Application Summary

DSPS is designed to facilitate phone to phone communications between individuals and groups. Its goals include:

- You don't have to know the phone number of the person you want to contact.
- With a rotating support schedule 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.

Features

- Intuitive name referencing - use someone's name to contact them, not a phone number
- Group paging - use a single name to page a large 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 its 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

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 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, though DSPS will prefix each message with its sender's name.

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 60 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

Types of Users

There are two types of users in DSPS, human users and system users. System users should be used to submit pages from crons, automation scripts, Nagios, etc. System users:

- cannot be pulled into a room by name — they have to pull you in with them (the exception is the :maint command)
- won't appear in the Audience list for the room unless specifically specified in the config file
- can have their pages blocked by various system filters — humans are never blocked

On Call Schedules

Your users are divided into groups and groups can optionally have an On Call schedule. The group name can 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 on a 1 week. 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 without having to know who it is.

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. Escalations can optionally tie-in to RT (if you use Request-Tracker) and log the entire paging conversation in a new ticket for each event.

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.

System Monitoring - Nagios, Zabix, etc

DSPS was originally written to integrate with Nagios and you'll see many of the filtering commands have Nagios in the command name. But the configuration file now supports two regular expressions (nagios_problem_regex & nagios_recovery_regex). By changing those you can easily make DSPS work with any monitoring system that sends the standard two types of alerts. More detail is in the SysAdmin section below.

User Commands

Users interact with DSPS by sending text messages to the system, or by sending text commands. The currently supported commands are:

COMMAND	SCOPE	DESCRIPTION
FILTERING		
:vacation T	You	Specify the amount of time (T) you're on vacation starting now. While on vacation you are removed from group pages and on-call schedules/ escalations. You can still be paged directly by name.
:norecovery	You	Disable all Nagios RECOVERY pages.
:recovery	You	Re-enable all Nagios RECOVERY pages.
:smartrecovery	You	Enable the Smart Recovery option. When enabled you only get a Nagios RECOVERY page if: 1) it's during normal "awake" hours, *or* 2) it's within three minutes of the last PROBLEM page. The theory is you want all recoveries during the day. And at night you want them if them come right after an issue (indicating a host/service hiccuped but you don't really need to get out of bed), but not if something is really down, you got up to fix it, and the recovery comes in hours later, potentially waking you up a second time.
:noregex T REGEX	Global	Block all Nagios pages that match /REGEX/ for the specified time (T).
:nore	Global	(same as :noregex)
:sleep	Global	Block all Nagios LOAD and RECOVERY pages for 3 hours; This can be used at the end of maintenance or a fix when the server's load averages haven't returned to normal yet. It means you're not interested in the subsequent load pages or when Nagios notices they're back to normal.
:snooze	Global	Block **all** Nagios pages for 15 minutes.
:nonagios T	Global	Block **all** Nagios pages for the specified time (T).
:nagios	Global	Unblock all Nagios pages (negates :nonagios)
CONVERSATIONS		
+NAME	Room	Send the room's chat history (up to the last 5 messages) to the specified person and pull them into the room.
:email ADDR	Room	Send the room's chat history to the specified email address.
:leave	You	Leave the chat room.
-	You	(same as :leave)
:disband	Room	Disband the room, kicking everyone out of it. The room is destroyed.
:pull PEOPLE	Room	Disband the current room and create a new one, pulling the specified people into it with you; used to shrink a room.
:autoreply T TEXT	You	Set an auto reply (TEXT) which is sent on your behalf any time you're pulled into a room for the specified amount of time (T). When set you still see all messages that you normally would and can interact/reply.
:ar	You	(same as :autoreply)
:summary DSC; IMT	Room	Summarize the conversation with a description (DSC) and impact (IMT). This is used if you enable room logging and choose to publish a weekly report of the issues handled by DSPS. The summary command will be used to describe what the conversation was about. The semi-colon is required as it distinguishes the two text fields.
:sum	Room	(same as :summary)

COMMAND	SCOPE	DESCRIPTION
CONVERSATIONS		
!TEXT	Room	Send the message to the room without parsing for names, groups or commands. The message is also not logged to RT.
^TEXT	Room	Place the room in Broadcast mode and send the specified message. In Broadcast mode, instead of replies going to everyone in the room, they only go to the initial sender (the person that set Broadcast mode). Useful for roll calls or larger audiences.
QUERIES		
?oncall	You	Show the on call schedule for all defined escalations.
?vacation	You	Show currently configured vacation time for all paging users.
?rooms	You	Show all rooms and participants in all current conversations.
?rt	You	Show the RT ticket number associated with your current room (if one exists).
?NAME	You	Define the given name. If it's a group or alias, display the individuals members. If it's an individual member's name, display their phone number.
?groups	You	Show all groups and aliases that aren't hidden.
?filters	You	Show the system filters that are currently in effect.
?triggers	You	Show the currently defined triggers and their statuses.
MISC		
:arm TRIGGER	Global	Arm all triggers that match the name TRIGGER.
:disarm TRIGGER	Global	Disarm all triggers that match the name TRIGGER.
:swap NAME	You+	Swap on call schedules with the person NAME. You and the target person have to share at least one on call schedule. The swapped time period (defaults to a week) will be the next week each of you have. If you're already in your week and less than 2 days into it, your current week is swapped.
:maint USER	Room	Creates a maintenance window room and pulls the specified USER in with you. USER is optional. See the detailed explanation on the next page.

Times

All time specifications in DSPS commands (such as for :vacation, :noregex, :autoreply, etc) take the form of numbers followed by a unit specification. For example, 3h to specify three hours. Valid units are 's'econds, 'm'inutes, 'h'ours, 'd'ays and 'w'eeks. If no unit is specified, seconds are assumed.

Maintenance Window Rooms

A maintenance window room is used in a situation when you expect to get pages. When you're doing intentional maintenance and maybe you don't know exactly what might break so you can't specifically tell Nagios not to care about a particular service (by setting Downtime). By setting up the Maint Window room you can catch everything. It essentially tells DSPS that regardless of who the monitoring system (Nagios or otherwise) *wants* to page, it should page you instead. You can use the :maint command in two ways. One is to issue it with the name of a system user (like nagios). That pulls the system user into the room with you. The other way is to not specify a user, in which case the "default maint" user is pulled it. The default maint user is specified in the config file. Here's the important part: *when a system user is in a maintenance room, it cannot pull additional people in or start escalations*. Only humans can pull additional people into maintenance rooms. This means if I start a maint room with myself and the nagios user, then all nagios pages will only go to me (unless I elect to pull additional people in too) for the duration of the room. The room will expire 60 minutes after the last page, as usual.

Architecture

The DSPS code includes both a server daemon and CLI client portion. The daemon must be running to accept and process pages. If you want to install DSPS on multiple servers in your network you probably only want one daemon. That way you have a centralized via of all paging rooms/audiences. You start the daemon on that main server ("dsps -d" or via the init.d script). When you run dsps from the command line you're always running the client code. By default it looks for the daemon on localhost:2176. If you set sys:dsps_server (detailed below) it will look on another machine/port.

SysAdmin Configuration

Groups & Users

Users can belong to multiple aliases but they can have (and must have) exactly one group. If you don't want to use groups then just make one and put all of your users in it. We use groups as functional teams, like Ops, Dev, or named after a particular product. A group is defined in the config file by just giving it a name.

Users require four fields to be defined. They are:

- Name
- Regex
- Mobile Number
- Access Level
- Options (optional)

Here's a config example with syntax:

```
group: Ops
  u: Rick, rick|ennis, 1234567890, 100
  u: DavidF, davidf|franz, 1234567891, 100
  u: Sean, sean|shawn|smith, 1234567892, 100
```

Note: A user's regex *must* include their actual name.

As mentioned on page 3, there are human users and system users. System users are defined with an exclamation mark at the beginning of their name. This tells DSPS that they're a system user and doesn't allow them to be pulled into conversations by their name. Here are two of ours:

```
group:!cli
        # standard system user for nagios - normal rules apply
        u:!nagios, nagios, 9999999999, 0

        # redirect directive for emergency, as described below
        u:!emergency, emergency, 9999999998, 0, redirect:srcall
```

Notice the group name begins with an exclamation mark also to exempt it from being referenced in a conversation. As a result the regex field is basically irrelevant but I kept them consistent for readability. Phone number is still important because it has to be globally unique among all users.

The emergency user also demonstrates the 5th field, options, that can be in a user definition. To date the "redirect:" directive is the only valid option. Redirect does two things:

- Automatically pulls that person/tag/etc (in this case, escalation) into the conversation any time this user submits a message.
- More importantly, it tells DSPS to *ignore any name/group/escalation references* in the message text itself.

For example, we have an 800 number where a call center transcribes our emergency calls into emails and sends them to a specific email address on our DSPS server. The mail daemon pipes those messages into DSPS as user "emergency." Those messages include the name and phone number of the customer that's calling us about whatever emergency is they're experiencing. Suppose the customer that's calling is named Sara. I don't want the word "Sara" being in that message to make DSPS send the page to my coworker Sara, as it automatically would under normal circumstances. So the "redirect:" directive tells DSPS to ignore all references in those messages and send them to the "srcall" person instead.

Aliases

Aliases are a quick and dirty way to use a single name and have it page multiple people. The people that are paged can be across different groups or unrelated in any way. Aliases be as simple as a name and who to page, called the referent (abbreviated with an “r”):

```
alias: westcoasters
        r: sara, sabrina, davidf
```

Or an alias can have an additional option to make it hidden. When hidden an alias is removed from a message before it goes out. For example, if I have this alias defined:

```
alias: allstaff
        r: rick, davidf, sean, davidc, sara, janeen, sabrina, jeff
        o: hidden
```

And I send the text message “Attention allstaff today’s announcement is moved to 2pm” what everyone will actually receive will be: “Attention today’s announcement is moved to 2pm”.

Escalations

An escalation performs these two functions in one:

- Gives a single tag/name to page which pages a different person depending on the on call schedule that’s been configured for that tag.
- Provides a way to page a larger audience if the initial page recipient doesn’t respond within a specified amount of time.

Let’s start with an example:

```
escalation: SRCall
        timer: 300
        escalate_to: stationrelations, ops
        alert_email: stationrelations@mycompany.us, techdept@mycompany.us
        swap_email: stationrelations@mycompany.us
        cancel_msg: Escalation canceled.
        options: rt_queue: Emergency
        options: rt_subject: EMERGENCY CALL
        schedule: 20140108 Sabrina
        schedule: 20140115 Sara
```

schedule: 20140122 auto/sara,jeff,janeen,sabrina

This defines the “SRCall” tag (Station Relations call, in our world). All commands, names and other identifiers are case-insensitive when used in a page. So “I need some help from srcall” will work just as easily.

The schedule lines define who is on call for srcall and when. On call schedules are always 1 week long but you can choose which day of the week to transition. We use Wednesdays. Schedules lines are YYYYMMDD so they sort numerically. This example shows Sabrina is on call starting on 1/8/2014 through 1/15/2014, at which point Sara takes over. The following week on 1/22/2014 the admin has specified a recurring schedule. Recurring schedules use “auto/“ after the date and are followed by the people in the order that you want them on call. Each person will get a 1 week rotation, with the first one starting on the date at the beginning of the line. After the last person the first one will come up again and the rotation starts over. That way you can write a single schedule line and be done if you like. Or in this case we’ve hardcoded a few specific ones then added the auto schedule.

When someone pages srcall the page goes only to the person that the schedule specifies for the current date (unless that person is already in a room with other people, in which case the other room occupants will see the page too). Assuming the on call person is not in another room for a previously existing conversation, the system will wait 5 minutes (the [timer] setting) for a reply. The reply for the on call person can be anything. If a reply is received DSPS confirms by sending them the [cancel_msg]. If no reply is received within 300 seconds DSPS resends the original page to the larger [escalate_to] group, which can consist of multiple names/groups/aliases. After firing the Escalation is complete. No additional “automatic” pages are sent on that event.

If [alert_email] is set DSPS sends an email to those addresses any time the escalation is used.

If the [options:rt_queue] & [options:rt_subject] values are set (and the RT integration is configured in the “System Settings” section below) then a new ticket is created in Request Tracker each time the escalation is used. All pages that take place in a room that where an escalation fired are subsequently logged to the RT ticket.

If [swap_email] is specified those addresses are notified any time someone on the srcall on call list uses the :swap command.

[min_to_abort] (not shown in the above example) is an integer setting. If specified and the on call person is already in a room with x number of people when the escalation page comes in, if x is greater than or equal to min_to_abort, the escalation is skipped. [min_to_abort] defaults to 2. In practice that means if I’m on call for srcall, and I’m already in a DSPS room with 1 other person when an srcall page comes through, the system will conclude there are 2 people in the room where I (the on call) person is, and that’s enough to not escalate the page. In other words, two people have seen the message; there’s no need to continue escalating to more people. You can set [min_to_abort] higher to change this behavior.

You can define as many escalations as you like. We tend to use one per team that’s on call.

System Settings

The system section of the configuration file is used to set options that affect DSPS as a whole. All of these options begin with “sys:”. They are listed here with syntax and then as an example:

- sys:default_maint: USERS/GROUPS/ALIASES
- sys:default_maint: !nagios

default_maint specifies who will be pulled into the room when the :maint command is issued with no user. e.g. ":maint" instead of ":maint jim"

- sys:require_at: BOOLEAN
- sys:require_at: 0

If require_at is false, any user, group, alias or escalation can be referenced directly in a page. "hey jim I need some help." If require_at is true, then it has to be "hey @jim I need some help." This is particularly useful for large user lists or names that are English words. NOTE: If you set this to true, then any reference to a user, group or alias in your config file (aside from it's original definition) must include an at sign as well. For example, if you have an Escalation, entries in the escalate_to field will need @s. If you have a user that includes the "redirect:" directive, the entries in the redirect will require @s.

- sys:show_nonhuman: BOOLEAN
- sys:show_nonhuman: 0

If true, system users (those whose names start with an exclamation mark) will appear in the ? rooms" command.

- sys:gateway_url: URL
- sys:gateway_url: http://app.signalhq.com/messages/send_individual_message

Specify the HTTP URL to POST to when placing an outgoing page.

- sys:gateway_params: [PARAM][PARAM]
- sys:gateway_params: [a=1234567890][c=012345][mdn=$CELLNUMBER][m=$MESSAGE]

Specify POST parameters for the gateway_url. Each parameter should be enclosed in square brackets. The variables $CELLNUMBER and $MESSAGE are substituted in with their actual values before the POST is made.

- sys:fallback_email: EMAIL_ADDRESS
- sys:fallback_email: 1234567890@vtext.com

If a page fails to send in a way that DSPS can detect — i.e. the POST to gateway_url does not return 200— then DSPS sends the page as an email to fallback_email. The idea here is to use one of those email to text gateways that most phone carriers provide. Note: this email is passed directly to sendmail so you can only specify a single address.

- sys:recovery_regex: REGEX
- sys:recovery_regex: ^[-+!]{0,1}RECOVERY\s+
- sys:nagios_problem_regex: REGEX
- sys:nagios_problem_regex: /^[-+]{0,1}PROBLEM/

These two regexes define the problem and recovery pages for a Nagios-style monitoring system. By changing these you can adapt DSPS to other monitoring systems. **Note: Due to internal requirements, both regexes need to allow for DSPS to prepend a plus or minus sign to the front of the message.** If you're using Nagios, the default regexes will work perfectly.

- sys:dsps_server: IP_ADDRESS:PORT
- sys:dsps_server: localhost:2176

If you want to have multiple servers that can all submit pages locally then it will be advantageous to run the DSPS daemon one machine that all the client machines can submit to. By having a single copy of the DSPS daemon it can track rooms/audiences for your entire network. The dsps_server directive is used on the client machines to tell them where to find the daemon. For a single machine, use localhost. If you want to use SSL see the next section.

- sys:server_listen: IP_ADDRESS:PORT
- sys:server_listen: 0.0.0.0:2176

Tell DSPS what IP address and port it should listen on. You can use this to lock it down to a specific IP. By default it listens on all local IP addresses (0.0.0.0).

- sys:rt_connection: RT_COMMAND
- sys:rt_connection: /usr/bin/ssh automation@rtserver.mycompany.us /usr/bin/rt

If you want to have DSPS interface with Request Tracker to create tickets for particular Escalations and log escalated rooms to RT, rt_connection is used to specify the path to the RT CLI (generally /usr/bin/rt). If you setup ssh keys it can safely be on a remote host.

- sys:smtp_server: HOSTNAME
- sys:smtp_server: mail.mycompany.us

In order for DSPS to send email smtp_server should be the name of the SMTP host that accepts messages.

- sys:smtp_from: EMAIL_ADDRESS
- sys:smtp_from: noreply@mycompany.us

If DSPS sends out email, this is the address the message will say it's from.

- sys:admin_email: EMAIL_ADDRESSES

- sys:admin_email: ops@mycompany.us, rick@mycompany.us

On several email events DSPS automatically Cc's the Admins. In test mode (dsps -t) all emails and rerouted to the Admins.

- sys:override_user: PAGING_USER
- sys:override_user: !signal

DSPS works most efficiently with the daemon listing on a port, waiting for clients to submit requests via HTTP. But some applications —and some paging gateway companies— don't provide HTTP clients. In these cases it's sometimes easier to receive the request as an email. To facilitate email message submission DSPS has two specific options. One is override_user. This is used to tell DSPS that when messages are submitted via one user it should substitute in another. For example, suppose your /etc/aliases had *pageme: "| dsps -u viamail -e"* to say that emails to pageme@yourdomain should be submitted to DSPS as the "viamail" paging user. You'd define that user as "!viamail" in the config file to mark it as a system user. Then you'd set set "sys:override_user: !viamail". That way DSPS knows when it gets those messages it should look elsewhere for the actual paging user to use. The actual paging user comes from the next option below.

- sys:override_regex: REGEX
- sys:override_regex: ^Subject: sms_reply:(\d+)

In the case of a message that's submitted by the override_user, then override_regex tells DSPS where to find the actual user name. It should be a regular expression with a single set of capturing parenthesis.

- sys:rt_link: URL
- sys:rt_link: http://help.ds.npr.org?/Ticket/Display.html?id=

If you're using RT and you want a link with the new ticket number automatically appended to the end of notification emails for Escalations that have an RT association, then rt_link is where you define that link. DSPS will automatically append the ticket number on the end.

Using SSL

DSPS uses a client-side SSL library to connect to any HTTPS URLs, including the DSPS daemon itself. But it doesn't currently have the server library setup. So if you want to run the daemon over SSL, have it listen on an alternative port and put an SSL end-point (like nginx) on the main port, forwarding connections.

Nagios Integration

Nagios can submit pages to DSPS via dedicated paging users. In most setups a single "! naigos" paging user works well. That way if Nagios pages person A about one issue and a few minutes later pages person B about another, the two people end up combined in the same chat

room (because !nagios can only be in one room at a time — see “room dynamics”). If that’s not desirable behavior you can make multiple users that Nagios submits pages as. Here’s the simple example as defined in nagios3/misccommands.cfg:

```
define command {
   command_name    notify-opscall-via-dsps
   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/dsps3/dsps -u nagios -s -
f }
}
```

Note how who to page (opscall) is hardcoded at the end of the message. That way any alert that Nagios sends to this command will page our opscale escalation, getting whoever is on call at that moment.

Gateway Account

In order to send and receive messages with mobile phones you’ll want to hire an SMS gateway company. I recommend “www.signal.co” (ask Jeff, jjudge@signal.co).

1. Setup your company with Signal (for example)
   - You’ll be assigned a short-code, account number and campaign number
   - They can bill you monthly for number of texts sent/received
2. Have them setup each new user that you want configured in DSPS.
   - Have the user opt-in by sending a Signal specified KEYWORD from the user’s cell phone (text message) to the provided short-code.
   - Contact Signal, provide your campaign number, account number and the user’s cell phone number. Explain that you want the “Public Interactive Lock In” but with your campaign number.
   - Repeat for each user you want to enroll.

Debugging

The easiest way to debug DSPS issues, bugs or even configurations is to run it in Test mode:

- dsps -t

In Test mode DSPS:

- won’t send any pages
- won’t send any emails
- logs everything to the console

When you’re done with debug mode you can use control-C to exit. At that point you need to restart the daemon in the background if you want it to continue running (dsps -d).