Rave Alert

From Anywhere. To Anywhere.

CAP Inbound Messaging API

Developer's Guide

Rave Alert

Rave Campus Messenger

November 2013

Rave MOBILE SAFETY Safe. Secure. Connected.™

Table of Contents

Introduction

Rave Alert and Rave Campus Messenger (Rave Alert) is a powerful multi-modal communications platform, providing a way to rapidly distribute messages through a variety of outlets. Rave Alert includes an *Admin Console*, providing a template-based alert authoring flow. The *Admin Console* provides Rave Alert Administrators with quick and easy way to manage the content, methods, and recipients for a given Broadcast Alert.

This document describes Rave Alert's Common Alerting Protocol (CAP) Inbound Messaging Interface. Rave's Inbound CAP Interface provides a means to automatically trigger a Broadcast Alert upon the receipt of a CAP-formatted XML message meeting criteria you define.

The CAP XML standard is managed by OASIS' Emergency Management Technical Committee. CAP is an XML message format designed to carry alert events between systems. Many alert notification systems, such as digital signage and siren platforms, look to CAP as their interface standard. Most notably, the adoption of the CAP standard by the Federal Emergency Management Agency (FEMA) and the National Emergency Number Association (NENA) has sealed CAP as the standard-of-choice for describing notification events.

Rave can accept CAP messages from any platform that can send a CAP-compliant XML document to Rave's HTTP BASIC authenticated REST web-service via HTTPS. CAP message originators can include an off the shelf application already under your control, or a custom application developed by you.

Requirements

To undertake development against Rave Alert using the Inbound CAP interface, you must have access to the following:

1. A Rave Alert domain, configured to enable the *CAP Inbound* interface.
2. Credentials to log into the *Rave Alert Admin Console* as a *Domain Administrator*
3. A working knowledge of how to send a Broadcast Alert from within the *Rave Alert Admin Console*, including an understanding of how to configure each notification method you plan on triggering via the *CAP Inbound* interface.
4. Where you are developing your own CAP message producer:
    a. A working knowledge of web-services application development in the language of your choice [*sample code is provided for Java*]
    b. A working knowledge of how to populate a CAP-compliant XML document.
5. Where you are configuring system capable of generating a CAP document:
    a. An originating-system which is able to send a CAP XML document (versions 1.1 or 1.2) posted to a REST web service using HTTPS and Basic authentication.
    b. Working knowledge of the configurations necessary to work within the above system(s) targeted to initiate a Rave Alert
    c. Access to those who support your desired CAP message producing system where you require further information on how to configure the platform.

Design Considerations

The CAP Interface is a powerful, yet fairly complex specification. Furthermore, there are a number of settings within the Rave Alert *Admin Console*, which when combined with the CAP message, work together to dictate the content, behavior, communications modes, and alert recipients for a given notification event.

Prior to starting this integration effort, we recommend you collect and document the following:

- Under what circumstances will you trigger an alert?
- Who must receive each alert? Do the recipients need to vary based on the circumstances or content of the alert?
- What alert modes (Text, Voice, eMail, etc.) should be employed for each alert?
- For each alert mode, what unique configurations are necessary?

We recommend you study your requirements, and carefully compare them to the Rave Alert configurations and the CAP XML message content described in this document. By organizing your CAP sending system behavior, CAP document content, and Rave Alert configurations in a way that minimizes the number of *CAP Notification Rules* and *CAP Notification Templates* configured within Rave Alert, you will simplify your testing and configuration tasks.

Information Architecture

The Rave Alert environment is controlled via the Inbound CAP interface through four separate components:

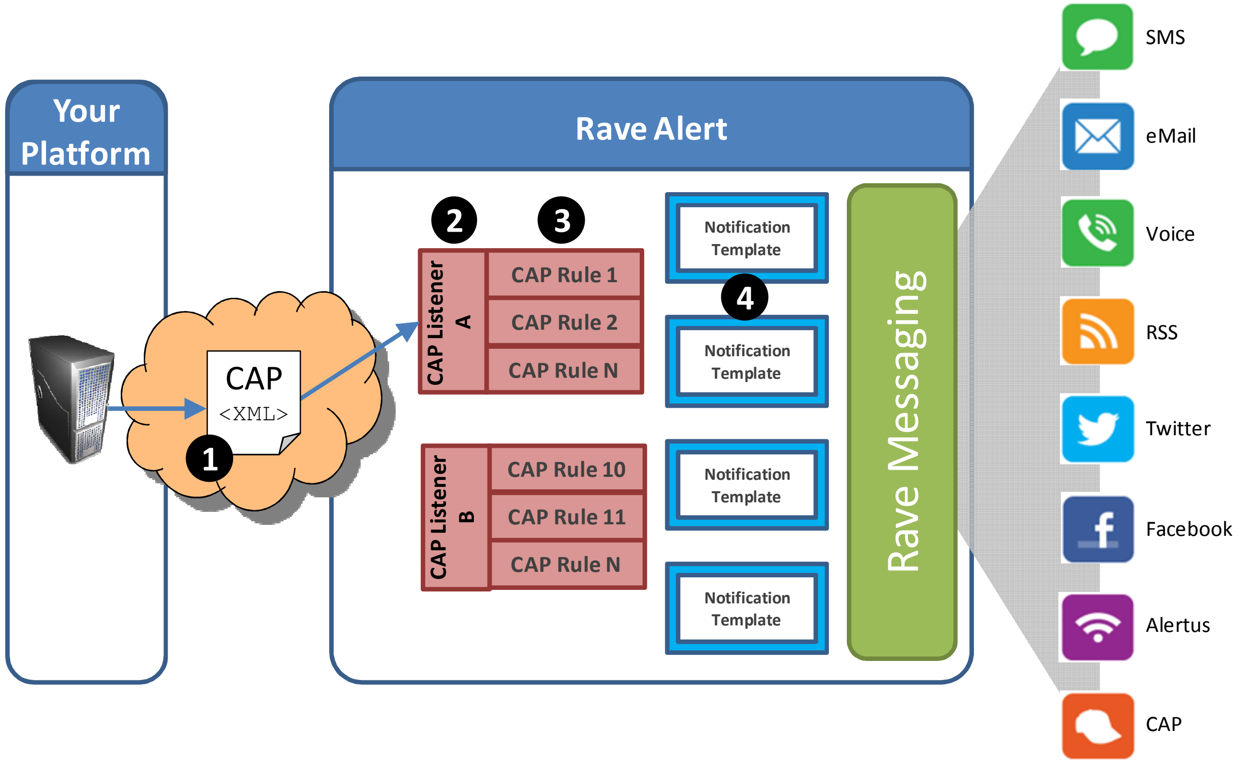


Figure 1: Rave Alert Inbound CAP Functional Architecture

1. <u>*CAP XML Document*</u> which carries the alert content, and contains fields and field values which drive notification logic within Rave Alert

2. <u>*Inbound CAP Listener:*</u> exposes a web interface which validates the message sender's credentials, and accepts the *CAP XML Document*

3. <u>Inbound CAP Notification Rules:</u> are referenced by the Rave Alert Rules Engine to determine if a given CAP message should result in an alert notification, which *CAP Notification Template* should be used for the alert, and how to map the content within the *CAP XML Document* to the message sent via each alert mode.

4. <u>CAP Notification Template:</u> identifies the notification methods to be used for a given alert, specifies any options unique to each selected notification method, and defines who should receive the notification.

Rave Alert and CAP XML

OASIS Specifications

A CAP formatted XML document is used to externally trigger a notification event within Rave Alert. Rave Alert supports both OASIS CAP versions 1.1 and 1.2. Where both versions are supported by your platform, we recommend the use of CAP version 1.2.

Rave Alert adheres to the CAP specification. Therefore Rave validates your CAP XML documents against the usage rules (e.g. validation of required versus optional fields) and valid values defined by OASIS.

The intent of this Developer Guide is to define the Rave Alert's use of CAP XML. This document does not replace the CAP XML specifications published by OASIS, which are available here:

OASIS CAP 1.2 Specification (Recommended)
http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html

OASIS CAP 1.1 Specification
http://www.oasis-open.org/committees/download.php/15135/emergency-CAPv1.1-Corrected_DOM.pdf

Rave Alert Implementation

Rave Alert requires only a subset of the fields defined in the OASIS specifications. Rave's usage requirements are defined in the following table. This table is based on CAP 1.2.

Field	Rave Usage	Description
CAP "Alert" Group [One Alert Group per Document]		
Identifier	**Required**	A cast by your system to uniquely identify the CAP document. This value MUST be unique across all messages sent by your platform to a given Rave Alert *Inbound CAP Listener*.
Sender	**Required**	Required as per OASIS documentation, but not used by Rave Alert
Sent	**Required**	Required as per OASIS documentation, but not used by Rave Alert
Status	**Required**	Valid values as per OASIS documentation
Msg Type	**Required**	For REST listeners, Rave supports: • Alert • Cancel For ATOM listeners, Rave supports: • Alert All other Msg Types are not supported
Source	Not Used	Not used by Rave Alert

Scope	**Required**	Valid values as per OASIS documentation
Restriction	Not Used	Not used by Rave Alert
Addresses	Not Used	Not used by Rave Alert
Code	Optional	A string set by the sending system. Can be used to direct a CAP message to a specific *Inbound CAP Rule*
Note	Not Used	Not used by Rave Alert
References	Conditional	Required when Msg Type = "Cancel" Format as per OASIS specification
Incidents[1]	Optional	Conditionally mapped to outbound Alert content.
CAP "Info" Group [One Info Group per Document Max]		
Language	Not Used	Not used by Rave Alert
Category	**Required**	Valid values as per OASIS documentation
Event[1]	**Required**	Format as per OASIS documentation
Response Type	Optional	Valid values as per OASIS documentation
Urgency	**Required**	Valid values as per OASIS documentation
Severity	**Required**	Valid values as per OASIS documentation
Certainty	**Required**	Valid values as per OASIS documentation
Audience	Not Used	Not used by Rave Alert
eventCode	Optional	One or more label/value pairs populated by the sending system. Must be formatted per OASIS documentation
Effective	Not Used	Not used by Rave Alert
Onset	Not Used	Not used by Rave Alert
Expires	Not Used	Not used by Rave Alert
Sender Name	Not Used	Not used by Rave Alert
Headline[1]	Optional	Conditionally mapped to outbound Alert content. As per OASIS document, this field should not exceed 160 characters in length.
Description[1]	Optional	Conditionally mapped to outbound Alert content.
Instruction[1]	Optional	Conditionally mapped to outbound Alert content.
Web[1]	Optional	Conditionally mapped to outbound Alert content.
Contact[1]	Optional	Conditionally mapped to outbound Alert content.

[1] See Table 2: CAP to Alert Mode Message Mapping which identifies how these fields are uses by alert mode

		Parameter	Optional	One or more label/value pairs populated by the sending system. Must be formatted per OASIS documentation
	CAP “Resource” Group [Not Used by Rave Alert]			
		Al l Resource Group Elements	Not Used	Not Used by Rave Alert
	CAP “Area” Group [Optional: One Area Group Per Document Max]			
		areaDesc	Conditional	Required where the Area Group is populated. Not used by Rave Alert
		Polygon	Not Used	Not Used by Rave Alert
		Circle	Not Used	Not Used by Rave Alert
		Geocode	Optional	One or more label/value pairs populated by the sending system. Must be formatted per OASIS documentation
		Altitude	Not Used	Not Used by Rave Alert
		Altitude	Not Used	Not Used by Rave Alert

Table 1: Rave Alert profile of OASIS CAP v1.2

[1] See Table 2: CAP to Alert Mode Message Mapping which identifies how these fields are uses by alert mode

Rave Alert Configurations

Only Rave Alert Domain Administrators are able to access the Inbound CAP configurations. If upon logging into your Rave Alert Domain as a *Domain Administrator* you do not see the options described below, contact Rave Support to enable the Inbound CAP feature.

About these Examples

The *CAP Notification Rules* shown in this document are for illustrative purposes only. You are responsible for developing a full understanding of how your CAP Publisher generates and populates CAP messages, and the detailed configurations required to ensure Rave Alert behaves as expected.

- When first creating and testing a *CAP Notification Rule*, it is recommended that you initially specify a limited distribution list, containing only the contacts necessary for testing, within the *CAP Notification Template*. You can always change this at a later date once you confirm the rules are working as intended.
- It is always best to use the designated CAP Publisher to create your test messages. This may be possible where you, or someone you have a working relationship with, controls the CAP producer, and can generate messages to meet your specific criteria.
- In some cases, live testing with a CAP producer may not be feasible. This is particularly true with the feeds generated by the National Weather Service. In such cases, you may need to monitor your State or County feed using a small distribution list, until you are certain it is working as expected.

Configuring your Domain to support the CAP XML API

To enable your Rave Alert domain to receive and act upon CAP XML documents, you need to create and configure three entities within the Rave Alert Admin Console:

- ○ A *CAP Listener*, which defines the web service exposed to receive CAP XML documents from an external system.
- ○ One or more *CAP Notification Templates*, which define the alert modes (Text, eMail, Voice, etc) and alert recipients to be messaged upon the receipt of an expected CAP XML document.
- ○ One or more *CAP Notification Rules* which inspect each CAP XML document received by the *CAP Listener*, and trigger a CAP *Notification Template* when the CAP XML document meets the criteria defined in the *CAP Notification Rules.*

Instructions on how to configure each of these entities is provided below.

1. Configure a CAP Listener

The first step in establishing an Inbound CAP interface is to configure an *Inbound CAP Listener*. This step defines how the system generating the CAP message will communicate with Rave Alert.

NOTE: ***This document assumes your application will post CAP documents to a REST Web Service****!*
While there is an option to poll and ATOM feed containing CAP XML documents, its primary purpose is to monitor National Weather Service alerts. We do not recommend ATOM for custom integrations.

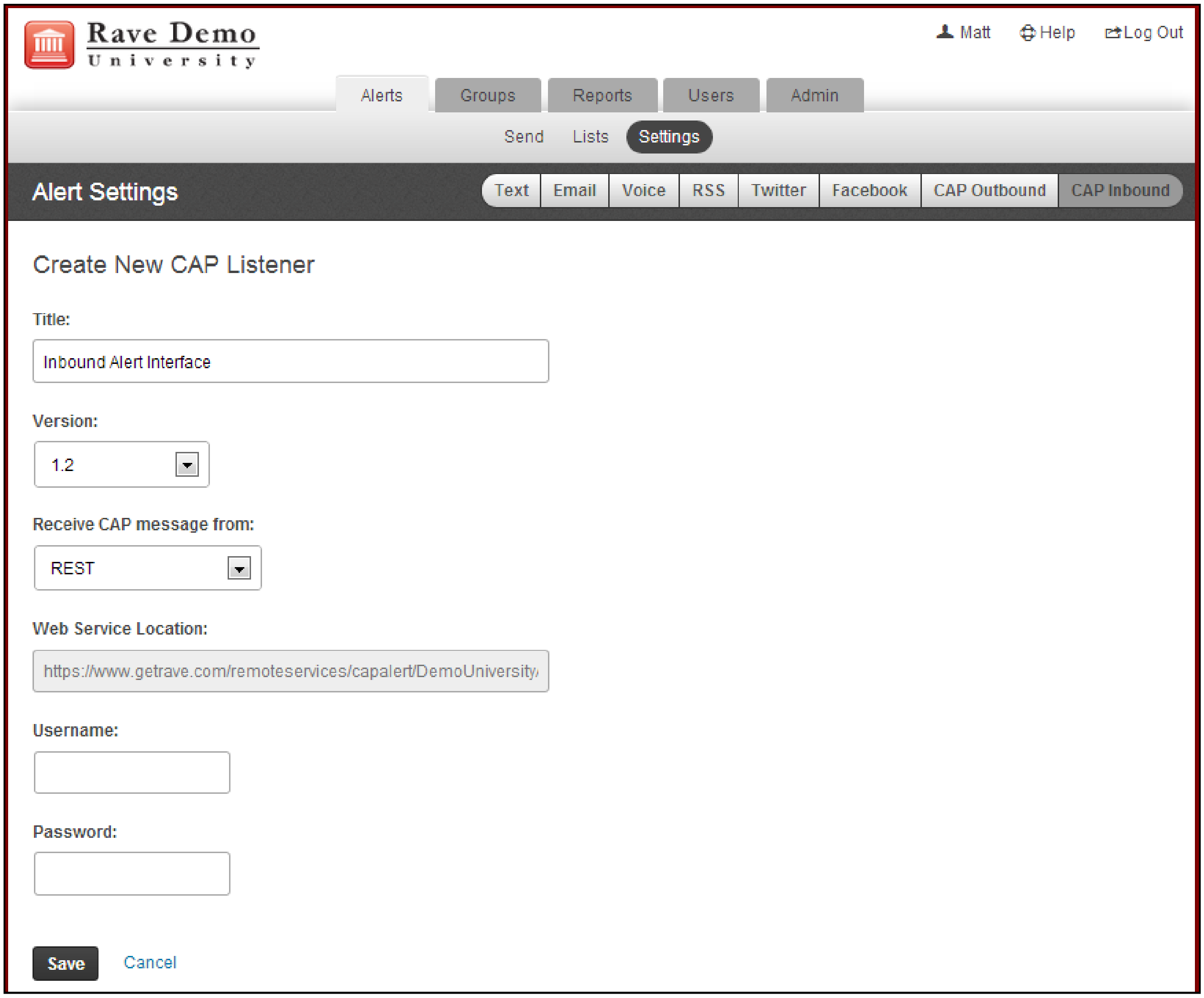
1. Specify a *Title* for your Listener. The title you provide is for your future reference.

2. Select the OASIS CAP *Version* your platform's XML documents will conform to. Version 1.2 is recommended.

3. Specify the message protocol that Rave Alert will *Receive CAP messages from*. The use of REST is *strongly recommended*.

4. Enter a *Username* and *Password*. The REST-configured CAP Listener will authenticate these credentials using HTTP BASIC authentication.

5. Record the provided *Web Service Location*. This value is determined by Rave Alert. Your system should post CAP XML documents intended for this listener to this URL.



Alert Settings

Create New CAP Listener

Title:

Inbound Alert Interface

Version:

1.2

Receive CAP message from:

REST

Web Service Location:

https://www.getrave.com/remoteservices/capalert/DemoUniversity/

Username:

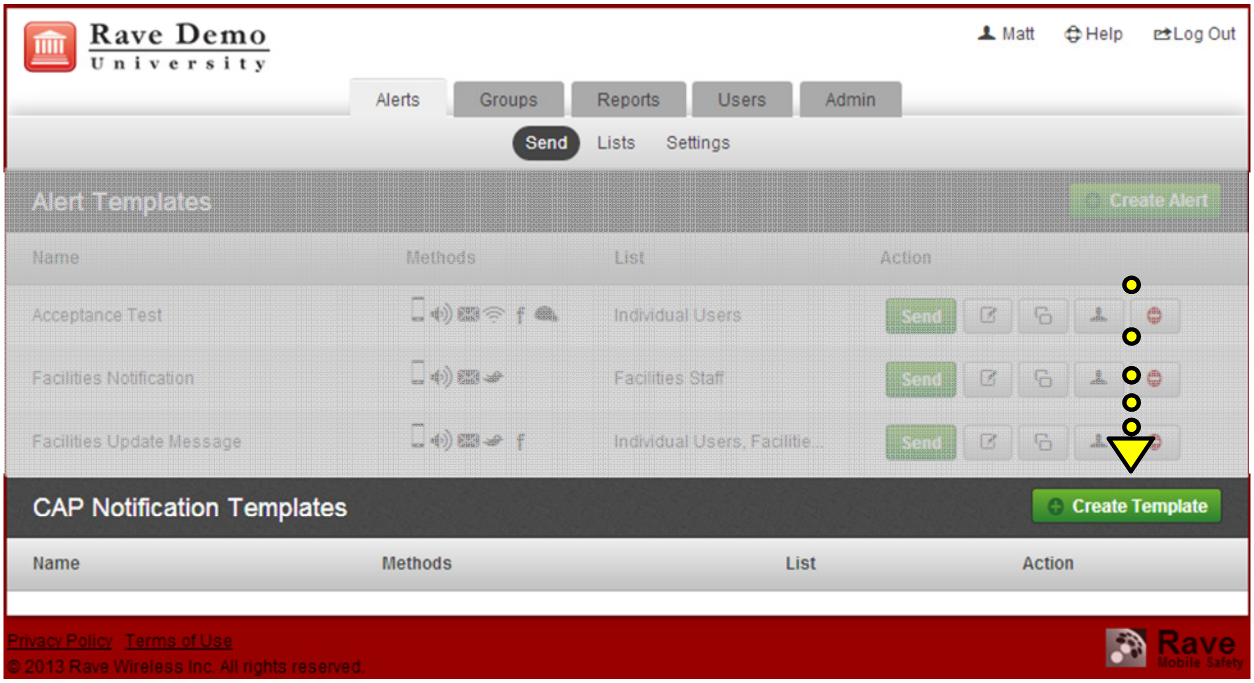
Password:

Figure 2: Creating a Web-Services CAP Listener

2. Create a CAP Notification Template

The second step is to create one or more Inbound *CAP Notification Templates*. You must create one *CAP Notification Template* for each unique combination of alert modes, alert mode settings, and alert recipients identified during the design phase of your project.

To create a *CAP Notification Template*, Navigate to the *Alerts → Send* tab. Scroll down below the list of *Alert Templates* to a section called *CAP Notification Templates*. Click the *Create Template* button within this heading:



Alert Templates

Name	Methods	List	Action
Acceptance Test		Individual Users	Send
Facilities Notification		Facilities Staff	Send
Facilities Update Message		Individual Users, Facilitie...	Send

CAP Notification Templates

Name	Methods	List	Action

Figure 3: Creating a CAP Notification Template

There three steps to populating a *CAP Notification Template*. This process is similar to how a manually-triggered *Alert Template* is created. The primary difference is that when triggered via CAP XML, the content for the Broadcast Alert messages is provided by your CAP XML document, rather than through this user interface.

The steps follow. If you need more information on how to populate mode-specific settings within this form, please refer to the *Admin Documentation* for sending a Broadcast Alert, which is accessed via the link in the upper right corner of the *Admin Console*.

1. Provide a *Template Name*.

   Hint: You may want to select name that is as generic as possible, as you can reference a given *CAP Notification Template* within any number of *CAP Notification Rules*.

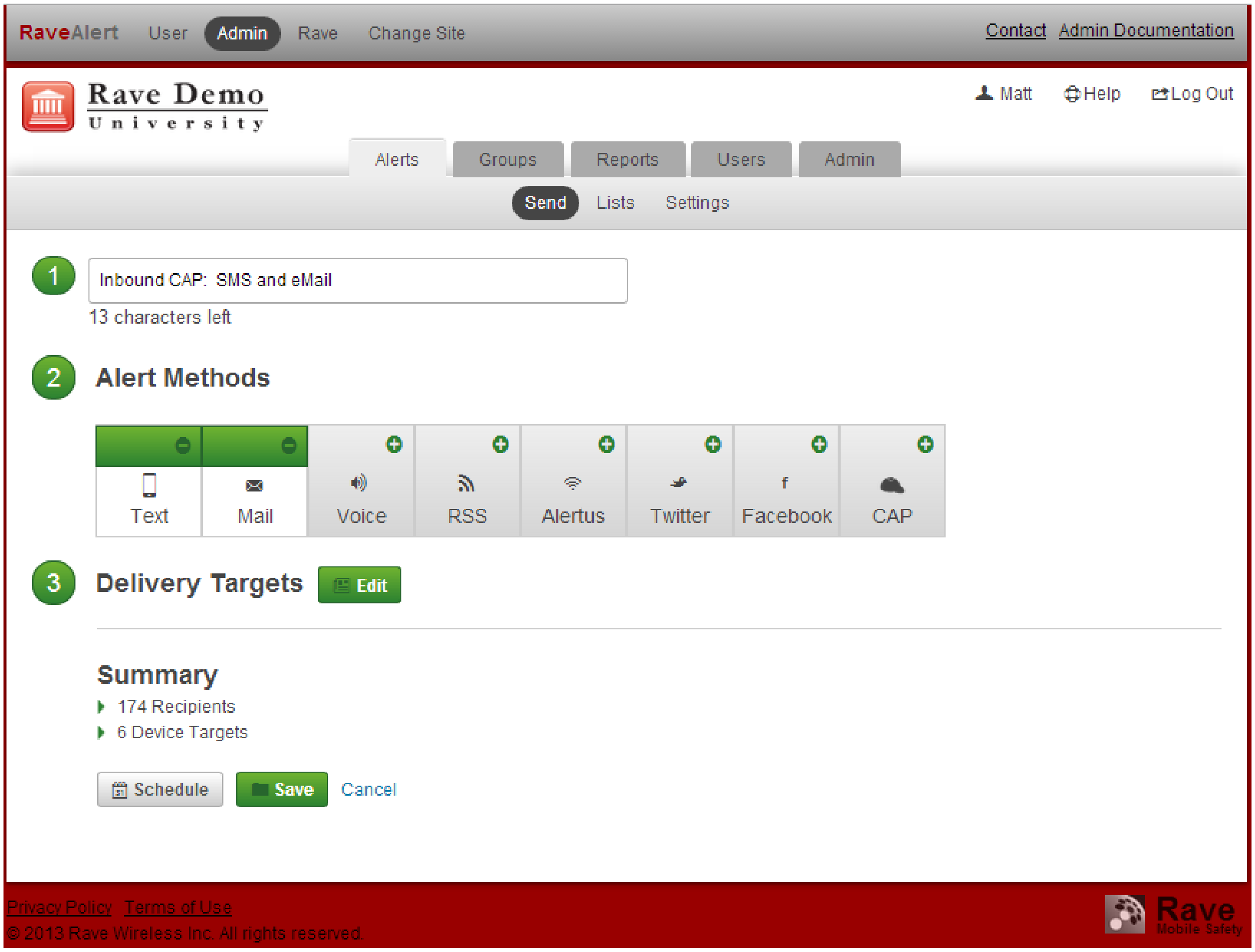
2. Select and Configure the *Alert Methods*.

    a. Click on the tab representing each desired *Alert Method*, until either a green (complete) or yellow (requires further configuration) bar appears across the top of the *Alert Method* tab.

    b. Some *Alert Methods* require further configuration. Populate configurations as you require. These configurations are identical to those specified for a manually triggered Broadcast Alert.

3. Select your *Delivery Targets. Delivery Targets* define the individuals, contact lists, and devices to be targeted by Broadcast Alerts created through this *CAP Notification Template*. Further details can be found in the Rave Alert's *Admin Documentation.*

4. *Save* your *CAP Notification Template* once all three steps meet your requirements, and have moved to the complete (green) state.



1 Inbound CAP: SMS and eMail

13 characters left

2 Alert Methods

3 Delivery Targets

Summary

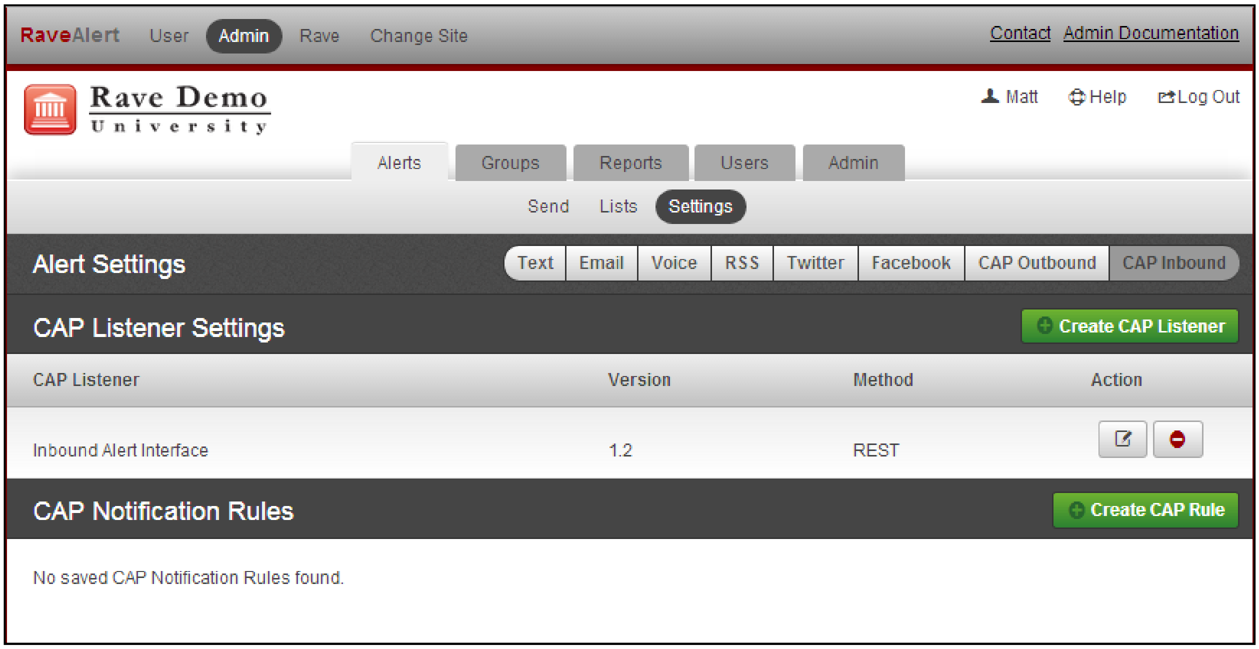
- 174 Recipients
- 6 Device Targets

Figure 4: Completed CAP Notification Template

3. Create a CAP Notification Rule

The last step in defining your Inbound CAP interface is to configure at least one set of *CAP Notification Rules*. You must create one *CAP Notification Rule* each time you have a new set of requirements for when to trigger a given *CAP Notification Template* and how to map CAP message content to the selected alert modes.

To create a *CAP Notification Rule*, Navigate to the *Alerts* → *Send* tab. Select the *Settings* menu and click the *Create CAP Rule* button.



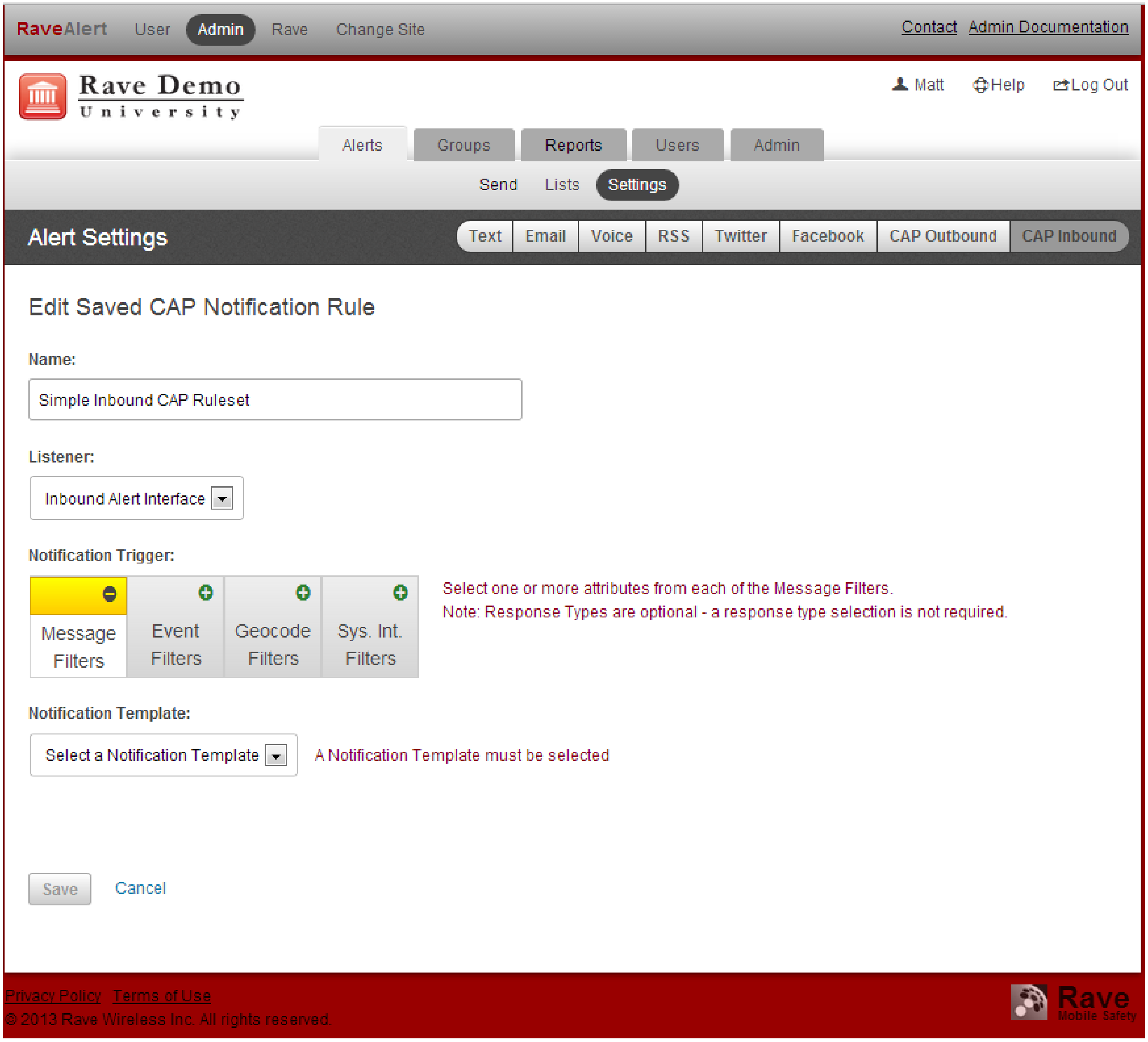
CAP Listener	Version	Method	Action
Inbound Alert Interface	1.2	REST	

No saved CAP Notification Rules found.

Figure 5: Creating a CAP Notification Rule

There are 6 steps to create a *CAP Notification Rule*:

1. *Name* your rule
2. Select a the previously configured CAP *Listener* from the drop-down



Alert Settings

Edit Saved CAP Notification Rule

Name: Simple Inbound CAP Ruleset

Listener: Inbound Alert Interface

Notification Trigger:

Select one or more attributes from each of the Message Filters.
Note: Response Types are optional - a response type selection is not required.

Notification Template: Select a Notification Template

A Notification Template must be selected

© 2013 Rave Wireless Inc. All rights reserved.

Figure 6: Populating a CAP Notification Rule

3. Configure the Notification Triggers. Notification triggers allow you to specify the criteria an XML document must meet before triggering the selected Notification Template. Notification Triggers are organized into four categories:

   Hint: The rules defined here can be complex. We recommended that you configure the minimum filters required to prevent an unnecessary alert.

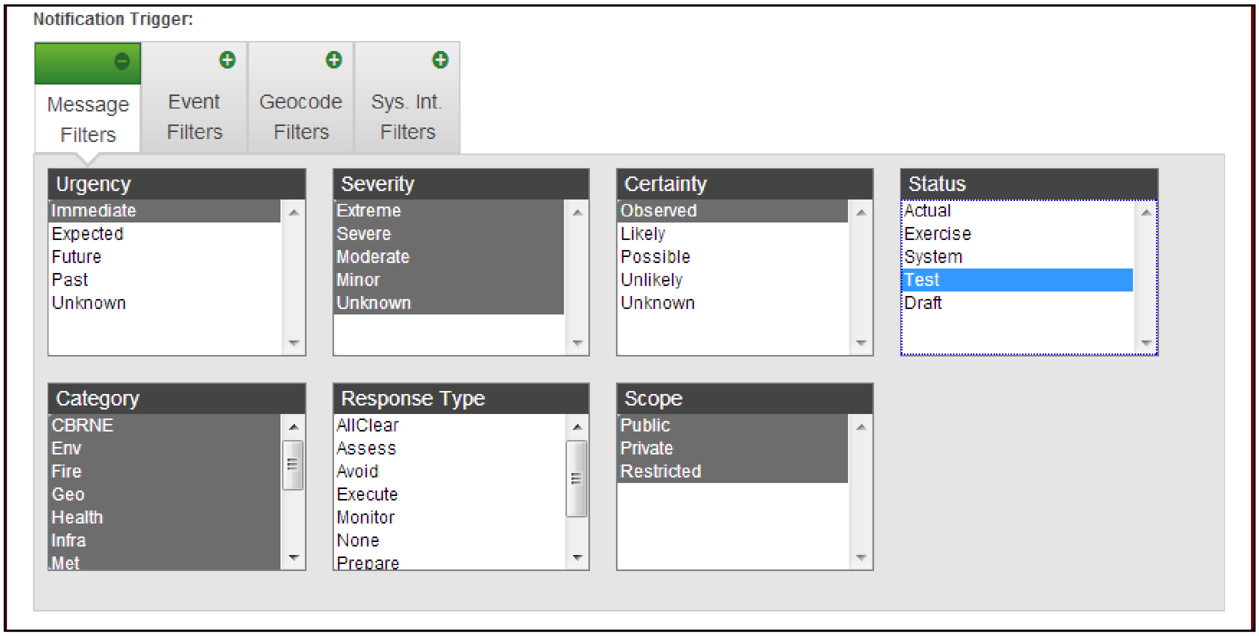
   a. *Message Filters:* All of the *Message Filters*, with the exception of *Response Type*, are required.

      ○ To select multiple values within a given list, hold down the Ctrl key when clicking on the desired value.

      ○ Values selected within a given category (such as *Severity*) are evaluated using a logical 'OR'. In the example, an inbound CAP can have any *Severity* value.

      ○ The configuration across each of the Message Filters is evaluated using the logical AND operator. For example, a given CAP message would need to meet one of the selected *Urgency* AND *Severity* AND *Certainty* AND *Status* AND *Category* AND *Scope* value be processed.

      ○ **HINT:** Unless you only want to trigger an alert when specific *Response Type* values are populated, you should leave this filter blank.



Notification Trigger:

Urgency: Immediate, Expected, Future, Past, Unknown

Severity: Extreme, Severe, Moderate, Minor, Unknown

Certainty: Observed, Likely, Possible, Unlikely, Unknown

Status: Actual, Exercise, System, Test, Draft

Category: CBRNE, Env, Fire, Geo, Health, Infra, Met

Response Type: AllClear, Assess, Avoid, Execute, Monitor, None, Prepare

Scope: Public, Private, Restricted

Figure 7: Configuring Message Filters

b. *Event Filters* allow you to filter on specific, pre-defined "Event" values. These valid values are dictated by the National Weather Service. The configuration of *Event Filters* is optional. Leaving *Event Filters* blank will cause a CAP message to be processed regardless of *Event* field content.

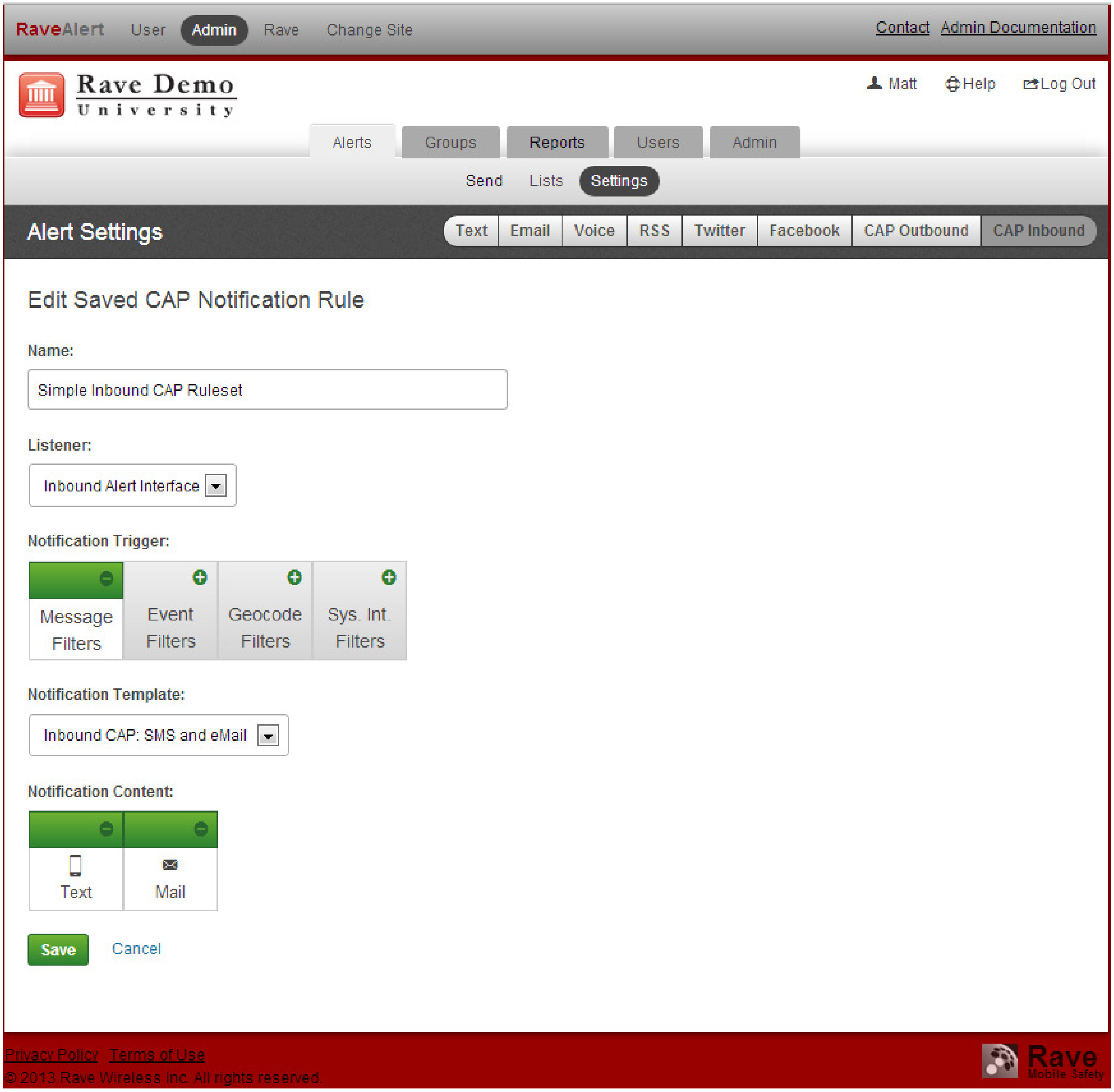
c. *Geocode Filters* filter inbound messages containing specific a tag-value pair(s) in the CAP area.geocode XML structure. Where not specified, Rave will process a CAP message containing *any* Geocode value, including no Geocode value.

d. *System Int(egration) Filters* are the most flexible aspect of the configuration, as they allow you to inspect the CAP `alert.code`, `alert.info.parameter`, and `alert.info.eventCode` fields to determine when your established *CAP Notification Rule* should trigger a given *CAP Notification Template.*

   - As with the configuration of *Message Filters*, selections of values within a *System Integration Filter* is evaluated with the logical OR operator, and selections across fields are evaluated with the logical AND operator.

   - Specifying *System Integration Filters* is optional. If no *System Integration Filters* are specified, CAP messages will be processed regardless of the content within these fields.

4. Select the *Notification Template* you wish to trigger when a CAP XML document meeting the above *Notification Triggers* is received via the selected *Inbound CAP Listener.*



Edit Saved CAP Notification Rule

Name: Simple Inbound CAP Ruleset

Listener: Inbound Alert Interface

Notification Trigger:

Notification Template: Inbound CAP: SMS and eMail

Notification Content:

Figure 8: A Simple Notification Rule, Ready to be Saved

5. Specify the Notification Content to be populated within the Broadcast Alert messages triggered by this Notification Rule. The notification content is specified for each alert mode enabled within the selected Notification Template. In general, the rules for populating a given Broadcast Alert message are specified in one of four ways. Not all alert modes support all mapping types (see Table 2: CAP to Alert Mode Message Mapping).

   Note: Not all Rave Alert clients have access to all *Alert Modes*.

   a. *Introductory Text* allows you to specify a fixed string which will be pre-pended to other message content mapping from the CAP message. This field is optional.
   b. Some Alert Modes, such as *Email* and *RSS*, require you to specify the *Subject*, which can be sourced from either the *Notification Template Name* or CAP `Event`
   c. You can enable checkboxes to identify which CAP XML fields should be mapped to the alert message. Mapped content is written to the message in the order displayed here.
   d. When you enable a checkbox, you are provided with an option for specifying a *Label.* This allows you to provide some context for the information presented to the alert recipient. For Example, when including the CAP `instruction` field in an *eMail* message, you may want to first write out a *Label* such as "`Be sure to:` " before writing out the content of the CAP `instruction` field.

NOTE: If you have access to the Voice notification method, the voice message will be created by Rave Alert's text-to-speech engine, using the content of the selected CAP XML fields as input.

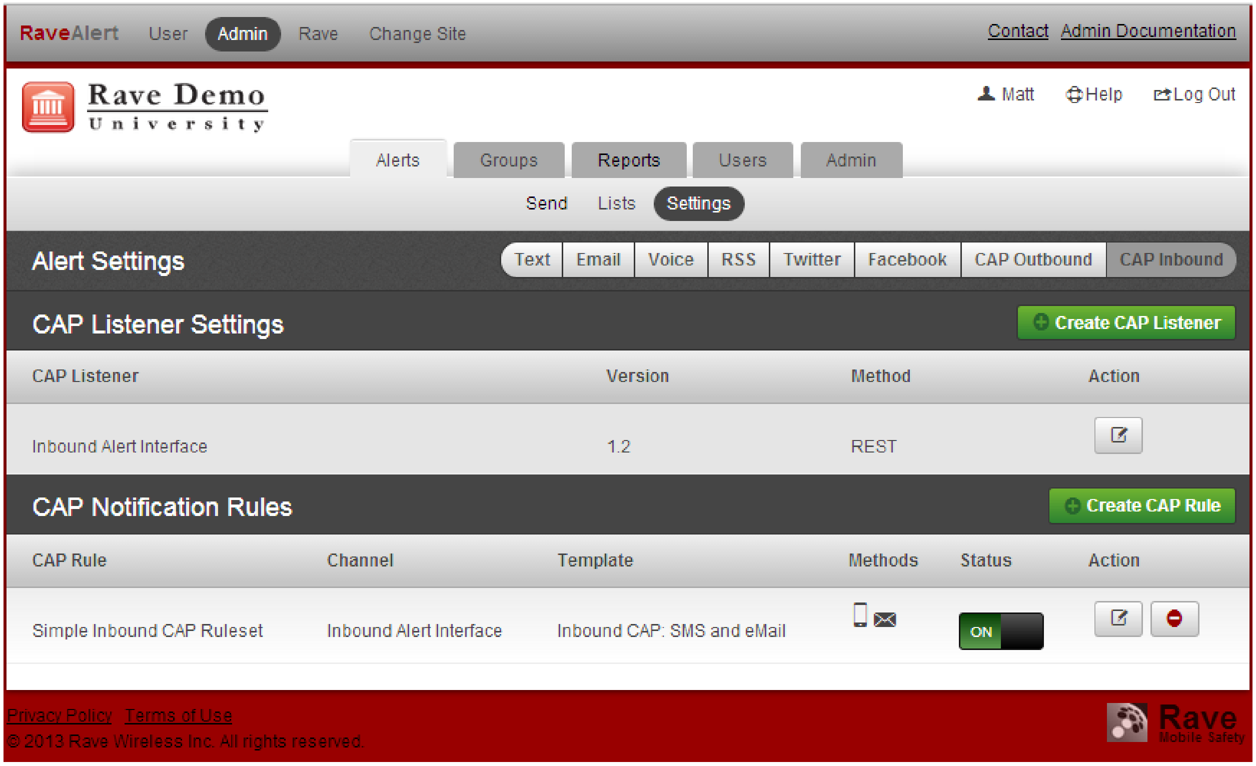
Alert Mode	CAP XML Document Fields						
	Incidents	Event	Headline	Description	Instruction	Web	Contact
SMS		✓	✓				
eMail	✓		✓	✓	✓	✓	✓
Voice	✓	✓	✓	✓	✓		✓
RSS	✓	✓	✓	✓	✓	✓	✓
Twitter		✓	✓				
Facebook	✓	✓	✓	✓	✓	✓	✓
Alertus			✓	✓			
CAP	✓	✓	✓	✓	✓	✓	✓

Table 2: CAP to Alert Mode Message Mapping

6. *Save* your completed *Notification Rule.* You will not be able to *Save* a rule set until all required fields are populated, and the configurations for all enabled notification methods are completed / green.

Maintaining Inbound CAP Configurations

Upon saving your *CAP Notification Rule*, you will be returned to the *Alerts* → *Settings* → *CAP Inbound* tab, where your newly created rule will be listed within *CAP Notification Rules*:



CAP Listener Settings

CAP Listener	Version	Method	Action
Inbound Alert Interface	1.2	REST	

CAP Notification Rules

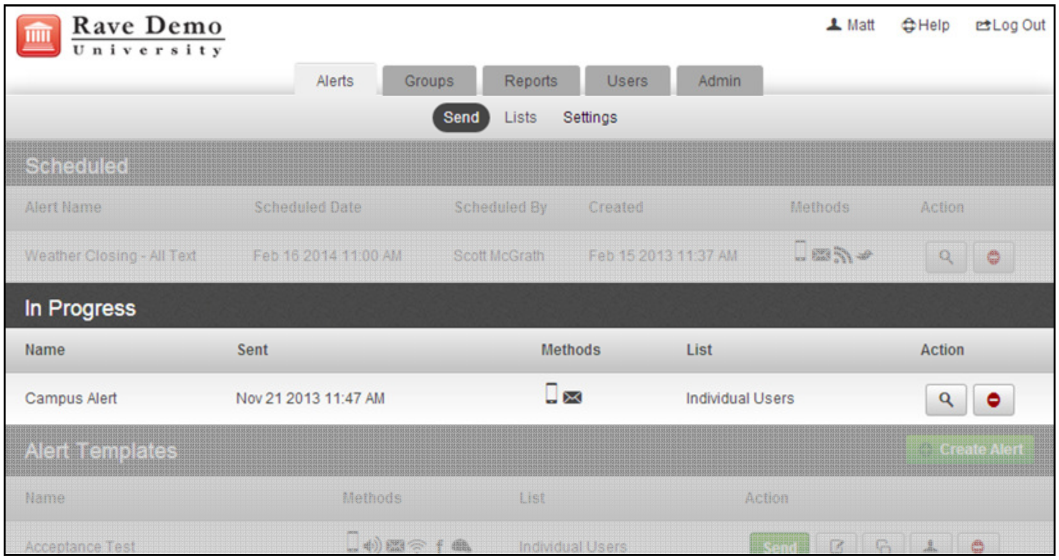
CAP Rule	Channel	Template	Methods	Status	Action
Simple Inbound CAP Ruleset	Inbound Alert Interface	Inbound CAP: SMS and eMail	📱 ✉	ON	

Once saved, the *CAP Notification Rule* is immediately activated. If the designated *CAP Listener* receives a CAP message meeting the criteria specified in the rule set, the identified *CAP Notification Template* will be triggered, the message content will be populated per the *CAP Notification Rules*, and the resulting messages will be sent to the distribution list(s) specified within the *CAP Notification Template.*

To suspend the generation of a Broadcast Alert via a given *CAP Notification Rule*, toggle the *Status* for the *CAP Notification Rule* to *OFF.*

In-Progress CAP-Triggered Broadcast Alerts

In-progress Broadcast Alerts triggered via *Inbound CAP* are listed in the *Admin Console* between any *Scheduled* alerts, and the list of *Alert Templates*.



Scheduled

Alert Name	Scheduled Date	Scheduled By	Created	Methods	Action
Weather Closing - All Text	Feb 16 2014 11:00 AM	Scott McGrath	Feb 15 2013 11:37 AM		

In Progress

Name	Sent	Methods	List	Action
Campus Alert	Nov 21 2013 11:47 AM		Individual Users	

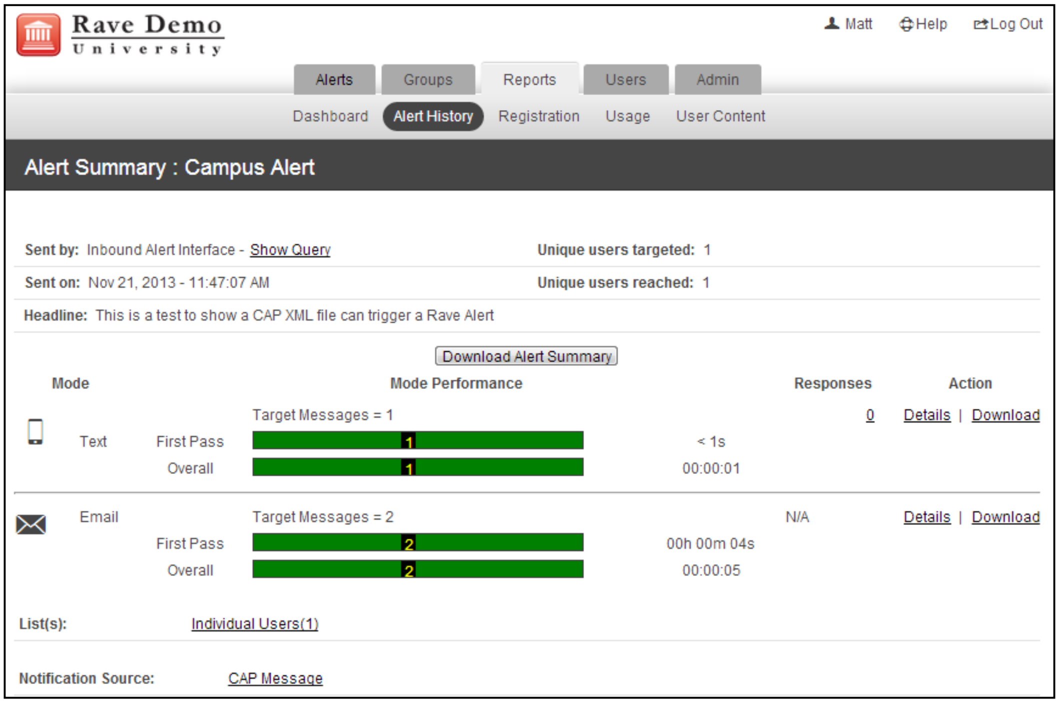
Alert Templates

Name	Methods	List	Action
Acceptance Test		Individual Users	

Figure 9: An In Progress Alert triggered by an Inbound CAP document

Reporting on CAP-Triggered Broadcast Alerts

The outcome of a CAP-Triggered Broadcast Alertsis documented within the *Reports* → *Alert History* tab. In addition to showing typical Broadcast Alert metrics, you can view the CAP XML document via the *Notification Source,* and the *CAP Notification Rules* via the *Show Query* link.



Alert Summary : Campus Alert

Sent by: Inbound Alert Interface - Show Query

Unique users targeted: 1

Sent on: Nov 21, 2013 - 11:47:07 AM

Unique users reached: 1

Headline: This is a test to show a CAP XML file can trigger a Rave Alert

Mode		Mode Performance		Responses	Action
Text		Target Messages = 1		0	Details | Download
	First Pass	1	< 1s		
	Overall	1	00:00:01		
Email		Target Messages = 2		N/A	Details | Download
	First Pass	2	00h 00m 04s		
	Overall	2	00:00:05		

List(s): Individual Users(1)

Notification Source: CAP Message

Figure 10: Sample Report for CAP-triggered Broadcast Alert

Creating a CAP XML Client

Overview

Since CAP XML is a widely accepted standard, and HTTPS REST Web Services using BASIC authentication are fairly standard, it is likely that you already have a system on-site that can be configured to send Rave Alert a CAP message

In the event that is not the case, we provide the following sample Java code as a starting point for the development of your own *Inbound CAP* client. This code example is not intended provide you with a production-ready client. Instead this example is intended to get you started as quickly as possible. Specifically, this example was designed to:

- Compile w/ JDK
- Limit dependencies on libraries
- Leverage the basic HTTP URL object
- Support simple execution from the command line

Furthermore, you may consider modifying this to:

- Use the HTTP Client Class, which includes more advanced connection management, such as support for multiple connections and better multi-threading
- Include more information in HTTP headers, such as user agent and content length
- Include gzip compression to minimize data sent over-the-wire

Functional Considerations

CAP Message Types

Rave's REST-based *Inbound CAP Listener* supports two CAP `message types` :

- `Alert`: to be used for a new notification event
- `Cancel`: to be used when attempting to terminate an in-progress notification event

Given the nature of multi-modal alerting, `cancel` message types are processed on a "best effort" basis. For example, already processed email, SMS, and voice notifications cannot be retracted; social media posts that have been applied to the target channels will need to be manually deleted.

For similar reasons, Rave Alert does not support the `Update` CAP message type. Should you need to update a previously issued alert, you must issue a `Cancel` message type, followed by a new `Alert` message type containing the updated message content.

HTTP Response Codes

Success

`200 OK` is returned upon receiving a valid CAP document which passes CAP XML schema validation and is persisted by our system.

`200 OK` DOES NOT necessarily mean the XML document triggered a Broadcast Alert. This response code is also returned when a CAP duplicate message id has been received, when the XML document did not trigger a *Notification Rule* or when the target *Status* of the targeted *Notification Rule* was set of *OFF*

Errors

`400 Bad Request` is returned when the submitted CAP XML document does not pass CAP Schema Validation.

`401 Unauthorized` is returned when the HTTP Basic credentials are not the same as those configured for the targeted *CAP Listener.*

You should plan to receive and handle other standard HTTP errors.

Sample Java CAP Listener Client

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;

import sun.misc.BASE64Encoder;

public class SimpleCAPSender
{
    private StringBuilder response;

    public int sendCAP(String raveServerURI, File pathToCAPXML, String user, String password)
    throws IOException
    {
        String cap = readCAPXML(pathToCAPXML);

        // Open a connection to the Rave Alert Service
        HttpURLConnection conn = (HttpURLConnection) new URL(raveServerURI).openConnection();

        // Set connection and read timeouts appropriately
        conn.setConnectTimeout(500);
        conn.setReadTimeout(30000);

        conn.setDoOutput(true);
        conn.setUseCaches(false);

        // Set the content type to xml
        conn.setRequestProperty("Content-Type", "text/xml; charset=\"utf-8\"");
        conn.setRequestProperty("Content-length", "" + cap.length());
        conn.setRequestMethod("POST");

        // Support Basic authentication if the app configured this helper with a user/password.
        if (user != null)
        {
            conn.setRequestProperty("Authorization", "Basic " + new BASE64Encoder().encode((user
            + ":" + password).getBytes()));
        }

        // Send the cap document
        post(conn, cap);

        // Read the response document
        readResponse(conn);

        // Return the response code
        return conn.getResponseCode();
    }

    private String readCAPXML(File pathToCAPXML) throws IOException
    {
        StringBuilder cap = new StringBuilder();
        BufferedReader reader = null;
        try
        {
            reader = new BufferedReader(new FileReader(pathToCAPXML));

            String line;
            while ((line = reader.readLine()) != null)
            {
                cap.append(line).append('\n');
            }
        }
        finally
        {
            if (reader != null)
            {
                reader.close();
            }
        }
        return cap.toString();
    }

    private void post(HttpURLConnection conn, String cap) throws IOException
    {
        DataOutputStream postStream = null;
        try
        {
            postStream = new DataOutputStream(conn.getOutputStream());
            postStream.writeBytes(cap);
        }
        finally
        {
            if (postStream != null)
            {
                postStream.flush();
                postStream.close();
            }
        }
    }

    private void readResponse(HttpURLConnection conn) throws IOException
    {
        BufferedReader responseStream = null;
        try
        {
            responseStream = new BufferedReader(new InputStreamReader(conn.getInputStream()));

            // Parse the response from the server
            response = new StringBuilder();
            String str;
            while ((str = responseStream.readLine()) != null)
            {
                response.append(str);
                response.append("\n");
            }
        }
        finally
        {
            if (responseStream != null)
            {
                responseStream.close();
            }
        }
    }

    public StringBuilder getResponse()
    {
        return response;
    }

    private static String dateTheWayCAPLikesIt(Date d)
    {
        // Format expected by CAP.
        SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ssZ");

        // The one bit this format doesn't have out of the box is the colon in the timezone (-
        0500 vs -05:00)
        // so we add it here...
        StringBuilder sb = new StringBuilder(df.format(d));
        sb.insert(sb.length() - 2, ':');
        return sb.toString();
    }


    public static void main(String[] args) throws Exception
    {
        if (args.length != 4)
        {
            System.out.println("usage: SimpleCAPSender raveServerURI user password
            pathToCAPXML");
            System.exit(-100);
        }

        // Contact Rave for these parameters
        String raveServerURI = args[0];
        String user = args[1];
        String password = args[2];
        String pathToCAPXML = args[3];

        // Send the CAP XML
        SimpleCAPSender sender = new SimpleCAPSender();

        int responseCode = sender.sendCAP(raveServerURI, new File(pathToCAPXML), user, password);

        if (responseCode == 200)
        {
            System.out.println("OK");
        }
        else
        {
            System.out.println("ERROR: received [" + responseCode + "] as response code, body
            follows...");
            System.out.println(sender.getResponse());
        }

    }
}
```

Running the Java CAP Listener Client

Once you have compiled the above sample code, it can be run from the command line as:

```
java SimpleCAPSender <Listener URL> <User> <Password> <Input CAP  XML>
```

Where:

- **<Listener URL>** is the URL cast by Rave for your *CAP Listener*
- **<User>** is the value you selected when creating your *CAP Listener*
- **<Password>** is the value you selected when creating your *CAP Listener*
- **<Input CAP XML>** is the path to a file containing the CAP XML document you wish to submit to the Rave (e.g. /Downloads/SimpleCAPSender.minimal.xml)

Sample CAP XML Files

Provided below are Sample CAP XML files. These files include the subset of OASIS CAP v1.2 fields leveraged by Rave Alert.

The first sample XML document includes all fields recognized by Rave Alert. We recommend you pare this document down to include only those fields required by your implementation.

The second sample XML document represents the minimum required dataset. You will observe there is practically no functional message content in this example. When using a dataset this small, the default "Introductory Text" defined within the CAP Notification Ruleset triggered by this CAP message will define the content of the resulting Broadcast Alert messages.

Sample CAP XML Document: Maximum Dataset

```
<?xml version = "1.0" encoding = "UTF-8"?>
<alert xmlns = "urn:oasis:names:tc:emergency:cap:1.2">
   <identifier>TEST1384898273186</identifier>
   <sender>CapXmlTestClient</sender>
   <sent>2013-11-19T16:57:53-05:00</sent>
   <status>Test</status>
   <msgType>Alert</msgType>
   <scope>Public</scope>
   <code>SampleCapClientv1.0</code>
   <incidents>CapXML-0001</incidents>
      <info>
         <category>Other</category>
         <event>Inbound CAP Message Test</event>
         <responseType>Assess</responseType>
         <urgency>Immediate</urgency>
         <severity>Severe</severity>
         <certainty>Observed</certainty>
         <eventCode>
             <valueName>MyCode</valueName>
             <value>TEST</value>
         </eventCode>
         <headline>This is a test to show a CAP XML file can trigger a Rave Alert</headline>
         <description>This is a test of the Rave Alert inbound CAP Interface.  This is a flexible
         and powerful feature that allows an external system to trigger a multi-modal broadcast
         alert on the Rave Alert platform.</description>
         <instruction>No action is required.  This is just a test.</instruction>
         <web>http://www.ravemobilesafety.com</web>
         <contact>Your helpful IT staff</contact>
         <parameter>
```

```
            <valueName>Param</valueName>
            <value>True</value>
         </parameter>
         <area>
            <areaDesc>Rave Mobile Safety HQ</areaDesc>
            <geocode>
                <valueName>FIPS</valueName>
                <value>25017</value>
            </geocode>
            <geocode>
                <valueName>ZIP</valueName>
                <value>01701</value>
            </geocode>
         </area>
      </info>
</alert>
```

Sample CAP XML Document: Minimum Dataset

```
<?xml version="1.0" encoding="UTF-8"?>
<alert xmlns="urn:oasis:names:tc:emergency:cap:1.2">
   <identifier>TEST1384898273185</identifier>
   <sender>TEST</sender>
   <sent>2013-11-19T16:57:53-05:00</sent>
   <status>Actual</status>
   <msgType>Alert</msgType>
   <scope>Public</scope>
      <info>
         <category>Rescue</category>
         <event>Campus Alert</event>
         <urgency>Immediate</urgency>
         <severity>Moderate</severity>
         <certainty>Observed</certainty>
      </info>
</alert>
```

Sample CAP XML Document: Cancel

```
<?xml version="1.0" encoding="UTF-8"?>
<alert xmlns="urn:oasis:names:tc:emergency:cap:1.2">
   <identifier>TEST1384898273190</identifier>
   <sender>TEST</sender>
   <sent>2013-11-19T16:57:53-05:00</sent>
   <status>Actual</status>
   <msgType>Cancel</msgType>
   <scope>Public</scope>
   <references>TEST,TEST1384898273185,2013-11-19T16:57:53-05:00</references>
      <info>
         <category>Rescue</category>
         <event>Campus Alert</event>
         <urgency>Immediate</urgency>
         <severity>Moderate</severity>
         <certainty>Observed</certainty>
      </info>
</alert>
```

Support

Rave Alert is supported by Rave Mobile Safety’s best-in-class 24x7 support operations.

- General support issues should be emailed to: techsupport@ravemobilesafety.com
- 24x7 support hot line for critical issues: 1-888-605-7163