DSPS - Digital Services Paging System

Overview

DSPS is designed to facilitate phone to phone communications between employees. Its goals include:

- You don't have to know the phone number of the person you want to contact.
- You don't have to know who's "on call" in order to contact that person.
- When responding to an off-hours page, you don't have to know who received the initial alert to keep everyone in the loop.
    - It's 3am and depending on the system that died, say 5 people were automatically paged. If you're going to respond you want to tell all of the other 4 that you're on it (i.e. they can go back to sleep). But you don't want to tell anyone beyond that, as you'd be waking additional people up. Having the system automatically track who got the initial alert and sending your reply only to those people maximizes communication and sleep.
- Like any group chat facility, it should be transparent about who you're chatting with.
- It should be intuitive to use.

Interface - How It Looks On Your Phone

DSPS uses a single 6-digit phone number to send and receive all pages (i.e. text messages). It's what the industry calls a "short code," much like what you'd text to if you wanted to vote on American Idol. Once your phone number has been setup in the paging system, you may want to add DSPS' number to your phone's Contacts with a familiar name, like "Work Paging." That way when you receive texts you'll know where they're coming from. All pages from anyone using the system will appear to be from that number.

Features

- Intuitive name referencing - use someone's name to contact them, not a phone number
- Group paging - use a single name to page a larger group
- Audience tracking - remember who's in the existing conversation
- Storm throttling so you don't get 20 pages in a row
- Nagios Recovery silencing - woken at 3am, you fix a problem, don't won't to get woken up half an hour later when Nagios notices it's fixed
- History sharing - text your conversation's history to someone new
- Emergency support
    - Maintain internal rotating "on call" schedules so it knows who to page
    - Auto escalation when initial page isn't responded to
    - RT ticket integration to log conversation history with the original call data

Simple Conversations

Contacting a person or entire group is as easy as sending a text message to DSPS and simply mentioning their name. For example, if I sent the text "Can you give me a hand Jim?" the system would page Jim. [It's worth noting that things obviously get more complicated with name collisions; you can configure some users by first name, some by last or just make up a unique identifier for them.] What really happens when I send the above text is that Jim and I get placed in a group chat room. Any texts he sends to DSPS will get copied to me and vice versa. If one of us mentions another person by name ("Hey Sean, come join us.") that person will get pulled in as well. You can also start a group conversation (page) an entire team at once. For example, we define an ops team, a dev team and a number of others. That enables us to send a message like "I need some dev help" and the message will instantly go out to everyone in my dev team. Messages are always prefixed with the sender's name.

Room Dynamics

There are a few guiding principals for how rooms work:

- Messages in a room are broadcast to everyone there.
- Every time the number of people in a room ("the audience") changes, everyone in the room is immediately informed.
- Rooms automatically expire 90 minutes after the last message was sent.
- Any number of rooms can exist at a time
- A given user can only be in one room at a time

On Call Schedules

Your users are divided into teams (making it easy to page the entire team) and teams can optionally have an On Call schedule. The team name can always be used to page everyone but the On Call option lets you page a particular tag (think of it as another name) and generically get whoever's on call that day. For example we have an Operations team that consists of 3 members and we do 1 week on call rotations. By sending a page with our special tag in it (we use "opscall") someone can reach which ever of us is on call at that moment.

Escalation Scenarios

There are times when you might want a page to have an escalation time associated with it. Generally if a computer sends the page (some sort of automated system, like nagios) and it's only sending it to one person, you'll care about escalation. What if that person is out of cell range or just doesn't reply? The system or customer is now dead in the water. So in these situations it's helpful to have DSPS automatically escalate the page to a larger audience if it's not replied to in a certain amount of time. DSPS support two types of escalation scenarios, which are very similar. One is called an emergency page, which acts differently because it can provide an interface to additional functions, such as sending an email to a given group of people and creating an RT ticket (if you use Request-Tracker) all when the page comes in. The other type is generic escalation. In both cases, after the time limit is up the page is resent to the specified larger audience.

DSPS Commands

In addition to direct text that can mention people's names or group names, automatically pulling those people/groups into the conversation, DSPS supports a handful of commands:

Silencing	Description
:norecovery **x**	Ignore the next **x** nagios RECOVERY pages to you
:norecovery	Ignore all nagios RECOVERY pages to you
:recovery	Enable all nagios RECOVERY pages to you
:vacation **n**	Set **n** vacation days (ignore all group/on-call pages for x days - you can still be paged by direct name)
:autoreply **n msg**	Set an auto reply **msg** for the next **n** time
:ar **n msg**	Shortcut for :autoreply
:noregex **n re**	Block all nagios pages that match /**re**/ for the next **n** time
:nore **n re**	Shortcut for :noregex
:sleep	Ignore all nagios LOAD and RECOVERY pages for 3 hours - used post-maintenance
:snooze	Ignore all nagios pages for 15 minutes
:nonagios **n**	Ignore all nagios pages for **n** time
:leave	Leave the room
:disband	Disband the room for everyone
RT	**Description**
?rt	Look up the associated ticket number
!**Txt**	**Txt** is sent to the room but not logged in the ticket
:stoprt	Stop logging to the current ticket for the duration of the room
Status	**Description**
?rooms	Show occupants of all rooms
?vacation	Show defined vacation settings
?oncall	On call
?**Name**	Show the details for Name (can be a person, group or alias)
?filter	Show the current filter/suppression rules that are in effect
Misc	**Description**
:swap **Name**	Swap next OnCall weeks with **Name**
:history **Name**	Send room history (last 5 messages) to **Name**; **Name** has to be a person, not a group/alias
:email **Addy**	Send room history (last 5 messages) to email address **Addy**
!**Txt**	Send the **Txt** to the room literally - do not parse the message for commands, names, groups or aliases
?	Show the help screen
^**Txt**	Send the **Txt** to the room in Broadcast mode - replies only go back to the sender, not the entire room

---

Command Times

A number of the above commands accept a time value (denoted by the variable "**n**"). In those commands **n** defaults to units of hours, but you can add a suffix to specify alternative units. For

example, ":autoreply 30m I'm driving" will set a 30 minute auto reply. Units can be 's'econds, 'm'inutes, 'h'ours, 'd'ays or 'w'eeks.

Emergency Pages

Emergency pages are a special class of escalation pages. You configure which team answers emergency pages with the $CFGsEmergencyGroup directive. It should point to a group that has a schedule defined. The only differences from a standard escalation is that emergency pages can also create and update an associated RT ticket, as well as send the original page to a pre-configured email address. If you configure the $CFGRT…. directives then an emergency page and all subsequent messages in that room will be logged to the ticket.

Rooms & Coalescing

DSPS can support as many simultaneous chat rooms as are desired. So if people A, B and C are in a room talking, D, E, and F may be in a second room talking (there can be a third, fourth, etc). For sanity sake, a given person can only be in one room at a time. Suddenly person B says “hey F!” The expectation is that F would be pulled into the room with A, B and C and part of the group conversation – and they will. But F is already talking to D and E. D and E also have expectations about an existing conversation. So when B calls on F we really have to pull F and everyone from F’s room into B’s, effectively combining the two rooms. It makes the most sense for everyone involved. At that point everyone will get a status update (a text) saying who’s now in the room.

Configuration Examples

```
our %hStaff = (


‘ops’ => {
        ‘members’ => {
                ‘Jim|jones|jjones’ => ‘5555551212’,
                ‘John|paul’ => ‘5555551213’,
                ‘Steve|roberts’ => ‘5555551214’,
                ‘BigBoss’ => ‘5555551216’,
        }

        ‘escalation’ => {
                ‘tag’ => ‘opscall’,
                ‘timer’ => 120,
                ‘on_expire_to’ => ‘ops’,
                ‘cancel_msg’ => “[Escalation canceled. You’re on your own. Good luck!]”,
        }

        ‘schedule’ => {
                ‘20130101’ => ‘John’,
                ‘20130105’ => ‘Steve’,
                ‘20130108’ => ‘auto’,
```

```
        }
}
```

There's quite a bit in there so let's take it apart. This defines 1 team, our ops team. It links the team members with their phone numbers. A basic regex is supported for people's names. In this case I've said people are addressable by their first or last names, and in one case also by his unix login name. If I had two Johns, I might just leave those two guys' first names out and only list them by last names. Each entry does need to be unique. Capitalization doesn't matter (matches are all case-insensitive) but the first part of the name, up to the first pipe, is used when someone sends a message. For example a page from 555555121 might go out as "Jim: Hey what's up?" For many groups you may just have the "members" section. That's the only required part, unless you're defining an alias, in which case you don't even need members (more on that below). The "escalation" and "schedule" sections are optional.

Escalation

Escalation and schedule are closely linked; you can't have one without the other. If someone pages 'opscall" it won't know which person to send it to without the schedule. Timer and on_expire_to are optional. When they're specified the tag can be used in 2 ways:

- A human initiated page that includes the tag – in this case it's just an alias to the oncall person's name and allows you to reach them without knowing **who** is on call. If I send "I need help from opscall" that puts me and the opscall person in a chat room (two people).
- An automated page that includes the tag – in this case just a single person is in the chat room, the opscall person. In this situation the timer starts counting down. If it runs out without any reply page from the opscall person (presumably them saying something like "I'm on it!") then the system escalates the page to the "on_expire_to" person/groups. Escalation can only happen once for the room.

The "cancel_msg" –if defined—is sent to the on call person if they do reply to the initial page before the timer has expired.

Scheduling

The schedule section is a date (YearMonthDay) pointing to who goes on call on that date. The name portion of the line can be anything that matches the regex defining that person (first name, last, whatever). The special reserved word "auto" can be used to have the system automatically cycle through the team members starting on that date giving everyone a 1 week rotation. Their order will be random but consistent. Another permutation can be used to specify the order or to even leave people out of the schedule. For example:

```
'schedule' => {
        '20130101' => 'John',
        '20130105' => 'auto/jim,john',
}
```

The above schedule says John is on call from 01/01 to 1/05. On 01/05 Jim goes on call for a week, then John for a week, then it repeats (back to Jim and so on). Note how BigBoss, who wanted to be part of the team to see all the group pages, isn't included in the pager rotation. In other words BigBoss will get all pages destined for "ops" but none of the ones for "opscall" with **this** version of "auto".

Aliases

Individual teams (like “ops” above) allow us to use a single tag to page a bunch of people. Teams are grouped with the escalation and scheduling directives. But sometimes you want to page a group of people that aren’t necessarily on the same team. To simplest example is say you want to page everyone. Suppose I have two teams:

```
‘ops’ => {
        ‘members’ => {
                ‘Jim|jones|jjones’ => ‘5555551212’,
                ‘John|paul’ => ‘5555551213’,
                ‘Steve|roberts’ => ‘5555551214’,
                ‘BigBoss’ => ‘5555551216’,
        }
},
‘dev => {
        ‘members’ => {
                ‘Bert’ => ‘5555551217’,
                ‘Ernie’ => ‘5555551218’,
                ‘Oscar|grouch’ => ‘5555551219’,
                ‘Elmo’ => ‘5555551210’,
        }
},

‘911’ => { ‘alias’ => ‘ops, dev’ },

‘srmgrs’ => { ‘alias’ => ‘john, ernie, elmo’ },
```

Now I have an alias called 911 that when it’s mentioned in a page will send the message to everyone on my ops and dev teams. And one called srmgrs that I can use for specific people, in my case, members of my senior management team.

Name References (@)

The system can be made to scale regardless of the number of users you have. In a small environment like ours, listing people by their first and last names works well, omitting names that are duplicates. But in a larger environment where there are many dupes, you may want to list people by something unique, like their network ID or login name. One additional feature you can use is a toggle in the configuration file ($CFGbRequireAtForNames) that will make the system only recognize names or groups if they’re preceded with an @ sign, much like twitter. For example, if you have a Will Smith and you want him to be addressable as Will, this is probably a good idea. Otherwise any sentence like “will you do something” would end up copying Mr. Smith.

System Paging Integration

Nagios

In order to integration nagios paging with DSPS it helps to have nagios running on the same machine as the pagingd daemon, though it’s not required (you can always fake it via ssh). When nagios wants to send a page (nagios3/misccommands.cfg) it should pipe the MESSAGE into the pagingd command with a parameter of nagios. Don’t worry that pagingd is already running as a daemon and this will be executing a second instance of it from the command line. The command

line version will send the command to the daemon. Let's look at an example nagios3/misccomands.cfg entry:

```
define command {
   command_name    notify-opscall-via-paging-system
   command_line    /usr/bin/printf "%b" "$NOTIFICATIONTYPE$ alert for service: $SERVICEDESC$\nHost: $HOSTNAME$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\nInfo: $SERVICEOUTPUT$\nDate: $LONGDATETIME$ opscall " | /usr/local/bin/pagingd nagios
   }
```

Notice two things about piping the message into pagingd. For one, we have the parameter nagios passed in on the end. That's a hardcoded parameter that DSPS looks for to understand that its getting a message with no human sender and that the message is the exact text it's being fed (i.e. no headers). Also, we end the message content with the string "opscall", which matches our Escalation configuration (look immediately under "Configuration Examples" above). In this case it's just like someone sending a message with a person's name in it, except the name we're having nagios include is "opscall." DSPS will look in the Schedule section of the same group (in this case, Ops) to figure out which person is on call and send it to that person. It'll also start a timer (due to the Escalation part of the config) to make sure someone responds in time.

Emergency

Another integration piece is configuring your system to submit emergency pages to DSPS. These are expected to come in formatted as an email, meaning it will have headers that DSPS should strip away. $CFGsEmergencyEmailTrigger is used to specify the condition that classifies an incoming email an emergency page. For example, we use

```
$  CFGsEmergencyEmailTrigger = ''^Subject: EMERGENCY';
```

so if an email has a subject containing the word Emergency, it matches. Once a message has matched, DSPS treats it as if the message text contains the Escalation "tag" for the group that's defined in $CFGsEmergencyGroup. Here's an example:

```
'StationRelations' => {

        'members' => {
          'Rakiesha'                => '5555551234',
          'Janeen|smiley'       => '5555551234',
          'Sara|hopey'          => '5555551234',
          'Sabrina|miller'      => '5555551234',
        },

        'escalation' => {
```

```
            'tag' => 'cscall',
            'timer' => 299,
            'on_expire_to' => 'ops',
            'cancel_msg' => "[Escalation canceled.  You're on your own.  Good luck!]",
        },

        'schedule' => {
          '20121018'  => 'Sabrina',
          '20121024'  => 'Janeen',
          '20121031'  => 'Rakiesha',
          '20121107'  => 'Sara',
          '20121114'  => 'Sabrina',
          '20121121'  => 'Janeen',
          '20121128'  => 'Rakiesha',
          '20121205'  => 'Sara',
          '20121212'  => 'Janeen',
          '20121219'  => 'Sabrina',
          '20130102'  => 'Sara',
          '20130109'  => 'auto',
        }
    },
```

and we have:

$CFGsEmergencyGroup = ‘StationRelations’;

So in our world, an email that comes into DSPS with a subject of Emergency, would be treated as if the word “cscall” was written in the message (even though it’s not) and the Schedule from our StationRelations group would be invoked to determine who to page, complete with a 5 minute timer. Because we’ve also configured RT integration, all texts in that room will be logged to the associated RT ticket. Initially it will just be a room with the on call person in it. But if they require help and reply with someone else’s name (pulling that person into the room), there will be more to log.

This begs the question of how to pipe emails into DSPS in the first place so it can look for, in this case, matching Subject lines. If you’re using a sendmail compatiable MTA, a simple entry in /etc/aliases will work:

```
dsps:    | /usr/local/bin/pagingd
```

Then emails to dsps@mydomain.com will automatically be forwarded. You can make an address that’s more intuitive for your customers, but you’ll undoubtedly soon get over-excited customers that page you with every little detail. One way to filter the input is to employ a call center to

accept calls, apply some filters you setup (types of questions/calls they accept), and then have the call center send the email to the paging system.

Request Tracker

DSPS support ticket creation and ticket commenting via RT. It's only enabled for emergency pages, as classified in the above section. The configuration looks like:

```
our $CFGsRTConnectionString = '/usr/bin/ssh automation\@rt.myCompany.com /usr/bin/rt';
our $CFGsRTTicketQueue = 'Emergency-Tickets';
```

The first of those settings specifies the path to the RT CLI command. If it's on the local box something like "/usr/bin/rt" is enough. But in our case RT is on a remote server. So we include the full ssh command (public key authentication happens behind the scenes) to execute the command remotely. To use the remote ssh version you'll need to make sure the DSPS server can kick off the RT command without being prompted for a password.

The remaining config parameter sets the name of the Queue that RT should create the emergency ticket in.

New DSPS Users

To use DSPS you need to be configured as an existing user in the paging system. There are two types of setup you'll need:

1. Setup your company – get an account with signalhq.com
    - You'll be assigned a short-code, account number and campaign number
    - They can bill you monthly for number of texts sent
2. Setup each new user that you want configured with the paging system
    - Have the user opt-in by sending a signalhq specified KEYWORD from the user's cell phone (text message) to the provided short-code
    - Contact signalhq, provide your campaign number, account number, and the user's cell phone number. Explain you want the "Public Interactive Lock In" but with a different campaign number.
    - Repeat for each user of the system – each cell phone