Nagios Reactor – Technical Overview and Terminology Definitions

Nagios®

The Industry Standard in IT Infrastructure Monitoring

Purpose

This document is designed to provide an explanation of what Nagios Reactor is, how it can be used, and to define terminology related to it.

Target Audience

This document is intended for use by Nagios Reactor users and administrators.

High Level Overview

Nagios Reactor is an application that allows users to define and manage complex automation and workflow rules easily. What that means in plain English is that it can be used to automatically execute multi-step tasks that would otherwise have to be performed manually or by (potentially) complex scripts or programs.

An Example Application

An obvious application where Nagios Reactor could be used would be as an advanced system to handle problems that are detected by either Nagios Core or Nagios XI.

Both Nagios Core and Nagios XI are capable of running special “event handler” commands when problems are detected with network infrastructure components (e.g. if a website or server goes down). System administrators often write event handler scripts to try and automatically fix problems without human interaction. This is useful when simple or common problems can be fixed easily without requiring an administrator to be contacted after hours to fix a problem themselves.

An example event handler is a script that attempts to automatically restart a web server if it has stopped serving web pages. It would detect if Apache or a requisite service has stopped running on the web server and send an ssh command with user/password and trigger an init script that would restart the service as needed. If this kind of operation would continuously monitor the web server and report on how the triggered init script resolved, you could potentially never have to log into the server manually until Nagios Reactor indicates to the user that the fail safe script failed.

Less Obvious Applications

Nagios Reactor can do much more than just execute “fix” scripts to resolve IT infrastructure problems. It can be also used to perform any multi-step task that can be automated by using the automation features of Nagios Reactor workflow. Internal company processes such as automated order fulfillment, application deployment, and testing can also be implemented using Nagios Reactor.

A General Purpose Machine

For the technically inclined, Nagios Reactor is really a general purpose application development platform. Nagios Reactor contains the equivalents of things you would use in writing an application using many scripting and programming languages. There are equivalents to functions, function arguments, local and global variables, boolean logic, if-then-else logic, while loops, and foreach loops.

The Benefits of Nagios Reactor Over Old School Scripts

Nagios Reactor can be leveraged in order to make existing automation scripts easier to understand and easier to maintain by both highly skilled technical administrators and other less technically experienced users in an organization. Nagios Reactor's event chains, blocks, conditions and actions are easier to create, understand, and maintain than complex scripts that are often utilized by system administrators. Administrators can also stash blocks, export and import actions and conditions and other customizable tools to create a specific tool set for any user on a Nagios Reactor server. This way, the administrator engineers the operations and shares them with other users, resulting in a much lower margin for error.

Revision 1.0 – August, 2014

Reactor Terminology

Event Chains

Event chains can be thought of as standalone programs or scripts. They might be designed to fix a problem, automate a common task, or handle a security alert. Any process or procedure that 1) consists of multiple steps, 2) has a basic logic flow or decision making process (if this, then that, else that), and 3) can be accomplished with scripts or programs that can theoretically be converted to a Nagios Reactor event chain.

Each event chain consists of one or more steps (called “Blocks”).

Event chains can be launched/started/triggered by the following methods:

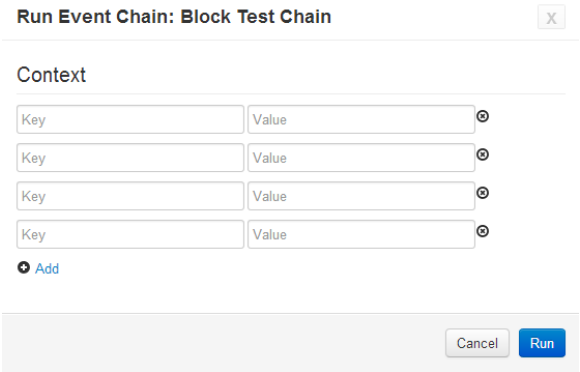
- Manually by the user through the Nagios Reactor web interface
- By a scheduled task (on a Windows box) or a cron job (on a Linux/Unix box)
- By a “Cron” (an internal Nagios Reactor scheduler)
- By an external application through Nagios Reactor API

Individual event chains can be secured to allow only specific users the ability to edit or launch the chain.

Event chains can be thought of in two ways - either as single functions in a scripting or programming language or as entire programs that consist of multiple sub-functions and steps.

Context Variables

Context variables are optional data that can be supplied to event chains when they are launched (to the right is the context variable model that will launch anytime you click the **Run Chain** button in Nagios Reactor). Context variables consist of a key and value pair. They are used to pass dynamic information to an event chain. Event chains can examine context variables and take different actions based on the value of a variable key or value. Context variables can be modified by event chains as the chain is executing.



Context variables are local in scope to the event chain they were passed to. In plain English, this means the variables are not accessible or available to other event chains.

Context variables are like arguments that can be passed to functions in scripting or programming languages.

Examples of some possible context variables (key/value pairs) are shown below:

Name	Value
hostname	www.nagios.com
address	192.168.5.1
state	OK
URL	http://www.nagios.com/products

Workflow

A workflow can be thought of as all the steps ("blocks") that comprise an event chain, as well as the order in which they are executed. Each event chain has one and only one workflow. These are described in more detail in the Event Chain and Block Types documentation.

Blocks

Blocks are steps within an event chain. They comprise the event chain's workflow. Blocks are evaluated/processed/executed in the order they are defined within an event chain. They can be a basic block with a set number of actions or a logic block with a If-then-else else style workflow, which is more capable of complex state checking and operation execution.

An example of how event chains and blocks relate to each other is the order placement process you go through when buying something from Amazon.com. The entire order process could be replicated as a Nagios Reactor event chain. The individual steps of the order process (e.g. adding a product to the shopping cart, clicking the checkout button, entering shipping and billing data, and submitting the order) could each be though of as a block.

Blocks can be added/edited/deleted from the chain by editing the **Workflow** tab of the event chain within the Reactor interface. Once you select an event chain from the list of current event chains, it is the second tab above the **Edit Details** button. Blocks can also be saved through the 'stash' panel on the right side of the user interface - while inside the workflow tab. This allows users to stash their own blocks for use in other locations, while working on an event chain, to be able to repeat a block with a few new variables made by that user or by making them global so that any user on that Nagios Reactor server can access and use the block. This is very useful for administrators that give out tasks to other users and wish to specify what blocks to use for what is needed in the even chain.

Basic Blocks

Basic blocks consist of only one or more actions that are taken in sequential order. There is no if-then-else logic present in the processing of actions. A basic block that defines steps to send an email might be to simply take the following actions in order: address the email to one or more persons, set a subject line, compose the message, and click the **send** button. Basic blocks can also perform large amount of actions and are great for executing commands in the Linux command line.

Logic Blocks

Logic blocks consist of both conditions and actions to form a more complex and capable set of actions with if-then-else logic.

Logic blocks work by evaluating one or more conditions. If one or more conditions are true, a certain set of actions can be taken. If the conditions are false, another different set of actions can be taken. This is where the power of the logic block lies. You can simply take the output of an operation to launch tasks - based on how the logic finished the previous step.

Examples of logic blocks in a real world process can be explained in how our business operates. Here are a few:

1. **If** today is Friday (condition), **then** the office closes at 4pm (true action), **else** the office closes at 5pm (false action).
2. **If** today is Tuesday (condition), **then** we have a team meeting (true action). Note that no false action is defined in this example – it isn't required in Nagios Reactor either).

Stashed Blocks

Stashed blocks are simply a handy way to move one or more blocks out of the event chain into a "stash" (or "toolbox" if you will) for later use. Blocks that are stashed can be inserted back into the event chain (from which they were stashed) at a later time. Stashed blocks in one event chain cannot be seen or used in other event chains.

Global Blocks

Global blocks are a method of saving one or more blocks in a special area that allow them to be used in multiple event chains by all users on that server. They can be useful if administrators want to define complex or commonly used blocks that are intended for use in multiple event chains. By utilizing global blocks, users can define event chains more quickly and create a workflow of blocks - making everything more efficient.

Conditions

Conditions are the “if” part of the if-then-else logic in blocks. They are usually tests that Reactor performs to check whether something is true, something exists, or something “works”. Each block can have multiple conditions. Some example conditions that currently exist in Reactor include tests for:

1. What day of the week it is
2. What time of the day it is
3. Whether a particular file exists in a specific directory
4. HTTP request evaluates to
5. Script evaluates to
6. Many more..

Most conditions are capable of checking context variables to make their decision as to whether or not something is true or false. There is also a 'Context variable evaluates to' condition to check for a expected return or error value.

Actions

Actions are the parts that make the Nagios Reactor functionality work. They are the “worker” portion of Nagios Reactor. Each block within an event chain can have one or more actions that are executed if a set of conditions is or is not met. Some example actions that currently exist in Nagios Reactor include:

1. Send an email
2. Fetch a web page
3. Create a file
4. Delete a file
5. Open a new incident in Nagios Incident Manager
6. Many more..

Metadata

Metadata is simply data that event chains can store, modify, and reference at a later point. Metadata – unlike context variables – is accessible between different event chains. Metadata is useful for storing variable/value data pairs for reference in a future run of an event chains. One useful metadata usage case is that it can be used to increment or decrement variable values to accomplish with loops within an event chain - or across multiple event chains.

For the more technical, metadata is akin to global variables or shared memory in an application. Metadata – like global variables – can be referenced and modified by different functions (event chains) or threads of an application.

Crons

A Cron is one method of launching or starting individual event chains on a regular, scheduled basis – similar to a cron job on a Linux system or a scheduled task on a Windows system. Crons are defined by an administrator in the Nagios Reactor web interface under **Administration** → **Cron**.

Event Chain Import / Export

Event chains can be exported from Nagios Reactor for the purposes of making a backup or sharing the chain with another user. Event chains can also be imported into Reactor from a previously exported chain (or properly crafted JSON string). This is a fantastic feature that you can use to facilitate some end-user innovation. Users will eventually be able to share their event chains (along with custom conditions and actions) on Nagios Exchange. Refer to the Exporting and Importing Event Chains in Nagios Reactor documentation for a detailed description on how to create, save and use this functionality effectively.

Finishing Up

If you have questions about Nagios Reactor or its capabilities, contact our support team via our online forum at: http://support.nagios.com/forum

Nagios Enterprises, LLC
P.O. Box 8154
Saint Paul, MN 55108
USA

US: 1-888-NAGIOS-1
Int'l: +1 651-204-9102
Fax: +1 651-204-9103

Web: www.nagios.com
Email: sales@nagios.com