Probe Guide for PagerDuty Gateway
pagerdutygtw 1.01
Contact CA

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product
## Contents

Contact CA ❄️

Contents ❄️

Documentation Changes ❄️

### Chapter 1: Overview

Limitations ................................................................. 6
Architecture ...................................................................... 7
    Probe Deployment ....................................................... 8
About This Guide .......................................................... 9
Related Documentation .................................................. 9
Release Summary .......................................................... 9
Files ............................................................................. 9
Pre-configuration Requirements ......................................... 10
Supported Platforms ..................................................... 10
System Requirements .................................................... 10
Software Requirements .................................................. 10
Probe Deployment Information ........................................ 10

### Chapter 2: Configuration Details

Probe GUI ....................................................................... 11
Probe Configuration ..................................................... 11
    Setup ........................................................................ 11
    Setup / Listener ......................................................... 13
    Setup / Security ......................................................... 13
    Setup / Endpoints ....................................................... 16
    Setup / Subscription .................................................. 17
    PagerDuty / Auth ....................................................... 18
    PagerDuty / Polling .................................................... 19
    PagerDuty / Field Map ............................................... 20
    PagerDuty / Alarm Triggers ........................................ 21
    Alarm Trigger Criteria ............................................... 22
    PagerDuty / Assignment Triggers ............................... 25
Chapter 3: Procedures 26
UIM Setup Instructions ........................................................................................................26
Attach Queue .......................................................................................................................26
Enable Admin Rights ...........................................................................................................28
Configure PagerDuty API .................................................................................................29
  Obtain PagerDuty Information .........................................................................................29
  Set probe API credentials ...............................................................................................32
  Restart and Verify Configuration ....................................................................................34
  Mapping PagerDuty Users to UIM ..................................................................................36
Clear Persistent Cache .........................................................................................................37

Chapter 4: Known Issues 38
Port Conflicts and Orphaned Processes ............................................................................38

Chapter 5: Troubleshooting / FAQ 39
HTTP Messaging Protocol .................................................................................................39
Probe Startup Failures .........................................................................................................39
Robot Communication ........................................................................................................39
Port Conflicts .....................................................................................................................40
Invalid Configuration ..........................................................................................................40
Successful Probe Startup ...................................................................................................40

Appendix 42
Probe Utility Callbacks ......................................................................................................42
set_api.................................................................................................................................42
Bug Fixes .............................................................................................................................43
Regular Expression Constructs ..........................................................................................44
## Documentation Changes

This table describes the version history for this document.

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>What's New?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>8/30/15</td>
<td>Refactoring service URLs and configuration of probe</td>
</tr>
<tr>
<td>Alpha</td>
<td>8/10/15</td>
<td>Alarm subscription and creation of PagerDuty Incidents</td>
</tr>
<tr>
<td>Alpha</td>
<td>8/4/15</td>
<td>Connectivity and client for PagerDuty REST API</td>
</tr>
<tr>
<td>Alpha</td>
<td>8/30/15</td>
<td>Initial package and prototype</td>
</tr>
<tr>
<td>Beta</td>
<td>1/23/16</td>
<td>Review and cleanup for demo</td>
</tr>
<tr>
<td>1.00</td>
<td>2/28/16</td>
<td>GA release.</td>
</tr>
<tr>
<td>1.01</td>
<td>3/2/16</td>
<td>Patch to address stale UIM session</td>
</tr>
<tr>
<td></td>
<td>3/5/16</td>
<td>Added first time installation and setup instructions. Completed configuration documentation for alarm_triggers, assignment_triggers and PagerDuty API settings.</td>
</tr>
</tbody>
</table>
Chapter 1: Overview

This document is a reference guide to installing and configuring the PagerDuty Gateway (pagerdutygtw) probe.

General Description
The PagerDuty Gateway probe can be used to trigger new PagerDuty Incidents based on UIM alarms, either by specifying matching alarm criteria, or by assigning UIM alarms to a designated UIM user. The probe will also acknowledge and resolve PagerDuty Incidents, assign and close UIM alarms, and post notes to the UIM alarm when new notes are added to the PagerDuty Incident.

Capabilities
Bi-directional synchronization between UIM and PagerDuty.

Limitations

Probe Limitations
The Probe requires internet connectivity to access the PagerDuty API services. TCP Port 443 for HTTPS communication from the Robot where the probe is installed to various PagerDuty URLs is required. Customers who wish to use the PagerDuty Gateway probe must have a valid and current PagerDuty account.
Architecture

The PagerDuty Gateway subscribes to a UIM Queue to receive alarm information from the NAS (Alarm Server).

Alarm messages (alarm_new, alarm_update, alarm_assign, alarm_close) are matched against configurable criteria within the probe to trigger new Incidents via the PagerDuty REST API.

PagerDuty Gateway will poll the REST API for updates made through the PagerDuty web interface. The following updates made to Incidents that were triggered by a UIM alarm will be reflected in the UIM Alarm Console:

- Incident Acknowledgement
- Incident Assignment
- Incident Notes
- Incident Resolution
Probe Deployment

The recommended location for installation of the PagerDuty Gateway probe is on or near the Primary Hub.
About This Guide

This guide is for the CA UIM Administrator to help understand the configuration of the PagerDuty Gateway probe.

Related Documentation

For related information that may be of interest, see the following material:

Related Documentation

- Documentation for other versions of the pagerdutygtw probe
- User documentation for the Admin Console
- Monitor Metrics Reference Information for CA UIM Probes
  (http://docs.nimsoft.com/prodhelp/en_US/Probes/ProbeReference/index.htm)

Release Summary

Please refer to the UIM Compatibility Support Matrix for the latest information on supported platforms. See also the Support Matrix for UIM Probes for additional specific information.

Files

The PagerDuty Gateway contains or will create the following files during runtime:

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pagerdutygtw.log</td>
<td>1Mb</td>
<td>This is the probe log file and will grow to the size configured in the probe configuration file and be rolled to a new file. Log file rotation is a behavior of UIM probes.</td>
</tr>
<tr>
<td>pagerdutygtw.cfg</td>
<td></td>
<td>Probe configuration file.</td>
</tr>
<tr>
<td>pagerdutygtw.cfx</td>
<td></td>
<td>The package configuration file that the pagerdutygtw.cfg file is created from.</td>
</tr>
<tr>
<td>data/persist.dat</td>
<td>3Mb</td>
<td>Data that is preserved between probe restarts, but is managed by the probe is stored in this file. Client heartbeat registrations and QoS definitions are stored in this file.</td>
</tr>
<tr>
<td>lib/</td>
<td></td>
<td>This directory contains all of the jar files for the running probe.</td>
</tr>
</tbody>
</table>
Pre-configuration Requirements

- The probe must be given admin rights (use set_admin callback).
- The PagerDuty API connection must be configured (use set_api callback).
- Probe must be installed on a Robot with internet access to https://<yoursubdomain>.pagerduty.com.
- See Procedures for required UIM setup and instructions.

Supported Platforms

The PagerDuty Gateway probe is supported on the same set of operating systems as the UIM Server solution. Please refer to the UIM Compatibility Support Matrix for the latest information on supported platforms.

System Requirements

The PagerDuty Gateway probe should be installed on systems with the following minimum resources:
- Memory: 2-4GB of RAM. The default configuration requires 64MB of RAM
- CPU: 3GHz dual-core processor, 32-bit or 64-bit

Software Requirements

The PagerDuty Gateway probe requires the following software environment:
- UIM Robot 7.80 or later
- Java Virtual Machine 1.7 or later (typically installed with NMS 8.0 and above)

Probe Deployment Information

There are two ways to distribute archive packages. You can distribute the package within the web-based Admin Console (for supported probes), from within Infrastructure Manager. See Deploy Probe for more information on deploying probes.
Chapter 2: Configuration Details

This section contains configuration details specific to the PagerDuty Gateway probe.

Probe GUI

Configure the probe using raw configure.

Future releases of the probe will use the built-in webserver to provide a friendly web interface for configuration of the probe.

Probe Configuration

The probe configuration file is broken into several sections under setup.

```
<setup>
  <subscription>
    Configuration for the probe’s connection to the UIM message bus to receive alarm information.
  </subscription>
  <admin>
    Configuration for the probe’s administrative rights to UIM.
  </admin>
  <listener>
    Configuration for the network port that the web server will listen on.
  </listener>
  <security>
    Client security options (for access to the internal webserver).
  </security>
  <endpoints>
    Probe servlet configuration, URL mappings.
  </endpoints>
  <pagerduty>
    <assignment_triggers>
      UIM user assignment mapping to defined PagerDuty Service.
    </assignment_triggers>
    <alarm_triggers>
      Alarm mapping to defined PagerDuty Service.
    </alarm_triggers>
    <auth>
      PagerDuty API connection and authentication credentials.
    </auth>
    <service_urls>
      PagerDuty API service URL settings
    </service_urls>
  </pagerduty>
</setup>
```

Setup

The setup section contains basic runtime parameters for the probe such as log level, subsystem and source override behavior. It also contains sub-sections that control the base functions of the probe.
### Key Values

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>loglevel</td>
<td>0-5</td>
<td>Level of detail for log messages. Level 3 will show more detailed information about messages being read, processed and published.</td>
</tr>
<tr>
<td>cycle</td>
<td>Seconds &gt; 0</td>
<td>Frequency to run manager thread. Normally users should not need to adjust this value. Large values for the manager cycle will cause delays in alarming for clients that have failed to transmit a heartbeat recently. Default is 150 seconds.</td>
</tr>
<tr>
<td>subsystem</td>
<td>Subsystem ID</td>
<td>This is the default subsystem that the probe will use when sending alarms, unless the subsystem is specified by the client in the alarm message.</td>
</tr>
<tr>
<td>logsize</td>
<td>KB</td>
<td>Maximum size of the log file in KB before it is rolled. Default is 10240.</td>
</tr>
<tr>
<td>persistent_cache</td>
<td>Filename</td>
<td>Specify the path and filename relative to the probe directory where the persistent cache data should be stored. In very high performance environments, consider locating this file on a ramdisk.</td>
</tr>
</tbody>
</table>
## Setup / Listener

The listener section describes what port the HTTP server will bind to on the local system.

![Raw Configure](image)

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Port</td>
<td>1024-65535</td>
<td>The numeric TCP port that the HTTP listener will bind to. This must be an integer and greater than zero. Standard and reserved ports are defined in <code>/etc/services</code>, choose a port that is suitable to your needs. Typically, you should choose a port that is also greater than 1024, because ports 1-1024 require administrative privileges and are considered reserved ports. <strong>The default for this configuration is 2082.</strong></td>
</tr>
<tr>
<td>address</td>
<td>IP Address</td>
<td>Reserved for future use.</td>
</tr>
</tbody>
</table>

## Setup / Security

Notice: The PagerDuty Gateway Probe version 1.x does not ship with this section in the configuration file.

To limit client access, the probe can be configured to only allow clients that match a whitelist.
Edit the probe security configuration and describe regular expressions to be matched against the client IP address. If the IP address of the client does not match any patterns in the `authorized_clients` section, then the probe will respond with HTTP 401 (unauthorized) to any request.

Restricting access to the probe by client IP address is a reasonable security measure to take. Especially if you intend to expose this probe to an untrusted network, it is best practice to limit the access to network resources as much as possible. There will, of course, be situations where leaving the probe open to any client is a desirable behavior.

The pattern matcher used in this implementation is the native `java.util.regex.Pattern` class provided by Oracle as part of the Java JRE. Java pattern matching is a strict implementation of regular expression parsing. Consider reviewing the capabilities and limitations of this implementation: [http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html](http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html).

```xml
<security>
  <authorized_clients>
    active= yes
    <localhost>
      active= yes
      ip= 127\0.0\1
      deny= no
    </localhost>
    <vlan10>
      active= yes
      ip= 10\1.10\[0-9]*
      deny= no
    </vlan10>
    <vlan90>
      active= yes
      ip= 10\90\10\[0-9]*
      deny= yes
  </authorized_clients>
</security>
```
</vlan90>
<any>
    active= no
    ip= .*
</any>
</authorized_clients>
</security>

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active = yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>ip= pattern</td>
<td>Java Regex Pattern</td>
<td>A regular expression that will be matched against the client IP address to determine if this rule should be considered to authorize or deny access to the internal webserver.</td>
</tr>
<tr>
<td>active= yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>deny= no</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
Setup / Endpoints

PagerDuty Gateway: This section will exist, but should not be modified. Future releases of the PagerDuty Gateway probe will expand on this section and bring new capabilities.

Note: The endpoints section contains all of the defined endpoints, no key values are stored directly under the <endpoints> section.

Endpoints

Endpoints are pieces of software that are loaded by the probe’s internal webserver. The user can re-configure and change the base URL paths for each of the provided function calls, however this would be considered advanced use.

The base URL for any endpoint under the PagerDuty Gateway probe will conform to the following convention:

http://<hostname>:<port>/pd/<endpoint_url_path>/<function>

The endpoint path is described in the configuration file as url_path.
Setup / Subscription

The probe will subscribe to a queue created on the same Hub where the nas is installed. You must create this queue before the probe will function correctly. See Attach Queue instructions under procedures to create this queue prior to installing the PagerDuty Gateway probe.

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>UIM Address /domain/hub/robot</td>
<td>The UIM Address (/domain/hub/robot) of the Hub that contains the attach queue that this probe will subscribe to. This configuration key may be left empty (blank) if the pagerdutygtw probe is installed on the same Hub as the nas probe.</td>
</tr>
<tr>
<td>queue</td>
<td>pagerduty</td>
<td>The name of the queue created to attach the following messages published by the nas: alarm_new, alarm_update, alarm_assign, alarm_close</td>
</tr>
<tr>
<td>nas</td>
<td>Probe name of the alarm server (nas)</td>
<td>The name of the alarm server probe (nas). This configuration key should not be changed unless CA releases a new alarm server that is not named ‘nas’. This configuration is here for future compatibility.</td>
</tr>
</tbody>
</table>
PagerDuty / Auth

The auth configuration does not need to be changed by a UIM administrator manually, this section will be created by executing the set_api callback (see setup instructions).

This section will be created when the callback set_api is executed using the Probe Utility.

Important: An administrator should never need to modify the values manually in this configuration section.

Note: The probe must have already been authorized with an appropriate UIM user with admin rights using the set_admin callback before calling set_api.

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>subdomain</td>
<td></td>
<td>The subdomain assigned to your PagerDuty account.</td>
</tr>
<tr>
<td>tld</td>
<td></td>
<td>The top level domain (tld). This should always be pagerduty.com.</td>
</tr>
<tr>
<td>api_key</td>
<td>Encrypted</td>
<td>The encrypted PagerDuty API key.</td>
</tr>
<tr>
<td>confirmed</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
**PagerDuty / Polling**

The **polling** section controls the frequency at which the probe will query PagerDuty’s API for updates and changes to incidents that have been created by the probe.

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>triggers</td>
<td>Seconds &gt; 0</td>
<td>Speed, in seconds at which to trigger new PagerDuty Incidents once a UIM alarm has been detected and matches the specified alarm trigger criteria.</td>
</tr>
<tr>
<td>updates</td>
<td>Seconds &gt; 0</td>
<td>Frequency to query the PagerDuty API for updates and changes such as acknowledgement, assignment, and notes for PagerDuty Incidents that have been previously created by the probe. This setting is shipped by default at 150 seconds (2 minutes, 30 seconds).</td>
</tr>
<tr>
<td>max_retries</td>
<td>Number &gt; 0</td>
<td>If the probe fails to communicate with the PagerDuty API, this configuration controls how many times the probe will attempt to retry.</td>
</tr>
<tr>
<td>triggers_help</td>
<td>N/A</td>
<td>Comment contained within the probe configuration to provide brief documentation for the triggers setting.</td>
</tr>
<tr>
<td>updates_help</td>
<td>N/A</td>
<td>Comment contained within the probe configuration to provide brief documentation for the updates setting.</td>
</tr>
</tbody>
</table>
PagerDuty / Field Map

The field_map section allows an administrator to set which UIM alarm fields are updated when the probe creates a new PagerDuty Incident. This section supports the substitution of information returned from PagerDuty upon new incident creation.

The default configuration that ships with probe version 1.x is show below:

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom_1</td>
<td>String</td>
<td>Set UIM alarm custom field 1 upon creation of new PagerDuty Incident.</td>
</tr>
<tr>
<td>Custom_2</td>
<td>String</td>
<td>Set UIM alarm custom field 2 upon creation of new PagerDuty Incident.</td>
</tr>
<tr>
<td>Custom_3</td>
<td>String</td>
<td>Set UIM alarm custom field 3 upon creation of new PagerDuty Incident.</td>
</tr>
<tr>
<td>Custom_4</td>
<td>String</td>
<td>Set UIM alarm custom field 4 upon creation of new PagerDuty Incident.</td>
</tr>
<tr>
<td>Custom_5</td>
<td>String</td>
<td>Set UIM alarm custom field 5 upon creation of new PagerDuty Incident.</td>
</tr>
</tbody>
</table>

The following variables are available to use in this configuration from the PagerDuty Incident:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(status)</td>
<td>$(html_url) The URL that links to the PagerDuty Incident.</td>
</tr>
<tr>
<td>$(incident_number)</td>
<td>The PagerDuty Incident Number.</td>
</tr>
<tr>
<td>$(id)</td>
<td>The PagerDuty ID assigned to the Incident.</td>
</tr>
</tbody>
</table>
PagerDuty / Alarm Triggers

The alarm trigger configuration is flexible, allowing an administrator to carefully control which UIM alarms will be used to create new PagerDuty Incidents.

Each alarm trigger configuration must be configured with the specific PagerDuty service_key. This service_key is used when triggering new PagerDuty Incidents. Each alarm_trigger may be configured to issue Incidents under a different PagerDuty Service.

The name of each alarm trigger section is arbitrary and can be chosen by an administrator during probe configuration.

No keys are used in the criteria section itself, keys must exist in each sub-section under criteria.

Within each alarm trigger, a section named criteria must exist. Within each criteria section, an arbitrary number of sections (sub-sections) may exist, they must be named numerically beginning with 0 (zero).
### Alarm Trigger Criteria

The alarm criteria sub-sections and keys are highly configurable within each alarm trigger.

The criteria fields in each sub-section support a variety of ways to easily match UIM alarm fields.

The keys in this configuration section that do not begin with an underscore (_) are treated as instructions to the matching engine. Keys may be the exact UIM alarm field name, or be suffixed with an instruction, separated by a dot (.)

Special note: if the key does not have an instruction suffix, then the matching engine uses a literal string comparison of the configuration value and the alarm field value.

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>_rule</td>
<td>include</td>
<td>exclude</td>
</tr>
</tbody>
</table>
Instructio

<table>
<thead>
<tr>
<th>Instruction Suffix</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>.matches</td>
<td>Pattern matching against the UIM alarm field. See regular expression constructs in the appendix for specific use of regular expressions.</td>
</tr>
<tr>
<td>.not</td>
<td>UIM alarm field does not match the value specified in this configuration (uses regular expressions, same as .matches)</td>
</tr>
<tr>
<td>.contains</td>
<td>Value is contained within the UIM alarm field.</td>
</tr>
<tr>
<td>.ends_with</td>
<td>UIM alarm field ends with.</td>
</tr>
</tbody>
</table>

**Instruction Suffix** | **Explanation** | **Comparison** |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>.starts_with</td>
<td>UIM alarm field starts with.</td>
<td>Literal String</td>
</tr>
</tbody>
</table>

Note: To understand more about what fields are available to build your criteria rules with, use the Dr. Nimbus message sniffer available with the installation of Infrastructure Manager. Listen for message subject alarm_new or alarm_update. UIM alarm field names are case-sensitive.

<table>
<thead>
<tr>
<th>Alarm Field</th>
<th>Description</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>severity</td>
<td>The UIM alarm severity string.</td>
<td>critical, major, minor, warning, information</td>
</tr>
<tr>
<td>level</td>
<td>The numeric UIM alarm severity.</td>
<td>5, 4, 3, 2, 1</td>
</tr>
<tr>
<td>subsys</td>
<td>The last string separated by dots (.) of the subsystem.</td>
<td>Reference the subsystem table defined in the nas probe configuration</td>
</tr>
<tr>
<td>sid</td>
<td>The numerical, dot-separated notation of the subsystem defined by the nas.</td>
<td>Example: 1.1.2</td>
</tr>
<tr>
<td>domain</td>
<td>The UIM domain that the alarm originated from.</td>
<td>Value will be specific to your environment.</td>
</tr>
<tr>
<td>origin</td>
<td>The Origin of the alarm as defined by the Robot configuration, or if not specified by the Robot, the origin will be the name of the Hub that the robot is attached to.</td>
<td>Value will be specific to your environment and depend on each Robot’s configuration.</td>
</tr>
<tr>
<td>hub</td>
<td>The Hub that the Robot is attached to.</td>
<td>Value will be specific to your environment.</td>
</tr>
<tr>
<td>robot</td>
<td>The name of the UIM Robot that generated the alarm.</td>
<td>Value will be specific to your environment.</td>
</tr>
<tr>
<td>source</td>
<td>The source IP of the alarm as defined by the probe.</td>
<td>Value will be specific to your environment.</td>
</tr>
<tr>
<td>hostname</td>
<td>The hostname as resolved by the nas using the IP address in the alarm source field.</td>
<td>Value will be specific to your environment.</td>
</tr>
<tr>
<td>prid</td>
<td>The name of the Probe that generated the alarm.</td>
<td>UIM Probe name.</td>
</tr>
<tr>
<td>user_tag1</td>
<td>Alarm user tag 1.</td>
<td>Value will be specific to your environment.</td>
</tr>
<tr>
<td>user_tag2</td>
<td>Alarm user tag 2.</td>
<td>Value will be specific to your environment.</td>
</tr>
<tr>
<td>supp_key</td>
<td>The alarm suppression key as defined by the Probe that generated the alarm.</td>
<td>Value is determined by the Probe that generated the alarm.</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>visible</td>
<td>Numeric 1 or 0 depending on the alarm visibility.</td>
<td>1 = visible</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = invisible</td>
</tr>
</tbody>
</table>
PagerDuty / Assignment Triggers

The PagerDuty Gateway probe can also detect alarm assignments and trigger the creation of a new PagerDuty Incident.

This behavior is similar to many other UIM integrations with ticketing solutions.

<table>
<thead>
<tr>
<th>Key</th>
<th>Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>service_key</td>
<td>PagerDuty Service Key</td>
<td>Alarms matched by this alarm trigger will be created under the PagerDuty Service identified by this service_key.</td>
</tr>
<tr>
<td>uimuser</td>
<td>UIM username</td>
<td>When an alarm is assigned to this user in UIM, the probe will trigger a PagerDuty Incident.</td>
</tr>
</tbody>
</table>
Chapter 3: Procedures

This section contains procedures to install, setup and manage the probe.

UIM Setup Instructions

A new attach queue must be created on the Primary Hub, or the Hub with a NAS installed where alarms will be generated to be used by the probe to create and update PagerDuty Incidents.

Overview of UIM setup:

1) Create an attach queue on the Hub where the nas probe is also running.

2) Install the PagerDuty Gateway probe on or near the Hub where the nas is running.

3) Execute the pagerdutygtw probe callback set_admin by either using the probe utility GUI from within Infrastructure Manager, or the command line program pu.exe (Windows), or pu (Linux) depending on the host operating system you are most familiar with.

4) Execute the pagerdutygtw probe callback set_api by either using the probe utility GUI from within Infrastructure Manager, or the command line program pu.exe (Windows), or pu (Linux) depending on the host operating system you are most familiar with.

5) Restart the probe.

6) Configure the user mappings for PagerDuty to UIM users in the probe.

7) Configure and enable the alarm triggers that will create PagerDuty Incidents.

Attach Queue

Create an attach queue on the Hub in your environment where the nas is installed, normally this will be your Primary Hub. If you also have a Secondary Hub, you may create a similar queue to be activated during a failure of your Primary Hub. If you have strategically placed nas probes throughout your environment, you may also choose to create similar Attach queues where appropriate. The nas will publish messages as alarms are created, updated, assigned and closed in UIM. This queue will make those messages available to the PagerDuty Gateway probe.

The recommended name for the queue is pagerduty although, you may name this queue differently if you choose. If the name of this queue is not pagerduty then the probe’s configuration must also be updated to use the same name.

| Queue Attributes |

© PagerDuty, Inc. 2016
Name: pagerduty
Type: attach
Address: N/A
Subject: alarm_new, alarm_update, alarm_assign, alarm_close
Bulk Size: N/A

The following screenshots are provided for reference:

Queues tab in Hub configuration GUI (Infrastructure Manager)

Pagerduty queue details:
Enable Admin Rights

The probe requires UIM administrator privileges to subscribe to interact with the Alarm Server (nas). This step is required as part of the initial installation and setup of this probe.

Once installed, an administrator must use the probe utility to execute `set_admin` with an existing UIM account that has an administrative ACL. These elevated privileges will allow the pagerdutygtw probe to query other probes to gather and update UIM Alarm information, such as the nas.
Configure PagerDuty API

In order for the probe to communicate with the PagerDuty API, the probe must be configured using an API key and a user account from PagerDuty.

Prior to executing the `set_api` callback in the probe using the probe utility, you will need to generate a new API key within PagerDuty, or use a pre-existing API key. PagerDuty API keys should be considered sensitive and treated like a password. In the PagerDuty web interface, you are only able to view the API key once upon creation. If you lose this API key, it is very easy to simply generate a new one.

Obtain PagerDuty Information

In the PagerDuty web interface (https://pagerduty.com) use the Configuration menu and navigate to API Access where you can manage and create new API keys.

In the upper-right corner of the API Access page within PagerDuty, give a description for the new API key you are about to generate. The description is arbitrary, so you may follow any naming conventions you wish. It is recommended that you give some indication in the description that the key is used by UIM.
Once you have created a new API key, you will be given the chance to copy the key. This is the only time that the full API key will be made available, make sure to copy the key and store it safely until you are ready to configure the PagerDuty Gateway in UIM.

Once you have obtained an API key from PagerDuty, you will also need to find the ID of the PagerDuty user that the probe will also use to authenticate to PagerDuty.

Use the Configuration menu in PagerDuty to navigate to the Users area.

From the view of your users in PagerDuty, select the user that you wish to use for the PagerDuty Gateway probe’s access to the API. In the below example, the user uim_pagerduty has been created as a service account user for the probe to use. You do not need to create a separate service account user; however, it may be recommended for easy identification when PagerDuty Incidents are created.
Click on the User to navigate to the details page, where you can obtain the PagerDuty user ID from the URL in your browser.

Obtain the PagerDuty User ID from the URL on the user’s details page.
Set probe API credentials

After you have successfully created an API key and obtained the User ID from PagerDuty to use for the integration, use the Probe Utility (select the pagerdutygtw probe and press `ctrl+p`) to call the `set_api` callback.
Configure PagerDuty API

Example of using the Probe Utility to call set_api.

```
Provision utility [UI/IM/CORE01/ /_hub/pagerdutygtr]

Provision commandset

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>The ID of the PagerDuty</td>
<td>The PagerDuty user ID to be used by the probe for authentication to the PagerDuty API services.</td>
</tr>
<tr>
<td>subdomain</td>
<td>The subdomain for your PagerDuty account.</td>
<td></td>
</tr>
<tr>
<td>api_key</td>
<td>The API key generated from the PagerDuty web interface that the probe will use to authenticate and interact with the PagerDuty API services.</td>
<td></td>
</tr>
</tbody>
</table>
Configure PagerDuty API

| tld | pagerduty.com | TLD (top level domain). This can be left empty and will be defaulted to pagerduty.com. |

Special note: the probe must already have been configured with a UIM administrative account using the `set_admin` callback.

Upon executing the `set_api` callback using the Probe Utility, the configuration will be written to the probe configuration file and saved. You may want to make a backup of the probe configuration at this moment by dragging the probe to your archive in Infrastructure Manager.

**Restart and Verify Configuration**

Once the UIM administration privileges and PagerDuty API credentials have been set using the `set_admin` and `set_api` callbacks, restart the probe and verify that the probe configuration is correct.

View the probe log file after restarting, the probe will query the PagerDuty API and retrieve any configured PagerDuty Services as well as configured Users.
In addition to viewing the probe log file, open the probe configuration to also see that the Services and Users were retrieved from the PagerDuty API.

The definedPagerDuty Services will be written to the probes configuration file under the pagerduty/services section.

Here you will have a reference to the PagerDuty service_key that will be used to configure the probe alarm triggers.
Mapping PagerDuty Users to UIM

Under the `pagerduty/users` configuration, the probe will create sections for each user found in PagerDuty.

For each of these sections, add a new key named `uimuser`, set the value to the appropriate UIM user

![Image of Raw Configure window](image)

Important:

By configuring this user mapping, the probe will be able to correctly determine the corresponding UIM user when a PagerDuty Incident is created, acknowledged, assigned, and resolved.
Clear Persistent Cache

Although only encountered in development, it may be necessary to manually delete the persistent cache file that the probe keeps to preserve dynamic information between restarts.

If the contents of this file becomes corrupt or unreasonably large for some reason, it is safe to delete. Deletion of this file would result in any existing alarms that are being synchronized between PagerDuty and UIM to be lost.

Navigate to the probe installation directory on the Robot host.  
Windows default location: C:\Program Files (x86)\Nimsoft\probes\marketplace\pagerdutygtw\data  
Linux default location: /opt/nimsoft/probes/marketplace/pagerdutygtw/data

Under the data directory, remove the file named persist.dat.
Chapter 4: Known Issues

This section contains a list of known issues in this release.

Port Conflicts and Orphaned Processes

In rare cases, a defunct copy of the probe can still be attached to the listening port. Be sure to look for and kill any remaining Java process that is the pagerdutygtw probe before re-starting.

Use the task manager in Windows to locate a duplicate process.

The process for the pagerdutygtw probe can be identified on linux systems in this way:

```
ps -ef | grep -i nim | grep pagerdutygtw
```

```
root      8542  2016 32 20:27 ?        00:00:04 nimbus(pagerdutygtw) -Xmx64M -cp lib/* pagerdutygtw
```
Chapter 5: Troubleshooting / FAQ

This section contains troubleshooting information for the pagerdutygtw probe.

HTTP Messaging Protocol

RFC2616 – Hypertext Transfer Protocol -- HTTP/1.1


RFC3986 – Uniform Resource Identifier (URI): Generic Syntax


RFC7230 - Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing


Probe Startup Failures

Max Restarts

Usually if the probe will not startup at all it means there is something wrong with the JRE installation or the probe dependencies are missing.

From the probe directory <Nimsoft>/Probes/marketplace/pagerdutygtw directory, open a command line shell and run the probe manually:

```
# ..\..\..\jre\jre6\bin\java.exe -cp "lib/**" pagerdutygtw
```

Take a note of any exceptions on the command line.

Robot Communication
The probe will fail to startup if there is not a UIM robot installed and running on the host.

Port Conflicts

The default port that the probe will listen on is 2082. If there is already a process attached to TCP port 2082, then the probe will fail to work properly.

Use the operating system’s netstat command to see if there is already a process bound to the port when the probe is not running.

Invalid Configuration

An invalid configuration file can also cause the probe not to start or run properly.

Successful Probe Startup

If the probe starts up successfully, this is what the log messages should look like:

```
Mar 02 00:19:23:389 [main, pagerdutygtw] *****************[ Starting ]*****************
Mar 02 00:19:23:406 [main, pagerdutygtw] port=48005
Mar 02 00:19:23:443 [main, pagerdutygtw] Only restricted security available, no login
Mar 02 00:19:54:006 [Thread-0, pagerdutygtw] Context Path: /pd
Mar 02 00:19:54:006 [Thread-0, pagerdutygtw] Webapp Path: srvr/pagerduty
Mar 02 00:19:54:006 [Thread-0, pagerdutygtw] Listen Port: 2082
Mar 02 00:19:54:114 [Thread-0, pagerdutygtw] Loading admin
com.pagerduty.probe.endpoints.ProbeAdmin /admin/*
Mar 02 00:19:54:116 [Thread-0, pagerdutygtw] Loading services
com.pagerduty.probe.endpoints.Services /services/*
Mar 02 00:19:54:116 [Thread-0, pagerdutygtw] Loading root inc.morsecode.core.Gateway /*
Mar 02 00:19:54:116 [Thread-0, pagerdutygtw] Starting HTTP Server
Mar 02 00:20:01:286 [Thread-0, pagerdutygtw] HTTP GET
https://morsecode-incorporated.pagerduty.com/api/v1/services?limit=49&offset=0 HTTP/1.1
```
Probe Utility Callbacks

Aug 30 12:00:06:249 pu: SSL - init: mode=0, cipher=DEFAULT, context=OK

Address: pagerdutygtw Request: _command

======================================================
Address: pagerdutygtw Request: set_api

set_api

Configure the probe with the appropriate pagerduty API information.

Prompt$> PDUSERID=<the ID of the pagerduty user>
Prompt$> SUBDOMAIN=<the subdomain for your pagerduty account>
Prompt$> APITOKEN=<the api token for your pagerduty account>

Prompt$> pu pagerdutygtw set_api $PDUSER $SUBDOMAIN $APITOKEN

Simultaneously watch the probe log file for confirmation that the API credentials were accepted.

PPUFVOK:8tdt 11c9dd0291504509b02bd7225b9f99f6 (/services/PPUFVOK)
PAPCXT0:outpost c64252179f5b4acda00483091dadcd7a (/services/PAPCXT0)

HTTP/1.1 HTTPClient
GET https://morsecode-incorporated.pagerduty.com/api/v1/users/PXXX999A

User Information:
{
  "id": "PXXX999",
  "time_zone": "Mountain Time (US & Canada)",
  "invitation_sent": "false",
  "avatar_url": "https://secure.gravatar.com/...",
  "color": "purple",
  "email": "user@yourdomain.com",
  "name": "Brad",
  "role": "owner",
  "user_url": "/users/PXXX999",
  "marketing_opt_out": "false"}
After a few minutes, the probe will write the user information to the persistent cache.

C:\Users\bcmorse\git\probe-groovy-framework\groovy_framework\pagerduty\data\persist.dat
Prompt$> cat data/persist.dat
<cache>
  <pagerdutygtw.pduser>
    <PXXX999A>
      id = PXXX999A
      time_zone = Mountain Time (US & Canada)
      avatar_url = https://....
      invitation_sent = false
      email = user@pagerduty.com
      color = purple
      name = PD User
      role = owner
      user_url = /users/PXXX999A
      marketing_opt_out = false
    </PXXX999A>
  </pagerdutygtw.pduser>
</cache>

Bug Fixes

2016-03-02
Version 1.01 – Fix to address stale UIM session. This defect presented itself after running for an extended period of time, the probe would be prevented from accessing the NAS for alarm field updates due to an expired session id (SID).
Regular Expression Constructs

This reference table is an abbreviation of the documentation provided by Oracle about pattern matching in the Java JRE: http://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html

Characters

- **x** The character x
- **\** The backslash character
- **\t** The tab character (\u0009)
- **\n** The newline (line feed) character (\u000A)
- **\r** The carriage-return character (\u000D)

Character classes

- **[abc]** a, b, or c (simple class)
- **[^abc]** Any character except a, b, or c (negation)
- **[a-zA-Z]** a through z or A through Z, inclusive (range)
- **[a-d[m-p]]** a through d, or m through p: [a-dm-p] (union)
- **[a-z&[def]]** d, e, or f (intersection)
- **[a-z&[^bc]]** a through z, except for b and c: [ad-z] (subtraction)
- **[a-z&[^m-p]]** a through z, and not m through p: [a-lq-z](subtraction)

Predefined character classes

- **.** Any character (may or may not match line terminators)
- **\d** A digit: [0-9]
- **\D** A non-digit: [^0-9]
- **\s** A whitespace character: [ \t\n\x0B\f\r]
\S A non-whitespace character: [^s]
\w A word character: [a-zA-Z_0-9]
\W A non-word character: [^\w]

**Boundary matchers**

^ The beginning of a line
$ The end of a line

**Greedy quantifiers**

X? X, once or not at all
X* X, zero or more times
X+ X, one or more times
X{n} X, exactly n times
X{n,} X, at least n times
X{n,m} X, at least n but not more than m times

**Reluctant quantifiers**

X?? X, once or not at all
X*? X, zero or more times
X+? X, one or more times
X{n}? X, exactly n times
X{n,}? X, at least n times
X{n,m}? X, at least n but not more than m times

**Possessive quantifiers**
Regular Expression Constructs

X?+   X, once or not at all
X*+   X, zero or more times
X++   X, one or more times
X(n)+  X, exactly n times
X(n,)+ X, at least n times
X(n,m)+ X, at least n but not more than m times

Logical operators

XY     X followed by Y
X|Y     Either X or Y
(X)    X, as a capturing group