Nagios XI - Integrating ServiceNow

Article Number: 552 | Rating: 5/5 from 2 votes | Last Updated: Fri, Aug 14, 2020 at 4:17 PM

Overview

This guide will show you how to integrate Nagios XI with ServiceNow.

ServiceNow is an online application that provides incident management. Nagios XI can be configured to automatically send notifications to ServiceNow to appear as incidents.

Note: Because development on the Service Now Ticketer package on the Exchange stopped in 2014, the integration between Nagios XI and ServiceNow using the Service Now Tick

Download Package

You will need to download the Service Now Ticketer from here:

https://exchange.nagios.org/directory/Plugins/Helpdesk-and-Ticketing/Service-Now-Ticketer/details

Once downloaded, extract the files as you'll need to upload them as you follow this guide.

ServiceNow Account Details

You will need a username, password and URL to insert to the configuration file.

The URL will be something like **customername**.service-now.com and will be required in the configure section later.

Install Perl Pre-Requisites

Open an SSH / Console session to your Nagios XI server.

Perl modules need to be installed using these commands:

```
yum install -y perl-DBI perl-XML-Simple perl-SOAP-Lite perl-Config-IniFiles perl-Digest-MD5-File
```

Some additional Perl modules need installing through CPAN (they are not available through yum as a package).

First run cpan to make sure it's configured:

```
cpan
```

If you are presented with the **cpan[1]>** prompt then it is configured, you can exit:

```
exit
```

The install commands are after these following steps.

If you get output such as:

```
Sorry, we have to rerun the configuration dialog for CPAN.pm due to
some missing parameters...

The following questions are intended to help you with the
configuration. The CPAN module needs a directory of its own to cache
important index files and maybe keep a temporary mirror of CPAN files.
This may be a site-wide or a personal directory.
```

Then CPAN is not configured yet. All you need to do is press **Enter** to accept the default option provided (there are several prompts).

If prompted for a CPAN mirror, type the following:

```
http://www.perl.com/CPAN/
```

When you are finished you will arrive at the **cpan[1]>** prompt.

Type the following to save the settings you just made:

```
o conf commit
```

Now you can exit as the install commands are done outside of cpan.

```
exit
```

Installation of Perl modules using CPAN is performed using these commands:

```
cpan Getopt::Whatever
```

Wait while the module is installed. If you run the command again it will confirm they are installed with *Getopt::Whatever is up to date (x.xx)*.

```
cpan Config::INI::Reader
```

You will receive this message:

You will receive this message:

```
---- Unsatisfied dependencies detected during ----
----       RJBS/Config-INI-0.025.tar.gz       ----
    Mixin::Linewise::Writers [requires]
    Mixin::Linewise::Readers [requires]
    Test::More [requires]
Shall I follow them and prepend them to the queue
of modules we are processing right now? [yes]
```

Press **Enter** to install the dependencies *(you may be presented with this multiple times)*.

Wait while the module is installed. If you run the command again it will confirm they are installed with *Config::INI::Reader is up to date (x.xxx)*.

Create Database

The ServiceNow integration requires a database to be created which will be called **nagsnt**.

There will also be a dedicated database user account called **servicenow** with the password **servicenow_password** *(we suggest you use a more secure password)*.

The storage location of the database will be the default location that MySQL / MariaDB uses, this can be changed however it is not covered in this guide. If your database is offloadec

Nagios XI has the default root password of '**nagiosxi**', if this has been changed you will need to use your password in the following commands.

This command will connect to the local MySQL / MariaDB database engine interface.

```
mysql -u root -p'nagiosxi'
```

Now execute these four commands *(press **Enter** after each command)*:

```
CREATE DATABASE nagsnt DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
CREATE USER 'servicenow'@'localhost' IDENTIFIED BY 'servicenow_password';
```

```
GRANT USAGE ON *.* TO 'nagsnt'@'localhost' IDENTIFIED BY 'servicenow_password' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 M
```

```
GRANT ALL PRIVILEGES ON nagsnt.* TO 'servicenow'@'localhost' WITH GRANT OPTION ;
```

Now you can exit the local MySQL / MariaDB database engine interface.

```
\q
```

Run this command to ensure that the database has been created:

```
echo 'show databases;' | mysql -u servicenow -p'servicenow_password' -h localhost
```

The last command should output something like:

```
Database
information_schema
nagsnt
test
```

Install Script + Config Files

You will now upload the files you extracted earlier.

Open Nagios XI and Navigate to Admin > System Extensions > **Manage Plugins**.

Click the **Browse** button

Browse the the extracted files and upload **config.ini**

Repeat these steps for **field_map.ini** and **sn_ticker.pl**

In your SSH / Console session execute the following commands to edit the **config.ini** file:

```
cd /usr/local/nagios/libexec
vi config.ini
```

When using the vi editor, to make changes press **i** on the keyboard first to enter insert mode. Press **Esc** to exit insert mode.

You will need to change these settings:

```
; Enable ticket track.
enable_tickettrack = 0
```

```
; The database settings are only required if you are using ticket track and the tt_type = db
[database]
db_type = mysql
db_username = servicenow
db_password = servicenow_password
db_port = 3306
db_address = 127.0.0.1
db_database = nagsnt

; The credentials required to log in to service-now
[servicenow]
sn_username = servicenow_user
sn_password = password
sn_url = myinstance.service-now.com
```

Specifically these settings relate to the database that was created earlier:

```
db_username = servicenow
db_password = servicenow_password
db_address = 127.0.0.1
```

Specifically these settings relate to the Service-Now information that you obtained earlier:

```
sn_username = servicenow_user
sn_password = password
sn_url = myinstance.service-now.com
```

When you have finished, save the changes in vi by typing **:wq** and pressing **Enter**.

Execute the following command to set the **field_map.ini** file to defaults:

```
sed -i '/^; This is an example entry./Q' field_map.ini
```

Build Database + Configure

You will now run a script to build the database.

In your SSH / Console session execute the following command:

```
/usr/local/nagios/libexec/sn_ticketer.pl --builddb
```

If the script worked the following output is produced:

```
Database table created succesfully.
```

Execute the following command that will generate the mappings for the **field_map.ini** file:

```
/usr/local/nagios/libexec/sn_ticketer.pl --page='incident.do' --buildmap
```

If the script worked the the **field_map.ini** file will contain the following:

```
;
; The INI Heading is the page to edit and the fields to enter data into are the options for that heading.
; The following special characters are evaluated in the order listed, they must not contain spaces!
; $WORD$ is a substitution option to be specified as an argument
; &EXPRESSION& should be mathematical expression that will be evaluated.
; %PAGE:FIELD:WORD% Using this will lookup the object ID of another service-now entry.
; ^ Fields prefixed with this symbol will be updated
; @ Arguments of fields prefixed with ^ and this symbol will be appended instead of being overwritten.
; WHITESPACE BETWEEN META CHARACTERS IS NOT ALLOWED.


[incident.do]
; string
closed_at = $closed_at$
; integer
incident_state = $incident_state$
; string
group_list = $group_list$
; string
closed_by = $closed_by$
; string
assignment_group = $assignment_group$
; string
category = $category$
; string
expected_start = $expected_start$
; string
description = $description$
```

```
; string
watch_list = $watch_list$
; string
upon_reject = $upon_reject$
; integer
reopen_count = $reopen_count$
; string
subcategory = $subcategory$
; string
rfc = $rfc$
; string
delivery_task = $delivery_task$
; string
work_notes = $work_notes$
; string
comments_and_work_notes = $comments_and_work_notes$
; string
time_worked = $time_worked$
; string
number = $number$
; string
resolved_by = $resolved_by$
; string
resolved_at = $resolved_at$
; string
sla_due = $sla_due$
; string
correlation_display = $correlation_display$
; string
delivery_plan = $delivery_plan$
; string
activity_due = $activity_due$
; boolean
knowledge = $knowledge$
; boolean
made_sla = $made_sla$
; integer
hold_reason = $hold_reason$
; string
opened_at = $opened_at$
; string
parent_incident = $parent_incident$
; string
approval_set = $approval_set$
; integer
priority = $priority$
; integer
reassignment_count = $reassignment_count$
; string
comments = $comments$
; integer
state = $state$
; integer
order = $order$
; string
assigned_to = $assigned_to$
; string
business_duration = $business_duration$
; integer
child_incidents = $child_incidents$
; integer
calendar_stc = $calendar_stc$
; string
parent = $parent$
; string
location = $location$
; boolean
active = $active$
; string
calendar_duration = $calendar_duration$
; string
follow_up = $follow_up$
; string
short_description = $short_description$
; string
work_start = $work_start$
; string
due_date = $due_date$
; string
user_input = $user_input$
; integer
impact = $impact$
; string
approval_history = $approval_history$
; string
opened_by = $opened_by$
; string
problem_id = $problem_id$
; string
business_service = $business_service$
```

```
; string
upon_approval = $upon_approval$
; string
company = $company$
; string
caused_by = $caused_by$
; integer
notify = $notify$
; integer
severity = $severity$
; string
additional_assignee_list = $additional_assignee_list$
; integer
business_stc = $business_stc$
; integer
urgency = $urgency$
; string
close_code = $close_code$
; string
approval = $approval$
; string
contact_type = $contact_type$
; string
close_notes = $close_notes$
; string
work_end = $work_end$
; string
cmdb_ci = $cmdb_ci$
; string
correlation_id = $correlation_id$
; integer
escalation = $escalation$
; string

caller_id = $caller_id$
; string
work_notes_list = $work_notes_list$
```

Information on what all this means is explained at the end of this guide under the "Field Mapping" section.

Create Notification Commands

You will now create the notification commands in Nagios XI.

Open Nagios XI and Navigate to Configure > **Core Configururation Manager** (CCM).

Expand Commands and click **>_Commands**

Click the **+ Add New** button

Command Name:

```
notify-host-by-snt
```

Command Line:

```
$USER1$/sn_ticketer.pl --page="incident.do" --host="$HOSTNAME$" --state="$HOSTSTATE$" --category="Nagios" --short_description="$
```

Command Type:

misc command

Active:

Checked

Click **Save**

Click the **+ Add New** button

Command Name:

```
notify-service-by-snt
```

Command Line:

```
$USER1$/sn_ticketer.pl --page="incident.do" --host="$HOSTNAME$" --state="$HOSTSTATE$" --category="Nagios" --short_description="$
```

Command Type:

misc command

Active:

Checked

Click **Save**

Create ServiceNow Nagios XI User (Contact)

To be able to send to ServiceNow you need a contact object created in CCM. The simplest way to do this is to create a Nagios XI user account for ServiceNow and that in turn will cr

Navigate to Admin > Users > **Manage Users**.

Click **Add New User**

- General Settings
  - Username: **ServiceNow**
  - Password & Repeat Password *(accept the auto generated password, you won't need to use it)*
  - Force Password Change at Next Login: **Un-check**
  - Email User Account Information: **Un-check**
  - Username: **ServiceNow**
  - Email: **ServiceNow@localhost** *(it's not important and is not used)*
  - Create as Monitoring Contact: **Checked**
  - Enable Notifications: **Checked**
  - Account Enabled: **Checked**
- Security Settings
  - Authorization Level: **User**
  - Can see all objects: **Checked**
  - Has read-only access: **Checked**
- Click the **Add User** button to create the user

The new user should appear in the list.

Assign Notification Commands To Contact

You will now assign the notification commands in CCM to the ServiceNow contact.

Navigate to Configure > **Core Configururation Manager** (CCM).

Expand Alerting and click **Contacts**

Click the **servicenow** contact to edit it

- **Alert Settings** tab
  - Click the **Manage Host Notification Commands** button
    - In the **left** pane select the **notify-host-by-snt** command
    - Click the **Add Selected >** button
    - Click the **Close** button
  - Click the **Manage Service Notification Commands** button
    - In the **left** pane select the **notify-service-by-snt** command
    - Click the **Add Selected >** button
  - Click the **Close** button
- Click the **Save** button

Click the **Apply Configuration** button

Assign Contact To Host And Service Objects

The last part of this step is to assign the contact to host and service objects that you want to send notifications to ServiceNow.

In this example, the built in service "Service Status - ntpd" for the localhost object will be used. We can easily stop the service to force the Nagios service to fail and hence the notifica

Navigate to Configure > **Core Configururation Manager** (CCM).

Expand Monitoring and click **Services**

Click the **Service Status - ntpd** service to edit it

- **Alert Settings** tab
  - Click the **Manage Contacts** button
    - In the **left** pane select the **servicenow** contact
    - Click the **Add Selected >** button
    - Click the **Close** button

Click the **Save** button

Click the **Apply Configuration** button

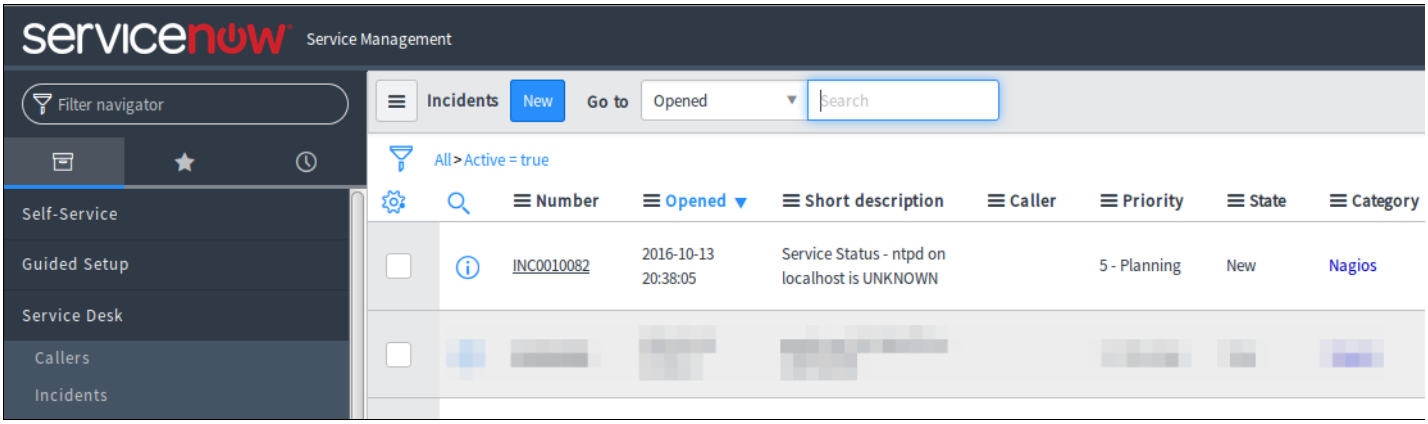
Now to cause the service to fail by stopping the service:

```
service ntpd stop
```

Watching the `/usr/local/nagios/var/nagios.log` we can see it fail and send the alert:

```
[1476416134] SERVICE ALERT: localhost;Service Status - ntpd;UNKNOWN;SOFT;1;ntpd is stopped
[1476416182] SERVICE ALERT: localhost;Service Status - ntpd;UNKNOWN;SOFT;2;ntpd is stopped
[1476416240] SERVICE ALERT: localhost;Service Status - ntpd;UNKNOWN;SOFT;3;ntpd is stopped
[1476416283] SERVICE ALERT: localhost;Service Status - ntpd;UNKNOWN;HARD;4;ntpd is stopped
[1476416283] SERVICE NOTIFICATION: nagiosadmin;localhost;Service Status - ntpd;UNKNOWN;xi_service_notification_handler;ntpd is stopped
[1476416283] SERVICE NOTIFICATION: servicenow;localhost;Service Status - ntpd;UNKNOWN;notify-service-by-snt;ntpd is stopped
```

You will see something similar on the ServiceNow Incidents page:



Now to let the service recover by starting the service:

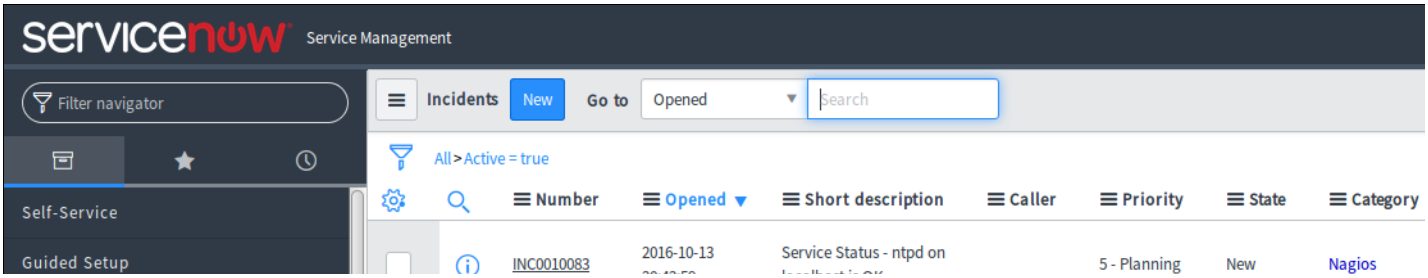
```
service ntpd start
```

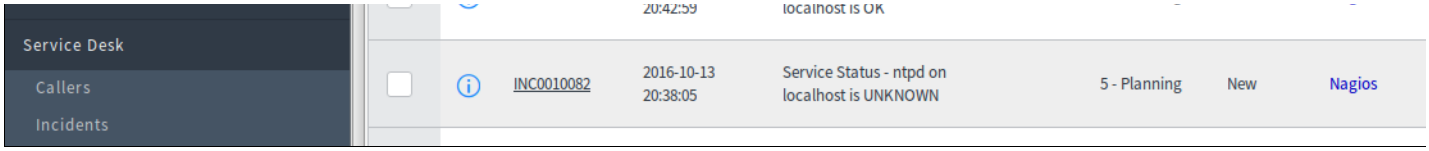
Watching the `/usr/local/nagios/var/nagios.log` we can see it recover and send the alert:

This is a success/output message from running a command:

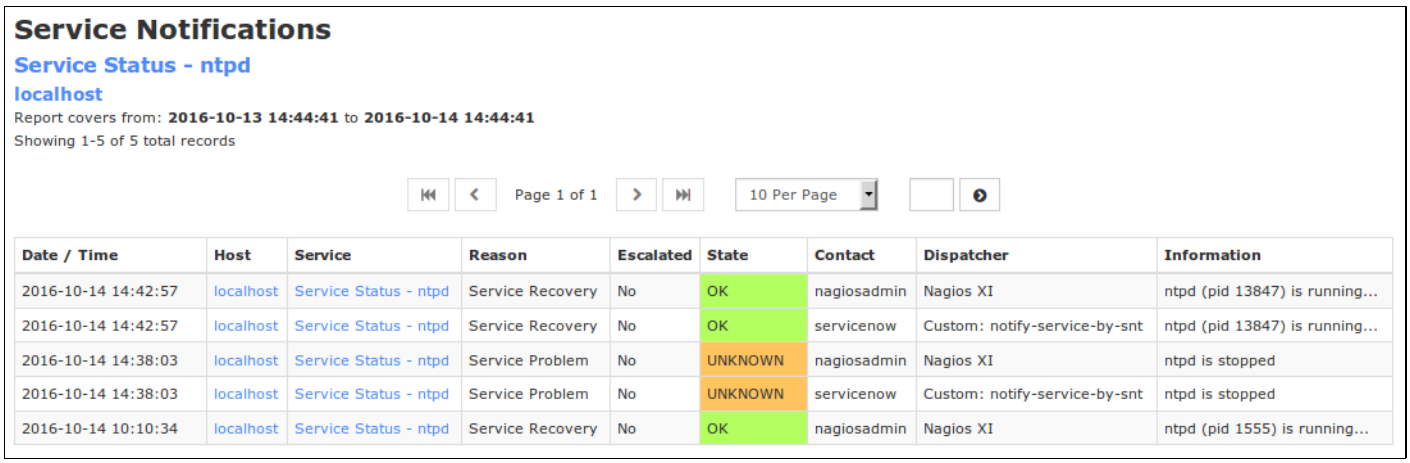
```
[1476416577] SERVICE ALERT: localhost;Service Status - ntpd;OK;HARD;4;ntpd (pid  13847) is running...
[1476416577] SERVICE NOTIFICATION: nagiosadmin;localhost;Service Status - ntpd;OK;xi_service_notification_handler;ntpd (pid  13847) is
[1476416577] SERVICE NOTIFICATION: servicenow;localhost;Service Status - ntpd;OK;notify-service-by-snt;ntpd (pid  13847) is running...
```

You will see something similar on the ServiceNow Incidents page:





You can also see the Service Notifications in Nagios XI to see that the ServiceNow contact was notified:



Service Notifications

Service Status - ntpd

localhost

Report covers from: **2016-10-13 14:44:41** to **2016-10-14 14:44:41**
Showing 1-5 of 5 total records

Page 1 of 1 | 10 Per Page

Date / Time	Host	Service	Reason	Escalated	State	Contact	Dispatcher	Information
2016-10-14 14:42:57	localhost	Service Status - ntpd	Service Recovery	No	OK	nagiosadmin	Nagios XI	ntpd (pid 13847) is running...
2016-10-14 14:42:57	localhost	Service Status - ntpd	Service Recovery	No	OK	servicenow	Custom: notify-service-by-snt	ntpd (pid 13847) is running...
2016-10-14 14:38:03	localhost	Service Status - ntpd	Service Problem	No	UNKNOWN	nagiosadmin	Nagios XI	ntpd is stopped
2016-10-14 14:38:03	localhost	Service Status - ntpd	Service Problem	No	UNKNOWN	servicenow	Custom: notify-service-by-snt	ntpd is stopped
2016-10-14 10:10:34	localhost	Service Status - ntpd	Service Recovery	No	OK	nagiosadmin	Nagios XI	ntpd (pid 1555) is running...

Next Steps

At this point you should go and assign the contact to the required host and service objects in CCM.

Field Mapping

Field mapping is how the data from Nagios is assigned to the appropriate fields in ServiceNow.

Looking at an extract from the command **notify-service-by-snt** :

```
--short_description="$SERVICEDESC$ on $HOSTNAME$ is $SERVICESTATE$"
```

Looking at an extract from the **/usr/local/nagios/libexec/field_map.ini** file:

```
; string
short_description = $short_description$
```

When Nagios executes the **sn_ticketer.pl** script it provides the argument **--short_description="xxxx"**. Everything between the double quotes is dynamic information from Na

Any of the field mappings defined in the **field_map.ini** file can be referenced by the **sn_ticketer.pl** script, this allows you to customise what information Nagios sends to Servic

All of the available Nagios macros can be found at this link:

Standard Macros In Nagios

Here is another example. Looking this command:

```
$USER1$/sn_ticketer.pl --page="incident.do" --host="$HOSTNAME$" --state="$HOSTSTATE$" --category="Nagios" --short_description="$HOSTNAM
```

You can append any new variable from Nagios to match one from your mappings. For example, if you want to populate the **severity** field in service now, and you have a mapping av

```
; integer
severity = $severity$
```

You can use the **$HOSTSTATEID$** macro in Nagios XI to populate the **severity** field in service now, this is done by appending this to the end of the command:

```
--severity="$HOSTSTATEID$"
```

The total command is now:

```
$USER1$/sn_ticketer.pl --page="incident.do" --host="$HOSTNAME$" --state="$HOSTSTATE$" --category="Nagios" --short_description="$HOSTNAM
```

Final Thoughts

For any support related questions please visit the Nagios Support Forums at:

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

Posted by: **tlea** - Thu, Oct 13, 2016 at 7:11 PM. This article has been viewed 8738 times.

Online URL: https://support.nagios.com/kb/article/nagios-xi-integrating-servicenow-552.html