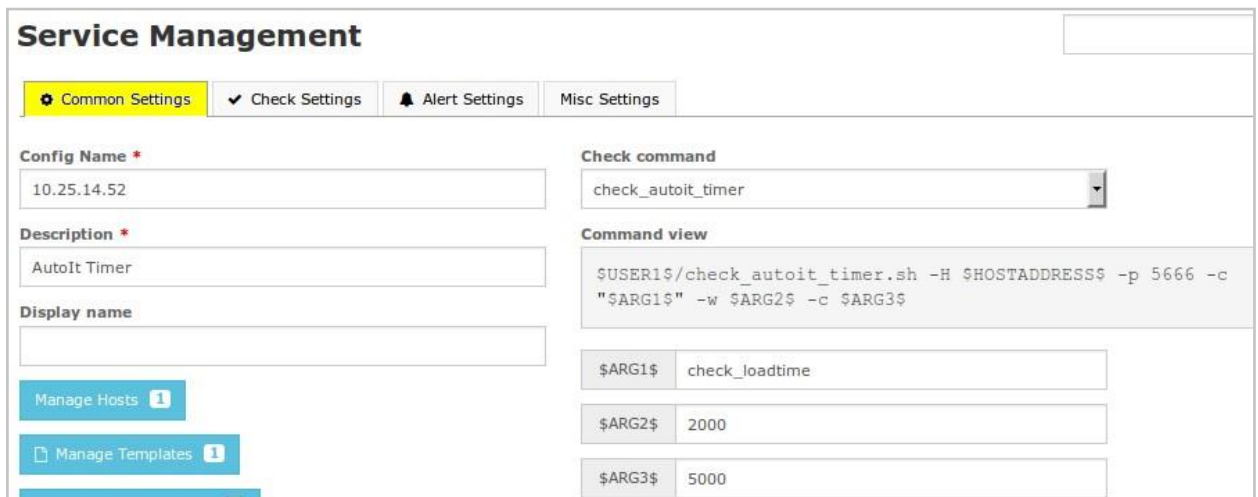
The screenshot shows the Nagios XI Core Config Manager (CCM) interface. The top navigation bar includes Home, Views, Dashboards, Reports, Configure, Tools, Help, and Admin. The left sidebar shows the Core Config Manager (CCM) menu with sections for Quick Tools, Monitoring, Alerting, Templates, and Commands. The 'Monitoring' section is expanded, and 'Services' is selected and circled in blue. The main content area displays the 'Services' page for host 10.25.14.52. A '+ Add New' button is circled in blue. Below the table, there is an 'Apply Configuration' button and a 'Go' button. The table lists the following services:

<input type="checkbox"/>	Service Name	Service Description
<input type="checkbox"/>	10.25.14.52	CPU Usage
<input type="checkbox"/>	10.25.14.52	Drive C: Disk Usage
<input type="checkbox"/>	10.25.14.52	Drive D: Disk Usage
<input type="checkbox"/>	10.25.14.52	Memory Usage
<input type="checkbox"/>	10.25.14.52	Uptime

3. Configure the **Common Settings** tab:
 - o **Config Name:** 10.25.14.52
 - o **Description:** Autolt Timer
4. Assign the service to a host:
 - o Click **Manage Hosts**.
 - o Select **10.25.14.52**, click **Add Selected >**, then **Close**.

Integrating AutoIT With Nagios XI 5

5. Apply the generic-service template:
 - Click **Manage Templates**.
 - Select **generic-service**, click **Add Selected >**, then **Close**.
6. Configure **Check Command**:
 - **Command:** check_autoit_timer
 - **\$ARG1\$:** check_loadtime
 - **\$ARG2\$:** 2000 (Warning threshold in ms)
 - **\$ARG3\$:** 5000 (Critical threshold in ms)
 - **Active:** Checked



The screenshot displays the 'Service Management' interface in Nagios XI. The 'Check Settings' tab is active. The configuration includes:

- Config Name:** 10.25.14.52
- Description:** AutoIt Timer
- Display name:** (empty)
- Check command:** check_autoit_timer
- Command view:**

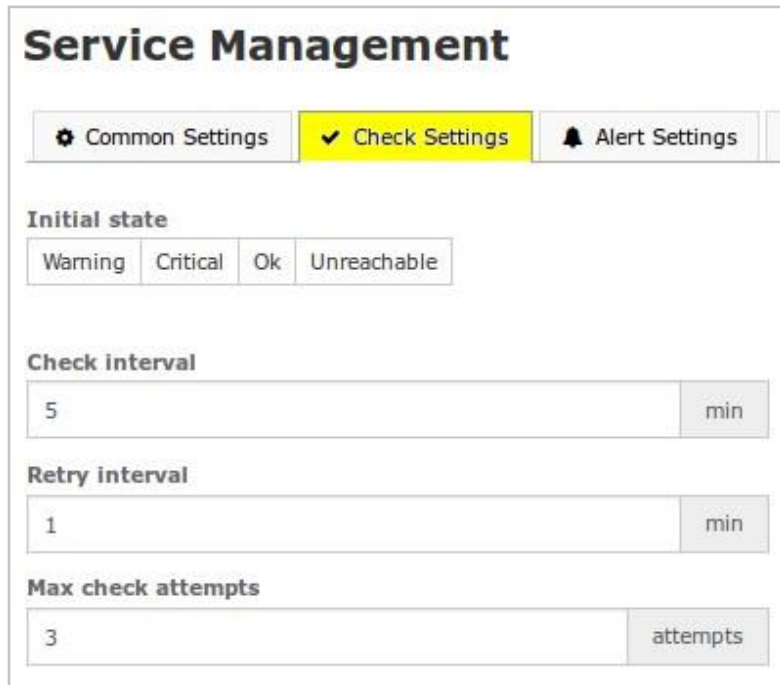
```
$USER1$/check_autoit_timer.sh -H $HOSTADDRESS$ -p 5666 -c "$ARG1$" -w $ARG2$ -c $ARG3$
```
- \$ARG1\$:** check_loadtime
- \$ARG2\$:** 2000
- \$ARG3\$:** 5000

Buttons for 'Manage Hosts' and 'Manage Templates' are visible at the bottom left.

Integrating AutoIT With Nagios XI 5

7. Set Check Intervals

- **Check interval:** 5
- **Retry interval:** 1
- **Max check attempts:** 3



Service Management

Common Settings **✓ Check Settings** Alert Settings

Initial state

Warning Critical Ok Unreachable

Check interval

5 min

Retry interval

1 min

Max check attempts

3 attempts

8. Click **Save**, then **Apply Configuration**.

Final Result

Once the service has been created, navigate to **Home > Service Detail** and search for the service. If the check has been configured correctly, it will appear like the example below.

Host	Service	Status	Duration	Attempt	Last Check	Status Information
10.25.14.52	AutoIt Timer	Warning	23s	1/3	2016-11-01 12:36:01	WARNING: Script took 4.2690 seconds to complete: time=4269ms:2000:5000:0

The Warning status shows that the **AutoIt** script is running more than 2000 milliseconds to load the test webpage.

Integrating AutoIT With Nagios XI 5

Finishing Up

This completes the documentation on integrating AutoIT with Nagios XI. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

[Visit Nagios Support Forum](#)

[Visit Nagios Knowledge Base](#)

[Visit Nagios Library](#)