

## The Industry Standard in IT Infrastructure Monitoring

### Purpose

This document describes how to monitor websites effectively with Nagios® XI™. There are three different wizards to help you monitor your websites. The Website Wizard is used to monitor the steady-state aspects of a website. The Website URL is very similar and allows you to monitor the status and content of a specific URL. Finally, the Web Transaction Wizard monitors transactions and other interactive activities on your website. All three are very useful for users to monitor the health of their websites, and to be notified when unexpected changes occur on the website or processes are not working as expected.

### Target Audience

This document is intended for use by both Nagios Administrators and end-users.

### Considerations

When monitoring websites, it is often recommended to check the operational status of several key metrics, including:

- HTTP response validity and load time
- DNS resolution and IP address match
- Website content
- SSL certificates
- Web transaction success and run time

Your monitoring needs will vary depending on the complexity of your website, its purpose, and its intended end-user.

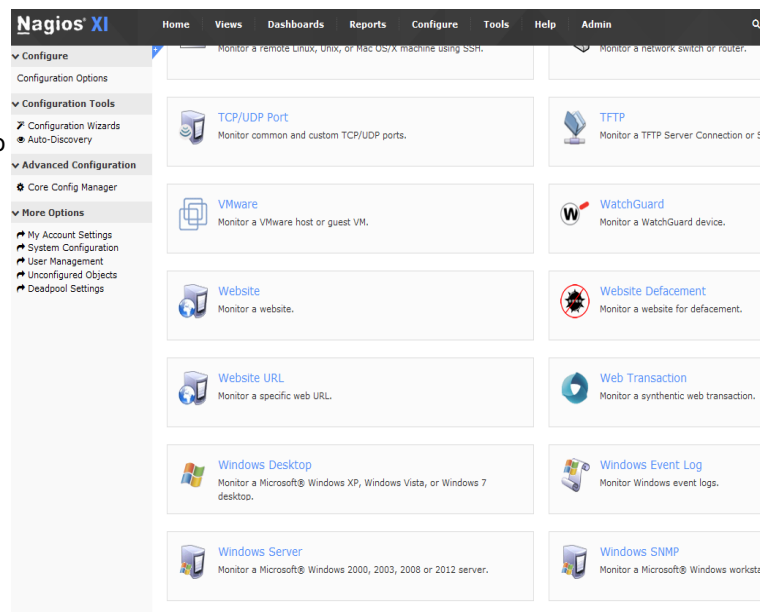
### Getting Started

Monitoring websites is made simple through the use of configuration wizards shipped with Nagios XI. There are two different ones for this category, with one checking more steady-state aspects of a web site and another for testing transactions and other interactive, dynamic activities on your site, named the **Website** and **Web Transaction** configuration wizards, respectively.

### The Website Wizard

This is the wizard you will use for most types of sites, where you are checking common server/site metrics. The best way to understand its capabilities is to see them, so a walk through of using this wizard follows.

Begin by visiting the configuration wizards page at **Configure** → **Monitoring Wizard**. Select the **Website** option.



Next, enter the URL to your website. This can be either the front page of your the domain or any sub-page. The latter will only have a purpose of checking of existence of that page and content monitoring on it.



## Configuration Wizard: Website - Step 1

Monitor a website.

Website URL:   
The full URL of the website you'd like to monitor.

The next step is where most of your options will be set. Here you can define which services you want to add for this site, including whether:

- To use SSL (HTTPS) and what port to use
- It returns a valid HTTP OK message
- It responds to ping
- DNS resolution appears to be working
- The DNS response matches what you had while running the wizard
- A particular string is found on the page (either literally or as a regular expression)
- The SSL certificate's expiry date is sufficiently far away

The **Use SSL** option and **SSL Certificate** check will be enabled if the URL you gave in Step 2 began with `https://`, and will not for other URLs. It is also possible to specify authentication credentials for if the page you are checking is protected by Basic authentication.



## Configuration Wizard: Website - Step 2

Don't Apply Configuration

### Website Details

Website URL:

Host Name:   
The name you'd like to have associated with this website.

IP Address:   
The IP address associated with the website fully qualified domain name (FQDN).

### Website Options

**Use SSL:**  Monitor the website using SSL/HTTPS.

**Port:**   
The port to use when contacting the website.

**On Redirect:**    
The port to use when contacting the website.

**Credentials:**    
**Basic authentication only.** The username and password to use to authenticate to the website (optional)

### Website Services

Specify which services you'd like to monitor for the website.

- HTTP**  
Includes basic monitoring of the website to ensure the web server responds with a valid HTTP response.
- Ping**  
Monitors the website server with an ICMP ping. Useful for watching network latency and general uptime of your web server. Not al

The next two steps are all similar to other wizards, and define the usual monitoring and notifications settings. You will want to keep in mind that each of the service checks is a hit on your web server, and you may not want to have those checks run too frequently if there is a risk of interfering with normal traffic. The defaults are likely fine for most situations and at any time you can accept the defaults by clicking **Finish**.

**Configuration Wizard: Website - Step 3** Don't Ask

Monitoring Settings

Define basic parameters that determine how the host and service(s) should be monitored.

**Under normal circumstances:**

Monitor the host and service(s) every  minutes.

**When a potential problem is first detected:**

Re-check the host and service(s) every  minutes up to  times before generating an alert.

**Configuration Wizard: Website - Step 4**

Notification Settings

Define basic parameters that determine how notifications should be sent for the host and service(s).

**When a problem is detected:**

Don't send any notifications

Send a notification immediately

Wait  minutes before sending a notification

**If problems persist:**

Send a notification every  minutes until the problem is resolved.

**Send alert notifications to:**

Myself ([Adjust my settings](#))

Other individual contacts

- Default Contact (xi\_default\_contact)
- Holden Smith (hsmith)

Specific contact groups

- All Contacts (xi\_contactgroup\_all)
- Nagios Administrators (admins)

Step 6 allows you to associate the Website to a Host Group, Service Group or identify a Parent Host.

**Configuration Wizard: Website - Step 5**

Host Groups

Define which hostgroup(s) the monitored host should belong to (if any).

- Linux Mod Gearman Workers (Gearman Workers)
- Linux Servers (linux-servers)
- Linux VM example group (Linux VMs)
- Windows Desktop PCs (Windows Desktops)

Service Groups

Define which servicegroup(s) the monitored service(s) should belong to (if any).

Parent Host

Define which host(s) are considered the parents of the the monitored host (if any).  
Note: Typically only one (1) host is specified as a parent.

192.168.4.11 (192.168.4.11)

The Final Step page allows you to click **Apply** which is the same as pressing the **Finish** button on previous screens.

**Configuration Wizard: Website - Final Step**

Final Settings

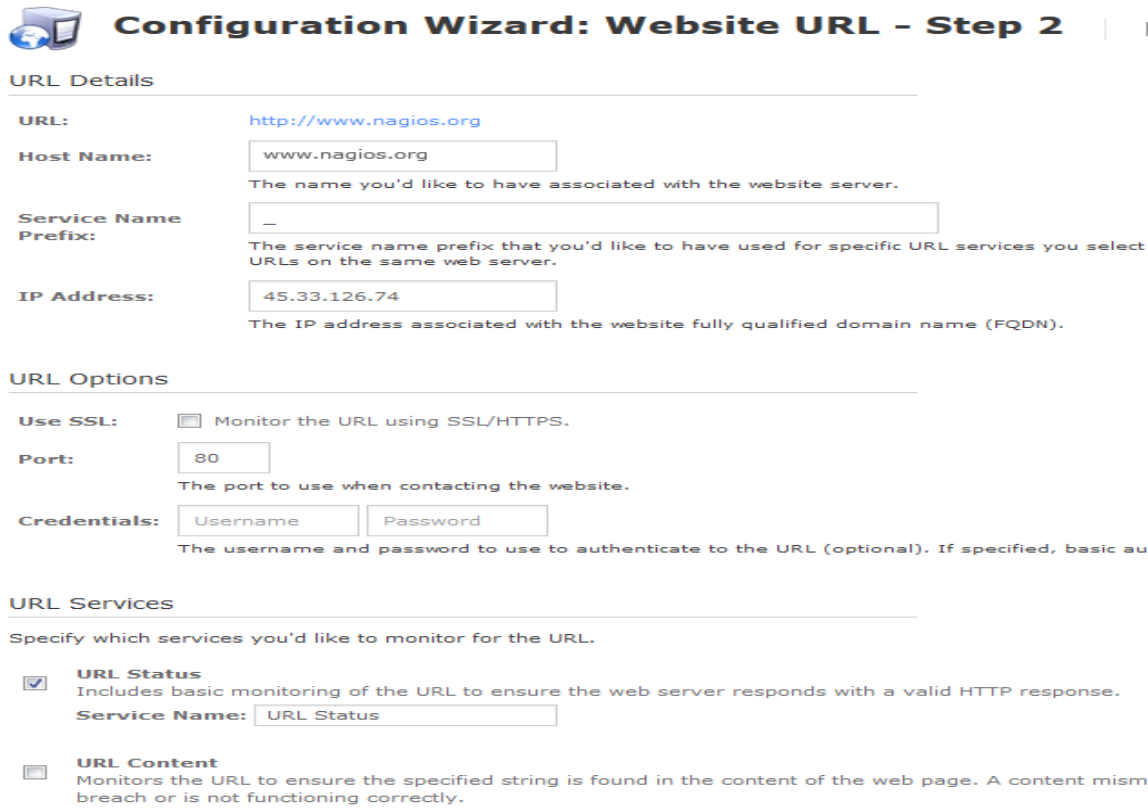
Click **Apply** to add your new configuration.

Once these steps are completed and the checks have run, you should see the results on the **Service Status** dashboard:

Host	Service	Status	Duration	Attempt	Last Check	Status Information
www.nagios.com	DNS IP Match	Ok	8s	1/5	2015-09-09 14:52:16	DNS OK: 0.090 seconds response time. www.nagios.com returns 45.33.179
	DNS Resolution	Ok	4s	1/5	2015-09-09 14:52:20	DNS OK: 0.008 seconds response time. www.nagios.com returns 45.33.179
	HTTP	Ok	25s	1/5	2015-09-09 14:51:59	HTTP OK: HTTP/1.1 200 OK - 59645 bytes in 0.287 second response time
	Ping	Ok	42s	1/5	2015-09-09 14:51:42	OK - www.nagios.com: rta 43.499ms, lost 0%

## The Website URL Wizard

The Website URL wizard is very similar to the Website wizard. It provides a URL status (does it return a valid HTTP OK message), as well as the ability to verify a particular string is found on the page (either literally or as a regular expression).



**Configuration Wizard: Website URL - Step 2**

**URL Details**

**URL:** <http://www.nagios.org>

**Host Name:**   
The name you'd like to have associated with the website server.

**Service Name Prefix:**   
The service name prefix that you'd like to have used for specific URL services you select URLs on the same web server.

**IP Address:**   
The IP address associated with the website fully qualified domain name (FQDN).

**URL Options**

**Use SSL:**  Monitor the URL using SSL/HTTPS.

**Port:**   
The port to use when contacting the website.

**Credentials:**    
The username and password to use to authenticate to the URL (optional). If specified, basic au

**URL Services**

Specify which services you'd like to monitor for the URL.

**URL Status**  
Includes basic monitoring of the URL to ensure the web server responds with a valid HTTP response.  
**Service Name:**

**URL Content**  
Monitors the URL to ensure the specified string is found in the content of the web page. A content mismatch or is not functioning correctly.

All subsequent steps are the same as the steps in the Website Wizard. Once all steps are completed and the checks have run, you should see the results on the **Service Status** dashboard:

Host	Service	Status	Duration	Attempt	Last Check	Status Information
www.nagios.org	_URL Status	Ok	1m 16s	1/5	2015-09-09 14:56:54	HTTP OK: HTTP/1.1 301 Moved Permanently - 534 bytes in 0.081 second response time

## The Web Transactions wizard

A more complex use case of website monitoring would be if you expect the content to be dynamic with user input and actions, and want to test that those actions complete as expected. For instance, you might test that a search box works (and what results are returned), whether the purchase and checkout process of your web store is behaving properly, or that a user can log in successfully. The **Web Transaction** wizard can be used for these types of checks. Additionally, it allows for checking all three of those in succession, and other multi-step procedures where each stage may be dependent on the previous one.

## Configuration Wizard: Web Transaction - Step 1

Web Information

Monitoring a synthetic web transaction is a process which may involve several steps, including the submission so you must be familiar with its syntax before monitoring a transaction.

Transaction Name:   
The name you'd like to have associated with this synthetic transaction test.

Primary URL:   
The primary URL that this transaction is associated with.

This wizard relies on a tool called *WebInject*, which handles the transition logic between stages of the transaction. Therefore you will need to understand how to write the configuration XML in the WebInject syntax to configure these kinds of checks. Some examples are given below, and the WebInject manual can be found online at <http://www.webinject.org/manual.html>.

Note that certain special characters need to be escaped. For instance, the < should be replaced with \x3C so as not to interfere with the XML, and within POST data URL escapes are used, so for instance @ becomes %40.

This syntax will go into the large **Test Case Data** input box in Step 3 of the wizard.

### Example #1: Searching The Nagios Support Forum

```
<testcases repeat="1">
<case
  id="1"
  url="http://support.nagios.com/forum/"
/>
<case
  id="2"
  method="post"
  url="http://support.nagios.com/forum/./search.php"
  postbody="keywords=foobar&submit=Search"
  verifypositive="\x3Cp>No posts were found because the word \x3Cstrong>foobar\x3C/strong> is
not contained in any post.\x3C/p>"
/>
</testcases>
```

In this example, first the main forum page is loaded, which will make sure it appears to be present and working. The second step submits a search for the word “foobar”, and checks to make sure that the results say that no posts exist using that phrase. Instead of “foobar” you might use something like “Internet Explorer”, such that you could alert your CSS guru when someone reported something that was broken.

## Configuration Wizard: Web Transaction - Step 2

Transaction Host Details

Primary URL:   
The primary URL that this transaction is associated with.

Host Name:   
The name you'd like to have associated with the primary URL.

IP Address:   
The IP address associated with the primary URL's fully qualified domain name (FQDN).

Transaction Details

Specify the details of how the transaction should be monitored.

Transaction Name:   
The name you'd like to have associated with this synthetic transaction test.

Test Case Data:   
Transaction test case data must be formatted according to WebInject standards. Read the WebInject test case documentation for more information on creating test case data.

```
<id="1"
url="http://support.nagios.com/forum/"
/>
<case
id="2"
method="post"
url="http://support.nagios.com/forum/./search.php"
postbody="keywords=foobar&submit=Search"
verifypositive="\x3Cp>No posts were found because the word \x3Cstrong>foobar\x3C/strong>
not contained in any post.\x3C/p>"
/>
</testcases>
```

Timeout:  seconds  
The response timeout for each test case.

Global Timeout:  seconds

**Example #2: Using An Online Store**

```

<testcases repeat="1">
  <testvar varname="USER">1rc94d+86yw3m9jrqj18@sharklasers.com</testvar>
  <testvar varname="PASS">holden123</testvar>

  <case
    id="1"
    description1="Login page"
    url="https://members.oreilly.com/account/login"
    parseresponse='_authentication_token' type="hidden" value="|" |escape'
    verifypositive="Sign in"
  />
  <case
    id="2"
    description1="Sign in"
    url="https://members.oreilly.com/account/login"
    method="post"
    postbody="email=${USER}&password=${PASS}&_authentication_token={PARSEDRESULT}"
    verifypositive="https://members.oreilly.com/account/benefits"
    parseresponse="found at |;"
  />
  <case
    id="3"
    description1="Members page"
    url="{PARSEDRESULT}"
    verifypositive="view or edit your account information"
  />
  <case
    id="4"
    description1="Book price"
    url="http://oreilly.com/catalog/9781593271794/"
    verifypositive="59.95"
  />
  <case
    id="5"
    description1="Book added to cart"
    url="https://epoch.oreilly.com/shop/cart.orm?prod=9781593271794.BOOK"
    verifypositive="Nagios, 2Ed"
  />
  <case
    id="6"
    description1="Book still in cart"
    url="https://epoch.oreilly.com/shop/cart.orm"
    verifypositive="Nagios, 2Ed"
    verifynegative="Backorder"
  />
  <case
    id="7"
    description1="Logout"
    url="https://members.oreilly.com/account/logout"
    verifypositive="http://oreilly.com/"
    parseresponse="found at |;"
  />
  <case
    id="8"
    description1="Main page"
    url="{PARSEDRESULT}"
    verifypositive="News & Commentary"
  />

```

</testcases>

This obviously more complicated example begins to show the power WebInject offers using O'Reilly Media's web site. The first step confirms that the login page loads. The second provides your authentication credentials and then checks that they were accepted, and follows the redirect to the members page in step 3. Step 4 checks the price on “[Nagios, Second Edition \(by Wolfgang Barth\)](#)”, and step 5 adds it to your shopping cart, with step 6 confirming it remains in your cart properly after that and appears to be in stock. Finally, the last two steps log you out and check that the home page loads. By carefully crafting the different steps and plenty of sufficiently specific verifypositive and verifynegative parameters, a great deal of information can be confirmed through this single Nagios service.

## The Result

Once you have gone through all of the steps above, Nagios XI will have created two service checks. The results should look something like this:

Host	Service	Status	Duration	Attempt	Last Check	Status Information
oreilly.com	Web Transaction	Ok	2m 7s	1/5	2015-09-09 15:18:35	WebInject OK - All tests passed successfully in 6.931 seconds
support.nagios.com	Web Transaction	Ok	2m 20s	1/5	2015-09-09 15:18:22	WebInject OK - All tests passed successfully in 0.897 seconds

## Finishing Up

This completes our tutorial of how to monitor a website with Nagios XI. If you still have questions about how to use the website wizards, or for any other support related questions, please visit the [Nagios Support Forums](#) at:

<http://support.nagios.com/forum/>