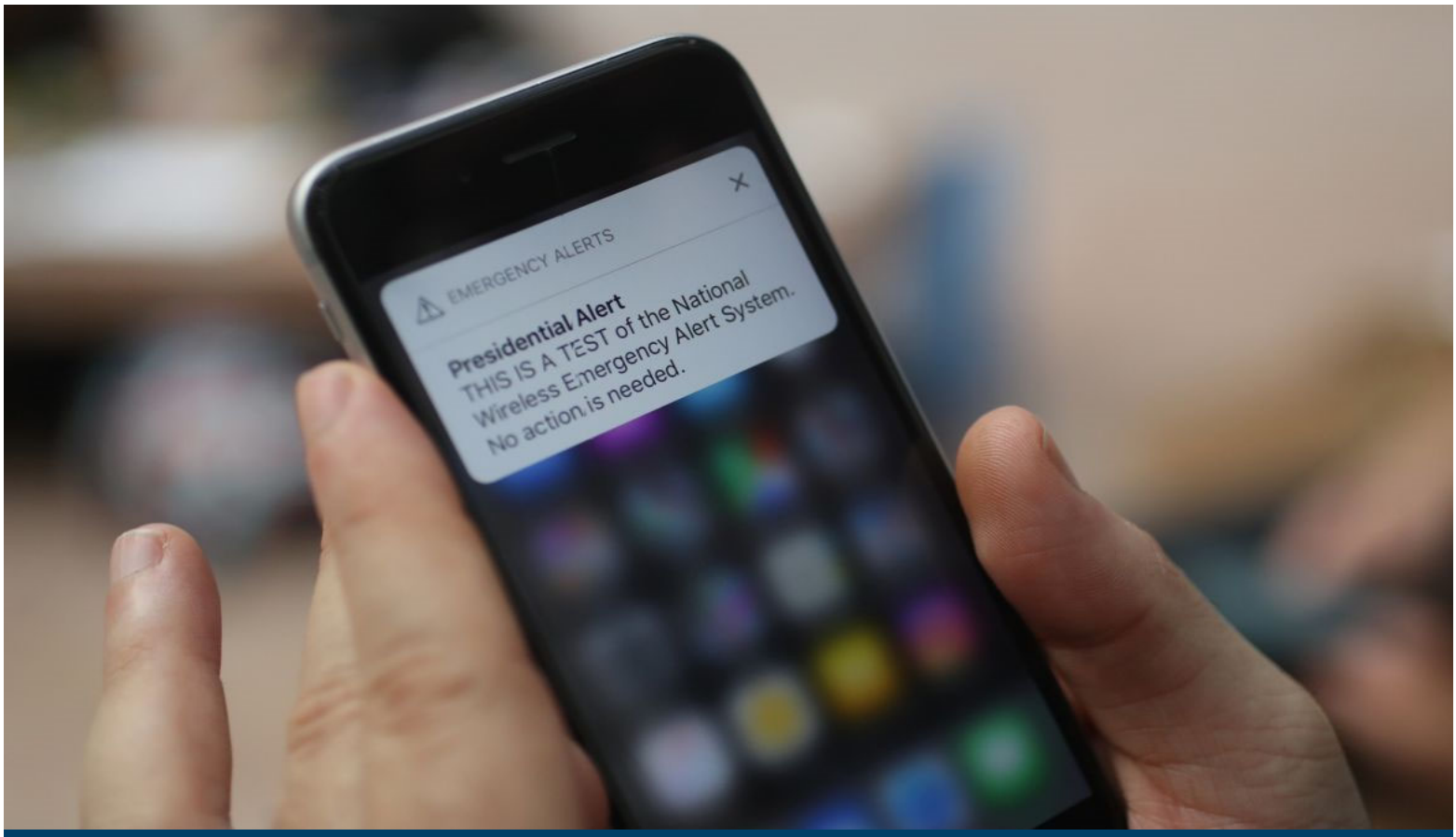


Web-Service Interface Design Guidance (IDG)

Integrated Public Alert and Warning System (IPAWS)

Open Platform for Emergency Networks (OPEN)

IPAWS-OPEN v4.02.06 DRAFT

September 2024

FEMA

TABLE OF CONTENTS

LIST OF TABLES

List of Figures

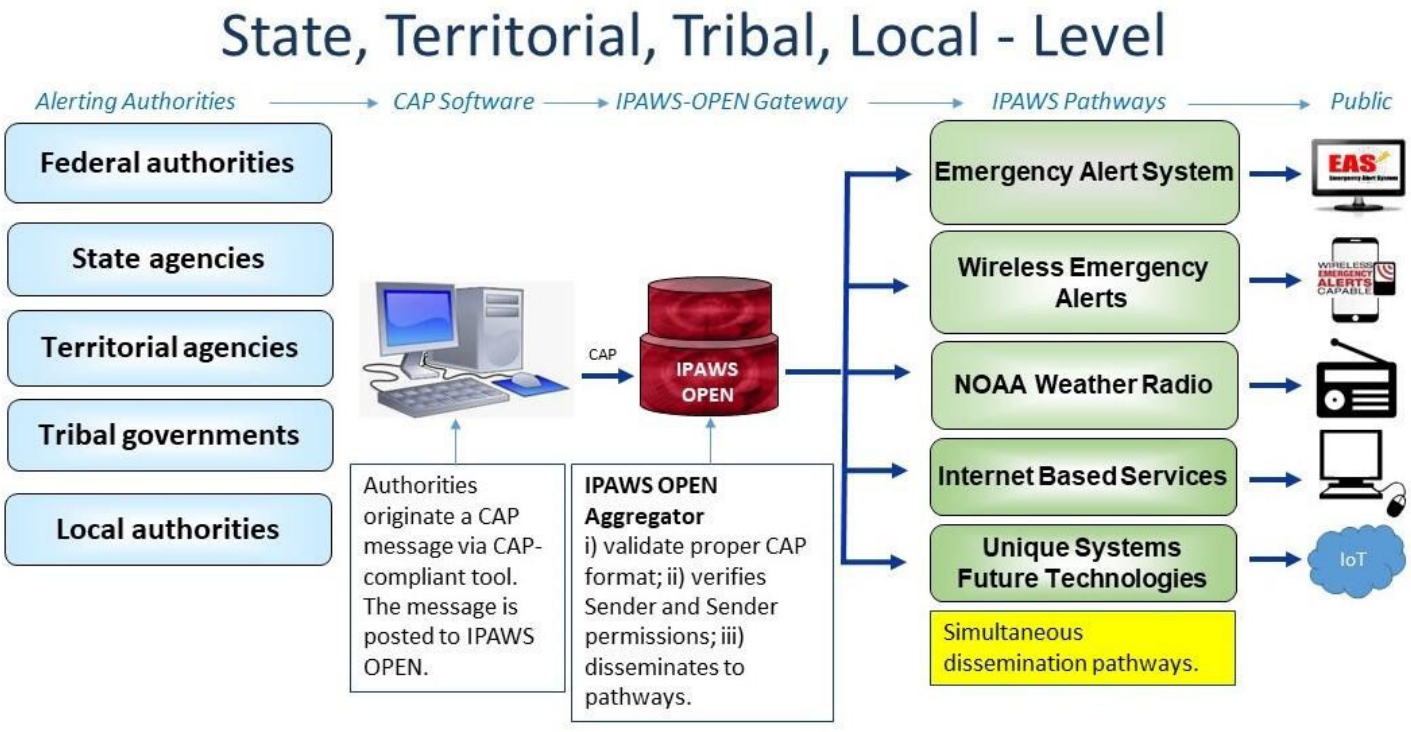
1. IPAWS Introduction

The Integrated Public Alert and Warning System (IPAWS) program was created by the Federal Emergency Management Agency (FEMA) in 2006, per Executive Order 13407. The United States initiated this effective, reliable, integrated, flexible, and comprehensive system to alert and warn the American people in situations of war, terrorist attack, natural disaster, or other hazards to public safety and well-being (public alert and warning system) taking appropriate account of the functions, capabilities, and needs of the private sector and of all levels of government in our Federal system, and ensures that under all conditions the President can communicate with the American public.

In 2007, the Federal Emergency Management Agency (FEMA) began modernizing the nation's public alert and warning system by integrating new technologies into the existing alert systems. The new system became operational in 2011. Known as the Integrated Public Alert and Warning System Open Platform for Emergency Networks (IPAWS-OPEN), it is an IP-based network that disseminates messages through the Common Alerting Protocol (CAP), an international standard used by IPAWS to send public alerts and warnings.

IPAWS is a network of systems for Federal, State, Tribal, Territorial, and Local (FSTTL) Alerting Authorities (AA) to send geographically targeted alerts. IPAWS-OPEN receives and authenticates messages from AA's and routes them to IPAWS-compliant public alerting systems such as the Emergency Alert System (EAS), NOAA Weather Radio (NWR) All Hazards, and Wireless Emergency Alerts (WEA) as well as internet-based and unique alerting systems.

Figure 1: Integrated Public Alert and Warning System Overview



IPAWS
Celebrating 10 Years of Wireless Emergency Alerts

In 2022, FEMA celebrated its 10th anniversary of Wireless Emergency Alerts (WEA). In the past 10 years, FEMA carried more than 70,000 messages from public safety authorities alerting people of threats posed by nearby extreme weather events, law enforcement incidents, missing persons, and many other hazards to the American public via cell phones.

This system ensures that under all conditions, the President of the United States can alert and warn the public. It consists of seventy-seven specially designated and highly resilient commercial and public radio broadcast stations who cooperatively participate with FEMA to provide emergency alert and warning information to the public. FEMA equips stations with backup communications equipment and power generators that enable them to continue broadcasting information to the public during and after an emergency event.

1.1 Purpose

The Interface Design Guide (IDG) is intended to help third-party programmers and system designers to understand the use of IPAWS-OPEN web service interfaces. This is primarily a technical "how-to" document, which is updated when new releases are deployed. The instructions provided in this design guide are not programming language specific. Follow industry best practices, based on the Web Service Description Language (WSDL), to build a successful web service client.

In 2021, IPAWS-OPEN v4.0 migrated to the Amazon Web-Services (AWS) GovCloud. The database migrated from Oracle to AWS RDS PostgreSQL. This migration did not change how Alert Originators (AO), Carriers, and other third parties interact with IPAWS. The release introduced new functionality based on requests from multiple stakeholders, alerting standards organizations, and the IPAWS Program Management Office (PMO).

1.2 Scope

This document introduces interface details to support cross-system information exchange using the following messaging interfaces:

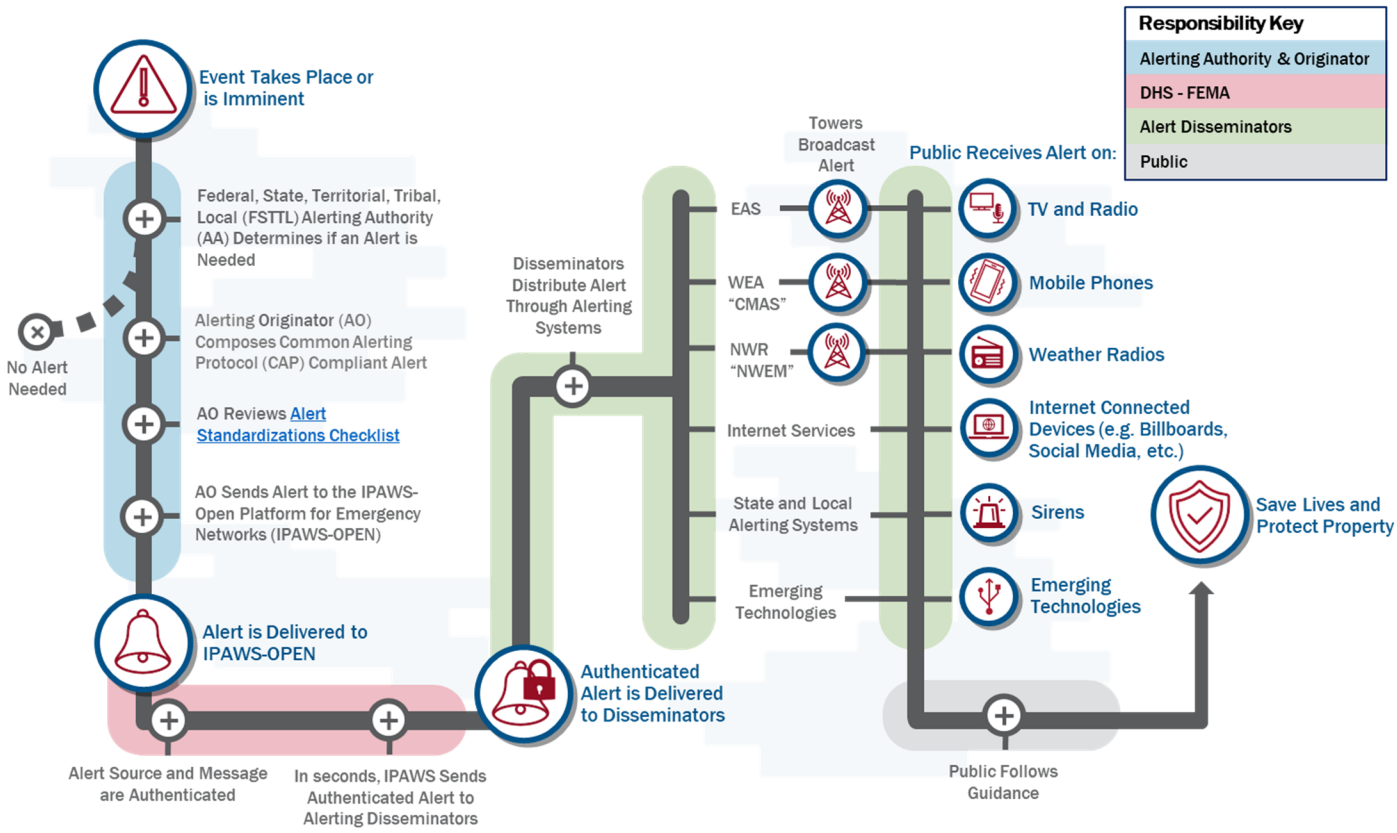
- Common Alerting Protocol (CAP) Version 1.2 Alert Aggregator
- NOAA Weather Radio (NWR) All Hazards, also referred to as the Non-Weather Emergency Message (NWEM) channel in IPAWS-OPEN
- Emergency Alert System (EAS)
- Commercial Mobile Alerting System (CMAS) for Wireless Emergency Alerts (WEA)
- PUBLIC Alerts
- Interface for CAP, NWR, EAS, WEA, and PUBLIC Alert Retrieval

Delivery to NWR also includes distribution of the alert in human consumable formats over some National Weather Service (NWS) dissemination systems such as NOAAPORT.

2. IPAWS Communication Channels

IPAWS-OPEN routes messages to IPAWS communications pathways, otherwise known as dissemination channels. Software and hardware developers create IPAWS-OPEN-compatible alert origination and dissemination tools that allow alerts to travel quickly to the Public through multiple pathways including EAS, WEA, NWR, internet services, State and Local alerting systems, and emerging technologies.

Figure 2: IPAWS Communication Channels



2.1 Emergency Alert System

The Emergency Alert System (EAS) is a national public warning system that requires radio and TV broadcasters, cable TV, wireless cable systems, satellite, and wireline operators to provide the President with capability to address the American people within 10 minutes during a national emergency.

Broadcast, cable, and satellite operators are the stewards of this important public service in close partnership with state, local, tribal, and territorial authorities.

FEMA, in partnership with the Federal Communications Commission and National Oceanic and Atmospheric Administration (NOAA), is responsible for implementing, maintaining, and operating the EAS at the federal level.

2.2 NOAA Weather Radio All Hazards

NOAA Weather Radio (NWR) All Hazards, also referred to as the Non-Weather Emergency Message (NWEM) channel in IPAWS, is a nationwide network of radio stations broadcasting continuous weather information directly from the nearest National Weather Service (NWS) office. NWR broadcasts official NWS alerts, warnings, watches, forecasts, and other hazard information 24 hours a day, 7 days a week.

Working with the Federal Communication Commission (FCC) Emergency Alert System, NWR is an "All Hazards" radio network, making it your single source for comprehensive weather and emergency information. In conjunction with Federal, State, and Local Emergency Managers and other public officials, NWR also broadcasts warning and post-event information for all types of hazards – including natural (such as earthquakes or avalanches), environmental (such as chemical releases or oil spills), and public safety (such as AMBER alerts or 911 Telephone outages). NWR includes more than 1000 transmitters, covering all 50 states, adjacent coastal waters, Puerto Rico, the U.S. Virgin Islands, and the U.S. Pacific Territories. NWR requires a special radio receiver or scanner capable of picking up the signal.

2.3 Wireless Emergency Alert

A Wireless Emergency Alert (WEA) is a short emergency messages from authorized federal, state, local, tribal, and territorial public alerting authorities that can be broadcast from cell towers to any WEA-enabled mobile device in a locally targeted area. Wireless providers primarily use cell broadcast technology for WEA message delivery. WEA is a partnership among FEMA, the Federal Communications Commission (FCC) and Commercial Mobile Service Providers (CMSP) to enhance public safety.

WEAs can be sent to mobile devices of those in harm's way, without the need to download an app or subscribe to a service. WEAs are messages that warn the public of an impending natural or human-made disaster. The messages are short and can provide immediate, life-saving information. Types of Wireless Emergency Alerts include:

- **Presidential "National" Alerts** are a special class of alerts only sent during a national emergency by the President of the United States or FEMA Administrator.
- **Imminent Threat Alerts** include natural or human-made disasters, extreme weather, active shooters, and threats & emergencies that are current or emerging.
- **Public Safety Alerts** contain information about a threat that may not be imminent or after an imminent threat has occurred. Public safety alerts are less severe than imminent threat alerts.
- **America's Missing: Broadcast Emergency Response** (**AMBER) Alerts** are urgent bulletins issued in child-abduction cases. Rapid and effective public alerts often play a crucial role in returning a missing child safely. An AMBER Alert instantly enables the entire community to assist in the search for and safe recovery of the child.
- **Test Messages (Opt-in)** assess the capability of state and local officials to send WEA. The message will state "this is a TEST."

2.4 CAP "PUBLIC" Alert-Feed

EAS previously only supported EAS alerting, but now includes all message types "CAP Alert-Feed" or PUBLIC dissemination channel. This channel supports all alert message types (CAPEXCH, NWEM (NWR), EAS, WEA, Public) that meet the IPAWS-Profile requirements and dissemination specific requirements.

3. Common Alerting Protocol (CAP)

The Common Alerting Protocol (CAP), a digital format for exchanging emergency alerts, allows a consistent alert message to be disseminated simultaneously over multiple communications pathways. FEMA formally adopted CAP and worked with the Organization for the Advancement of Structured Information Standards (OASIS) to develop the IPAWS Profile.

CAP provides a standard around which our nation's alert and warning capabilities are integrated. The following guidance and technical documents currently define CAP as it is implemented and used in IPAWS:

- OASIS Common Alerting Protocol Version 1.2, July 1, 2010 http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.pdf
- OASIS Common Alerting Protocol Version 1.2 USA Integrated Public Alert and Warning System Profile Version 1.0, Committee Specification 01, October 13, 2009 http://docs.oasis-open.org/emergency/cap/v1.2/ipaws-profile/v1.0/cd03/cap-v1.2-ipaws-profile-cd03.pdf
- EAS-CAP Industry Group (ECIG) Recommendations for CAP EAS Implementation Guide – Version 1.0, May 17, 2010 http://www.eas-cap.org/ecig-cap-to-eas_implementation_guide-v1-0.pdf

Every month the IPAWS Program Office distributes a "tip" to emergency managers and software vendors. The tips cover a wide range of topics, including best practices, recommendations, and current issues. After the tips are sent, they are posted for the public. For more information visit https://www.fema.gov/emergency-managers/practitioners/integrated-public-alert-warning-system/public-safety-officials/tips

3.1 Benefits of CAP

CAP is a simple but general format for exchanging all-hazard emergency alerts and public warnings over all kinds of networks. CAP allows a consistent warning message to be disseminated simultaneously over many different warning systems, thus increasing warning effectiveness while simplifying the warning task. CAP facilitates the detection of emerging patterns in local warnings of various kinds. CAP also provides a template for effective warning messages based on best practices identified in academic research and real-world experience. http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html

Through the integration of CAP in IPAWS-OPEN, a single emergency alert can trigger a variety of public warning systems, increasing the likelihood that people receive the alert by one or more communication pathways.

3.2 IPAWS Profile v1.0

In addition to the basic CAP standard, a supplemental IPAWS Profile technical specification was developed to ensure compatibility with existing warning systems used in the United States. The IPAWS Profile v1.0 of the XML-based Common Alerting Protocol (CAP) describes an interpretation of the OASIS CAP v1.2 standard necessary to meet the needs of the Integrated Public Alert and Warning System (IPAWS), the public alerting "system of systems." The IPAWS Profile's open standard facilitates integration by multiple industries and ensures interoperability among alert and warning systems across many different alert dissemination pathways.

Please visit http://www.fema.gov/IPAWS-OPEN for additional links and information about IPAWS-OPEN Web-Services.

3.3 IPAWS-OPEN Integration

Access to IPAWS is free. To send a message using IPAWS, however, an organization must procure its own IPAWS-compatible software. Alert Origination Software Providers (AOSP) furnish the software interface that alerting authorities use to generate Common Alerting Protocol (CAP) compliant messages with a Collaborative Operating Group (COG) that are sent to IPAWS-OPEN. The software then allows AAs to deliver messages through IPAWS-OPEN to communication channels of their choosing.

Private sector vendors and developers are encouraged to design IPAWS-compliant alert origination software and IPAWS-compatible products that distribute alerts to the public. AOSPs are required to test their product in the IPAWS-OPEN development environment to ensure they meet critical capabilities recommended by FEMA. A few industries that have chosen to develop IPAWS-compatible tools and technologies include transportation, utilities, healthcare, service providers, and web developers.

IPAWS does not certify or endorse any vendor product. A list of private-sector providers who have successfully demonstrated their IPAWS capabilities can be found here:
https://www.fema.gov/sites/default/files/documents/fema_alert-origination-software-providers-ipaws_102022.pdf

To receive approval to access the test system, in order to develop interfaces to IPAWS-OPEN web-services, the following steps are required:

- Register for an IPAWS Portal User Account
- Complete a Developer Application
- Review and sign the Memorandum of Agreement (MOA)
- Review and sign the Rules of Behavior (ROB)

4. Accessing IPAWS-OPEN

To begin the process, register for an account on the IPAWS User Portal at https://www.fema.gov/emergency-managers/practitioners/integrated-public-alert-warning-system/public-safety-officials/sign-up/registration

If you are approved for access, you can submit the developer application and all required documents via the portal. Once fully executed by the AOSP and FEMA, you will be provided with fully executed MOA and ROB documents and access information to include:

1. Digital Certificate
2. COG Profile Permissions
3. Web Service Endpoints

4.1 Digital Certificate

All IPAWS-OPEN operations require the use of a valid, unexpired digital certificate. A digital certificate will be sent upon registration and approval. Certificates are assigned by environment type using the following structure:

- Staging = 12xxxxx
- Demo/Test = 3xxxxxx
- Production = 2xxxxxxx

4.2 COG Profile Permissions

Access to IPAWS-OPEN is built on Collaborative Operations Groups (COGs). A COG is a virtual organization that holds membership in IPAWS-OPEN and manages system access within that membership. All IPAWS COGs are sponsored by a vetted Emergency Management Organization. Channel specific permissions will be created in IPAWS-OPEN and provided upon approval. Permissions are used for the validation of event codes and geocodes as implemented in the Aggregator Service.

- A channel-specific error code is returned if the COG is not authorized for the event code or geocode for a dissemination channel.
- Channel-specific event code and geocode authorization are checked first in the COG Profile.
- If that check is successful, the message is disseminated to that channel.
- If that check fails, the message is rejected only for that Dissemination Channel and a response is returned with the appropriate channel specific error code.

Table 1: IPAWS-OPEN Dissemination Channels

Allowed values for dissemination channels:	Response codes related to channel specific authorizations.
• **CAPEXCH** – CAP Exchange Dissemination Channel • **NWEM** – NWS NWR Dissemination Channel • **EAS** - EAS Dissemination Channel • **CMAS** - CMAS Dissemination Channel • **PUBLIC** - Non-EAS PUBLIC Dissemination Channel	• **215** - signer-not-authorized-for-event-code-CAPEXCH • **219** - signer-not-authorized-for-geocode-CAPEXCH • **415** - signer-not-authorized-for-event-code-NWEM • **419** - signer-not-authorized-for-geocode-NWEM • **515** - signer-not-authorized-for-event-code-EAS • **519** - signer-not-authorized-for-geocode-EAS • **615** - signer-not-authorized-for-event-code-CMAS • **619** - signer-not-authorized-for-geocode-CMAS • **815** - signer-not-authorized-for-event-code-Non-EAS-PUBLIC • **819** - signer-not-authorized-for-geocode-Non-EAS-PUBLIC

4.3 Web Service Endpoints

Web service endpoints differ for each IPAWS-OPEN environment. Endpoints referenced in this document and used in the IPAWS-OPEN Cloud Development & Test Environment (CDTE) or staging environment for AOSP (vendors) include:

Table 2: IPAWS-OPEN Staging Endpoints

Environment	Web Service URL
A Interface	https://tdl.integration.aws.fema.gov/IPAWS_CAPService/IPAWS
EAS Feed	https://tdl.apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/eas/recent/2023-08-21T11:40:43Z
EAS ATOM Feed	https://tdl.apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/feed
NWR/NWEM Feed	https://tdl.apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/nwem/recent/2023-08-21T11:40:43Z
WEA Feed	https://tdl.apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/PublicWEA/recent/2023-08-21T11:40:43Z
PUBLIC Feed	https://tdl.apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/public/recent/2023-08-21T11:40:43Z
PUBLIC NON_EAS	https://tdl.apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/public_non_eas/recent/2023-08-17T12:00:00Z

4.4 NWR (NWEM) Permissions

Operational NWR capability no longer requires additional COG level permission from the National Weather Service (NWS). COG's that wish to distribute some or all of their NWR (NWEM) alerts via NWS systems must establish this in the Memorandum of Agreement with FEMA IPAWS and should contact their local NWS Warning Coordination Meteorologist (https://www.weather.gov/stormready/contact) to coordinate the types of messages the NWS may expect to receive. For more information on the NOAA Weather Radio broadcast capability please visit https://www.weather.gov/NonWeatherAlerts/.

5. IPAWS-OPEN Architecture

The IPAWS-OPEN Cloud Environment (CE) is an enterprise messaging system that allows integration between different architectural standards. Each of these services include a commonly defined Request schema for requesting messages, message lists, and other data. These services also include a Response schema for returning both message and non-message data (e.g., message lists and value lists).

5.1 IPAWS-OPEN Services

IPAWS-OPEN uses Get and Post web-services on an Apache Tomcat framework, utilizing document/literal style binding. A Web Service Description Language (WSDL) binding style describes how a service is bound to the Simple Object Access Protocol (SOAP) messaging protocol. In general, a WSDL SOAP binding can either be a Remote Procedure Call (RPC) style binding or a document style binding. A SOAP binding can also have an encoded or a literal use.

> IPAWS-OPEN SOAP Interfaces are exclusively document literal.

The style dictates how to translate a WSDL binding to a SOAP message. There is no formal mapping for document/literal style between the SOAP messages and programming languages. The SOAP message should conform to a particular XML Schema. The various Web- Service development kits provide tools that will generate code that

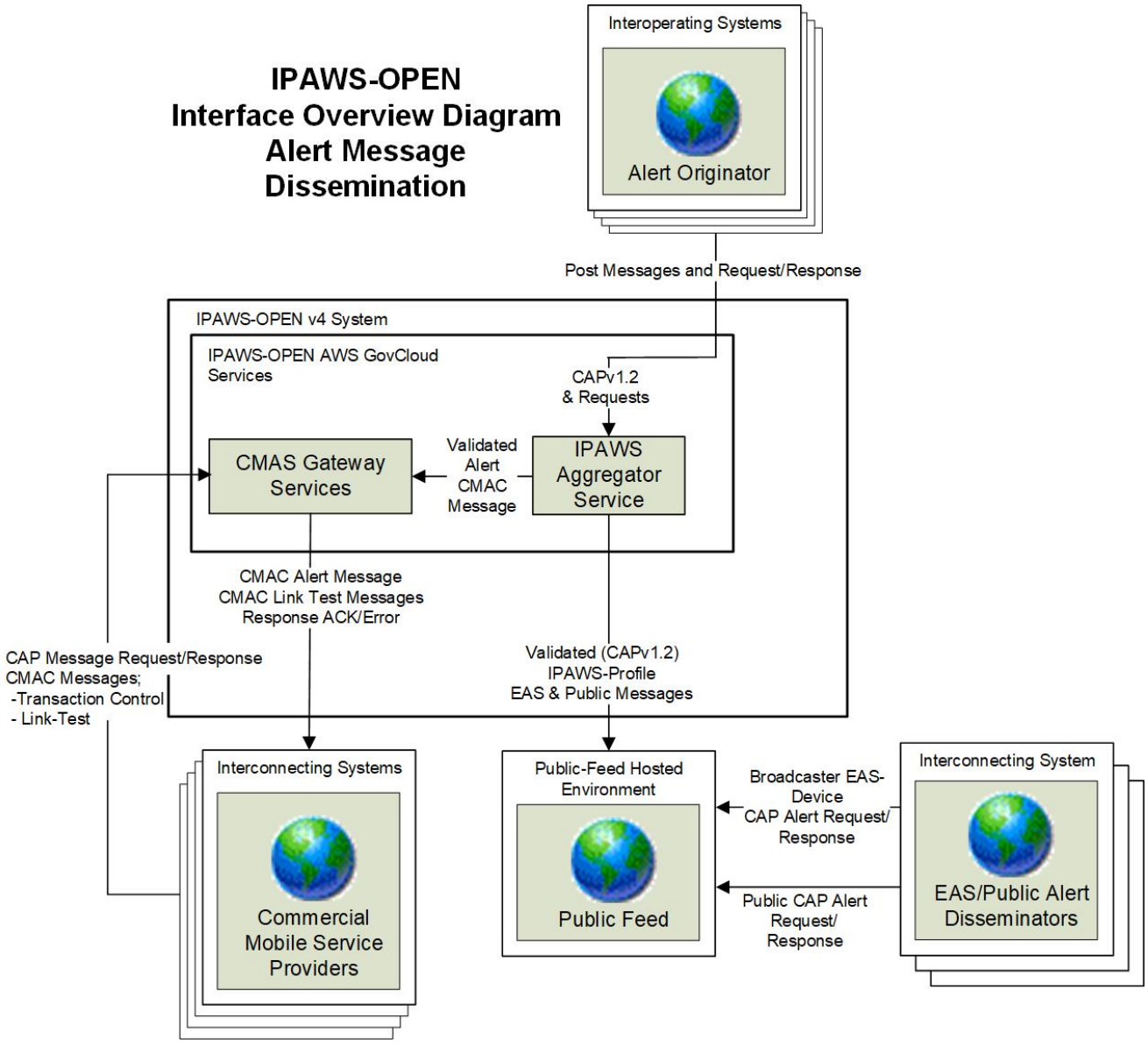
maps from the SOAP message to the programming language being used based on the XML Schema definition.

The messaging service CAP Alert Aggregator employs message schemas in accordance with applicable OASIS standards. Each of these services also include a commonly defined Request schema for requesting messages, message lists, and other data. These services also include a Response schema for returning non-message data (e.g., message lists and value lists).

5.2 Interface Design Diagram

The following provides a high-level introduction to the IPAWS-OPEN services. It illustrates the posting of Alert Messages from Alert Originators utilizing Interoperating Systems and dissemination through the IPAWS-OPEN services.

Figure 3: IPAWS Interface Design Diagram



The external Interoperating Systems use standards-based web-services to establish interfaces to IPAWS-OPEN. These systems can include other Commercial Off The Shelf (COTS) and Government Off The Shelf (GOTS) incident management systems and alert and message distribution systems. IPAWS-OPEN Cloud platform is an enterprise messaging system that allows integration between different architectures without needing to write code, using

interface adapters and data transformation services. The Cloud infrastructure allows various applications to exchange data with one another as they participate in business processes. The objective is to deliver business processes to end-users and other systems. IPAWS-OPEN primarily uses the following web-services:

- IPAWS Alert Aggregator Service (CAP Version 1.2) service provides interfacing systems with the ability to post Alerts and Warnings to IPAWS Dissemination Channels including NOAA Weather Radio (NWR), Emergency Alert System (EAS), Commercial Mobile Alerting System (CMAS) for Wireless Emergency Alerts (WEA), and PUBLIC Alerts. It also provides the capability to post CAP v1.2 messages for retrieval by other interoperating systems in the same manner as the other IPAWS-OPEN messaging services.
- All IPAWS-OPEN Web-Service interfaces support alert message exchange between COGs interfacing with external Interoperating Systems.
- The CAP Alert Aggregator Service provides the capability for an Alert Originator to post a single CAP v1.2 Alert Message, meeting the IPAWS Profile requirements, and then disseminate that message over multiple channels.
- Transformation from CAP v1.2, meeting IPAWS Profile and specific CMAS requirements, to Commercial Mobile Alert Message for Interface-C (CMAC v2.0), and dissemination to Commercial Mobile Service Provider (CMSP) Gateways.
- Dissemination of CAP v1.2, meeting IPAWS Profile to CAP Alert-Feed Platform:
    - If the alert meets specific NWEM requirements it will be flagged as a NWR Alert for retrieval.
    - If the alert meets specific EAS requirements it will be flagged as an EAS Alert for retrieval.
    - If the alert meets specific CMAS requirements it will be flagged as a WEA Alert for retrieval.
    - If the alert meets general IPAWS Profile requirements it will be flagged as a PUBLIC Alert for retrieval.

5.3 SOAP Messages and WS-Security

IPAWS-OPEN Web-Services are secured using the WS-Security v1.0 specification to include:

- **Authentication**: Each Interoperating System Client must:
    - Include Binary Security Token element (i.e., X509 v3 certificate encoded in base 64) per XML digital signature specification (http://www.w3.org/TR/XMLdsig-core/).
    - Digitally sign the SOAP Message and include XML digital signature elements in the SOAP Header.
- **Message Integrity and Non-Repudiation** for Posting to IPAWS-OPEN: This is achieved by digitally signing the SOAP Message using XML digital signature specification.
- **Message Confidentiality**: Interoperating System clients invoke IPAWS-OPEN Web- Services using Transport Layer Security (TLS) connection.

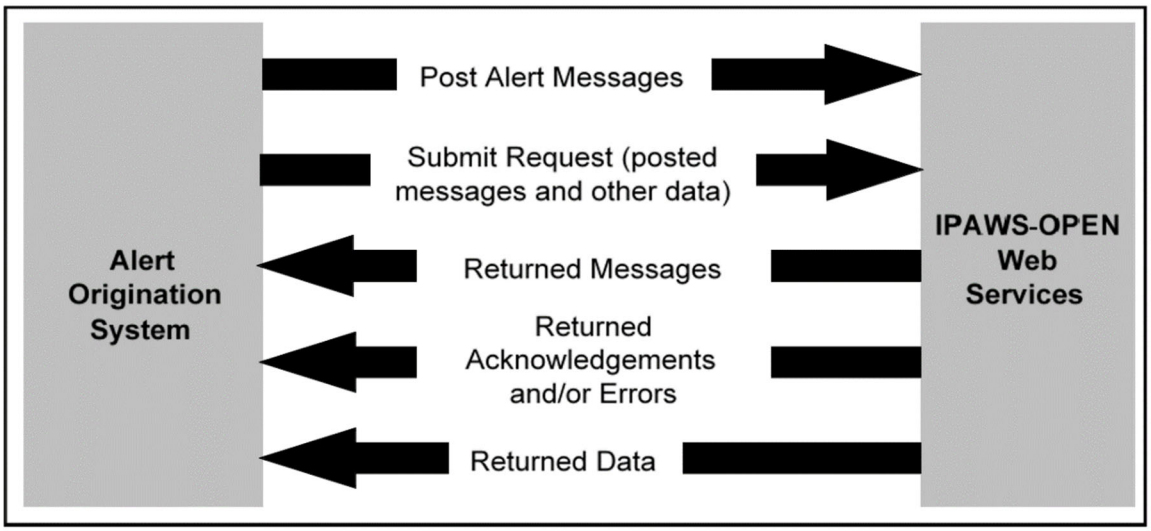
The response returned from IPAWS-OPEN will not be digitally signed.

5.4 IPAWS-OPEN Message Summary

The core set of message types exchanged in IPAWS-OPEN between an interoperating alert origination system and IPAWS-OPEN is shown in the figure below. The CAP v1.2 Aggregator service provides capabilities to publish alerts across multiple dissemination channels, and the capability to receive a consolidated message across all channels.

- Submit request or receive posted messages based on specified parameters.
- Submit request for other data.
    - Aggregator: Message lists, COG list, Value lists, system acknowledgement and server time.
    - Aggregator Request: Consolidated message processing status.

Figure 4: IPAWS-OPEN Message Processing Summary



- Posted/returned Messages based on message schema
- Submitted request based on Request Schema
- Returned Data responses based on Response Schema
- Returned Acknowledgments and errors based on defined Response Schema for aggregator (CAPv1.2) Service

The messaging services incorporate a common pattern for web-service operations and includes Get and Post services as described below.

5.5 Get Service Operations

There are a number of Get Service Requests available in the CAP Aggregator messaging Web Service. The following are IPAWS-OPEN Get operations. Refer to the Get Service section for more details.

1. **Get Message** operation to retrieve messages from IPAWS-OPEN based on parameters included in the request message. The actual operation name is **getMessage**.
2. **Get Common**- Service Request operation to retrieve non-message information from IPAWS-OPEN such as message lists and common-service requests. The actual operation name is **getRequest**. These common service requests include:

    - **Get Acknowledgement** to check connectivity to IPAWS-OPEN and authentication. The actual operation name is **getAck**.
    - **Get Server Info** to check server time to verify time synchronization and IPAWS-OPEN version. The actual operation name is **getServerInfo**.
    - **Get COG List** for use by interoperating system to enable end-users to select distribution for posted messages. The actual operation name is **getCOG**.
    - **Get Message Status** for use by an interoperating system to retrieve from the IPAWS Aggregator service. This provides a consolidated message status including response from all dissemination channels. The actual operation name is **getMessageStatus**.
    - **Get Message Lists** for use by an interoperating system to retrieve message lists based on parameters included in request message. The actual operation name is **getMessageList.**
    - **Get Message Time** is used to retrieve timestamps for when a message is received by IPAWS-OPEN along with the timestamps for all status codes recorded for a message. The actual operation name is **getMessageTime**.

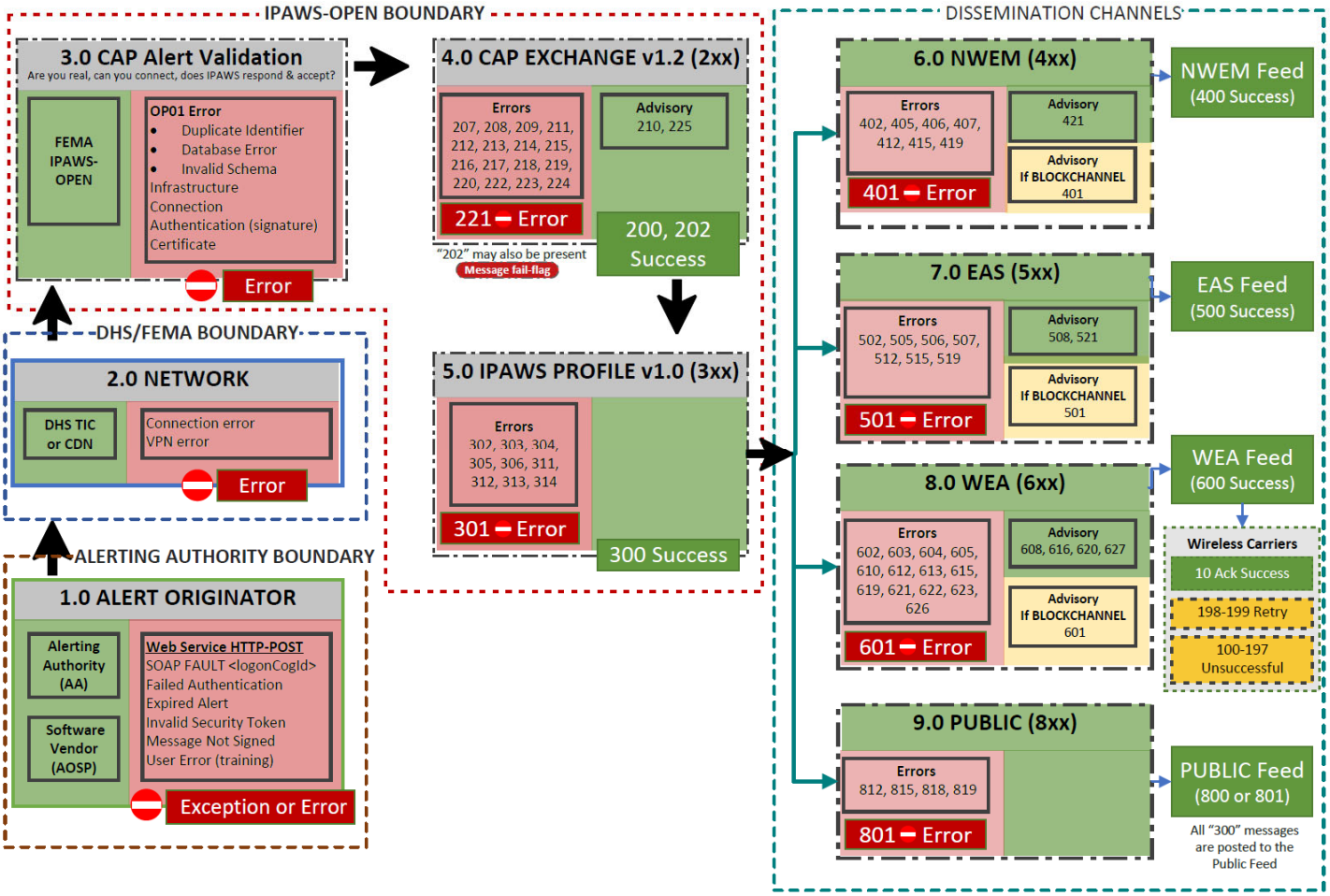
5.6 Post Service Operations

Post operation is to submit a NWEM/EAS/WEA alert to IPAWS-OPEN. A single alert must be posted in a single transaction in order to provide origination-to-dissemination (end-to-end) tracking, response, and proper processing for the posted alert. The operation for submitting messages to IPAWS-OPEN can be shared with other collaborative operating groups (COG) and one or more dissemination channels. This operation name is **postCAP**. Refer to Post Service section for more details.

5.7 IPAWS-OPEN Validation Workflow

All communications through IPAWS-OPEN are via HTTP Post transactions. The figure below highlights ways a posted transaction can get to and disseminate to distribution channels successfully with associated status codes. Alternatively, the red highlights potential reasons for failure and associated error codes for an alert not properly processed.

Figure 5: Message Validation Path



6. Get Service Overview

Get operations are used to check connectivity, retrieve a COG Profile or COG message and metadata from IPAWS-OPEN based on parameters included in the request message.

6.1 Connection Pre-requisites

The follow elements are required to successfully execute Get operations in IPAWS-OPEN.

- Digital Certificate (unexpired)
- Authorized (enabled) COG with permissions
- Stable internet connection
- Web-service endpoint
- Minimum required parameters and digital signature

6.2 Request Message Data Object Model

The Request and Response formats provide an openly defined digital message format for all types of information requests to the IPAWS-OPEN platform. The Web-Services API's employing these message formats provide an interfacing system with the ability to request posted messages and resource data. Requested resource data is used by the interfacing system to author standard messages that are posted to IPAWS-OPEN. The values for the following SOAP header elements should be provided for each request.

Table 3: SOAP Header Elements

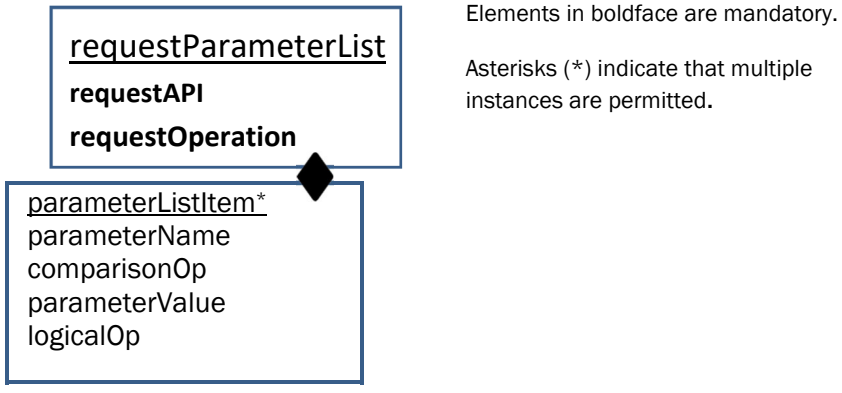
SOAP Header Element	Description	Example
logonUser	Logged on User Name	<soapenv:Header> <ipaws:CAPHeaderTypeDef> <ipaws:logonUser>IPAWSTest</ipaws:logonUser> <ipaws:logonCogId>XXX123</ipaws:logonCogId> </ipaws:CAPHeaderTypeDef> </soapenv:Header>
logonCogId	Logged on COG Identifier	

The Logged-on COG Identifier controls the retrieval of messages/message-lists by identifying the originating COG Identifier associated with message/message-list request.

6.3 Request Message Format

The following figure shows the Document Object Model (DOM) for the Request message format:

Figure 6: Request Message Format DOM



6.4 Structure of the IPAWS-OPEN Request Message

The IPAWS-OPEN Request message consists of a <requestParameterList> segment, which may contain one or more <ParameterListItem> segments as shown in the document object model.

For all services, the <requestParameterList> element maps to <getRequestTypeDef> element (for non-message requests) and <getMessageTypeDef> element (for message requests). From the Request XSD for all services, the <parameterListItem> element maps to <parameters> element.

<requestParameterList> Segment

The <RequestParameterList> segment identifies the specific request parameter list to be processed. This list is identified by the <requestAPI> and <requestOperation> attributes.

<ParameterListItem> Segment

The <ParameterListItem> segment identifies the parameter name and value, along with comparison and logical operations to be processed to execute the request identified by the request parameter list. There can be one or more parameter list item elements in a request parameter list. For specific parameter names there can be zero or one parameter value.

Table 4: requestParameterList Elements

Element Name	Optionality	Notes
requestAPI	Required	Identifies the Web-Service application programming interface (API) that will process the request. Examples include: • CAP12 - for CAPv12 Alert getMessage requests • REQUEST1 - for non-Alert resource data requests
requestOperation		Identifies the Web-Service operation that will process the request.Examples Include: • getMessage (CAP v1.2) • getMessageList (CAP v1.2) • getMessageStatus (CAP v1.2) • getCOGProfile (CAP v1.2) • getAck (CAP v1.2) • getServerInfo (CAP v1.2) • getCOG (CAP v1.2)
The “parameterListItem” Element and Sub-Elements (There can be one or more parameterListItem elements.) ParameterListItem is not required for the following requests: • getCOGProfile • getAck • getServerInfo • getCOG		
parameterName	Required	**parameterName**: The parameter name associated with the requestOperation. **comparisonOp**: Comparison Operation which has a value of “equalto” or “like”. Required for the following requests: • getMessage • getMessageList • getMessageStatus
comparisonOp	Required	
parameterValue	Required	Required for the following requests: • getMessage

Element Name	Optionality	Notes
		• getMessageList • getMessageStatus
logicalOp	Conditional	Logical Operation, which is assumed to be “AND” in future release will implement “OR” capability.

6.5 Request XML Schema

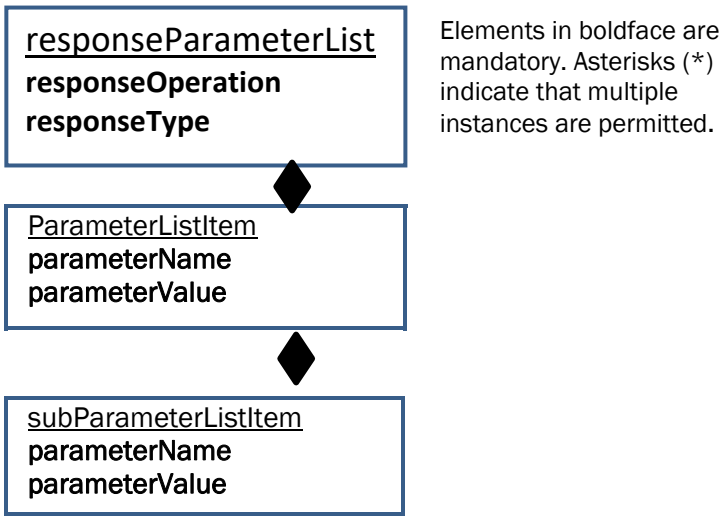
XML data structures are validated in application business rules according to the Request schema. This is done to provide flexibility in handling new information in Request messages.

```
<?XML version="1.0" encoding="UTF-8"?> <xsd:schema elementFormDefault="qualified"
targetNamespace="http://gov.fema.ipaws.services/caprequest"
XMLns:req="http://gov.fema.ipaws.services/caprequest"
XMLns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:complexType name="requestParameterList">
<xsd:sequence>
<xsd:element minOccurs="0" name="requestAPI" type="xsd:string"/>
<xsd:element minOccurs="0" name="requestOperation" type="xsd:string"/>
     <xsd:element maxOccurs="unbounded" minOccurs="0" name="parameters" type="req:parameterListItem"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="parameterListItem">
<xsd:sequence>
<xsd:element minOccurs="0" name="parameterName" type="xsd:string"/>
<xsd:element minOccurs="0" name="comparisonOp" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="parameterValue" type="xsd:string"/>
<xsd:element minOccurs="0" name="logicalOp" type="xsd:string"/>
</xsd:sequence></xsd:complexType> </xsd:schema>
```

6.6 Response Message Data Object Model

The following figure shows the document object model for the Response message format:

Figure 7: Response Message Format DOM



6.7 Structure of the IPAWS-OPEN ResponseMessage

The IPAWS-OPEN Response message consists of a <ResponseParameterList> segment, which may contain one or more <ParameterListItem> segments as shown in the document object model. In turn a <ParameterListItem> segment can contain zero, one or more<SubParameterListItems> segments.

<ResponseParameterList> Segment

The <ResponseParameterList> segment identifies the specific Response parameter list to be processed. This list is identified by the <ResponseType> and <ResponseOperation> attributes.

<ParameterListItem> Segment

The <ParameterListItem> segment identifies the parameter name and value that is included in the response associated with the ParameterListItem element. There can be one or more parameter list item elements in a response parameter list. For specific parameter names there can be zero or one parameter value.

<SubParameterListItem> Segment

The <SubParameterListItem> segment identifies the sub-parameter name and value that is included in the response associated with the <ParameterListItem> element. There can be one or more sub-parameter list item elements in a response parameter list item. For specific parameter names there can be zero or one parameter value.

Table 5: ResponseMessage

Element Name	Optionality	Notes
"**responseParameterList**" Element and Sub-Elements The getMessage Request returns only an Alert Message XML in the response.		
responseOperation	Present	A response message will always have a responseOperation element.

responseType	Optional	Identifies the service associated with the response.
		• No responseType value in the following response messages:
		  ○ getMessageStatus
		  ○ getCOGProfile
		  ○ getAck
		  ○ getServerInfo
		• Value of "REQUEST1" for responseType in the following messages:
		  ○ getMessageList (CAP v1.2)

"parameterListItem" Element and Sub-Elements (There can be one or more parameterListItem elements).		
parameterName	Present	The parameter name associated with the responseOperation
parameterValue	Optional	The parameter value associated with a parameterName and associated requestOperation. **NOTE:** If a request returns no data only a parameterName element will be returned.
"**subParameterListItem**" Element and Sub-Elements (There can be one or more subParameterListItem elements). The following requests will have subParameterListItem elements: • getMessageList • getMessageStatus • getCOGProfile • getCOG		
subParameterName	Present	The sub-parameter name associated with the Sub-Parameter List Item.
subParameterValue	Optional	The sub-parameter value associated with the Sub-Parameter Name.

A getMessageList response returns metadata associated with the posted Alert message XML. There may be a case where specific metadata information was not present in the Alert XML. As a result, only the parameterName may be included in the response message, with no parameterValue.

6.8 Get Response XML Schema

Data structures are validated in application business rules according to the Response schema. This provides flexibility in handling new information in Response messages.

```
<?XML version="1.0" encoding="UTF-8"?> <xsd:schema elementFormDefault="qualified"
targetNamespace="http://gov.fema.ipaws.services/capresponse"
XMLns:resp="http://gov.fema.ipaws.services/capresponse"
XMLns:xsd="http://www.w3.org/2001/XMLSchema"><xsd:complexType name="responseParameterList">
<xsd:sequence> <xsd:element maxOccurs="unbounded" minOccurs="0" ref="resp:parameterListItem"/>
<xsd:element minOccurs="0" name="ResponseOperation" type="xsd:string"/>
<xsd:element minOccurs="0" name="ResponseType" type="xsd:string"/>
</xsd:sequence> </xsd:complexType>
<xsd:element name="parameterListItem">
<xsd:complexType> <xsd:sequence>
<xsd:element minOccurs="0" name="parameterName" type="xsd:string"/>
<xsd:element minOccurs="0" name="parameterValue" type="xsd:string"/>
<xsd:element maxOccurs="unbounded" minOccurs="0" name="subParaListItem"
type="resp:subParameterListItem"/>
</xsd:sequence> </xsd:complexType>
</xsd:element>
<xsd:complexType name="subParameterListItem">
<xsd:sequence>
<xsd:element minOccurs="0" name="subParameterName" type="xsd:string"/>
<xsd:element minOccurs="0" name="subParameterValue" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

7. Get Service Requests

The Logged on User Name and COG Identifier are saved with post and request transaction records as part of the transaction history. The Logged-on COG Identifier also controls the post and retrieval of messages/message-lists by identifying the originating COG Identifier associated with a post or message/message-list request.

7.1 Common Get-Service Requests

There are a series of Common Service Requests available in the CAP Aggregator messaging Web Service. Refer to “Request Message Data Object Model” section for an explanation of the Request Schema. The Request Message Document is comprised of two container elements: ***getRequestTypeDef*** and ***parameters***. Each request message will have one ***getRequestTypeDef*** element. It is possible to have a request with zero, one, or more ***parameters*** elements, dependent on the complexity of the request.

7.2 Get System Acknowledgement (getAck) Request

The GetAck operation is used to check connectivity and authentication with IPAWS-OPEN. The ***getACK*** request only requires a ***getRequestTypeDef*** element. The following table depicts the request structure of ***getACK*** operation.

Table 6: Request getResponseTypeDef

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getACK
parameters	parameterName	Not Required
	comparisonOp	Not Required
	parameterValue	Not Required
	logicalOp	Not Required

Example of getACK

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML sections below.

Webservice Request Details

Below is the request parameters and values required to ping **getAck** operation.

Table 7: Request getAck

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getACK

Webservice Response Details

Below is the expected response for a successful connection.

Table 8: Response getResponseTypeDef

Complex Type Name	Attribute	Value
getResponseTypeDef	parameterName	ACK
	parameterValue	PONG
	ResponseOperation	getACK

Webservice Request & Response XML

Table 9: GetACK Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getAck</cap:requestOperation> > </ipaws:getRequestTypeDef> </soapenv:Body>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/ " XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>ACK</ns4:parameterName> <ns4:parameterValue>PONG</ns4:parameterValue> </ns4:parameterListItem> <ns4:ResponseOperation>getACK</ns4:ResponseOperation> </ns2:getResponseTypeDef> </soap:Body>

7.3 Get Server Info Request

The ***getServerInfo*** operation is used to check IPAWS-OPEN server time to accomplish time synchronization checks and verify current IPAWS-OPEN code version. The ***getServerInfo*** request only requires a ***getRequestTypeDef*** element. Only the <soap:Body> is shown. The following table depicts the request structure of getServerInfo operation.

Table 10: Request getServerinfo

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getServerInfo
parameters	parameterName	Not Required
	ComparisonOp	Not Required
	parameterValue	Not Required
	logicalOp	Not Required

Example of getServerInfo

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML sections below.

Webservice Request Details

Below is the request parameters and values required to ping getServerInfo operation.

Table 11: Values Required getServerinfo

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getServerInfo

Webservice Response Details

Below is the expected response for a successful **getServerInfo** operation

Table 12: Response getServerInfo

Complex Type Name	Attribute	Value
getRequestTypeDef	parameterName	servertime
	parameterValue	2023-06-08T15:51:05-00:00
	parameterName	code
	parameterValue	IPAWSv4.02.00
	ResponseOperation	getServerInfo

Webservice Request & Response XML

Table 13: getServerInfo Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getServerInfo</cap:requestOperation> </ipaws:getRequestTypeDef> </soapenv:Body>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>servertime</ns4:parameterName> <ns4:parameterValue>2023-06-08T15:51:05-00:00</ns4:parameterValue> </ns4:parameterListItem> <ns4:ResponseOperation>getServerInfo</ns4:ResponseOperation> </ns2:getResponseTypeDef> </soap:Body>

7.4 Get COG Request

The ***getCOG*** request is used to retrieve a list of all enabled COGs from IPAWS-OPEN. The ***getCOG*** request requires both a ***getRequestTypeDef*** and ***parameters*** element. Only the <soap:Body> is shown.

The following table depicts the request structure of ***getCOG*** operation.

Table 14: Request getCOG

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getCOG
parameters	parameterName	Not Required
	ComparisonOp	Not Required
	parameterValue	Not Required
	logicalOp	Not Required

Example of getCOG List

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML sections below.

Webservice Request Details

Below is the request parameters and values required to ping **getCOG** operation.

Table 15: Value Required getCOG List

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getCOG
parameters	parameterValue	All

Webservice Response Details

Below table mentions some of the expected response details for a successful getCOG operation.

Table 16: Response getCOGList

Complex Type Name	Attribute	Value
getRequestTypeDef	parameterName	coglist
	parameterValue	all
	ResponseOperation	getCOG

Webservice Request & Response XML

Table 17: getCOG Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getCOG</cap:requestOperation> <cap:parameters> <cap:parameterName/> <cap:comparisonOp/> <cap:parameterValue>ALL</cap:parameterValue> <cap:logicalOp/> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>coglist</ns4:parameterName> <ns4:parameterValue>all</ns4:parameterValue> <ns4:subParaListItem> <ns4:subParameterName>IPAWS UAT TEST</ns4:subParameterName> <ns4:subParameterValue>XXX123</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>IPAWS Test</ns4:subParameterName> <ns4:subParameterValue>XXX234</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>Test Group</ns4:subParameterName> <ns4:ResponseOperation>getCOG</ns4:ResponseOperation>

Request	Response
	<ns4:ResponseType>REQUEST1</ns4:ResponseType> </ns2:getResponseTypeDef> </soap:Body>

7.5 Get COG Profile Request

A **getCOGProfile** request can be used to retrieve all the COG Profile Authorizations for the requesting COG. This transaction allows interoperable systems to retrieve their COG Profile settings, and to determine which operations they are allowed to perform.

The following table depicts the request structure of **getCOGProfile** operation.

Table 18: Request getCOGProfile

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getCOGProfile
parameters	parameterName	Not Required
	comparisonOp	Not Required
	parameterValue	Not Required
	logicalOp	Not Required

Table 19: Response getCOGProfile

Parameter Name	Label	Description
cogid	COG Identifier	The COG Identifier Associated with the COG Profile
name	COG Name	The COG Name associated with the COG Profile
description	COG Description	The COG description associated with the COG Profile
categoryName	COG Category	The COG Category Name (Default Value: IPAWS-OPEN)
organizationName	Organization Name	The COG Organization Name – Values include: CIV - Civil authorities PEP - Primary Entry Point System EAS - Broadcast station or cable system WXR - National Weather Service
cogEnabled	COG Enabled	Enables a COG for the CAP Aggregator service to post messages or submit message or data requests.
email	email	Primary contact email for COG.
eventCodes	Allowed Event Codes	Authorizes the posting of an Alert by a COG to a specific set of Event Codes.
geoCodes	Allowed Geocodes	Authorizes the posting of an Alert by a COG to a specific set of Geocodes.

Example of getCOGProfile

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML sections below.

Webservice Request Details

Below is the request parameters and values required to ping **getCogProfile** operation.

Table 20: Values Required getCogProfile

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getCOGProfile

Webservice Response Details

Below table mentions some of the expected response details for a successful **getCogProfile** operation for the respective COG.

Table 21: Response getGOGProfile

Complex Type Name	Attribute	Value
getRequestTypeDef	parameterName	cogid
	parameterValue	XXX123
	ResponseOperation	getCOGProfile

Webservice Request & Response XML

Table 22: getGOGProfile Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getCogProfile </cap:requestOperation> </ipaws:getRequestTypeDef> </soapenv:Body>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>cogid</ns4:parameterName> <ns4:parameterValue>XXX123</ns4:parameterValue> </ns4:parameterListItem> <ns4:parameterListItem> <ns4:parameterName>name</ns4:parameterName> <ns4:parameterValue>Test IPAWS </ns4:parameterValue> </ns4:parameterListItem> <ns4:parameterListItem> <ns4:parameterName>description</ns4:parameterName> <ns4:parameterValue>Alerting Solutions</ns4:parameterValue> </ns4:parameterListItem> <ns4:parameterListItem> <ns4:parameterName>categoryName </ns4:parameterName> <ns4:parameterValue>IPAWS-OPEN</ns4:parameterValue>

Request	Response
	</ns4:parameterListItem> <ns4:parameterListItem> <ns4:parameterName>organizationName</ns4:parameterName> <ns4:parameterValue>CIV</ns4:parameterValue> </ns4:parameterListItem> <ns4:parameterListItem> <ns4:parameterName>cogEnabled</ns4:parameterName> <ns4:parameterValue>Y</ns4:parameterValue> </ns4:parameterListItem> <ns4:parameterListItem> <ns4:parameterName>broadcastAuthorized</ns4:parameterName> <ns4:parameterValue>N</ns4:parameterValue> </ns4:parameterListItem> <ns4:parameterListItem> <ns4:parameterName>email</ns4:parameterName> <ns4:parameterValue>test@fema.gov</ns4:parameterValue> </ns4:parameterListItem> <ns4:parameterListItem> <ns4:parameterName>eventCodes</ns4:parameterName> <ns4:subParaListItem> <ns4:subParameterName>PUBLIC</ns4:subParameterName> <ns4:subParameterValue>AVW</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>CAPEXCH</ns4:subParameterName> <ns4:subParameterValue>NPT</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> <ns4:parameterListItem> <ns4:parameterName>geoCodes</ns4:parameterName> <ns4:subParaListItem> <ns4:subParameterName>NWEM</ns4:subParameterName> <ns4:subParameterValue>XXX123</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> <ns4:ResponseOperation>getCogProfile</ns4:ResponseOperation> </ns2:getResponseTypeDef> </soap:Body>

Refer to sections below for the Get Message, Get Message-List and Get Message Status for details.

7.6 Get Message, Get Message-List and Get Message Status Requests

The Get Message and Get Message-List requests are used by an interoperating system to retrieve messages or message-lists based on parameters included in request message. There are both a common set (i.e., getting messages or message-lists by date) and Web-Service specific set of request parameters. Refer to the "Request Message Data Object Model" section for an explanation of the Request Schema. The Request Message is comprised of two container elements: ***getRequestTypeDef*** and ***parameters***. Each Get Message List request message will have one ***getRequestTypeDef*** element, and one or more ***parameters*** elements, dependent on the complexity of the request.

The search parameters for Get Message and Message-Lists are based on metadata for posted messages to IPAWS-OPEN, which is comprised of the required CAP Alert Sub-Elements along with other selected Sub-elements. In general, if non-required Sub-Elements are not included in posted messages then they will not be available for Get Message or Message-List requests, except as noted below.

7.6.1 Metadata for CAP Aggregator Service

The following table lists all CAP Alert message tags that:

- Are persisted in metadata.
- May be returned in message list requests.

The use of **<comparisonOp>** with the value of "**like**" or "**equalto**" can be utilized for all metadata queries, except where noted.

Table 23: CAP Alert Message Metadata

Parameter Name for Message Element	Returned Msg/List Request	Remarks
<alert> Element		
identifier	Y	The <identifier> element for the alert message. **(See NOTE-2)**
sender	Y	Text from the <sender> element identifying the sender for the alert message. **(See NOTE-2)**
sent	Y	The date and time from the <sent> element when the alert message was sent. **(See NOTE-2)**
status	Y	The code from the <status> element denoting the appropriate handling of the alert message (See CAP v1.2 standard for code values). **(See NOTE-2)**
msgType	Y	The code from <msgType> element denoting the nature of the alert message (See CAP v1.2 standard for code values). **(See NOTE-2)**
scope	Y	The code from <scope> element denoting the intended distribution of the alert message (See CAP v1.2 standard for code values). **(See NOTE-2)**
addresses	Y	The group listing of intended recipients from the <addresses> element of the alert message. For IPAWS-OPEN each recipient is identified by the COG-ID. Multiple space-delimited COG-ID addresses may be included in the alert. Each Address has its own metadata record. (**NOTE**: Required for CAP Exchange with other COGS. Not Required for CAP dissemination.)
headline	Y	The text headline from the <headline> element of the alert message. If the value for <headlines> element is not included in the alert, the following text is persisted in the metadata record and returned in the message list: "No Headline". **(See NOTE-3)**
code	Y	The value from the <code> element where value is equal to "IPAWSv1.0". If this <code> value is not included in the alert, the following shall be loaded in metadata record: 'IPAWS Profile Not In Use'. (See Note-3) NOTE: Use "profilecode" as the request parameter. Returned as "profilecode" in a Message List.
references	N	The group listing from the <references> element identifying earlier message(s) referenced by the alert message. A single metadata record shall be created for the <references> element. Multiple references shall be persisted in the metadata the same as provided in the alert (space delimited).

Parameter Name for Message Element	Returned Msg/List Request	Remarks
		If no <references> element is provided, then no value shall be persisted in metadata. (See Note-3)
incidents	Y	The group listing from <incidents> element naming the referenced incident(s) of the alert message. If the value for <incidents> element is not included in the alert, no record shall be persisted and no value shall be returned in the message list.
<info> element		
language	N	The first <language> element for the alert message. (See Note-1)
category	N	The code from the <category> element denoting the category associated with the subject event of the alert message (See CAP v1.2 standard for code values). (See Note-2)
event	N	The text from the <event> element denoting the type of the subject event associated with the alert message. (See Note-2)
responseType	N	The code from the <responseType> element denoting the type of action for the target audience for the alert message (See CAP v1.2 standard for code values). (See Note-1)
urgency	N	The code from the <urgency> element denoting the urgency associated with the subject event of the alert message (See CAP v1.2 standard for code values). (See Note-2)
severity	N	The code from the <severity> element denoting the severity associated with the subject event of the alert message (See CAP v1.2 standard for code values). (See Note-2)
certainty	N	The code from the <certainty> element denoting the certainty associated with the subject event of the alert message (See CAP v1.2 standard for code values). (See Note-2)
eventCode	Y	The code from the <eventCode> element identifying the event type of the alert message. (See Note-1) If there is a <info> block <eventCode><valueName> element equal to “SAME”, then the value in the corresponding <eventCode> <value> shall be persisted in the “eventcode” metadata value.
senderName	Y	The text from the <sendername> element naming the originator of the alert message. (See Note-1)
headline	Y	The text headline from the <headline> element of the alert message. If the value for <headlines> element is not included in the alert, the following text shall be persisted in the metadata record and returned in the message list: “No Headline”. (See Note-3)
parameter		A system specific additional parameter associated with the alert message. Persist where <parameter><valueName> equal to EAS-ORG, BLOCKCHANNEL, and CMAMtext.
	Y	EAS-ORG: If there is a <info> block <parameter><valueName> equal to “EAS-ORG”, then the value in the corresponding <parameter><value> shall be persisted in the “EAS-ORG” metadata value. NOTE: Use “EAS-ORG” as the request parameter.
	N	BLOCKCHANNEL: If there is one or more <info> block <parameter><valueName> elements equal to “BLOCKCHANNEL”, then the value in the corresponding <parameter><value> shall be persisted in the “BLOCKCHANNEL”

Parameter Name for Message Element	Returned Msg/List Request	Remarks
		metadata value. Each BLOCKCHANNEL parameter shall have its own metadata Record. NOTE: Use “BLOCKCHANNEL” as the request parameter.
	N	CMAMtext: If there is a <info> block <parameter><valueName> equal to “CMAMtext”, then the value in the corresponding <parameter><value> shall be persisted in the “CMAMtext” metadata value. NOTE: Use “CMAMtext” as the request parameter.
<area> element		
areaDesc	N	The text from the area description <areaDesc> element describing the affected area of the alert message. **(See NOTE-2)**
polygon	N	The paired value of points defining a polygon that delineates the affected area of the alert message. A value is persisted in a single metadata record as a Boolean (Metadata value is “Y” or “N”), indicating the existence or non- existence of polygon data. **(See NOTE-3)**
circle	N	The paired value of a point and radius defining a circle that delineates the affected area of the alert message. A value is persisted in a single metadata record as a Boolean (Metadata value is “Y” or “N”), indicating the existence or non-existence of circle data. **(See NOTE-3)**
geocode	N	The geographic code in the <geocode> element delineating the affected area of alert message. Only <geocode> elements with <valueName> equal to "SAME" are persisted in metadata. **(See NOTE-1)** Each <geocode> element has its own metadata record.
<resource> element		
resourceDesc	N	The text from the <resourceDesc> element describing the type and content of the resource file. **(See NOTE-2)**
mimeType	N	The code in the <mimeType> element provides the identifier of the MIME content type and sub-type describing the resource file.
uri	N	The code in the <uri> element providing the hyperlink for the resource file. If the value for the <uri> element is not included in the alert, the following text is persisted in the metadata record: “No URI’. **(See NOTE-3)**
derefUri	N	The code in the <derefUri> element contains base-64 encoded data content of the resource file. A value is persisted in a single metadata record as a Boolean (Metadata value is “Y” or “N”), indicating the existence or non- existence of derefUri data. **(See NOTE-3)**
Derived Metadata		
cogid	N	Sending COG-ID – The COG-ID associated with <logonCogId> element provided in the SOAP header.
cogname	Y	Sending COG Name - The COG Name associated with <logonCogId> element provided in the SOAP header.
ipawsProfile	Y	Enables the retrieval of CAP v1.2 messages and message lists based on whether the message is IPAWS-Profile compliant or not. (Metadata value is “Y” or “N”). **(See NOTE-3)**

Parameter Name for Message Element	Returned Msg/List Request	Remarks
signaturevalidated	Y	Enables the retrieval of CAP v1.2 messages and message lists based on whether the alert message has a valid signature. (Metadata value is "Y" or "N"). **(See NOTE-3)**

NOTE-1: If the value for the element is not included in the alert, no metadata record is persisted.
NOTE-2: Mandatory Element in Posted CAP Message.
NOTE-3: Metadata with alternative text, based on presence in CAP Alert XML, or validation result.

Programmers Notes - Request Messages and Parameters;

- **<parameterName> Element for getRequest Operations** - To ensure consistent request results, it is important to use the values for the <parameterName> as shown in this design guide. The use of invalid values for <parameterName> will result in no requested data being returned and no specific error message being raised.
- **<logicalOp> Element** - The only currently recognized value for the <logicalOp> element is "and." This tag is a placeholder for future capability. In the current processing <logicalOp> defaults to "and" even when empty.
- **Additional Data for Get Message List Operations** - Additional data is now returned in a message list including COG-Name for all message lists, <headline> element for CAP message lists. Also, the <code> element when equal to "IPAWSv1.0" and <parameter><valueName> element when equal to "EAS-ORG" will be returned in a CAP message list for those messages meeting the CAP IPAWS Profile.

7.7 GetMessage Operations using a Single Parameter

A **getMessage** request can be used to retrieve a CAP v1.2 message via the Aggregator Service. The Aggregator Service provides the same flexibility to submit queries based on various parameters.

The following examples for getMessage (using a single parameter) illustrates the request message for the Aggregator service. This request with only one parameterValue requires both a **getMessageTypeDef** and **parameters** element.

Table 24: Request Messages and Parameters

Complex Type Name	Attribute	Value
getMessageTypeDef	requestAPI	REQUEST1
	requestOperation	getMessage
parameters	parameterName	identifier
	comparisonOp	equalto
	parametervalue	TEST-IPAWS-CAE_2023

Webservice Response Details

Below table mentions some of the expected response details for a successful NOMESSAGEFOUND **getMessage(Identifier)** operation.

Table 25: getMessage Response - No Message Found

Complex Type Name	Attribute	Value
messageResponseTyoeDef	identifier	NOMESSAGEFOUND-200040856068

Webservice Request & Response XML

Table 26: getMessage Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getMessageTypeDef> <cap:requestAPI>Request1</cap:requestAPI> <cap:requestOperation>getMessage</cap:requestOperation> <cap:parameters> <cap:parameterName>identifier</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <cap:parameterValue>TEST-IPAWS-CAE_2023 </cap:parameterValue> </cap:parameters> </ipaws:getMessageTypeDef> </soapenv:Body>	<soap:Body> <ns2:messageResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns3:alert> <ns3:identifier>NOMESSAGEFOUND-200040856068</ns3:identifier> <ns3:sender>test@fema.gov</ns3:sender> <ns3:sent>2023-05-25T14:02:34-00:00</ns3:sent> <ns3:status>System</ns3:status> <ns3:msgType>Alert</ns3:msgType> <ns3:scope>Public</ns3:scope> <ns3:NOTE>NO MESSAGE FOUND</ns3:NOTE> </ns3:alert> </ns2:messageResponseTypeDef> </soap:Body>

Example of getMessage (by identifier) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessage(Identifier)** operation.

Table 27: Request GetMessage(identifier)

Complex Type Name	Attribute	Value
getMessageTypeDef	requestAPI	REQUEST1
	requestOperation	getMessage
parameters	parameterName	identifier
	comparisonOp	equalto

Complex Type Name	Attribute	Value
	parametervalue	TEST-IPAWS_202316153315

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessage(Identifier)** operation.

Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 28: Response getMessage(Identifier)

Complex Type Name	Attribute	Value
messageResponseTypeDef	Identifier	TEST-IPAWS_202316153315

Webservice Request & Response XML

Table 29: getMessage(Identifier) Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getMessageTypeDef> <cap:requestAPI>Request1</cap:requestAPI> <cap:requestOperation>getMessage </cap:requestOperation> <cap:parameters> <cap:parameterName>identifier</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <!--Zero or more repetitions:--> <cap:parameterValue>TEST-IPAWS202316153315</cap:parameterValue> </cap:parameters> </ipaws:getMessageTypeDef> </soapenv:Body>	<soap:Body><ns2:messageResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns3:alert> <ns3:identifier>TEST-IPAWS_202316153315</ns3:identifier> <ns3:sender>test</ns3:sender> <ns3:sent>2023-02-06T15:33:15-05:00</ns3:sent> <ns3:status>Actual</ns3:status> <ns3:msgType>Alert</ns3:msgType> <ns3:scope>Public</ns3:scope> <ns3:code>IPAWSv1.0</ns3:code> <ns3:info> <ns3:language>en-US</ns3:language> <ns3:category>Safety</ns3:category> <ns3:event>Evacuation Immediate</ns3:event> <ns3:responseType>Monitor</ns3:responseType> <ns3:urgency>Immediate</ns3:urgency> <ns3:severity>Extreme</ns3:severity> <ns3:certainty>Observed</ns3:certainty> <ns3:eventCode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>EAN</ns3:value>

Request	Response
	</ns3:eventCode> <ns3:expires>2023-02-06T17:33:15-05:00</ns3:expires> <ns3:senderName>COGID, CogName, Requesting Agency</ns3:senderName> <ns3:headline>WEA2.0 Test Message only Disregard please.</ns3:headline> <ns3:description>THIS is NOT an Actual Message. It is only a test. This is Descriptive text that defines the alert</ns3:description> <ns3:instruction>This is not an Actual message. It is only a Test. This is where the call to action for folks receiving the message should be provided.</ns3:instruction> <ns3:parameter> <ns3:valueName>EAS-ORG</ns3:valueName> <ns3:value>CIV</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>timezone</ns3:valueName> <ns3:value>EST</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>BLOCKCHANNEL</ns3:valueName> <ns3:value>NWEM</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>BLOCKCHANNEL</ns3:valueName> <ns3:value>EAS</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>WEAHandling</ns3:valueName> <ns3:value>Presidential</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>CMAMtext</ns3:valueName> <ns3:value>THIS IS A TEST of the National Wireless Emergency Alert System. No action is needed.</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>CMAMlongtext</ns3:valueName> <ns3:value>this is where the 360 character description in English would go..</ns3:value> </ns3:parameter> <ns3:area> <ns3:areaDesc>FAIRFAX COUNTY IN WA</ns3:areaDesc> <ns3:geocode>

Request	Response
	<ns3:valueName>SAME</ns3:valueName> <ns3:value>XXX000</ns3:value> </ns3:geocode> </ns3:area> </ns3:info> </ns3:alert> </ns2:messageResponseTypeDef> </soap:Body>

Example of getMessage (by Headline) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessage(Headline)** operation.

Table 30: Request getMessage(headline)

Complex Type Name	Attribute	Value
getMessageTypeDef	requestAPI	REQUEST1
	requestOperation	getMessage
parameters	parameterName	headline
	comparisonOp	equalto
	parametervalue	Test Earthquake Warning

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessage(Headline)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 31: Response messageReponseTypeDef

Complex Type Name	Attribute	Value
messageResponseTypeDef	identifier	Test-IPAWS20211220201032

Example of getMessage (by MsgType) Operation

The message type (msgType) value can be Alert or Cancel or Update.

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessage(MsgType)** operation.

Table 32: Request getMessage(MsgType)

Complex Type Name	Attribute	Value
getMessageTypeDef	requestAPI	REQUEST1
	requestOperation	getMessage
parameters	parameterName	headline
	comparisonOp	equalto
	parametervalue	Cancel

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessage(Cancel)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 33: Response getMessage(Cancel)

Complex Type Name	Attribute	Response
messageResponseTypeDef	identifier	Test_IPAWS_202348175102023

Webservice Request & Response XML

Table 34: getMessage(Cancel) Request & Response XML

Request	Response
<soap:Body> <ipaw:getMessageTypeDef> <cap:requestAPI>Request1</cap:requestAPI> <cap:requestOperation>getMessage</cap:requestOperation> <!--Zero or more repetitions:-->	<soap:Body> <ns2:messageResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/"

Request	Response
<cap:parameters> <cap:parameterName>msgType</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <!--Zero or more repetitions:--> <cap:parameterValue>Cancel</cap:parameterValue> </cap:parameters> </ipaw:getMessageTypeDef> </soap:Body>	XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns3:alert> <ns3:identifier>Test_IPAWS_202348175102023</ns3:identifier> <ns3:sender>test@fema.gov</ns3:sender> <ns3:sent>2023-05-10T15:08:12-04:00</ns3:sent> <ns3:status>Actual</ns3:status> <ns3:msgType>Cancel</ns3:msgType> <ns3:scope>Public</ns3:scope> <ns3:code>IPAWSv1.0</ns3:code> <ns3:references>w- test@fema.gov,Test_IPAWS_202348175102023,2023-05-10T14:56:00-04:00</ns3:references> </ns3:alert> </ns2:messageResponseTypeDef> </soap:Body>

This method may attempt to retrieve too many messages and timeout. Use additional criteria's like Sent date to retrieve a smaller record set.

Example of getMessage (by Scope) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessage(Scope)** operation.

Table 35: Request: getMessage(Scope)

Complex Type Name	Attribute	Value
getMessageTypeDef	requestAPI	REQUEST1
	requestOperation	getMessage
parameters	parameterName	scope
	comparisonOp	equalto
	parametervalue	Public

Webservice Response Details

There will be timeouts retrieving too many messages for the Scope criteria for **getMessage(Scope)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Webservice Request & Response XML

Table 36: getMessage(Scope) Request & Response

Request	Response
<soapenv:Body> <ipaws:getMessageTypeDef> <cap:requestAPI>CAP12</cap:requestAPI> <cap:requestOperation>getMessage</cap:requestOperation> <!--Zero or more repetitions:--> <cap:parameters> <cap:parameterName>scope</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <!--Zero or more repetitions:--> <cap:parameterValue>Public</cap:parameterValue> </cap:parameters> </ipaws:getMessageTypeDef> </soapenv:Body>	Timeout as this operation is trying to retrieve too many messages.

This method may attempt to retrieve too many messages and timeout. Use additional criteria's like Sent date to retrieve a smaller record set.

Example of getMessage (by Sender) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessage(Sender)** operation.

Table 37: Request getMessage(Sender)

Complex Type Name	Attribute	Value
	requestAPI	REQUEST1

Complex Type Name	Attribute	Value
getMessageTypeDef	requestOperation	getMessage
parameters	parameterName	sender
	comparisonOp	equalto
	parametervalue	alerts@test.org

Webservice Response Details

There will be timeouts retrieving too many messages with the sender criteria for **getMessage(Sender)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Webservice Request & Response XML

Table 38: getMessage(Sender) Request & Response XML

Request	Response
<soapenv:Body> <ipaw:getMessageTypeDef> <cap:requestAPI>Request1</cap:requestAPI> <cap:requestOperation>getMessage</cap:requestOperation> <!--Zero or more repetitions:--> <cap:parameters> <cap:parameterName>sender</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <!--Zero or more repetitions:--> <cap:parameterValue>alerts@test.org</cap:parameterValue> </cap:parameters> </ipaw:getMessageTypeDef> </soapenv:Body>	Timeout as this operation is trying to retrieve too many messages.

This method may attempt to retrieve too many messages and timeout. Use additional criteria's like Sent date to retrieve a smaller record set.

Example of getMessage (by Status) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessage(Status)** operation.

Table 39: Request getMessage(Status)

Complex Type Name	Attribute	Value
getMessageTypeDef	requestAPI	REQUEST1
	requestOperation	getMessage
parameters	parameterName	status
	comparisonOp	equalto
	parametervalue	Actual

Webservice Response Details

There may be timeouts as it is retrieving too many messages for Status criteria for **getMessage(Status)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Webservice Request & Response XML

Table 40: getMessage(Status) Request & Response XML

Request	Response
<soapenv:Body> <ipaw:getMessageTypeDef> <cap:requestAPI>CAP12</cap:requestAPI> <cap:requestOperation>getMessage</cap:requestOperation> <!--Zero or more repetitions:--> <cap:parameters> <cap:parameterName>status</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <!--Zero or more repetitions:--> <cap:parameterValue>Actual</cap:parameterValue> </cap:parameters> </ipaw:getMessageTypeDef>	Timeout as this operation is trying to retrieve too many messages.

Request	Response
</soapenv:Body>	

This method may retrieve lot of messages and will timeout. Use additional criteria's like Sent date to retrieve smaller record set.

Example of getMessage (by SentDateTime) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessage(Sent)** operation.

Table 41: Request getMessage(Sent)

Complex Type Name	Attribute	Value
getMessageTypeDef	requestAPI	REQUEST1
	requestOperation	getMessage
parameters	parameterName	sent
	comparisonOp	greaterthan
	parametervalue	2023-05-24T08:36:38-05:00

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessage(Sent)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 42: Response getMessage(Sent)

Complex Type Name	Attribute	Value
messageResponseTypeDef	identifier	Test_IPAWS_20235713160

Webservice Request & Response XML

Table 43: getMessage(Sent) Request & Response XML

Request	Response
<soapenv:Body> <ipaw:getMessageTypeDef> <cap:requestAPI> REQUEST1</cap:requestAPI> <cap:requestOperation>getMessage</cap:requestOperation> <!--Zero or more repetitions:--> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>greaterthan</cap:comparisonOp> <!--Zero or more repetitions:--> <cap:parameterValue>2023-05-24T08:36:38-05:00</cap:parameterValue> </cap:parameters> </ipaw:getMessageTypeDef> </soapenv:Body>	<soap:Envelope XMLns:soap="http://schemas.XMLsoap.org/soap/envelope/"> <soap:Body> <ns2:messageResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns3:alert> <ns3:identifier>Test_IPAWS_20235713160</ns3:identifier> <ns3:sender>alerts@test.org</ns3:sender> <ns3:sent>2023-06-07T13:16:00-04:00</ns3:sent> <ns3:status>Actual</ns3:status> <ns3:msgType>Alert</ns3:msgType> <ns3:scope>Public</ns3:scope> <ns3:code>IPAWSv1.0</ns3:code> <ns3:info> <ns3:language>en-US</ns3:language> <ns3:category>Geo</ns3:category> <ns3:event>Earthquake Warning</ns3:event> <ns3:responseType>Shelter</ns3:responseType> <ns3:urgency>Immediate</ns3:urgency> <ns3:severity>Severe</ns3:severity> <ns3:certainty>Observed</ns3:certainty> <ns3:eventCode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>EQW</ns3:value> </ns3:eventCode> <ns3:expires>2023-06-07T13:46:00-04:00</ns3:expires> <ns3:senderName>COGID, CogName, Requesting Agency </ns3:senderName> <ns3:headline>Earthquake Warning</ns3:headline> <ns3:description>Earthquake Detected! Drop, Cover, Hold On. Protect Yourself. </ns3:description> <ns3:parameter> <ns3:valueName>EAS-ORG</ns3:valueName> <ns3:value>CIV</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>timezone</ns3:valueName> <ns3:value>PST</ns3:value>

Request	Response
	</ns3:parameter> <ns3:parameter> <ns3:valueName>CMAMtext</ns3:valueName> <ns3:value>Earthquake Detected! Drop, Cover, Hold On. Protect Yourself. - </ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>BLOCKCHANNEL</ns3:valueName> <ns3:value>NWEM</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>BLOCKCHANNEL</ns3:valueName> <ns3:value>EAS</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>WEAHandling</ns3:valueName> <ns3:value>Earthquake</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>DBGFBYPASS</ns3:valueName> <ns3:value>TRUE</ns3:value> </ns3:parameter> <ns3:area> <ns3:areaDesc>Earthquake Detected! Drop, Cover, Hold On. Protect Yourself. -</ns3:areaDesc> <ns3:polygon>39.1000,-122.5000 38.9000,-122.5000 38.9000,- 123.1000 38.7000,-123.1000 38.7000,-123.7000 38.5000,-123.7000 38.5000,- 122.3000 39.1000,-122.3000 39.1000,-122.5000</ns3:polygon><ns3:geocode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>041033</ns3:value> </ns3:geocode><ns3:geocode><ns3:valueName>SAME</ns3:valueName> <ns3:value>041029</ns3:value></ns3:geocode> <ns3:geocode><ns3:valueName>SAME</ns3:valueName> <ns3:value>041015</ns3:value> </ns3:geocode> <ns3:geocode><ns3:valueName>SAME</ns3:valueName> <ns3:value>006103</ns3:value></ns3:geocode> </ns3:area></ns3:info> <ns3:info><ns3:language>es-US</ns3:language> <ns3:category>Geo</ns3:category> <ns3:event>Earthquake Warning</ns3:event> <ns3:responseType>Shelter</ns3:responseType>

Request	Response
	<ns3:urgency>Immediate</ns3:urgency> <ns3:severity>Severe</ns3:severity> <ns3:certainty>Observed</ns3:certainty> <ns3:eventCode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>EQW</ns3:value> </ns3:eventCode> <ns3:expires>2023-06-07T13:46:00-04:00</ns3:expires> <ns3:senderName>COGID, CogName, Requesting Agency </ns3:senderName> <ns3:headline>USGS Earthquake Warning</ns3:headline> <ns3:description>Terremoto detectado! </ns3:description> <ns3:parameter> <ns3:valueName>EAS-ORG</ns3:valueName> <ns3:value>CIV</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>timezone</ns3:valueName> <ns3:value>PST</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>CMAMtext</ns3:valueName> <ns3:value>Terremoto detectado!</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>BLOCKCHANNEL</ns3:valueName> <ns3:value>NWEM</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>BLOCKCHANNEL</ns3:valueName> <ns3:value>EAS</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>WEAHandling</ns3:valueName> <ns3:value>Earthquake</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>DBGFBYPASS</ns3:valueName> <ns3:value>TRUE</ns3:value> </ns3:parameter> <ns3:area>

Request	Response
	<ns3:areaDesc>Terremoto detectado! -</ns3:areaDesc> <ns3:polygon>39.1000,-122.5000 38.9000,-122.5000 38.9000,-123.1000 38.7000,-123.1000 38.7000,-123.7000 38.5000,-123.7000 38.5000,-124.5000 38.7000,-122.3000 39.1000,-122.5000</ns3:polygon> <ns3:geocode><ns3:valueName>SAME</ns3:valueName> <ns3:value>041033</ns3:value></ns3:geocode> <ns3:geocode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>XXX123</ns3:value> </ns3:geocode> <ns3:geocode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>XXX456</ns3:value> </ns3:geocode> <ns3:geocode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>XXX789</ns3:value> </ns3:geocode> <ns3:geocode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>XXX012</ns3:value> </ns3:geocode> </ns3:area> </ns3:info> </ns3:alert> </ns2:messageResponseTypeDef> </soap:Body> </soap:Envelope>

The best practice to pull a small set of messages no more than two days from the current date (current date – 2).

7.8 GetMessage Operations using Multiple Parameters

The following table depicts the request structure of GetMessage operation with multiple parameterName value pair for the Aggregator services.

Table 44: Request GetMessage

Complex Type Name	Attribute	Value	Comments
	requestAPI	CAP12	

Complex Type Name	Attribute	Value	Comments
getMessageTypeDef	requestOperation	getMessage	
Parameters (First Parameter)	parameterName	parametervalue	Accepted Parameter Values are.: • identifier • header • profile • sender • sent • status • scope
	comparisonOp	equalto or like • equalto requests are case-sensitive and compares total string • like requests are not case-sensitive and provides partial string search (“contains in”)	
	parameterValue	One of these values: CIV, PEP, EAS, or WXR	
	logicalOp	Not Required	And
Parameters (Second Parameter)	parameterName	parametervalue	Accepted Parameter Values are: • identifier • header • profile • sender • sent • status • scope
	comparisonOp	equalto or like • equalto requests are case-sensitive and compares total string • like requests are not case-sensitive and provides partial string search (“contains in”)	
	parameterValue	One of these values: CIV, PEP, EAS, or WXR	

The best practice is to Use equalto on <cap:comparisonOp>equalto </cap:comparisonOp> for an exact match instead of “like” to pull relevant records.

The following tables lists all the examples of GetMessage Operations illustrates the request message for the Aggregator service. This request with only one parameterValue requires both a getMessageTypeDef and parameters element.

Example of getMessage (by SentDateGreaterThanAndLessThan) Operation – No Message Found

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessage (SentDateTime)** operation.

Table 45: Request getMessage(SentDateTime)

Complex Type Name	Attribute	Value
getMessageTypeDef	requestAPI	REQUEST1
	requestOperation	getMessage
First Parameters	parameterName	sent
	comparisonOp	greaterthan
	parametervalue	2023-05-11T11:30:59-04:00
	logicalOp	and
Second Parameters	parameterName	sent
	comparisonOp	lessthan
	parametervalue	2023-04-11T14:50:59-04:00

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessage(SentDateTime)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 46: Response getMessageSentDateTime

Complex Type Name	Attribute	Value
messageResponseTypeDef	identifier	NOMESSAGEFOUND-200040947817

Webservice Request & Response XML

Table 47: getMessageSentDateTime Request & Response XML

Request	Response
<soapenv:Body> <ipaw:getMessageTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessage</cap:requestOperation> <!--Zero or more repetitions:--> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>greaterthan</cap:comparisonOp> <cap:parameterValue>2023-05-11T11:30:59-04:00 </cap:parameterValue> <cap:logicalOp>and</cap:logicalOp> </cap:parameters> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>lessthan</cap:comparisonOp> <cap:parameterValue>2023-04-11T14:50:59-04:00 </cap:parameterValue> </cap:parameters> </ipaw:getMessageTypeDef> </soapenv:Body>	<soap:Body> <ns2:messageResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns3:alert> <ns3:identifier>NOMESSAGEFOUND-200040947817</ns3:identifier> <ns3:sender>test@fema.gov</ns3:sender> <ns3:sent>2023-06-08T15:04:02-00:00</ns3:sent> <ns3:status>System</ns3:status> <ns3:msgType>Alert</ns3:msgType> <ns3:scope>Public</ns3:scope> <ns3:NOTE>NO MESSAGE FOUND</ns3:NOTE> </ns3:alert> </ns2:messageResponseTypeDef> </soap:Body>

Example of getMessage (by SentDateGreaterThanAndLessThan) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessage(SentDateTime)** operation.

Table 48: Request getMessage(SentDateTime)

Complex Type Name	Attribute	Value
getMessageTypeDef	requestAPI	REQUEST1
	requestOperation	getMessage
First Parameters	parameterName	sent
	comparisonOp	greaterthan
	parametervalue	2023-04-11T11:30:59-04:00
	logicalOp	and
Second Parameters	parameterName	sent
	comparisonOp	lessthan
	parametervalue	2023-04-11T14:50:59-04:00

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessage(SentDateTime)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 49: Resonse getMessage(SentDateTime)

Complex Type Name	Attribute	Value
messageResponseTypeDef	identifier	TEST-IPAWS-2023311144719

Webservice Request & Response XML

Table 50: getMessage(SentDateTime) Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getMessageTypeDef> <cap:requestAPI>Request1</cap:requestAPI> <cap:requestOperation>getMessage</cap:requestOperation> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>greaterthan</cap:comparisonOp> <cap:parameterValue>2023-04-11T11:30:59-04:00</cap:parameterValue> <cap:logicalOp>and</cap:logicalOp>	<soap:Envelope XMLns:soap="http://schemas.XMLsoap.org/soap/envelope/"> <soap:Body> <ns2:messageResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns3:alert> <ns3:identifier>TEST-IPAWS-2023311144719</ns3:identifier>

Request	Response
</cap:parameters> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>lessthan</cap:comparisonOp> <cap:parameterValue>2023-04-11T14:50:59-04:00</cap:parameterValue> </cap:parameters> </ipaws:getMessageTypeDef> </soapenv:Body>	<ns3:sender>IPAWS-TEST</ns3:sender> <ns3:sent>2023-04-11T14:47:19-04:00</ns3:sent> <ns3:status>Actual</ns3:status> <ns3:msgType>Alert</ns3:msgType> <ns3:source>IPAWS-Tester</ns3:source> <ns3:scope>Public</ns3:scope> <ns3:addresses>XXX123</ns3:addresses> <ns3:code>IPAWSv1.0</ns3:code> <ns3:info> <ns3:language>en-US</ns3:language> <ns3:category>CBRNE</ns3:category> <ns3:event>Evacuation Immediate</ns3:event> <ns3:responseType>Evacuate</ns3:responseType> <ns3:urgency>Immediate</ns3:urgency> <ns3:severity>Severe</ns3:severity> <ns3:certainty>Observed</ns3:certainty> <ns3:eventCode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>EVI</ns3:value> </ns3:eventCode> <ns3:effective>2023-04-11T14:47:19-04:00</ns3:effective> <ns3:expires>2023-04-11T15:47:19-04:00</ns3:expires> <ns3:senderName>COGID, CogName, Requesting Agency</ns3:senderName> <ns3:headline>WEA 2.0 Test. Test Message only Disregard please.</ns3:headline> <ns3:description>This is a Simulation - This is Only a Test. A Nuclear Accident has occurred at the Indian Head Nuclear Plant causing the release of significant amounts of radioactive material.</ns3:description> <ns3:instruction>All residents within a 10-mile radius of Indian Head, MD, MUST EVACUATE IMMEDIATELY. This is a Simulation. This is Only a Test.</ns3:instruction> <ns3:parameter> <ns3:valueName>EAS-ORG</ns3:valueName> <ns3:value>CIV</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>timezone</ns3:valueName> <ns3:value>EST</ns3:value> </ns3:parameter>

Request	Response
	<ns3:parameter> <ns3:valueName>WEAHandling</ns3:valueName> <ns3:value>Imminent Threat</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>CMAMtext</ns3:valueName> <ns3:value>This is where the 90 character English text to WEA goes. http://www.fema.gov</ns3:value> </ns3:parameter> <ns3:parameter> <ns3:valueName>CMAMlongtext</ns3:valueName> <ns3:value>this is where the 360 character description in English would go. Mandatory Evacuation Order for Hwy 74 east of Caspers Park see rivcoready.org or ocgov.com</ns3:value> </ns3:parameter> <ns3:area> <ns3:areaDesc>Alexandria</ns3:areaDesc> <ns3:polygon>38.8512,-77.1912 38.8107,-77.1908 38.8001,-77.0713 38.8503,-77.0701 38.8512,-77.1912</ns3:polygon> <ns3:geocode> <ns3:valueName>SAME</ns3:valueName> <ns3:value>XXX123</ns3:value> </ns3:geocode> </ns3:area> </ns3:info> </ns3:alert> </ns2:messageResponseTypeDef> </soap:Body> </soap:Envelope>

Recommendation: The best practice is to pull messages no greater than two days from the current date.

7.9 GetMessageList Operations using a Single Parameter

A getMessageList request can be used to retrieve a CAP v1.2 message via the Aggregator Service by using single parameter or multiple parameters criteria's. The following table depicts the request structure of GetMessageList operation with single parameterName value for the Aggregator services.

Table 51: GetMessageList Single parameterName

Complex Type Name	Attribute	Value	Comments
getRequestTypeDef	requestAPI	CAP12 OR Request1	
	requestOperation	getMessageList	
parameters	parameterName	parametervalue	Accepted Parameter Values are • identifier • header • profile • sender • sent • status • scope • addresses • senderName
	comparisonOp	equalto or like • equalto requests are case-sensitive and compares total string • like requests are not case-sensitive and provides partial string search (“contains in”)	
	parameterValue	One of these values: CIV, PEP, EAS, or WXR	
	logicalOp	Not Required	

Recommendation: Best practice is to Use equalto on <cap:comparisonOp>equalto </cap:comparisonOp> for exact match instead of like to pull relevant smaller record set.

The following tables lists all the examples of GetMessageList Operations illustrates the request message for the Aggregator service. This request with only one parameterValue requires both a getRequestTypeDef and parameters element.

Example getMessageList (by Identifier) Operation – No Message Found

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageList(Identifier)** operation

Table 52: Request GetMessageList(Identifier)

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageList
Parameter	parameterName	identifier
		equalto
	parametervalue	TEST-IPAWS-4.1-CAE_202316153315

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessageList(Identifier)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 53: Response getMessageList(Identifier)

Complex Type Name	Attribute	Value
getResponseTypeDef	parameterName	NO MESSAGE LIST FOUND
	ResponseOperation	getMessageList

Webservice Request & Response XML

Table 54: getMessageList(Identifier) Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList</cap:requestOperation> <cap:parameters> <cap:parameterName>identifier</cap:parameterName> <cap:comparisonOp> equalto </cap:comparisonOp> <cap:parameterValue>TEST-IPAWS-4.1-CAE_202316153315</cap:parameterValue> </cap:parameters>	<soap:Envelope XMLns:soap="http://schemas.XMLsoap.org/soap/envelope/"> <soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem>

Request	Response
</ipaws:getRequestTypeDef> </soapenv:Body>	<ns4:parameterName>NO MESSAGE LIST FOUND</ns4:parameterName> </ns4:parameterListItem> <ns4:ResponseOperation>getMessageList</ns4:ResponseOperation> </ns2:getResponseTypeDef> </soap:Body> </soap:Envelope>

Example of getMessageList (by Identifier) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageList(Identifier)** operation.

Table 55: Request getMessageList(Identifier)

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageList
Parameter	parameterName	identifier
	comparisonOp	equalto
	parametervalue	TEST-IPAWS-4.1-CAE_202316153315

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessageList(Identifier)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 56: Response getMessage(Identifier)

Complex Type Names	Attributes	Values
getResponseTypeDef	parameterName	msgid
	parameterValue	TEST-IPAWS-e3d00903-7cb3-448d

Complex Type Names	Attributes	Values
	ResponseOperations	getMessageList

Webservice Request & Response XML

Table 57: getMessage(Identifier) Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList</cap:requestOperation> <cap:parameters> <cap:parameterName>identifier</cap:parameterName> <cap:comparisonOp> equalto </cap:comparisonOp> <cap:parameterValue>TEST-IPAWS-e3d00903-7cb3-448d </cap:parameterValue> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>msgid</ns4:parameterName> <ns4:parameterValue> TEST-IPAWS-e3d00903-7cb3-448d </ns4:parameterValue> <ns4:subParaListItem> <ns4:subParameterName>cogname</ns4:subParameterName> <ns4:subParameterValue>COG IPAWSOPENXXX123</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>headline</ns4:subParameterName> <ns4:subParameterValue>26 Sep Test Message -Test Message only Disregard please.</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>sender</ns4:subParameterName> <ns4:subParameterValue>IPAWS-TEST</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>sent</ns4:subParameterName> <ns4:subParameterValue>2023-05-22T18:31:11-00:00</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>status</ns4:subParameterName> <ns4:subParameterValue>Actual</ns4:subParameterValue> </ns4:subParaListItem>

Request	Response
	<ns4:subParaListItem> <ns4:subParameterName>msgtype</ns4:subParameterName> <ns4:subParameterValue>Alert</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>scope</ns4:subParameterName> <ns4:subParameterValue>Public</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>incidents</ns4:subParameterName> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>profilecode</ns4:subParameterName> <ns4:subParameterValue>IPAWSv1.0</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>EAS-ORG</ns4:subParameterName> <ns4:subParameterValue>CIV</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ipawsProfile</ns4:subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>msgSize</ns4:subParameterName> <ns4:subParameterValue>6213</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>eventCode</ns4:subParameterName> <ns4:subParameterValue>RWT</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem><ns4:subParameterName>signaturevalidated</ns4 :subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>entryTimestamp</ns4:subParameterName> <ns4:subParameterValue>2023-05-22 18:31:12.577779</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> <ns4:ResponseOperation>getMessageList</ns4:ResponseOperation>

Request	Response
	<ns4:ResponseType>REQUEST1</ns4:ResponseType> </ns2:getResponseTypeDef> </soap:Body>

Example of getMessageList (by Headline) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageList(Headline)** operation.

Table 58: Request getMessageList(Headline)

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageList
Parameter	parameterName	headline
	comparisonOp	equalto
	parametervalue	Tornado Warning issued March 24 at 7:53PM CDT until March 24 at 9:00PM

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessageList(Headline)** operation.
Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 59: Response getMessageList(Headline)

Complex Type Name	Attribute	Value
getResponseTypeDef	parameterName	msgid
	parameterValue	TEST-IPAWS-WEA-q1202002-7cb3-448d
	ResponseOperation	getMessageList

Webservice Request & Response XML

Table 60: getMessageList(Headline) Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList </cap:requestOperation> <cap:parameters> <cap:parameterName> headline </cap:parameterName> <cap:comparisonOp> equalto </cap:comparisonOp> <cap:parameterValue> Tornado Warning issued March 24 at 7:53PM CDT until March 24 at 9:00PM </cap:parameterValue> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>msgid</ns4:parameterName> <ns4:parameterValue> TEST-IPAWS-q1202002-7cb3-448d</ns4:parameterValue> <ns4:subParaListItem> <ns4:subParameterName>cogname</ns4:subParameterName> <ns4:subParameterValue>COG IPAWSOPENXXX123</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>headline</ns4:subParameterName> <ns4:subParameterValue> Tornado Warning issued March 24 at 7:53PM CDT until March 24 at 9:00PM</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>sender</ns4:subParameterName> <ns4:subParameterValue>IPAWS-TEST</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>sent</ns4:subParameterName> <ns4:subParameterValue>2023-05-22T18:31:11-00:00</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>status</ns4:subParameterName> <ns4:subParameterValue>Actual</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>msgtype</ns4:subParameterName>

Request	Response
	<ns4:subParameterValue>Alert</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>scope</ns4:subParameterName> <ns4:subParameterValue>Public</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>incidents</ns4:subParameterName> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>profilecode</ns4:subParameterName> <ns4:subParameterValue>IPAWSv1.0</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>EAS-ORG</ns4:subParameterName> <ns4:subParameterValue>CIV</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ipawsProfile</ns4:subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>msgSize</ns4:subParameterName> <ns4:subParameterValue>6213</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>eventCode</ns4:subParameterName> <ns4:subParameterValue>RWT</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem><ns4:subParameterName>signaturevalidated</ns4:subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>entryTimestamp</ns4:subParameterName> <ns4:subParameterValue>2023-05-22 18:31:12.577779</ns4:subParameterValue> </ns4:subParaListItem>

Request	Response
	</ns4:parameterListItem> <ns4:ResponseOperation>getMessageList</ns4:ResponseOperation> <ns4:ResponseType>REQUEST1</ns4:ResponseType> </ns2:getResponseTypeDef> </soap:Body>

Example of getMessageList (by MsgType) Operation

The message type (msgType) comparison value can be Alert or Cancel or Update.

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageList(MsgType)** operation.

Table 61: Request getMessageList(MsgType)

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageList
Parameter	parameterName	msgType
	comparisonOp	equalto
	parametervalue	Cancel

Webservice Response Details

Below table mentions some of the expected response details for a successful **getMessageList(MsgType)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 62: Response getMessageList(MsgType)

Complex Type Name	Attribute	Value
getResponseTypeDef	parameterName	msgid
	parameterValue	TEST-IPAWS-1202002-7cb3-448d

Complex Type Name	Attribute	Value
	ResponseOperation	getMessageList

Webservice Request & Response XML

Table 63: getMessageList(MsgType) Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList</cap:requestOperation> <cap:parameters> <cap:parameterName>msgType</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <cap:parameterValue>Cancel</cap:parameterValue> </cap:parameters></ipaws:getRequestTypeDef> </soapenv:Body>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>msgid</ns4:parameterName> <ns4:parameterValue> TEST-IPAWS-1202002-7cb3-448d </ns4:parameterValue> <ns4:subParaListItem> <ns4:subParameterName>cogname</ns4:subParameterName> <ns4:subParameterValue>IPAWS Test COG</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>headline</ns4:subParameterName> <ns4:subParameterValue> Test Message only Disregard please.</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>sender</ns4:subParameterName> <ns4:subParameterValue>TEST@fema.gov</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>sent</ns4:subParameterName> <ns4:subParameterValue>2020-09-18T02:56:28-00:00</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>status</ns4:subParameterName> <ns4:subParameterValue>Actual</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>msgtype</ns4:subParameterName> <ns4:subParameterValue>Cancel</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem>

Request	Response
	<ns4:subParameterName>scope</ns4:subParameterName> <ns4:subParameterValue>Public</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>incidents</ns4:subParameterName> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>profilecode</ns4:subParameterName> <ns4:subParameterValue>IPAWSv1.0</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>EAS-ORG</ns4:subParameterName> <ns4:subParameterValue>CIV</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ipawsProfile</ns4:subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>msgSize</ns4:subParameterName> <ns4:subParameterValue>7278</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>eventCode</ns4:subParameterName> <ns4:subParameterValue>CDW</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>signaturevalidated</ns4:subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>entryTimestamp</ns4:subParameterName><ns4:subParameterValue>2020-09-18 02:55:52.887576</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> <ns4:ResponseOperation>getMessageList</ns4:ResponseOperation> <ns4:ResponseType>REQUEST1</ns4:ResponseType> </ns2:getResponseTypeDef> </soap:Body>

Recommendation: This method may retrieve too many messages and may timeout. Use additional criteria's like Sent date to retrieve a smaller record set.

Example getMessageList (by Scope) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageList(Scope)** operation.

Table 64: Request getMessageList(Scope)

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageList
Parameter	parameterName	scope
	comparisonOp	equalto
	parametervalue	Public

Webservice Response Details

The table below includes expected response details for a successful **getMessageList(Scope)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 65: Respond getMessageList(Scope)

Complex Type Name	Attributes	Value
getResponseTypeDef	parameterName	ERROR Message
	parameterValue	Error processing getReq. Error Code: OP01

Webservice Request & Response XML

Table 66: getMessageList(Scope) Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList</cap:requestOperation> <cap:parameters> <cap:parameterName>scope</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <cap:parameterValue>Public</cap:parameterValue> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	Timeout as it this operation is trying to retrieve too many messages. <soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="http://gov.fema.ipaws.services/capresponse" XMLns:ns4="urn:oasis:names:tc:emergency:cap:1.2"> <ns3:parameterListItem> <ns3:parameterName>ERROR</ns3:parameterName> <ns3:parameterValue>Y</ns3:parameterValue> <ns3:subParaListItem> <ns3:subParameterName>ERROR Message</ns3:subParameterName> <ns3:subParameterValue>Error processing getReq. Error Code: OP01</ns3:subParameterValue> </ns3:subParaListItem> </ns3:parameterListItem> </ns2:getResponseTypeDef> </soap:Body>

This method may retrieve too many messages and may timeout. Use additional criteria's like Sent date to retrieve a smaller record set.

Example getMessageList (by Sender) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageList(bySenderScope)** operation.

Table 67: Request getmessageList(Sender)

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageList

Complex Type Name	Attribute	Value
Parameter	parameterName	sender
	comparisonOp	equalto
	parametervalue	test@test.com

Webservice Response Details

The table below includes expected response details for a successful **getMessageList(Sender)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 68:Response getMessageList(Sender)

Complex Type Name	Attribute	Value
getResponseTypeDef	parameterName	ERROR Message
	parameterValue	Error processing getReq. Error Code: OP01

Webservice Request & Response XML

Table 69: getMessageList(Sender) Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList</cap:requestOperation> <cap:parameters> <cap:parameterName>sender</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <cap:parameterValue>test@test.com</cap:parameterValue> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	Timeout as it this operation is trying to retrieve too many messages. <soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>ERROR</ns4:parameterName> <ns4:parameterValue>Y</ns4:parameterValue> <ns4:subParaListItem> <ns4:subParameterName>ERROR Message</ns4:subParameterName> <ns4:subParameterValue>Error processing getReq. Error Code: OP01</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> </ns2:getResponseTypeDef>

Request	Response
	</soap:Body>

Use additional criteria's like Sent date to retrieve a smaller record set to avoid timeouts retrieving too many messages.

Example of getMessageList (by Status) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageListByStatusOperation**

Table 70: Request getMessageListbyStatusOperation

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageList
Parameter	parameterName	status
	comparisonOp	equalto
	parametervalue	Actual

Webservice Response Details

The table below includes expected response details for a successful **getMessageList(Sender)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 71: Response getMessageListbyStatusOperation

Complex Type Name	Attribute	Value
getResponseTypeDef	parameterName	ERROR Message
	parameterValue	Error processing getReq. Error Code: OP01

Webservice Request & Response XML

Table 72: getMessageListbyStatusOperation Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList</cap:requestOperation> <cap:parameters> <cap:parameterName>status</cap:parameterName> <cap:comparisonOp>equalto</cap:comparisonOp> <cap:parameterValue>Actual</cap:parameterValue> <cap:logicalOp></cap:logicalOp> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	Timeout as it this operation is trying to retrieve too many messages. <soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>ERROR</ns4:parameterName> <ns4:parameterValue>Y</ns4:parameterValue> <ns4:subParaListItem> <ns4:subParameterName>ERROR Message</ns4:subParameterName> <ns4:subParameterValue>Error processing getReq. Error Code: OP01</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> </ns2:getResponseTypeDef> </soap:Body>

This method may retrieve too many messages and may timeout. Use additional criteria's like Sent date to retrieve a smaller record set.

Example of getMessageList (by SentDateTime) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageListbySentByDateTime** operation.

Webservice Response Details

The table below includes expected response details for a successful **getMessageListSentByDateTime.** Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 73: Respond getMessageListBySentDateTime

Complex Type Name	Attribute	Value
getResponseTypeDef	parameterName	msgid
	parameterValue	TEST_IPAWS_202357105922
	ResponseOperation	getMessageList

Webservice Request & Response XML

Table 74: getMessageListBySentDateTime Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList</cap:requestOperation> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>greaterthan</cap:comparisonOp> <cap:parameterValue>2023-06-06T09:30:38-05:00</cap:parameterValue> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>msgid</ns4:parameterName> <ns4:parameterValue> TEST_IPAWS_202357105922</ns4:parameterValue> <ns4:subParaListItem> <ns4:subParameterName>cogname</ns4:subParameterName> <ns4:subParameterValue>IPAWS Test COG</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>headline</ns4:subParameterName> <ns4:subParameterValue>USGS Earthquake Warning</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>sender</ns4:subParameterName> <ns4:subParameterValue>test@fema.gov</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>sent</ns4:subParameterName> <ns4:subParameterValue>2023-06-07T14:59:22-00:00</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem>

Request	Response
	<ns4:subParameterName>status</ns4:subParameterName> <ns4:subParameterValue>Actual</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>msgtype</ns4:subParameterName> <ns4:subParameterValue>Alert</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>scope</ns4:subParameterName> <ns4:subParameterValue>Public</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>incidents</ns4:subParameterName> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>profilecode</ns4:subParameterName> <ns4:subParameterValue>IPAWSv1.0</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>EAS-ORG</ns4:subParameterName> <ns4:subParameterValue>CIV</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ipawsProfile</ns4:subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>msgSize</ns4:subParameterName> <ns4:subParameterValue>9958</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>eventCode</ns4:subParameterName> <ns4:subParameterValue>EQW</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>signaturevalidated</ns4:subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem>

Request	Response
	<ns4:subParaListItem> <ns4:subParameterName>entryTimestamp</ns4:subParameterName> <ns4:subParameterValue>2023-06-07 14:59:24.977794</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> <ns4:ResponseOperation>getMessageList</ns4:ResponseOperation> <ns4:ResponseType>REQUEST1</ns4:ResponseType> </ns2:getResponseTypeDef> </soap:Body>

Recommendation: The best practice is to pull messages no greater than 2 days from the current date.

Example of getMessageListBySentDateTime Operation – No Message Found

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageList(Sent)** operation.

Table 75: Request getMessageList(Sent)

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageList
First Parameter	parameterName	sent
	comparisonOp	greaterthan
	parametervalue	2023-05-11T11:30:59-04:00
	logicalOp	and
Second Parameter	parameterName	sent
	comparisonOp	lessthan

Complex Type Name	Attribute	Value
	parametervalue	2023-05-14T11:33:59-04:00

Webservice Response Details

The table below includes expected response details for a successful **getMessageList(Sent)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 76: Respond getMessageList(Sent)

Complex Type Name	Attribute	Value
getResponseTypeDef	parameterName	NO MESSAGE LIST FOUND
	ResponseOperation	getMessageList

Webservice Request & Response XML

Table 77: getMessageList(Sent) Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList</cap:requestOperation> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>greaterthan</cap:comparisonOp> <cap:parameterValue>2023-05-11T11:30:59-04:00</cap:parameterValue> <cap:logicalOp>and</cap:logicalOp> </cap:parameters> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>lessthan</cap:comparisonOp> <cap:parameterValue>2023-05-14T11:33:59-04:00</cap:parameterValue> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="http://gov.fema.ipaws.services/capresponse" XMLns:ns4="urn:oasis:names:tc:emergency:cap:1.2"> <ns3:parameterListItem> <ns3:parameterName>NO MESSAGE LIST FOUND</ns3:parameterName> </ns3:parameterListItem> <ns3:ResponseOperation>getMessageList</ns3:ResponseOperation> </ns2:getResponseTypeDef> </soap:Body>

7.10 GetMessageList Operations using Multiple Parameters

The following table depicts the request structure of GetMessageList operation with multiple parameterName value pair for the Aggregator services.

Table 78: Request GetMessageList

Complex Type Name	Attribute	Value	Comments
getRequestTypeDef	requestAPI	CAP12 OR Request1	
	requestOperation	getMessageList	
Parameters (First Parameter)	parameterName	parametervalue	Accepted Parameter Values are • identifier • header • profile • sender • sent • status • scope
	comparisonOp	equalto or like • equalto requests are case-sensitive and compares total string • like requests are not case-sensitive and provides partial string search (“contains in”)	
	parameterValue	One of these values: CIV, PEP, EAS, or WXR	
	logicalOp	Not Required	And
Parameters (Second Parameter)	parameterName	parametervalue	Accepted Parameter Values are • identifier • header • profile • sender • sent • status • scope
	comparisonOp	equalto or like	

Complex Type Name	Attribute	Value	Comments
		• equalto requests are case-sensitive and compares total string • like requests are not case-sensitive and provides partial string search (“contains in”)	
	parameterValue	One of these values: CIV, PEP, EAS, or WXR	

The best practice is to Use equalto on <cap:comparisonOp>equalto </cap:comparisonOp> for exact match instead of “like” to pull relevant records.

The following tables lists all the examples of GetMessage Operations illustrates the request message for the Aggregator service. This request with only one parameterValue requires both a getRequestTypeDef and parameters element.

Example of getMessageList (by DateGreaterThanAndLessThan) Operation

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details
Below is the request parameters and values required to ping **getMessageListByDateGreaterThanAndLessThanOperation,**

Table 79: Request getMessageListByDateGreaterThanAndLessThanOperation

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageList
First Parameter	parameterName	sent
	comparisonOp	greaterthan
	parametervalue	2023-05-11T11:30:59-04:00
	logicalOp	and

Complex Type Name	Attribute	Value
Second Parameter	parameterName	sent
	comparisonOp	lessthan
	parametervalue	2023-05-14T11:33:59-04:00

Webservice Response Details

The table below includes expected response details for a successful **getMessageList(Sent)** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 80: Request getMessageListByDateGraterThanAndLessThanOperation

Complex Type Name	Attribute	Value
getResponseTypeDef	parameterName	msgid
	parameterValue	TEST-IPAWS-202342412811
	ResponseOperation	getMessageList

Webservice Request & Response XML

Table 81: getMessageListByDateGraterThanAndLessThanOperation Request & Response XML

Request	Response
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageList </cap:requestOperation> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>greaterthan</cap:comparisonOp> <cap:parameterValue>2023-05-24T11:30:59-04:00</cap:parameterValue> <cap:logicalOp>and</cap:logicalOp> </cap:parameters> <cap:parameters> <cap:parameterName>sent</cap:parameterName> <cap:comparisonOp>lessthan</cap:comparisonOp> <cap:parameterValue>2023-05-24T15:33:59-04:00</cap:parameterValue> </cap:parameters>	<soap:Body> <ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>msgid</ns4:parameterName><ns4:parameterValue>TEST-IPAWS-202342412811</ns4:parameterValue> <ns4:subParaListItem> <ns4:subParameterName>cogname</ns4:subParameterName> <ns4:subParameterValue> TEST COG4</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>headline</ns4:subParameterName> <ns4:subParameterValue> Test Message only Disregard please.</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem>

Request	Response
</ipaws:getRequestTypeDef> </soapenv:Body>	<ns4:subParameterName>sender</ns4:subParameterName> <ns4:subParameterValue>IPAWS-TEST</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>sent</ns4:subParameterName> <ns4:subParameterValue>2023-05-24T16:08:11-00:00</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>status</ns4:subParameterName> <ns4:subParameterValue>Actual</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>msgtype</ns4:subParameterName> <ns4:subParameterValue>Alert</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>scope</ns4:subParameterName> <ns4:subParameterValue>Public</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>incidents</ns4:subParameterName> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>profilecode</ns4:subParameterName> <ns4:subParameterValue>IPAWSv1.0</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>EAS-ORG</ns4:subParameterName> <ns4:subParameterValue>CIV</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ipawsProfile</ns4:subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>msgSize</ns4:subParameterName> <ns4:subParameterValue>6229</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>eventCode</ns4:subParameterName> <ns4:subParameterValue>EVI</ns4:subParameterValue>

Request	Response
	</ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>signaturevalidated</ns4:subParameterName> <ns4:subParameterValue>Y</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>entryTimestamp</ns4:subParameterName> <ns4:subParameterValue>2023-05-24 16:08:14.415006</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> <ns4:ResponseOperation>getMessageList</ns4:ResponseOperation> <ns4:ResponseType>REQUEST1</ns4:ResponseType> </ns2:getResponseTypeDef> </soap:Body>

The best practice is to pull messages no greater than 2 days from the current date.

7.11 Get Message Status

A **getMessageStatus** request can be used to retrieve the same information contained in the consolidated response along with the CMAS Channels where an Alert was successfully disseminated. The following table depicts the request structure of getMessageStatus operation with single parameterName value for the Aggregator services.

Table 82: Request GetMessageStatus for Aggregator Services

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageStatus
parameters	parameterName	Identifier
	comparisonOp	equalto or like • equalto requests are case-sensitive and compares total string • like requests are not case-sensitive and provides partial string search (“contains in”)
	parameterValue	For Example: CAP12-TEST-11-30-0001
	logicalOp	Not Required

The following example for getMessageStatus (by a single parameter) illustrates the request message for the Aggregator service. This request with only one parameterValue requires both a ***getRequestTypeDef*** and ***parameters*** element.

Example of GetMessageStatus

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageStatus** operation.

Table 83: Request getMessageStatus

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageStatus
First Parameter	parameterName	identifier
	comparisonOp	equalto
	parametervalue	TEST-IPAWS-2023425144722

Webservice Response Details

The table below includes expected response details for a successful **getMessageStatus** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 84: Webservice Request & Response

Complex Type Name	Attitude	Value
getResponseTypeDef	parameterName	identifier
	parameterValue	TEST-IPAWS-2023425144722
	ResponseOperation	getMessageList

Table 85: getMessageStatus Request & Response XML

Request	Respond
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation>getMessageStatus</cap:requestOperation> <cap:parameters> <cap:parameterName>identifier</cap:parameterName> <cap:comparisonOp> equalto </cap:comparisonOp> <cap:parameterValue> TEST-2023425144722</cap:parameterValue> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	<ns2:getResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest" XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2" XMLns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>identifier</ns4:parameterName> <ns4:parameterValue> TEST-IPAWS-2023425144722</ns4:parameterValue> </ns4:parameterListItem> <ns4:parameterListItem> <ns4:subParaListItem> <ns4:subParameterName>CHANNELNAME</ns4:subParameterName> <ns4:subParameterValue>CAPEXCH</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUSITEMID</ns4:subParameterName> <ns4:subParameterValue>200</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ERROR</ns4:subParameterName> <ns4:subParameterValue>N</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUS</ns4:subParameterName> <ns4:subParameterValue>Ack</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>CHANNELNAME</ns4:subParameterName> <ns4:subParameterValue>CAPEXCH</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUSITEMID</ns4:subParameterName> <ns4:subParameterValue>202</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ERROR</ns4:subParameterName> <ns4:subParameterValue>N</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUS</ns4:subParameterName>

Request	Respond
	<ns4:subParameterValue>alert-signature-is-valid</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>CHANNELNAME</ns4:subParameterName> <ns4:subParameterValue>IPAWS</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUS</ns4:subParameterName> <ns4:subParameterValue>Ack</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>CHANNELNAME</ns4:subParameterName> <ns4:subParameterValue>EAS</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUSITEMID</ns4:subParameterName> <ns4:subParameterValue>500</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ERROR</ns4:subParameterName> <ns4:subParameterValue>N</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUS</ns4:subParameterName> <ns4:subParameterValue>Ack</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>CHANNELNAME</ns4:subParameterName> <ns4:subParameterValue>CMAS</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUSITEMID</ns4:subParameterName> <ns4:subParameterValue>601</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ERROR</ns4:subParameterName> <ns4:subParameterValue>N</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUS</ns4:subParameterName>

Request	Respond
	<ns4:subParameterValue>message-not-disseminated-as-CMAS-Version-2</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>CHANNELNAME</ns4:subParameterName> <ns4:subParameterValue>PUBLIC</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUSITEMID</ns4:subParameterName> <ns4:subParameterValue>801</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>ERROR</ns4:subParameterName> <ns4:subParameterValue>N</ns4:subParameterValue> </ns4:subParaListItem> <ns4:subParaListItem> <ns4:subParameterName>STATUS</ns4:subParameterName> <ns4:subParameterValue>message-not-disseminated-as-non-EAS-public</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> <ns4:ResponseOperation>getMessageStatus</ns4:ResponseOperation> </ns2:getResponseTypeDef> </soap:Body>

7.12 Get Message Time

A **getMessageTime** request can be used to retrieve the timestamps for when a message is received by IPAWS-Open along with the timestamps for all status codes recorded for a message. The following table depicts the request structure of **getMessageTime** operation with single parameterName value for the Aggregator services.

Table 86: Request **GetMessageTime** for Aggregator Services

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageTime
	parameterName	Identifier

Complex Type Name	Attribute	Value
parameters	comparisonOp	equalto or like • equalto requests are case-sensitive and compares total string • like requests are not case-sensitive and provides partial string search ("contains in")
	parameterValue	For Example: CAP12-TEST-11-30-0001
	logicalOp	Not Required

The following example for **getMessageTime** (by a single parameter) illustrates the request message for the Aggregator service. This request with only one parameterValue requires both a ***getRequestTypeDef*** and ***parameters*** element.

Example of GetMessageTime

Webservice Request and Response details section mentions details of Request & Response parameters and their respective values used in the example. Also, parameters and respective values are highlighted on the Webservice Request and Response sections and as well on the Webservice Request & Response XML section below.

Webservice Request Details

Below is the request parameters and values required to ping **getMessageTime** operation.

Table 87: Request **getMessageTime**

Complex Type Name	Attribute	Value
getRequestTypeDef	requestAPI	REQUEST1
	requestOperation	getMessageTime
First Parameter	parameterName	identifier
	comparisonOp	equalto
	parametervalue	TEST-IPAWS-2023425144722

Webservice Response Details

The table below includes expected response details for a successful **getMessageTime** operation. Please refer to response XML on the Webservice Request & Response XML section for more details.

Table 88: Webservice Request & Response

Complex Type Name	Attitude	Value
getResponseTypeDef	parameterName	identifier
	parameterValue	TEST-IPAWS-2023425144722
	ResponseOperation	getMessageTime

Table 89: **getMessageTime** Request & Response XML

Request	Respond
<soapenv:Body> <ipaws:getRequestTypeDef> <cap:requestAPI>REQUEST1</cap:requestAPI> <cap:requestOperation> getMessageTime </cap:requestOperation> <cap:parameters> <cap:parameterName>identifier</cap:parameterName> <cap:comparisonOp> equalto </cap:comparisonOp> <cap:parameterValue> TEST-2023425144722</cap:parameterValue> </cap:parameters> </ipaws:getRequestTypeDef> </soapenv:Body>	<ns2:getResponseTypeDef xmlns="http://gov.fema.ipaws.services/caprequest" xmlns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" xmlns:ns3="urn:oasis:names:tc:emergency:cap:1.2" xmlns:ns4="http://gov.fema.ipaws.services/capresponse"> <ns4:parameterListItem> <ns4:parameterName>msgid</ns4:parameterName><ns4:parameterValue>TEST-IPAWS-2023425144722</ns4:parameterValue><ns4:subParaListItem> <ns4:subParameterName>cogname</ns4:subParameterName> <ns4:subParameterValue>COG2</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>headline</ns4:subParameterName> <ns4:subParameterValue>WEA 2.0 Test. Test Message only Disregard please.</ns4:subParameterValue></ns4:subParaListItem> <ns4:subParaListItem><ns4:subParameterName>sender</ns4:subParameterName> <ns4:subParameterValue>IPAWS-Test</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>receivedTimestamp</ns4:subParameterName> <ns4:subParameterValue>2023-10-05 13:31:52.187</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>processingTime</ns4:subParameterName> <ns4:subParameterValue>599 ms</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>202 CAPEXCH</ns4:subParameterName> <ns4:subParameterValue>2023-10-05 13:31:52.218</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>200 CAPEXCH</ns4:subParameterName> <ns4:subParameterValue>2023-10-05 13:31:52.279</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>300 IPAWS</ns4:subParameterName> <ns4:subParameterValue>2023-10-05 13:31:52.299</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>501 EAS</ns4:subParameterName>

Request	Respond
	<ns4:subParameterValue>2023-10-05 13:31:52.765</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>401 NWEM</ns4:subParameterName> <ns4:subParameterValue>2023-10-05 13:31:52.772</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>800 PUBLIC</ns4:subParameterName> <ns4:subParameterValue>2023-10-05 13:31:52.778</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>600 CMAS</ns4:subParameterName> <ns4:subParameterValue>2023-10-05 13:31:52.786</ns4:subParameterValue> </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>10 ATT_LAB_2-111</ns4:subParameterName> <ns4:subParameterValue>2023-10-05 13:31:53.103</ns4:subParameterValue> </ns4:subParaListItem> </ns4:parameterListItem> <ns4:ResponseOperation>getMessageTime</ns4:ResponseOperation> <ns4:ResponseType>REQUEST1</ns4:ResponseType> </ns2:getResponseTypeDef> </soap:Body>

7.13 Get Request Best Practices

The following recommendations will ensure success when using IPAWS-OPEN Get Requests.

- Ensure COG certificate is valid.
- Use GetAck to test connectivity before proceeding with other Get Operations.
- Use GetMessage or GetMessageStatus operation to retrieve a single relevant message record using filters like Identifier with equalto Operator OR use multiple parameter filters like Sent (datetime) with lessthan and greaterthan operators.
  It's better to retrieve messages for only a day or two, refer to Example of getMessageByDateGreaterThanAndLessThan Operation section for example.
- Use GetMessageList to retrieve multiple relevant record set using multiple parameter filters like Sent with lessthan and greaterthan operators. It is best to retrieve messages no greater than during a day period, refer to Example of getMessageListByDateGreaterThanAndLessThan Operation section for example.

Table 90 Get Request Best Practices

Request Type	Acceptable Use	Automated or Manual Execution	Acceptable Frequency	Recommendations	Required Element(s)
getACK	Used to check connectivity, certificate, and authentication with IPAWS-OPEN. This is the ONLY request that should be used as a frequent, automated "heartbeat" or connectivity check.	Okay for automated or user initiated execution	No more frequent than once every 60 seconds		getRequestTypeDef (requestOperation)
GetCogProfile	Used to retrieve the COG Profile Authorizations (permissions) for the requesting COG. COG permissions do not change regularly, perhaps only once per year at most.	Okay for automated or user initiated execution	No more frequent than once upon logon or restart for automated execution		getRequestTypeDef (requestOperation)
getMessageList	Used to retrieve metadata for messages either sent by your COG or addressed to the COG (to retrieve COG-to-COG messages)	Okay for automated or user initiated execution	No more frequent than every 60 seconds	Establish algorithm to go back 24 hrs. the first time (upon logon or restart) then go-back 2 minutes with frequency = every 1 minutes thereafter	getRequestTypeDef (requestOperation) parameterName parameterValue comparisonOp
getMessage	Used to retrieve full CAP XML message (or messages) either sent by your COG or addressed to the COG (to retrieve COG-to-COG messages)	Okay for user initiated execution only		Create a list of alerts using getMessageList. User may manually select a particular alert. Software will execute getMessage by identifier only.	getRequestTypeDef (requestOperation) parameterName parameterValue comparisonOp
getMessageStatus	Used to retrieve the same information contained in the consolidated response. Also used to retrieve individual CMAS Channel status where an Alert was successfully disseminated to carriers.	Okay for automated or user initiated execution	No more frequent than every 60 seconds (once the response is retrieved, it is not necessary to repeat this request)	The consolidated message response is automatically sent by IPAWS. Therefore getMessageStatus returns the same, repeated information. getMessageStatus should only be used to: a) retrieve message status	

Request Type	Acceptable Use	Automated or Manual Execution	Acceptable Frequency	Recommendations	Required Element(s)
				b) refresh slow or incomplete consolidated message response data c) retrieve individual CMAS Channel status.	
getMessageTime	Used to retrieve timestamps for when a message is received by IPAWS-OPEN along with the timestamps for all status codes recorded for a message	Okay for automated or user initiated execution	No more frequent than every 60 seconds		
getCOG	Used to retrieve a list of all COGs for COG-to-COG CAP exchange.	Okay for automated or user initiated execution	Once per 24 hours, or upon logon or restart	This is only useful for COG-to-COG messaging for which you need a list of COG IDs to insert into the <addresses> element. If your software does not support COG-to-COG messaging DO NOT use the getCOG request.	
getServerInfo	Used to check IPAWS-OPEN server time to accomplish time synchronization checks and verify current IPAWS-OPEN release version.	Okay for automated or user initiated execution	Once per 24 hours, or upon logon or restart		

7.14 Get Request Testing

Use testing tools, such as SOAP UI, for testing Get operations. The following are instruction for one method.

- Download SOAPUI.
- Configure COG Certificate (ensure certificate is valid and unexpired)
- Point to proper web-services endpoint
- Select one of the Get examples as mentioned above

The logon CogId in the SOAP Header element must match the configured COG Certificate.

8. Post Services

This operation will POST (send) a NWR (NWEM), EAS, WEA, or PUBLIC alert. CAP Alert messages must be posted one at a time. A single Alert must be posted in a single transaction in order to provide origination-to-dissemination (end-to-end) tracking, response, and processing but can be sent to one or multiple channels simultaneously.

8.1 Connection Prerequisites

The following elements are required to successfully post to IPAWS-OPEN.

- Digital Certificate (unexpired)
- Authorized (enabled) COG with permissions
- Stable internet connection
- Web-service endpoint
- CAP v1.2 message containing minimum required parameters and digital signature

We recommend using an internet browser to first test the IPAWS Webservice URL and ensure the WSDL is rendering.

8.2 Digital Signature

All CAP v1.2 Alert messages to be disseminated to NWEM (NWR), EAS, CMAS, or PUBLIC via the aggregator service are required to have the CAP message digitally signed and the digital signature verified by IPAWS-OPEN. The same algorithms for digitally signed SOAP messages apply. The certificates issued for digitally signing SOAP messages for IPAWS-OPEN can be utilized for digitally signing CAP messages.

- Digital signing of CAP v1.2 Alert messages will provide message integrity and non- repudiation throughout the message flow from Alert Originator to all dissemination channels.
- To ensure successful verification of digitally signed CAP messages it is important that Canonicalization (Exclusive) form be utilized.
- Several mechanisms can invalidate the digital signature after signing of the message:
    - Any Data Change.
    - Any change to whitespace between tags, such as formatting changes associated with “pretty print.”
- Name space label changes and namespaces have no effect on an “Exclusive” signature.

If an alert has an invalid signature, error codes 208 (Alert-signature-is-not-valid) and 221 (invalid-CAPEXCHANGE-message) are generated and no further validation takes place. The following table summarizes the algorithms for digital signatures.

Table 91: Digital Signatures

XML Signature Algorithm	Requirement	References
Signature Algorithm	RSA SHA-256	http://www.w3.org/2001/04/XMLdsig-more#rsa-sha256
Canonicalization	Exclusive	http://www.w3.org/TR/XML-exc-c14n/
Digest	SHA-256	http://www.w3.org/2001/04/XMLenc#sha256
Transforms	Enveloped Signature	http://www.w3.org/2000/09/XMLdsig#enveloped-signature
Certificate	X.509	http://www.ietf.org/rfc/rfc5280.txt

In the Test environment, FEMA Generated test certificates will be provided for digitally signing SOAP Messages. The target implementation in the test and production environments utilizes WS-Security with digital signatures as shown in the following example. The following example illustrates getting an acknowledgement from the CAP service to check connectivity and authentication with IPAWS-OPEN.

CAP Acknowledgement with WS Security Header

```
<soap:Envelope
xmlns:edx="http://gov.fema.ipaws.services/IPAWS_CAPService/"
xmlns:req="http://gov.fema.ipaws.services/caprequest"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurityutility-1.0.xsd">
<soap:Header>
<wsse:Security soap:mustUnderstand="1" xmlns:wsse="http://docs.oasisopen.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
<wsse:BinarySecurityToken EncodingType="http://docs.oasisopen.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-tokenprofile-1.0#X509v3" wsu:Id="CertId-
58E2BAFBAF42B4D9B31534267482814457">
MIIGaTCCBVGgAwIBAgIQCgFCggAAAVLM2OvnGxrN8DANBgkqhkiG9w0BAQsFADBuMQswCQYDVQQGEwJVUzEfMB0GA1UECgwWSWRlbIRydXN0IFNIcnZ
pY2VzIExMQzEgMB4GA1UECwwXSWRlbIRydXN0IEdsb2JhbCBDb21tb24xHDAaBgNVBAMME1BURSBJR0MgU2VydmVyIENBIDEwHhcNMTYwMjEwMjAy
MTU0WhcNMTkwMjA3MjAyMTU0WjCBpjELMAkGA1UEBhMCVVMxEzARBgNVBAoTCkZFTUEgSVBBV1MxJTAjBgNVBAsTHE5hdGlvbmFs==</wsse:Binary
SecurityToken>
<ds:Signature Id="Signature-305" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsasha256"/>
<ds:Reference URI="#id-306">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>qVpbqaDBMcHPbDE7wTBcZ+FZTiU=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
eyjiKC++B0H4wbVXV1Me/88fRvCyS33yS6j+Gd2432+rWgC+h4Za78/6zNYXI2az4w+U3ZLrZEmwh0ZRx3VER1TQSGojpUVaYtkR+Y35Y+DRh/NRGJ2S
4Sw4k8tfyIWDXmFGrOtYqfAIXwMkTbLjQQipd/ZgIBdUrC8qGm6T6f8001a+8IihRVtBPfTHBmK9XgKBqY3lozCSQ8+aaJ5KmiAkhkxHoMHQ72Tm
82IdrEbSegOSdEMtMjBjPVIBDNW70UKRRegEhBxzRPT/T1A4hXt2n463tlSEvEQYm3OqYD35BzyQnCVlbrqjKoX0qD97DixfuzBSKLB5PIQDb+JxuQ==
</ds:SignatureValue>
<ds:KeyInfo Id="KeyId-58E2BAFBAF42B4D9B31534267482814458">
<wsse:SecurityTokenReference wsu:Id="STRId 58E2BAFBAF42B4D9B31534267482814459"><wsse:Reference URI="#CertId
58E2BAFBAF42B4D9B31534267482814457" ValueType="http://docs.oasis open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile1.0#X509v3"/></wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature></wsse:Security>
<edx:CAPHeaderTypeDef>
<edx:logonUser>ipawstester1</edx:logonUser>
<edx:logonCogId>XXX123</edx:logonCogId>
</edx:CAPHeaderTypeDef>
```

```
</soap:Header>
<soap:Body wsu:Id="id-306">
<edx:getRequestTypeDef>
<req:requestAPI>REQUEST1</req:requestAPI>
<req:requestOperation>getAck</req:requestOperation>
</edx:getRequestTypeDef>
</soap:Body>
</soap:Envelope>
```

The SOAP message above consists of a SOAP envelope that includes a SOAP header and a SOAP body. In the above example, the SOAP body contains a request message to get an acknowledgement to check connectivity and authentication from IPAWS-OPEN. The SOAP message should only contain a single SOAP header and body element.

See the two <elements> in the above example. It is important that the **Signature Reference URI# and SOAP Body Id** have the same value.

- Reference URI="id-306"
- Body wsu:Id="id-306"
- The SOAP header also provides additional information that is used by the IPAWS-OPEN Web-Services, and which is defined by CAPHeaderTypeDef in the CAPService
- WSDL

The SOAP header also provides additional information that is used by the IPAWS-OPEN Web-Services, which is defined by CAPHeaderTypeDef in the CAPService WSDL

Custom SOAP Header (Logged on User-ID and COG-ID)

```
<env:Envelope ..... >
<env:Header>
<edx:CAPHeaderTypeDef>
<edx:logonUser>ipawsopentester</edx:logonUser>
<edx:logonCogId>XXX123</edx:logonCogId>
</edx:CAPHeaderTypeDef>
. . . . .
</env:Header>
<env:Body>
. . . . .
</env:Body>
</env:Envelope>
```

The following table provides information about the additional SOAP header elements.

Table 92: SOAP Header Elements

SOAP Header Element	Description
logonUser	Logged on User Name
logonCogId	Logged on COG Identifier

The Logged on User Name and COG Identifier are saved with post and request transaction records as part of the transaction history. The Logged-on COG Identifier controls the post by identifying the originating COG Identifier associated with a post. In addition, the Distinguished Name is extracted from Certificate (<BinarySecurityToken> element) and the Common Name attribute (CN) is compared with the <logonCogId> element. The CN must contain the logonCogId.

Digital Signature with Valid Message

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><alert xmlns="urn:oasis:names:tc:emergency:cap:1.2"><identifier>WEA</identifier><sender>IPAWS-PMO</sender><sent>2023-07-25T15:50:12-04:00</sent><status>Actual</status><msgType>Alert</msgType><source>IPAWS-Tester</source><scope>Public</scope><addresses>XXX123 XXX124</addresses><code>IPAWSv1.0</code><info><language>en-US</language><category>Safety</category><event>Avalanche Warning</event><responseType>Monitor</responseType><urgency>Immediate</urgency><severity>Severe</severity><certainty>Observed</certainty><eventCode><valueName>SAME</valueName><value>RWT</value></eventCode><effective>2023-07-25T15:50:12-04:00</effective><expires>2023-07-25T16:50:12-04:00</expires><senderName>COGID, CogName, Requesting Agency</senderName><headline> 26 Sep Test Message -Test Message only Disregard please.</headline><description> 23 July Test Message. THIS is NOT an Actual Message.</instruction><web>https://grandpaham.com</web><parameter><valueName>EAS-ORG</valueName><value>CIV</value></parameter><parameter><valueName>BLOCKCHANNEL</valueName><value>NWEM</value></parameter><parameter><valueName>BLOCKCHANNEL</valueName><value>EAS</value></parameter><parameter><valueName>WEAHandling</valueName><value>WEA Test</value></parameter><parameter><valueName>CMAMtext</valueName><value>@This is a Test Imminent Threat Alert for #ipaws </value></parameter><parameter><valueName>CMAMlongtext</valueName><value>@This is a Test Imminent Threat Alert. </value></parameter><parameter><valueName>DBGFBYPASS</valueName><value>TRUE</value></parameter><resource><resourceDesc>EAS Broadcast Content</resourceDesc><mimeType>audio/x-ipaws-audio-mp3</mimeType><derefUri>XYZ</derefUri></resource><area><areaDesc>Washington</areaDesc><polygon>33.44,-111.93 33.42,-111.93 33.42,-111.90 33.45,-111.90 33.44,-111.93</polygon><circle>38.87,-77.027 100.4</circle><geocode><valueName>SAME</valueName><value>051059</value></geocode><geocode><valueName>ZIP</valueName><value>22193</value></geocode></area></info><capsig:Signature xmlns:capsig="http://www.w3.org/2000/09/xmldsig#"><capsig:SignedInfo><capsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/><capsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/><capsig:Reference URI=""><capsig:Transforms><capsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/></capsig:Transforms><capsig:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/><capsig:DigestValue>JsOsaais=</capsig:DigestValue></capsig:Reference></capsig:SignedInfo><capsig:SignatureValue>j0ICiHQM5QA37hEfG3ke3hfsZbAGApbRzw6yLt/Ox50xuYycm6Rfxyd9UraxGu3+LhlEHbx90MDhPO5MhM5KM//3HrvVKNZzdVbmdGcDclqEKkz98fqi9sepK3Exb7JFRRjTRUTOwYK4OZjKnpLUZj0n/LQc10q8S7wQ0MuzhsQm3JXZtHw+YpiOrfMOh6AOY/A==</capsig:SignatureValue><capsig:KeyInfo><capsig:X509Data><capsig:X509SubjectName>CN=IPAWSOPENXXX123,OU=A01234,OU=Devices IPAWS,OU=National Continuity Programs,O=FEMA IPAWS,C=US</capsig:X509SubjectName><capsig:X509Certificate>Y2VydGlmaWNhdGVzL3BvbGljeS9JR0MvaW5kZXguaHRtbDCB4AYIKwYBBQUHAgIwgdMMgdBUZXN0IENlcnRpZmljYXRlLiBEbyBObyBSZWx5LiBDZXJ0aWZpY2F0ZSB1c2UgcmVzdHJpY3RlZCB0byBSZWx5aW5nIFBhcnR5KHMpIGluIGFjY29yZGFuY2Ugd2l0aCBJR0MtQ1AgKHNlZSBodHRwczovL3NlY3VyZS5pZGVudHJ1c3QuY29tL2NlcnRpZmljYXRlcy9wb2xpY3kvSUdDL2luZGV4Lmh0bWwpLiBJ2T/EX/VUKOr0R5O/qS/RzTgi+wZsXaysuEva</capsig:X509Certificate></capsig:X509Data></capsig:KeyInfo></capsig:Signature></alert>

8.2.1 Digital Signature Errors

The following are probable error responses related to WS- Security and Digital Signatures for SOAP messages.

- **Certificate mismatch**: Header cogId different than certificate Common Name (CN). SOAP Message does not contain the correct value for <logonCogId> element. This is an issue with the SOAP Header <logonCogId> element not matching the CERT CN.
- **Improperly Signed**: Signed with invalid certificate
- **Expired Certificate**: Signed with invalid certificate. Check validity date range of certificate

SOAP Message Improperly Signed

```
<soap:Body>  <ns2:postCAPResponseTypeDef XMLns="http://gov.fema.ipaws.services/caprequest"
XMLns:ns2="http://gov.fema.ipaws.services/IPAWS_CAPService/" XMLns:ns3="urn:oasis:names:tc:emergency:cap:1.2"
XMLns:ns4="http://gov.fema.ipaws.services/capresponse">       <ns4:parameterListItem>
<ns4:parameterName>identifier</ns4:parameterName>          <ns4:parameterValue>TEST-IPAWS-4.1-All-Testj1234-
Naveenqzqeewe</ns4:parameterValue>    </ns4:parameterListItem><ns4:parameterListItem><ns4:subParaListItem>
<ns4:subParameterName>CHANNELNAME</ns4:subParameterName>
<ns4:subParameterValue>CAPEXCH</ns4:subParameterValue>
 </ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>STATUSITEMID</ns4:subParameterName>
<ns4:subParameterValue>208</ns4:subParameterValue>
</ns4:subParaListItem> <ns4:subParaListItem>
 <ns4:subParameterName>ERROR</ns4:subParameterName>
<ns4:subParameterValue>Y</ns4:subParameterValue>
</ns4:subParaListItem>   <ns4:subParaListItem>
  <ns4:subParameterName>STATUS</ns4:subParameterName>
 <ns4:subParameterValue>alert-signature-not-valid</ns4:subParameterValue>
 </ns4:subParaListItem><ns4:subParaListItem>
  <ns4:subParameterName>CHANNELNAME</ns4:subParameterName>          <ns4:subParameterValue>CAPEXCH</ns4:subParameterValue>
</ns4:subParaListItem> <ns4:subParaListItem>
<ns4:subParameterName>STATUSITEMID</ns4:subParameterName><ns4:subParameterValue>221</ns4:subParameterValue>
</ns4:subParaListItem><ns4:subParaListItem><ns4:subParameterName>ERROR</ns4:subParameterName><ns4:subParameterValue>Y</ns4:sub
ParameterValue></ns4:subParaListItem><ns4:subParaListItem> <ns4:subParameterName>STATUS</ns4:subParameterName>
<ns4:subParameterValue>invalid-CAPEXCHANGE-message</ns4:subParameterValue></ns4:subParaListItem> </ns4:parameterListItem>
</ns2:postCAPResponseTypeDef> </soap:Body>
```

Expired Certificate

```
<soap:Body>
    <soap:Fault>
      <faultcode XMLns:env="http://www.w3.org/2003/05/soap-envelope">env:Receiver</faultcode>
      <faultstring>SOAPMessage signed with invalid certificate.  Check validity date range of certificate.</faultstring>
    </soap:Fault>
  </soap:Body>
```

8.3 Alert-Feed Dissemination Channels

The CAP Alert-Feed dissemination channel supports all alert message types that meet the IPAWS-Profile and dissemination specific requirements. It extends the reach of all CAP Alert message types by enabling authorized alert re-broadcasters to make these messages more widely available. Message dissemination channels can include:

- **NWEM (NWR)**: Alerts that meet NWEM channel specific requirements (currently this feed is utilized by National Weather Service for NWR broadcast and WMO teletype style plain language alert message dissemination)
- **EAS**: Alerts that meet EAS channel specific requirements
- **WEA**: Alerts that meet CMAS channel specific requirements
- **PUBLIC**: Alerts that meet IPAWS-Profile requirements

IPAWS CAP Service disseminates a message as a PUBLIC Alert if it meets IPAWS CAP Profile Requirements (Code 300 “Ack” response status code). The Public Alert channel authorizations for eventCode and geocodes by default include the superset assigned to the total of all other channels for a specific COG. **The Block Dissemination (BLOCKCHANNEL) functionality is not available for Public Alerts.**

8.4 Channel Validations

The dissemination of an alert is controlled by a combination of COG-Profile authorizations and alert message content. Channel validations enable IPAWS-OPEN to check each level of processing. Below is a summary of each Channel Validation structure.

Table 93: Channel Validation Summary

Status Item Block	Dissemination Channel
10 +	CMAS Dissemination ACK and Errors
200 +	CAP Exchange ACK and Errors
300 +	Core IPAWS-Profile ACK and Errors
400 +	NWEM (NWR) Specific ACK and Errors
500 +	EAS Specific ACK and Errors
600 +	CMAS Specific ACK and Errors
800 +	Public Non-EAS Specific ACK and Errors

Table 94: Channel Validation Summary

Status Item Block	Dissemination Channel Validations
200 +	This block of validations enables CAP Exchange and requires at a minimum that the CAP message must validate against the CAP v1.2 schema. • Fundamental validations and responses are aligned for CAPEXCH as a dissemination channel. Principal IPAWS Dissemination authorizations and validations are being enforced for CAPEXCH. • CAPEXCH validations and responses. A few examples include: o Check for valid value in <area> <circle> and <polygon> elements. o Check for valid value in <area> <geocode> element. o Check that the <area> <circle> or <polygon> element is inside authorized boundary for the Alert Originator COG. o Check that Alert Originator COG not authorized for <eventCode> or <geocode> elements in the CAP Alert message. ▪ If a COG is authorized for <eventCode> or <geocode> element in any dissemination channel it will be authorized for that code for CAPEXCH. ▪ Validations for <eventCode> and <geocode> elements will only apply if <valueName> is equal to “SAME”. ▪ This will allow for other <eventCode> and <geocode> schemes that are only meant for CAP Exchange and not IPAWS Dissemination. **A CAP Alert and Update intended for IPAWS dissemination requires at a minimum an <info> element.** • A CAP Alert is identified for IPAWS dissemination is identified by an <alert><code> element with value equal to “IPAWSv1.0”

Status Item Block	Dissemination Channel Validations
	• If an IPAWS Alert or Update is posted without an <info> element the following error is returned: ○ 221 – invalid-CAPEXCHANGE-message ○ If a 221-response code is raised, it prevents the alert from mistakenly being retrieved and redistributed as a "valid" IPAWS alert. • A CAP Alert and Update not identified for IPAWS dissemination can be posted without an <info> block element.
300 +	This block of validations enables further processing for message dissemination to NWEM (NWR), EAS, CMAS, PUBLIC messages. • Initial Step: Verification of CAP message Digital Signature. If this step fails, no further dissemination processing is executed. • If the digital signature is valid, then core IPAWS-Profile checks that are not specific to dissemination channels, are completed. • **NOTE**: If message passes Block 300 Validations, then Block 400, 500, 600, and 800 validations are completed.
400 +	This block of validations includes NWEM (NWR) specific checks needed for NOAA Radio dissemination. These validations include EAS validations because NWR messages are sometimes picked up for re-transmission by downstream EAS broadcasters.
500 +	This block of validations includes those IPAWS-Profile checks related to EAS, and other EAS specific checks.
600 +	This block of validations includes those IPAWS-Profile checks related to CMAS, and other CMAS specific checks.
800 +	This block of validations includes those IPAWS-Profile checks related to PUBLIC Alerts, and other PUBLIC Alert specific checks.

In addition to these block validations there are also the following checks against a posting COG's Profile:

- The following check is accomplished prior to Block 200 Validations:
  - **COG Enabled** – Enables a COG for the CAP Aggregator service to post messages or submit message or data requests. (Check is done prior to the Block 200 Validations).
- The following checks are accomplished as part of Block 300 Validations:
  - **Broadcast Message Authorized** – Enables a COG to post an Alert Message and have it posted to the Broadcast COG (COG-ID = 0, ALL IPAWS-Services COGs).
  - **Allowed Event Codes** – Authorizes the posting of an Alert by a COG to a specific set of Event Codes by dissemination channel: NWEM (NWR), EAS, CMAS, PUBLIC.
  - **Allowed Geocodes** - Authorizes the posting of an Alert by a COG to a specific set of Geocodes by dissemination channel: NWEM (NWR), EAS, CMAS, PUBLIC.

Refer to Status Item Response Details section for additional detail.

8.5 Block Dissemination Channels

IPAWS-OPEN provides the capability to restrict dissemination by channel (i.e., NWEM, EAS, and CMAS). If no channels are blocked, the dissemination of alert messages is controlled by a combination of COG-Profile authorizations and alert message content. This capability enables Alert Originators to explicitly control distribution of alerts, and prevent unintended transmission to specific dissemination channels.

If all Channels are blocked the IPAWS-Profile validations will not be executed and CAP Alert will be available only for CAP Exchange and PUBLIC. When a channel is blocked no validation is accomplished for that dissemination channel. A message status response code will be returned identifying that the message was not sent to a channel.

A <parameter> element with <valueName> element equal to **BLOCKCHANNEL** and one of the following three values will restrict a message dissemination channel:

- NWEM
- EAS
- CMAS

If an Alert meets all COG Profile validations it will be disseminated at a minimum as a PUBLIC Alert.

A single channel or multiple channels can be blocked as shown in the following examples.

Example: Single Block Channel for CMAS

```
<parameter>
<valueName>BLOCKCHANNEL</valueName>
<value>CMAS</value>
</parameter>
```

Example: Multiple Block Channels for CMAS and EAS

```
<parameter>
<valueName>BLOCKCHANNEL</valueName>
<value>CMAS</value>
</parameter>
<parameter>
<valueName>BLOCKCHANNEL</valueName>
<value>EAS</value>
</parameter>
```

The following status codes are returned in posted message status when blocking channels:

- **401** - message-not-disseminated-as-NWEM
- **501** - message-not-disseminated-as-EAS
- **601** - message-not-disseminated-as-CMAS

8.6 Element Mapping to Dissemination Channels

The following set of tables correlate each element for CAP v1.2 Alert messages with requirements for each dissemination channel.

Messages passing the Core IPAWS-Profile validations are published to the PUBLIC channel.

<ALERT> ELEMENT MAPPING TO DISSEMINATION CHANNELS

Table 95: <Alert> Element Mapping to Dissemination Channels

CAP v1.2 <alert> Element						
Element Name	CAP Exchange	IPAWS Profile	NWR/NWEM	EAS	CMAS	Comments

identifier	**Required**
	Check for Uniqueness
		Required
		No Additional Requirement
			Required
			No Additional Requirement
				Required
				No Additional Requirement
					Required	• **General:**
						  • Element <identifier> is required to be unique within a submitting COG.
						  • Element <identifier> must not include spaces, commas, or restricted characters (< and &)
						• **CMAS:**
						  • Included in transformation to CMAC message.

CAP v1.2 <alert> Element						
Element Name	CAP Exchange	IPAWS Profile	NWR/NWEM	EAS	CMAS	Comments
sender	**Required** Check for Restricted Characters	**Required** No Additional Requirement	**Required** No Additional Requirement	**Required** No Additional Requirement	**Required**	**General:** • Element <sender> must not include spaces, commas, or restricted characters (< and &). • **CMAS:** • Included in transformation to CMAC message.
sent	**Required** Check for valid date format	**Required** Additional Check	**Required** Additional Check	**Required** No Additional Requirement	**Required**	• **IPAWS-Profile Additional Checks**: • <expires> greater than <sent> • <sent> within +/- 5 minutes of current time. • **CMAS:** • Included in transformation to CMAC message.
status	**Required** Allowed Values “Actual” “Exercise” “System” “Test” “Draft”	**Required** Allowed Value “Actual”	**Required** Allowed Value “Actual”	**Required** Allowed Value “Actual”	**Required** Allowed Value “Actual”	• **General:** • The code denoting the appropriate handling of the alert message. • **CMAS:** • Included in transformation to CMAC message.
msgType	**Required** Allowed Values “Alert” “Update” “Cancel” “Ack” “Error”	**Required** Allowed Value “Alert” “Update” “Cancel”	**Required** Allowed Value “Alert” “Update” “Cancel”	**Required** Allowed Value “Alert” “Update” “Cancel”	**Required** Allowed Value “Alert” “Update” “Cancel”	• **General:** • For <msgType> element equal to “Update”, “Cancel”, “Ack” and “Error” also calls for a proper value in the <references> element. • **NWR (NWEM):** • Only “Alert”, “Update”, “Cancel” messages are processed by NWS for NWR. • **CMAS:** • Included in transformation to CMAC message.
source	**Optional**	**Optional**	**Optional**	**Optional**	**Optional**	• **General:** • Alerting software vendors should include their software name and version number in the value for the <source> CAP element. The purpose is to aid in troubleshooting and maintain compliance with FEMA policies. • The <source> value may be publicly presented as a human readable "signature line” in some delivery systems. • If <source> is missing or empty the advisory

CAP v1.2 <alert> Element						
Element Name	CAP Exchange	IPAWS Profile	NWR/NWEM	EAS	CMAS	Comments
						code 225 is presented **NWR (NWEM):** • The <source> element, when provided, is used by the NWS NWR broadcast capability in combination with the CAP <identifier> element to "sign" the transformed NWR (NWEM) message rendered from the CAP.
restriction	**Conditional**	**Conditional** No additional Requirement	**Conditional** No Additional Requirement	**Conditional** No Additional Requirement	**Not-Used**	• **General:** o If the value of the <scope> element is equal to "Restricted", a value in the <restriction> element is required. This is not enforced by CAP v1.2 XML schema or in Block 200 (CAP exchange) validations.
addresses	**Conditional**	**Conditional** No Additional Requirement	**Conditional** No Additional Requirement	**Conditional** No Additional Requirement	**Not-Used**	• **General:** Required when <scope> is "Private", optional when <scope> is "Public" or "Restricted". o Each recipient is identified by an identifier or an address. o Multiple space-delimited addresses MAY be included. Addresses including whitespace MUST be enclosed in double-quotes. • **CAP Exchange:** The <addresses> element is needed to specify other COG-ID(s) for posting of a CAP v1.2 Alert. o Include in the <addresses> a space delimited string of all COG-ID(s) that the message will be posted to. For example: <addresses>120001 120002</addresses>
code	**Optional**	**Required** Allowed value "IPAWSv1.0"	**Required** Allowed value "IPAWSv1.0"	**Required** Allowed value "IPAWSv1.0"	**Required** Allowed value "IPAWSv1.0"	• **General:** o Any user-defined flag or special code used to flag the alert message for special handling. Multiple instances MAY occur. o Checks are not conducted for duplicate <code> element values. All entries will be persisted.

CAP v1.2 <alert> Element						
Element Name	CAP Exchange	IPAWS Profile	NWR/NWEM	EAS	CMAS	Comments
						• **Core IPAWS-Profile:** Requires one <code> value equal to "IPAWSv1.0" or the Alert will not be disseminated to NWR, EAS or CMAS.
note	**Optional**	**Optional**	**Optional**	**Optional**	**Optional**	• **General:** The text describing the purpose or significance of the alert message • **CMAS:** o Used in the case when alert originator chooses to bypass geofencing (a method of geotargeting) to streamline processing of polygons and circles.
references	**Optional**	**Required** For "Update" or "Cancel" Message	**Required** For "Update" or "Cancel" Message	**Required** For "Update" or "Cancel" Message	**Required** For "Update" or "Cancel" Message	• **General:** The group listing identifying earlier message(s) referenced by the alert message. o It is in the form "sender,identifier,sent" of an earlier CAP message. If multiple messages are referenced, they are separated by whitespace. • **Core IPAWS-Profile Check:** o An error will be raised if element <reference> for an "Update" or "Cancel" Alert message is empty/missing or the format is not valid • **EAS:** o An "Update" or "Cancel" message that references a message already on the CAP Alert-Feed causes removal of the original message from the CAP Alert-Feed. • **EAS & CMAS:** o Both EAS and CMAS do not require an <info> element for a "Cancel" message. The latest <references> element value is used to obtain applicable <geocode> codes and <eventCode> code from the latest referenced "Alert" or "Update". o For an "Update" Alert, if the referenced alert is not found in IPAWS-OPEN, it will still be disseminated to CMAS. • **CMAS:**

CAP v1.2 <alert> Element						
Element Name	CAP Exchange	IPAWS Profile	NWR/NWEM	EAS	CMAS	Comments
						Included in transformation to CMAC
incidents	**Optional**	**Optional**	**Optional**	**Optional**	**Optional**	**General:** The group listing naming the referent incident(s) of the alert message o Used to collate multiple messages referring to different aspects of the same incident. If multiple incident identifiers are referenced, they are separated by whitespace. Incident names including whitespace are surrounded by double-quotes.
scope	**Required** Allowed Values "Public" "Private" "Restricted"	**Required** Allowed Values "Public" "Private" "Restricted"	**Required**	**Required**	**Required**	**General:** Can be "Private", "Public" or "Restricted"

Core Requirements for CAP v1.2 <info> Element:

- **CAP Exchange:** Multiple occurrences are permitted within a single <alert>. In addition to the specified sub-elements, MAY contain one or more <resource> blocks and/or one or more <area> blocks.
- **Core IPAWS-Profile:**
  - All <info> blocks in a single alert MUST relate to a single incident or update, with the same <category> and <eventCode> values.
  - An <info> block SHOULD contain only one <eventCode> with a <valueName> of "SAME", and which must be authorized in the Alert Originator's COG Profile.
  - All <info> blocks SHOULD be appropriate for immediate public release.
  - Multiple <info> blocks may be used to deliver content in different languages.
  - Exchange partners may elect to process only the first <info> block encountered in a language they support.
  - Other <eventCode> elements may also be present.
- **EAS and NWR (NWEM):** Follows core IPAWS-Profile requirements.

<INFO> ELEMENT MAPPING TO DISSEMINATION CHANNELS

- <info> is a child of <alert>. Example: <alert> <info></info></alert>.

- <info> can use the following child elements.

Table 96: CAPv1.2 <info> Element

CAP v1.2 <info> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWR/NWEM	EAS	CMAS	Comments
language	Optional	Optional	Optional	Optional	Required Allowed Values “en-US” “es-US”	• General: ○ Content of the <language> tag is not validated to match RFC 3066 by IPAWS-OPEN at the 200 (CAP Exchange) or the CAP schema. It can be any string, or empty, and the message will post successfully. The appropriate RFC 3066 language code should be used when creating new CMAS Messages. It will be validated for content, where needed by IPAWS push distribution channels. ○ If not present, an implicit default value of “en- US” Is assumed. ○ A null value in this element Is considered equivalent to “en-US.” • Core IPAWS-Profile: ○ Multiple <info> blocks may be used to deliver content in different languages. ○ Exchange partners may elect to process only the first <info> block encountered in a language they support • EAS: Follows Core IPAWS-Profile Requirements • CMAS: ○ See WEA Requirements section for additional detail
category	Required See Table for Allowed Values	Required No Additional Requirement	Required No Additional Requirement	Required No Additional Requirement	Required	• Core IPAWS-Profile: ○ All <info> blocks in a single alert MUST relate to a single incident or update, with the same <category> values. • CMAS: • Included in transformation to CMAC message.
event	Required	Required No Additional Requirement	Required No Additional Requirement	Required No Additional Requirement	Required	• General: • o The text denotes the type of the subject event for the alert message.

CAP v1.2 <info> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWR/NWEM	EAS	CMAS	Comments
responseType	**Optional** See Table for Allowed Values	**Optional**	**Optional**	**Optional**	**Optional** See Table for Allowed Values	• **General:** ○ The code denoting the type of action recommended for the target audience. • **CMAS:** ○ **NOTE:** The CAP v1.2 standards allow that multiple instances MAY occur within an <info> block. ○ If the <info> block contains multiple <responseType> elements only the first <responseType> element will be processed for CMAS
urgency	**Required** Allowed Values “Immediate” “Expected” “Future” “Past” “Unknown”	**Required**	**Required**	**Required**	**Required**	• **General:** ○ The code denoting the urgency of the subject event of the alert message. • **CMAS:** • See WEA **Requirements section** for additional details.
severity	**Required** Allowed Values “Extreme” “Severe” “Moderate” “Minor” “Unknown”	**Required**	**Required**	**Required**	**Required**	• **General:** ○ The code denoting the severity of the subject event of the alert message. • **CMAS:** • See WEA **Requirements section** for additional details.
certainty	**Required** Allowed Values “Observed” “Likely” “Possible” “Unlikely” “Unknown”	**Required**	**Required**	**Required**	**Required**	• **General:** ○ The code denoting the certainty of the subject event of the alert message. • **CMAS:** • See WEA **Requirements section** for additional details.
audience	**Optional**	**Optional**	**Optional**	**Optional**	**Not-Used**	• **General:** The text describing the intended audience of the alert message.
eventCode	**Optional** See Table for Values	**Required** See Table for Values	**Required** See Table for Values	**Required** See Table for Values	**Required** See Table for Values	• **General:** ○ A system-specific code identifying the event type of the alert message.

CAP v1.2 <info> Element						
Element Name	**CAP Exchange**	**IPAWS-Profile**	**NWR/NWEM**	**EAS**	**CMAS**	**Comments**
	Associated with SAME Code Domain	Associated with SAME Code Domain	Associated with SAME Code Domain	Associated with SAME Code Domain	Associated with SAME Code Domain	○ In the form: <eventCode> <valueName>valueName</valueName> <value>value</value> </eventCode> ▪ The content of "valueName" is a user-assigned string designating the domain of the code, and the content of "value" is a string denoting the value itself. ▪ Values of "valueName" that are acronyms SHOULD be represented in all capital letters without periods. ▪ Multiple instances MAY occur within an <info> block. • **Core IPAWS-Profile:** ○ All <info> blocks in a single alert MUST relate to a single incident or update, with the same <category> values. ○ An <info> block SHOULD contain only one <eventCode><value> with a <valueName> of "SAME", and which must be authorized in the Alert Originator's COG-Profile. • **EAS** All values for EAS Event Code are passed, even if the Event Code is not shown in FCC Part 11.31, as long as the value is a three- letter code. The Event Code still needs to be authorized and assigned in the COG Profile.
effective	**Optional** Check for valid date format	**Optional**	**Optional**	**Optional**	**Not Used**	• **General**: ○ The effective time of the information of the alert message. • **IPAWS-Profile:** ○ Ignored if present. Alerts Are effective upon issuance. • **NWR (NWEM) Additional Check:** ○ The <effective> element must be set to the same time as the <sent> element.

CAP v1.2 <info> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWR/NWEM	EAS	CMAS	Comments
onset	Optional Check for valid date format	Optional	Optional	Optional	Not Used	• General: o The expected time of the beginning of the subject event of the alert message. • IPAWS-Profile: • Ignored if present. Alerts Are effective upon issuance.
expires	Optional Check for valid date format	Required Additional Check	Required No Additional Requirement	Required No Additional Requirement	Required No Additional Requirement	• IPAWS-Profile Additional Checks: o Check that <expires> greater than <sent> • EAS: o Check that <expires> does not exceed <sent> by 99.5 hours. • CMAS: • Check that <expires> does not exceed <sent> by 24 hours.
senderName	Optional	Optional	Recommended	Optional	Optional	• General: o The text naming the originator of the alert message. • NWEM (NWR): o The senderName element is used to identify the agency or authority issuing the alert. The format is: <COGID>,<CogName>,<Requesting Agency> • Requesting Agency is used when the message is sent on behalf of another entity (e.g., State Agency sending on behalf of a county agency)
headline	Recommended	Recommended	Recommended	Recommended	Not-Used	• General: o The text headline of the alert message. • NWEM (NWR): • The <headline> element is broadcast by NWS systems at beginning of the transformed NWR message if present.
description	Optional	Optional	Required	Required	Not-Used	• General: o The text describing the subject event of the alert message. o IPAWS-OPEN maintains carriage returns and line feeds for all XSD:String fields after the first character and before the last character. All whitespace to include

CAP v1.2 <info> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWR/NWEM	EAS	CMAS	Comments
						carriage returns and line feeds is trimmed from the beginning and end of the content of XSD:String values. • **IPAWS-Profile:** o Messages SHOULD have meaningful values for the <description>. • The content in <description> may be truncated. It is recommended that essential information be addressed first.
instruction	Optional	Optional	Optional	Optional	Not-Used	• **General:** o The text describing the recommended action to be taken by recipients of the alert message. • **IPAWS-Profile:** o Messages SHOULD have meaningful values for the <instruction>. • The content in <instruction> may be truncated. It is recommended that essential information be addressed first.
web	Optional	Optional	Optional	Optional	Not-Used	• **General:** o The identifier of the hyperlink associating additional information with the alert message. • Validated to be an XSD token (string with no spaces). No further validation is attempted.
contact	Optional	Optional	Optional	Optional	Not-Used	• **General:** • The text describing the contact for follow-up and confirmation of the alert message.
EAS-ORG	Optional	Optional	**Required** Required For "EAS-ORG"	**Required** Required For "EAS-ORG"	Optional	• **General:** ▪ Values of "valueName" that are acronyms SHOULD be represented in all capital letters without periods. ▪ Multiple instances MAY occur within an <info> block. • **Core IPAWS-Profile:** o Messages intended for EAS or NWR dissemination must include an instance of <parameter> element with a <valueName> element equal to "EAS-ORG" and a <value>

CAP v1.2 <info> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWR/NWEM	EAS	CMAS	Comments
						of the originator's SAME organization code. ○ The originator's SAME organization code must be one of the following three-letter codes: ▪ **PEP** – Primary Entry Point System ▪ **EAS** – Broadcast station or cable system ▪ **WXR** – National Weather Service ▪ **CIV** – Civil authorities • **EAS and NWR:** ○ See Core IPAWS Profile comments for instance of <parameter> element with a <valueName> element equal to "EAS-ORG."

Allowed Values for <info> <category> element

Table 97: CAPv1.2 Allowed Values <info> <category>

CAP v1.2 Allowed Values <info> <category> Element				
Code Value	Code Description	NWEM	EAS	CMAS
"Geo"	Geophysical (inc. landslide)	**N/A**	Y	Y
"Met"	Meteorological (inc. flood)	**N/A**	Y	Y
"Safety"	General emergency and public safety	**N/A**	Y	Y
"Security"	Law enforcement, military, homeland, and local/private security	**N/A**	Y	Y
"Rescue"	Rescue and recovery	**N/A**	Y	Y
"Fire"	Fire suppression and rescue	**N/A**	Y	Y
"Health"	Medical and public health	**N/A**	Y	Y
"Env"	Pollution and other environmental	**N/A**	Y	Y
"Transport"	Public and private transportation	**N/A**	Y	Y
"Infra"	Utility, telecommunication, other non-transport infrastructure	**N/A**	Y	Y

CAP v1.2 Allowed Values <info> <category> Element				
Code Value	Code Description	NWEM	EAS	CMAS
"CBRNE"	Chemical, Biological, Radiological, Nuclear or High-Yield Explosive threat or attack	**N/A**	Y	Y
"Other"	Other events	**N/A**	Y	Y

Allowed Values for <info> <responseType> element

Table 98: CAP v1.2 Allowed Values <info> <responseType> Element

CAP v1.2 Allowed Values <info> <responseType> Element				
Code Value	Code Description	NWEM	EAS	CMAS
"Shelter"	Take shelter in place or per CAP Alert <instruction> element	**N/A**	Y	Y
"Evacuate"	Relocate as instructed in the CAP Alert <instruction> element	**N/A**	Y	Y
"Prepare"	Make preparations per the CAP Alert <instruction> element	**N/A**	Y	Y
"Execute"	Execute a pre-planned activity identified in CAP Alert <instruction> element	**N/A**	Y	Y
"Avoid"	Avoid the subject event as per the CAP Alert <instruction> element	**N/A**	Y	Y
"Monitor"	Attend to information sources as described in CAP Alert <instruction> element	**N/A**	Y	Y
"Assess"	Evaluate the information in this message. (This value SHOULD NOT be used in public warning applications.)	**N/A**	Y	N
"AllClear"	The subject event no longer poses a threat or concern, and any follow-on action is described in CAP Alert <instruction> element	**N/A**	Y	N
"None"	No action recommended	**N/A**	Y	N

Allowed Values for <info> <eventCode> Element (SAME Code Domain)

Click here to navigate back to <info> section.

Table 99: CAP v1.2 Allowed Values for <info> <eventCode> Element

CAP v1.2 Allowed Values for <info> <eventCode> Element Code Values - Specific Area Message Encoding (SAME) Code Domain				
Code Value	**Code Description**	**NWEM**	**EAS**	**CMAS**
"ADR"	Administrative Message	Y	Y	**N**
"AVA"	Avalanche Watch	Y	Y	**N**
"AVW"	Avalanche Warning	Y	Y	Y
"BLU"	Law Enforcement Blue Alert	Y	Y	Y
"CAE"	Child Abduction Emergency	Y	Y	Y
"CDW"	Civil Danger Warning	Y	Y	Y
"CEM"	Civil Emergency Message	Y	Y	Y
"DMO"	Practice/Demo Warning	Y	Y	Y
"EQW"	Earthquake Warning	Y	Y	Y
"EVI"	Evacuation Immediate	Y	Y	Y
"FRW"	Fire Warning	Y	Y	Y
"HMW"	Hazardous Materials Warning	Y	Y	Y
"LAE"	Local Area Emergency	Y	Y	Y
"LEW"	Law Enforcement Warning	Y	Y	Y
MEP	Missing and Endangered Person	Y	Y	Y
"NUW"	Nuclear Power Plant Warning	Y	Y	Y
"RHW"	Radiological Hazard Warning	Y	Y	Y
"RMT"	Required Monthly Test	**N**	Y	Y
"RWT"	Required Weekly Test	**N**	Y	Y
"SPW"	Shelter in Place Warning	Y	Y	Y
"TOE"	911 Telephone Outage Emergency	Y	Y	Y
"VOW"	Volcano Warning	Y	Y	Y

Table 100: FEMA Use Only Values for <info> <eventCode> Element

FEMA Use Only Values for <info> <eventCode> Element Code Values - Specific Area Message Encoding (SAME) Code Domain				
Code Value	**Code Description**	**NWEM**	**EAS**	**CMAS**
"EAN"	Presidential Alert	**N**	Y	Y
"NPT"	National Periodic Test	**N**	Y	Y

<resource> Element Mapping to Dissemination Channels

Table 101: CAP v1.2 <resource> Element

CAP v1.2 <resource> Element						
Element Name	**CAP Exchange**	**IPAWS-Profile**	**NWEM**	**EAS**	**CMAS**	**Comments**
resourceDesc	**Conditional**	**Conditional**	**Not-Used**	**Conditional**	**Not-Used**	• **General:** ○ The text describing the type and content of the resource file. • **EAS:** ○ A value of "EAS Broadcast Content" Is used to indicate that the elements of a <resource> block are intended for EAS broadcast. ○ EAS broadcast audio and video content SHOULD match the message's textual content. ○ The value of <resourceDesc> is case sensitive. ○ Content is identified by the <mimeType>.
mimeType	**Conditional**	**Conditional**	**Not-Used**	**Conditional**	**Not-Used**	• **General:** ○ The identifier of the MIME content type and sub-type describing the resource file. ○ The value for <mimeType> element is not validated beyond CAP v1.2 XML schema requirements. Developers using the <mimeType> field must assure that exchange partners will understand the value that is used. • **EAS:** ○ The <mimeType> element must be equal to one of the following defined mime-type

CAP v1.2 <resource> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWEM	EAS	CMAS	Comments
						phrases. This element is used to identify broadcast content for delivery to the public. ▪ “audio/x-ipaws-audio-mp3" ▪ "audio/x-ipaws-streaming-audio" ▪ "video/x-ipaws-video" ▪ "video/x-ipaws-streaming-video" o If broadcast content exceeds two minutes playing time it may be truncated by exchange partners except for Presidential Messages.
size	Optional	Optional	Not-Used	Optional	Not-Used	• General: o The integer indicating the size of the resource file. o Approximate size of the resource file in bytes. o For <uri> based resources, <size> SHOULD be included if available.
uri	Optional	Optional	Not-Used	Optional	Not-Used	• General: o The identifier of the hyperlink for the resource file. o A full absolute URI, typically a Uniform Resource Locator that can be used to retrieve the resource over the Internet OR o a relative URI to name the content of a <derefUri> element if one is present in this resource block. o Validated to be an XSD token (string with no spaces). No further validation is attempted.
derefUri	Conditional	Conditional	Not-Used	Conditional	Not-Used	• General: o The base-64 encoded data content of the resource file. o The value for <derefUri> element is not validated beyond CAP v1.2 XML schema requirements. o Maximum file size for a CAP message including base-64 encoded content should not exceed 1.5 MB. o NOTE: The CAP Specification calls for Base64 encoding but schema is XSD:String. Therefore, any extraneous content in this string will be maintained, which could cause an issue for some decoders. o MAY be used either with or instead of the <uri> element in messages transmitted over

CAP v1.2 <resource> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWEM	EAS	CMAS	Comments
						one-way (e.g., broadcast) data links where retrieval of a resource via a URI is not feasible. ○ See CAP v1.2 standard for additional information.
digest	Optional	Optional	Not-Used	Optional	Not-Used	• General: ○ The code representing the digital digest ("hash") computed from the resource file. ○ Calculated using the Secure Hash Algorithm (SHA-1) per [FIPS 180-2].

<area> Element Mapping to Dissemination Channels

- General:
  - Multiple occurrences are permitted, therefore the target area for the <info> block is the union of all the included <area> blocks.
  - MAY contain one or multiple instances of <polygon>, <circle> or <geocode>.
  - If multiple <polygon>, <circle> or <geocode> elements are included, the area described by this <area> block is represented by the union of all the included elements.

- IPAWS-Profile:
  - At least one <area> block MUST be present to meet IPAWS-Profile requirements for NWEM (NWR), EAS, CMAS, and PUBLIC.
  - At least one instance of <geocode> with a <valueName> of "SAME" and a value of a SAME 6-digit location (extended FIPS) SHOULD be used.
  - The more precise geospatial representations of the area, <polygon> and <circle>, SHOULD also be used whenever possible.
  - A SAME value of "000000" refers to ALL United States, Territories and Marine Zones.
  - If a SAME-based <geocode> is not present, IPAWS exchange partners unable to use a geospatial representation may ignore the message. Authorization for dissemination based on geocodes is incorporated in the COG Profile.

- EAS Alert Geocodes:
  - As part of the EAS broadcast protocol currently up to 31 geocodes are supported. They are processed in the order that they are encountered in the CAP Alert message. If an Alert message with more than 31 geocodes is posted for dissemination to the EAS channel only the first 31 geocodes in the Alert message will be processed by the EAS broadcaster.
  - An advisory response code will be returned if a EAS CAP Alert contains more than the geocode maximum limit for EAS. Response Code: 521 (exceeded-EAS-geocode-max-limit).

The alert will still be disseminated as EAS Alert but only the first geocodes up to the maximum limit will be processed.

- CAP allows multiple Area Blocks, but one area block is processed by EAS. The presence of more than one area block will not cause the message to be rejected or ignored.

- **Polygon and Circle Validations:**
  - An additional validation ensures that Alert <area> block <circle> and <polygon> elements are not outside the shape representing a COG's authorized area. A COG's authorized area includes the geographic area represented by FIPS assigned in the COG Profile.
- **Validation of CAP Alert "SAME" FIPS geocodes with a leading non-zero subdivision code.**
  - Validation allows a "SAME" 6-Digit FIPS geocode that has either a leading zero digit indicating no subdivision or the 1/9th area sub-division code. A "SAME" geocode maps to a format defined as PSSCCC:
    - **P** = County Subdivision 0-9
    - **0** = all, 1 = NW, 2 = N, 3 = NE, 4 = W, etc.
    - **SS** = State designation
    - **CCC** = County designation

Allowed values for <area> Element.

Table 102: <area> Element Mapping to Dissemination Channel

CAP v1.2 <area> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWEM	EAS	CMAS	Comments
areaDesc	Required	Required	Required	Required	Required	• **General:** o The text describing the affected area of the alert message. • **CMAS:** This element is required for a successful CMAS and will raise an error if not present.
polygon	Optional	Optional	Not-Used	Optional	Optional	• **General:** o The paired values of points defining a polygon that delineates the affected area of the alert message. o Code Values: The geographic polygon is represented by a whitespace-delimited list of [WGS 84] coordinate pairs. o A minimum of 4 coordinate pairs MUST be present (which defines a triangle) and the first and last pairs of coordinates MUST be the same. This is validated in the Aggregator service. o For example: <polygon>41.066,-73.8389 41.0669,-73.8012 41.0251,-73.7842 41.0208,-73.836 41.0516,-73.8632 41.066,-73.8389</polygon> o Multiple instances MAY occur within an <area> block. • **IPAWS-Profile** o The more precise geospatial representations of the area, <polygon> and <circle>, SHOULD also be used whenever possible. o Additional Check is accomplished to verify that the first and last pairs of coordinates are the same. • **CMAS:** o Should not exceed 10 shapes (polygons and/or circles) or over 100 total pair of coordinates across all included polygons. o Also, the decimal precision of Alert polygon shapes should not exceed 4 decimal point precision.
circle	Optional	Optional	Not-Used	Optional	Optional	• **General:** o The paired values of a point and radius delineating

CAP v1.2 <area> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWEM	EAS	CMAS	Comments
						the affected area of the alert message. ○ The value for <circle> element is not validated beyond CAP v1.2 XML schema requirements. ○ Code Values: The circular area is represented by a central point given as a [WGS 84] coordinate pair followed by a space character and a radius value in kilometers. ○ For example: <circle>41.066,-73.8389 1</circle> • Multiple instances MAY occur within an <area> block. • **IPAWS-Profile**: ○ The more precise geospatial representations of the area, <polygon> and <circle>, SHOULD also be used whenever possible. • **CMAS:** ○ Should not exceed 10 shapes (polygons and/or circles) or over 100 total pair of coordinates across all included polygons. • Also, the decimal precision of Alert circle shapes should not exceed 4 decimal point precision.
geocode	**Optional**	**Required**	**Required**	**Required**	**Required**	• **General:** ○ The geographic code delineating the affected area of the alert message. ○ Any geographically-based code to describe a message target area, in the form: <geocode> <valueName>valueName</valueName> <value>value</value> </geocode> ○ The content of "valueName" is a user-assigned string designating the domain of the code, and the content of "value" is a string (which may represent a number) denoting the value itself. ○ Values of "valueName" that are acronyms SHOULD be represented in all capital letters without periods (e.g., SAME, FIPS, ZIP). ○ Multiple instances MAY occur within an <area> block. • **IPAWS-Profile:** ○ At least one <area> block MUST be present to meet

CAP v1.2 <area> Element						
Element Name	CAP Exchange	IPAWS-Profile	NWEM	EAS	CMAS	Comments
						IPAWS-Profile requirements and dissemination to NWEM (NWR), EAS, and CMAS. • At least one instance of <geocode> with a <valueName> of "SAME" and a value of a SAME 6-digit location (extended FIPS) SHOULD be used.
altitude	Optional	Optional	Not-Used	Optional	Not-Used	• General: ○ The specific or minimum altitude of the affected area of the alert message. ○ If used with the <ceiling> element this value is the lower limit of a range. Otherwise, this value specifies a specific altitude. ○ The altitude measure is in feet above mean sea level per the [WGS 84] datum. • See NOTE-1
ceiling	Conditional	Conditional	Not-Used	Conditional	Not-Used	• General: ○ The maximum altitude of the affected area of the alert message. ○ MUST NOT be used except in combination with the <altitude> element. This requirement is not enforced; therefore, participating external systems should be able to handle a <ceiling> without an accompanying <altitude>. ○ The ceiling measure is in feet above mean sea level per the [WGS 84] datum. • See NOTE-1

NOTE-1: The CAP Specification requires type decimal for <altitude> and <ceiling> elements. This currently is not being enforced by the by Schema Validation or the web-service. The CAP Alert message is being persisted but without the <altitude> and <ceiling> elements if they contain non-decimal or non-integer values.

Core Requirements for CAP v1.2 <resource> Element:

- **General:**
  - Refers to an additional file with supplemental information related to a <info> element (e.g., an image or audio file).
  - Multiple instances MAY occur within an <info> block.
- **CMAS:**
  - Not used in CAP transformation for CMAS.

8.7 Alert <urgency>, <severity>, <certainty> Elements

All WEA (CMAC) alerts must contain one of the top two values for <urgency>, <severity>, and <certainty> (USC) elements. Allowed values include:

- **urgency**: Immediate or Expected
- **Severity**: Extreme or Severe
- **Certainty**: Observed or Likely

8.8 CAP-Reference Element Validation

CAP <reference> element validations address all message types that require a <reference> element. The validations address a range of error conditions including invalid <reference> element structure, missing or empty <reference> elements, and where the referenced Alert is expired or was not posted to IPAWS-OPEN.

The following table identifies the <reference> element error conditions for specific message types and describes the response code and how the Alert messages will be exchanged or disseminated.

Table 103: CAP Reference Element Validation

Message Type	Error Condition	Response Status Codes
CAP Alert “Update”, “Cancel”, “Ack” or “Error” Message with no <reference> element	Element <reference> missing in Alert message.	207 - reference-element-missing *See NOTE-1*
CAP Alert “Update”, “Cancel”, “Ack” or “Error” Message with empty <reference> element	Element <reference> empty in Alert message.	207 - reference-element-missing *See NOTE-1*
CAP Alert “Alert”, “Update”, “Cancel”, “Ack” or “Error” Message without the proper<reference> element structure	Element <reference> structure does not match the structure: Format must be identifier,sender,sent (no spaces)	224 - reference-element-invalid *See NOTE-2 and See NOTE-4*
CAP Alert “Cancel” Message	Referenced CAP Alert message is expired	309 – message-references-expired-alert *See NOTE-2*
CAP Alert “Update”, “Cancel” “Ack” and “Error” Message	Referenced CAP Alert message cannot be found in IPAWS-OPEN.	218 – referenced-alert-not-found *See NOTE-3*
CAP Alert “Ack” and “Error” Message	Referenced CAP Alert message is expired	209 – message-references-expired-alert *See NOTE-3*

NOTE-1: With the specified error condition:

- Alert is not disseminated as an IPAWS message
- Alert is not available for CAP Exchange

NOTE-2: With the specified error condition:

- Alert is not disseminated as an IPAWS message
- Alert is available for CAP Exchange

NOTE-3: CAP Alert “Update”, “Cancel”, “Ack” and “Error” messages:

- By Specification these Alerts are not disseminated as an IPAWS message
- With the specified error condition, the Alert is available for CAP Alert Exchange

NOTE-4: If a CAP Message with <msgType> element equal to “Alert” with data in the <references> element, it still needs to have the proper structure to be disseminated as an IPAWS message.

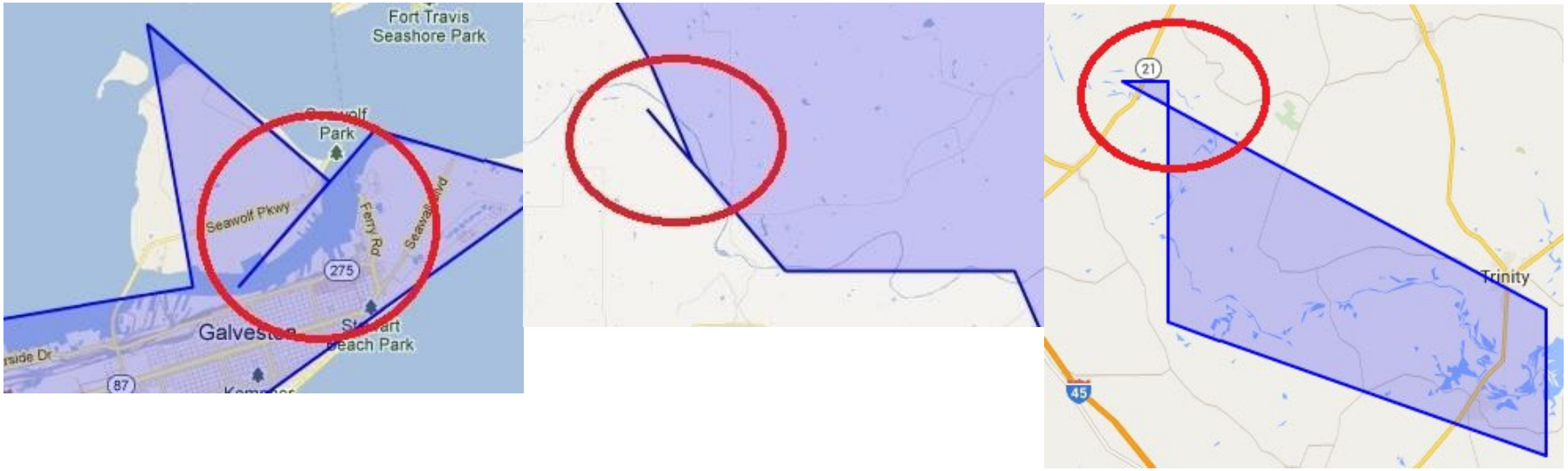
8.9 Circle and Polygon Validation

Additional validation ensures that Alert <area> block <circle> and <polygon> elements are not outside the shape representing the COG’s authorized area. The COG's authorized area includes the geographic area represented by FIPS Codes assigned in the COG Profile. This mitigates situations where an Alert Originator attempts to disseminate a CMAS Alert outside of their authorized FIPS Code range using the Alert <area> block <circle> and <polygon> elements.

It is important that the polygon is well formed in order to be properly validated and processed by the dissemination channels. Polygons with invalid geometry along with zero area polygons can cause issues with polygon processing in downstream dissemination systems. See the figures below for examples of Alert polygons with problem areas marked by a red circle.

Figure 8: Polygon and Circle Error Examples

Collocated Line Segment **Zero-area Boundary Crossover** **Flagpole**



Polygon Upper Limits

IPAWS-OPEN limits all polygons targeted for CMAS/WEA activation to the ATIS Specifications required upper limit of 100 nodes.

The node upper limit for non-CMAS Alerts is 200 nodes. The need for the original limit was due to the possible negative impact on downstream processing due to excessive number of nodes in the <polygon> element. See polygon element in "<area> Element Mapping to Dissemination Channels" table for additional details.

The 10/100 Rule validating alert <area> block for WEA Alerts is to ensure streamlined processing of the <area> block by Wireless Carriers.

- IPAWS CAP Service validates shapes in the alert <area> block. The IPAWS CAP Service will return an error if an Alert contains more than 10 shapes (polygons and/or circles) or over 100 total pair of coordinates (nodes) across all included polygons and/or circles. IPAWS CAP Service will return error Response Code 623 (exceeds-10-shapes-or-100-nodes).
- The IPAWS CAP Service also validates the decimal precision of Alert circle and polygon shapes and will return advisory code 627 (circle-polygon-coordinate-pairs-exceed-4-decimal- precision) if 4 decimal point precision is exceeded.

WEA device-based Geofencing/Geotargeting (DBGF) implemented by wireless carriers can bypass the Geofencing/Geotargeting to streamline WEA processing.

- If an Alert is posted to the CAP Service with parameter equal to valueName = DBGFBYPASS and value = TRUE, the following notification will be passed in the generated WEA Alert; Bypass Device-Based Geo-Fencing.

- Lower Case for valueName and value will also be accepted.
- The purpose of this parameter is to indicate to a Wireless Carrier to bypass “geofencing” (a method for geotargeting) for certain event codes and handling codes that require streamlined processing.
- A global authorization to the COG Profile will govern the authorization to use this feature.

NWR (NWEM) Polygon & Circle Processing

NWS does not accept circles. It is recommended to use polygon’s instead of circles if the NWR affected area is a geo-targeted alert. Other critical NWR-specific processing rules include:

- NWR accepts polygons, but only reads precisions to three decimal points. The second (or the one hundredth) is rounded based on the 3rd decimal point. As a result, the 4th decimal point and beyond are never looked at.
- NWR accepts 20 or less vertices (nodes) in polygons for NOAA downstream dissemination.
- NWR accepts multiple English/Spanish info blocks, but only supports English.
  - NWR ignores a Spanish language tag

8.10 Requirements for WEA

CAP messages intended for WEA distribution must include three additional CAP <parameter> elements.

1. An <info> block with a <language> element containing “en-US”.

For multi-language message (English and Spanish) a CAP Alert with two <info> blocks is required. One <info> block for English with <language> element equal to 'en-US' and one <info> block for Spanish with <language> element equal to 'es-US'.

2. A CMAMtext with up to 90 characters, whereas a 360-character CMAMtext parameter is optional.
3. A parameter element with <valueName> equal to WEAHandling, and corresponding parameter <value> element, which includes one from the following values: Presidential, Amber, Imminent Threat, Earthquake, Public Safety, WEA Test.

The WEA Handling Code is case sensitive.

To ensure proper processing of a CAP Alert intended to be disseminated as a WEA, ensure the <language> element in all CAP <info> blocks have a valid value and are not empty or missing.

Table 104: English & Spanish CMAMText Requirements

Language	CAP <language> Element	CAP CMAM Text Parameter	Remarks
English <info> Block	en-US	CMAMtext up to 90 characters	Required for WEA
		CMAMlongtext up to 360 characters	Conditional **(See NOTE-1)**
Spanish <info> Block	es-US	CMAMtext up to 90 characters	Required with Spanish <info> block **(See NOTE-2)**
		CMAMlongtext up to 360 characters	Conditional **(See NOTE-1)**

NOTE-1: If CMAMlongtext is missing or empty the 90-character CMAMtext will be utilized for the CMAMlongtext and passed with the CMAC Alert.
NOTE-2: If Spanish CMAMtext up to 90-characters is missing or empty, then the English CMAMtext up to 90 characters is used to populate Spanish CMAMtext up to 90-Chracters in the CMAC Alert.

English <info> Block

```
<info>
<language>en-US</language>
<parameter>
<valueName>WEAHandling</valueName>
<value>Imminent Threat</value>
</parameter>
<parameter>
<valueName>CMAMtext</valueName>
<value>This is where the 90 character English text for WEA is located</value>
</parameter>
<parameter><valueName>CMAMlongtext</valueName><value>This is where the 360 character English text
for WEA is located</value></parameter> </info>
```

Spanish <info> Block

```
<info><language>es-US</language>
<parameter>
<valueName>WEAHandling</valueName>
<value>Imminent Threat</value>
</parameter>
<parameter>
<valueName>CMAMtext</valueName>
<value> Aquí es donde se encuentra el texto en inglés de 90 caracteres para WEA. </value>
</parameter>
<parameter>
<valueName>CMAMlongtext</valueName>
<value> Aquí es donde aparece el texto en espanol de 360 caracteres para WEA </value>
</parameter>
</info>
```

If more than one English or Spanish language info block is contained in a CAP alert, only the first English <info> block and Spanish <info> block will be processed and disseminated as a WEA Alert.

8.10.1 Spanish & Special Characters

In June 2019, Wireless Emergency Alerts (WEA 2.0) began supporting both English and Spanish language messaging in a single alert. When creating an alert in Spanish, the alert must also be written in English (90-character English is required, Spanish is optional). This is to ensure that all legacy WEA 1.0 phones still receive an alert. The approved Spanish and special characters include:

Table 105: IPAWS Approved Spanish Characters

Character	Description
Á	Latin Capital letter A with acute
É	Latin Capital letter E with acute
Í	Latin Capital letter I with acute
Ó	Latin Capital letter O with acute
Ú	Latin Capital letter U with acute
à	Latin Small Letter A with grave

Character	Description
á	Latin Small Letter A with acute
è	Latin Small Letter E with grave
é	Latin Small Letter E with acute
ì	Latin Small Letter I with grave
í	Latin Small Letter I with acute
ò	Latin Small Letter O with grave
ó	Latin Small Letter O with acute
ù	Latin Small Letter U with grave
ú	Latin Small Letter U with acute

IPAWS-approved Spanish characters can be used in the value of either CMAMtext or CMAMlongtext fields under the Spanish language <info> block successfully with all wireless carriers. In addition, IPAWS approved characters can be used in headline, description, and instruction elements in Spanish language info blocks for EAS or NWR (NWEM) alert dissemination.

Special character to avoid for successful message dissemination include:

- Less-than sign (<) character or Greater-than (>) character may cause an IPAWS schema error
- "{", "}", "|", "\", "^", "~", "[", "]" in CMAMtext will raise a 621 error code (invalid-characters-in-element-CMAMtext)
- "{", "}", "|", "\", "^", "~", "[", "]" in <uri> and <web> will raise a 222 error code (restricted-characters-in-message)

Vendors should clearly explain to users what is required for their specific software when composing an alert in Spanish or using special characters.

8.10.2 WEA Handling Parameter

WEAHandling parameter is required for all WEA messages and are enforced for all WEA message categories as follows:

WEAHandling parameter	CMAC Version 2.0
Imminent Threat	Enforced
Public Safety	Enforced
WEA Test	Enforced
AMBER	Enforced
Presidential	Enforced
Earthquake[2]	Enforced

The matrix below shows how vendors should assign WEA Handling Parameters for successful message dissemination.

Table 106: WEA Handling Parameter & EventCode Matrix

WEA Handling Parameter							
Event Code	**Event Description**	**Imminent Threat**	**Public Safety**	**Amber** Child Abduction	**WEA Test** State/Local Test	**Presidential**	**Earthquake**
ADR	Administrative Message	Not Allowed	Not Allowed	Not Allowed	Not Allowed	Not Allowed	Not Allowed
AVA	Avalanche Watch	Not Allowed	Not Allowed	Not Allowed	Not Allowed	Not Allowed	Not Allowed
AVW	Avalanche Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
BLU	BLUE Alert	Optional	Recommended	Not Allowed	Not Allowed	Not Allowed	Not Allowed
CAE [1]	Child Abduction Emergency	Not Allowed	Not Allowed	Required	Not Allowed	Not Allowed	Not Allowed
CDW	Civil Danger Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
CEM	Civil Emergency Message	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
DMO	Practice/Demo Warning	Not Allowed	Not Allowed	Not Allowed	Required	Not Allowed	Not Allowed
EQW	Earthquake Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed [2]
EVI	Evacuation Immediate	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
FRW	Fire Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
HMW	Hazardous Materials Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
LAE	Local Area Emergency	Optional	Recommended	Not Allowed	Not Allowed	Not Allowed	Not Allowed
LEW	Law Enforcement Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed

WEA Handling Parameter							
Event Code	**Event Description**	**Imminent Threat**	**Public Safety**	**Amber** Child Abduction	**WEA Test** State/Local Test	**Presidential**	**Earthquake**
MEP	Missing and Endangered Person	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
NUW	Nuclear Power Plant Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
RHW	Radiological Hazard Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
RMT	Required Monthly Test	Not Allowed	Not Allowed	Not Allowed	Required	Not Allowed	Not Allowed
RWT	Required Weekly Test	Not Allowed	Not Allowed	Not Allowed	Required	Not Allowed	Not Allowed
SPW	Shelter in Place Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed
TOE	911 Telephone Outage Emergency	Optional	Recommended	Not Allowed	Not Allowed	Not Allowed	Not Allowed
VOW	Volcano Warning	Recommended	Optional	Not Allowed	Not Allowed	Not Allowed	Not Allowed

Note-1: For NCMEC and approved State agencies only
Note-2: For USGS use only

8.11 NWS Handling of CAP Messages

400-level validations include NWEM specific checks needed for NOAA Weather Radio dissemination. These validations mirror EAS validations because NWR messages and WMO teletype style plain language alert messages generated from the CAP are sometimes picked up for re-transmission by downstream EAS broadcasters.

- IPAWS CAP Service and NWR processing follow EAS validation rules, but with separate eventcode and geocode authorization
- IPAWS CAP Service sets a flag indicating a valid NWR alert and posts the CAP Alert to the NWR CAP Alert-Feed platform (currently this feed is only utilized by National Weather Service for NOAA Weather Radio dissemination)
- NWS Dissemination of Non-Weather Emergency Message (NWEM) over NWR and the NOAAPORT. To enable NWR broadcast, NWS must first generate a World Meteorological Organization (WMO) teletype style formatted version of the alert and transmit it to NWS offices via the NOAAPORT. NOAAPORT is also monitored by many third parties who may also redistribute the alert

The following comments outline specific requirements for processing a CAP v1.2 as an NWR message.

<alert> Element
Standard CAP v1.2 and EAS validations apply to NWEM Alerts for <alert> block.

<info> Element
Standard CAP v1.2 and EAS validations apply to NWR (NWEM) Alerts for <info> block. Correct population of the senderName element is

important because NWS populates the alert text broadcast over NWR and other NWS dissemination systems with information from the senderName element (text is intended to be bold). This is done to ensure proper attribution and clarity in the alert message. NWS makes a clear distinction between the alerting authority generating the alert and the alerting authority requesting the alert which are not always the same (e.g., COG generates an alert on behalf of another COG for which they have authority during a backup situation. Incorrect population of the senderName will result in misleading and confusing information being conveyed to alert message recipients..

The following format for distribution over the NWEM channel is recommended.

- Format is in three parts, comma delimited.
- The senderName element should have the comma delimited format “<COGID>,<CogName>,<Requesting Agency>”.
- Requesting Agency is used when the message is sent on behalf of another entity (e.g., State Agency sending on behalf of a county agency)

<area> Element

Standard CAP v1.2 and EAS validations apply to NWEM Alerts for <area> block.

NWR Event Codes and Geocodes

Standard CAP v1.2 validations apply to NWR (NWEM) Alerts Event Codes and Geocodes.

NWR processes and disseminates Specific Area Message Encoding (SAME)-formatted geocodes which enable programmable features on consumer NWR receivers and are used by broadcasters' EAS encoder/decoder equipment for EAS activation. The geocode for SAME uses the six digit PSSCCC format which uniquely describes each county or county equivalent. A P value of zero corresponds to the entire county. A non-zero P value means Partial County Alerting is being employed. After coordination with the local NWS office, broadcasters, State and Local Emergency Communications Committees and local officials, NWR will execute Partial County Alerting where predefined subdivisions (also known as partitions) of a county are encoded to convey the actual alert area more accurately. Messages using Partial County Alerting may contain more than one geocode per county equivalent in order to convey all the subdivisions included in the alert area. SS is the two digit state/equivalent territory identifier. CCC is the three digit county or equivalent area identifier.

NWS Handling of Polygons

NWS appends polygon information to WMO-formatted alert messages when possible in support of Partial County Alerting and as a courtesy to downstream users as a more accurate conveyance of the actual alert area. Current NWS system design and policies present limitations that NWS expects to address in the coming years. The following, current limitations on the inclusion of polygons by NWS systems should be noted; however, these limitations have no impact on the NWS' ability to convey the alert over NWR and via NOAAPORT.

- NWS will append the alert polygon to the WMO-formatted message when the alert is defined by only one polygon and the polygon contains 20 or fewer discrete vertices. Note: 20 Discrete vertices is equivalent to 21 total vertices in a CAP message because CAP requires the last vertex in the CAP message to be identical to the first vertex.
- If polygon vertices have precision to more than two decimal places, the second decimal place is rounded by NWS based on the 3rd decimal place.

NWS Handling of Circles

Current NWS system design and policies do not allow NWS to accept and convey alert areas defined as circles. NWS expects to address this limitation in the coming years.

NWS Handling of Spanish Alerts

NWS accepts English and Spanish <info> blocks, but only supports English. NWS ignores the Spanish <info> block as of the publication date of this document.

8.12 Language Translation

IPAWS does not provide language translation of alerts. Machine and web-based translations are generally not advised unless reviewed by a foreign speaking individual.

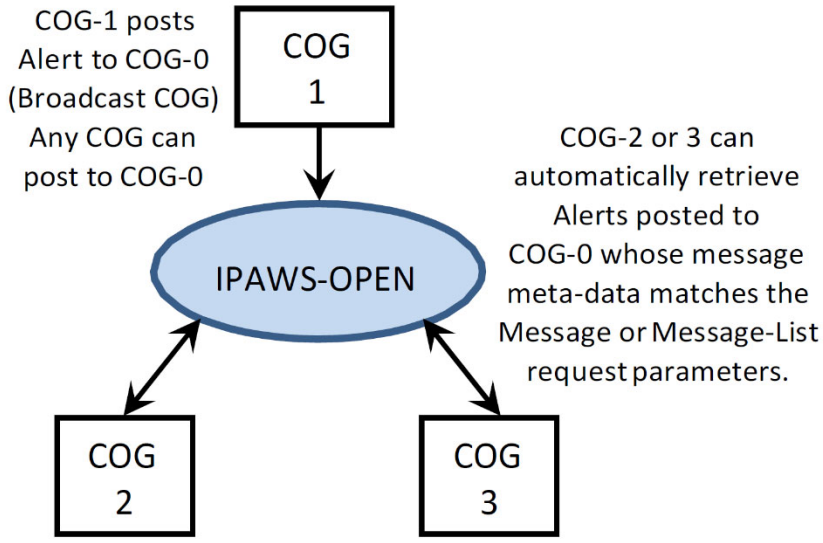
Only UTF-8 characters are accepted by IPAWS-OPEN. Alerting authorities should avoid using non-UTF-8 characters, such as symbols and special characters in non-English languages.

8.13 Multi-COG Public Alert Retrieval

A Broadcast COG feature enables the capability to retrieve Public Alerts from IPAWS-OPEN through the CAP Aggregator service. This capability will allow all COGs to retrieve CAP v1.2 Public Alerts without “directed” distribution. Refer to *Get COG List (getCOG) Section* to find all enabled COGs where a message can be broadcasted to.

The current capability to post to the Broadcast COG (COG-ID = 0, ALL IPAWS-Services COGs) is restricted and controlled by setting authorization in the COG Profile. The capability to download large number of Alerts is limited based on a configurable retrieval time period. The following diagram illustrates the current paradigm for posting to the Broadcast COG.

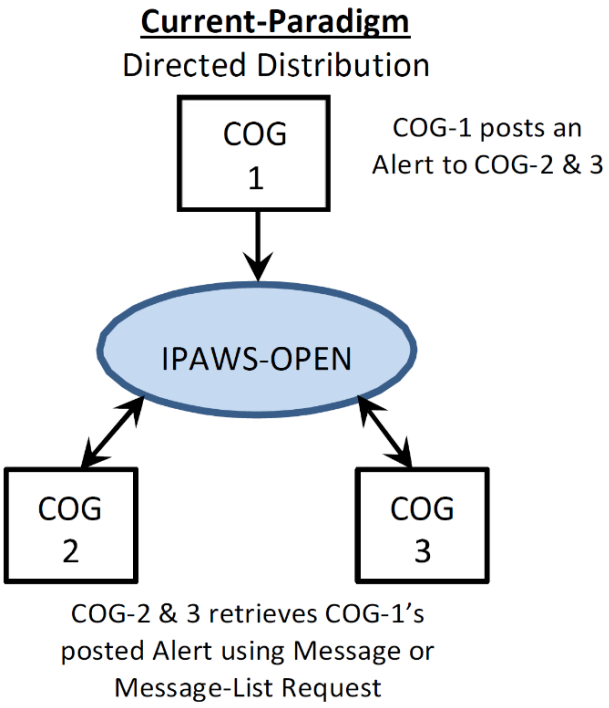
Figure 9: Broadcast COG Process



- Any COG can automatically retrieve Alerts from the Broadcast COG.
- Authorization to post to the Broadcast COG (COG-ID=0 "All IPAWS-Services COG") is restricted and managed in the COG Profile.
    - If an unauthorized COG attempts to post to the Broadcast COG, the following response will be returned: Advisory Code 210 (signer-not-authorized-for-broadcast- cog).
    - The unauthorized Broadcast Alert will not be automatically retrieved by all COGs

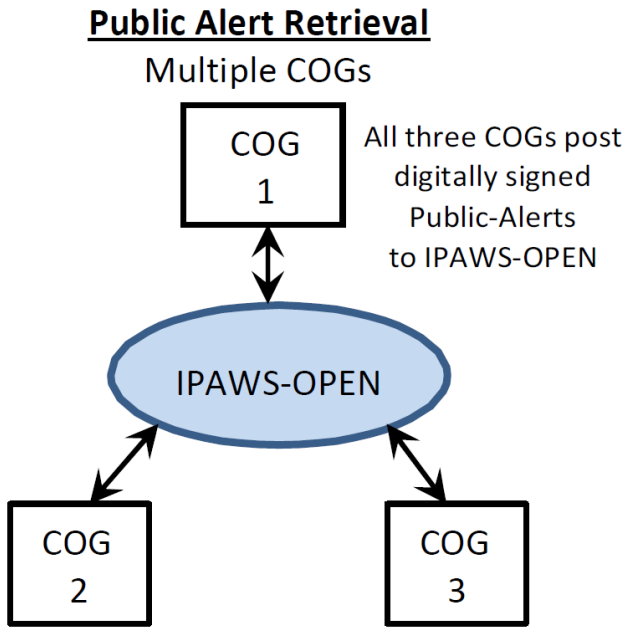
The following diagram illustrates the current paradigm for directed distribution.

Figure 10: Alert Retrieval Process



- A COG can post an Alert to one or more other COGs. Those COGs can then retrieve those alerts.
- The retrieval of Alerts is expanded beyond directed distribution to include the capability to directly retrieve CAP Alerts from any COG with element <scope> equal to “Public” and verified digital signature. The retrieval of CAP Alerts from multiple COGs is illustrated in the following diagram.

Figure 11: Public Alert Retrieval



All COGs can retrieve digitally signed Public-Alerts from IPAWS-OPEN posted by any other COG by using a Public Alert Message or Message-List Request

To improve the effectiveness of Multiple COG Public Alert retrievals the following metadata is persisted and made available as parameters in Message and Message List requests.

- COGs that posted an Alert message.
- Signature validated flag.
- <eventCode> <value> element with <valueName> equal to “SAME”.
- **Public Alert Requests:**

The current “Get Message” and “Get Message List” requests include additional parameters to accomplish a public alert retrieval from multiple COGs.

The following are additional parameters that can be added along with other parameters in either a **getMessage** or **getMessageList** request for Public Alert Requests.

```
<parameters>
<parameterName>requestType</parameterName>
<comparisonOp>equalto</comparisonOp>
<parameterValue>Public</parameterValue>
<logicalOp>and</logicalOp>
</parameters>
<parameters>
<parameterName>sourceCog</parameterName>
<comparisonOp>equalto</comparisonOp>
<parameterValue>XXX123 XXX456</parameterValue>
<logicalOp>and</logicalOp>
</parameters>
<parameters>
<parameterName>eventCode</parameterName>
<comparisonOp>equalto</comparisonOp>
<parameterValue>FLW TOR SVR</parameterValue>
<logicalOp>and</logicalOp>
</parameters>
```

The following table also includes the parameterName value for Public Alert requests, which can be added to any **getMessage** or **getMessageList** request.

Table 107: Public Alert Requests

Complex Type Name	Attribute	Value
(First) Required Parameter (Public Alert Request)	parameterName	requestType
	ComparisonOp	equalto
	parameterValue	Public
	logicalOp	and
	parameterName	sourceCog
	ComparisonOp	equalto
	parameterValue	For Example: XXX123
	logicalOp	Not Required
(Third) Optional Parameter (Public Alert Request)	parameterName	eventCode
	ComparisonOp	equalto
	parameterValue	For Example: FLW TOR SVR
	logicalOp	Not Required

- Adding the parameterName equal to “requestType” and parameterValue equal to “Public” identifies the request a public alert retrieval from multiple COGs.
- To specify the source COG(s) to retrieve from add parameterName equal to “sourceCog” and add for parameterValue the COG(s) in a space delimited string. This parameter is required if “requestType” equal to “Public” is in the request.

Multiple source COGs are handled as an “or” in the request query.

- To specify the event codes to retrieve, add parameterName = “eventCode” and add parameterValue the event codes in a space delimited string. This is an optional parameter.

Multiple event codes are handled as an “or” in the request query.

- The number of Public Alerts that can be retrieved is limited by a configurable retrieval time period.
    - The configurable time period is in minutes.
    - The initial “default” value is 24 hours (1440 minutes).
    - Only Public Alerts posted within the previous 24 hours can be retrieved.

9. CAP Alert Feeds

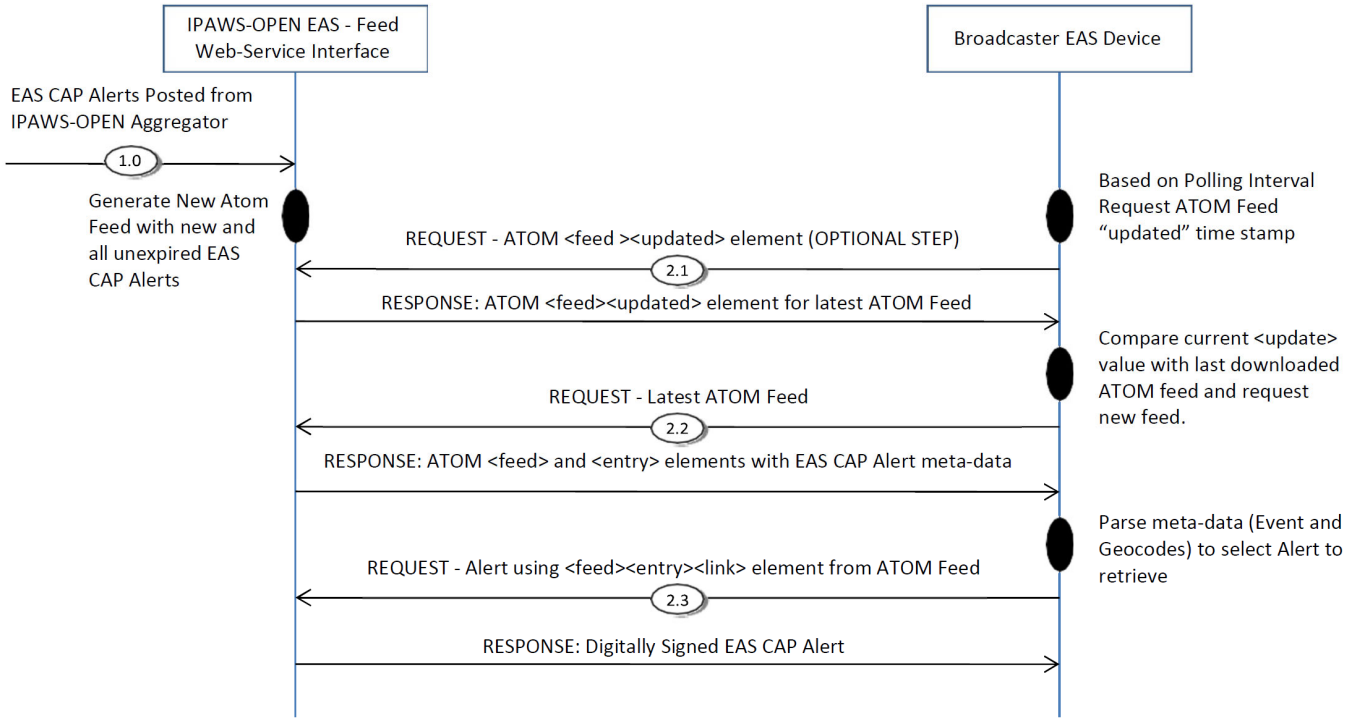
IPAWS-OPEN maintains alert feed interfaces for two specific purposes:

1. An **EAS “ATOM” Feed** for TV and Radio broadcasters to meet FCC regulatory requirements to monitor IPAWS-OPEN for alerts intended to activate EAS.
2. A **Public Feed** to make all IPAWS Profile conforming alerts available to alert consumers.
    - Public feeds use a direct URL path for each alert type NWR (NWEM), EAS, WEA, Public. A consumer visits the URL corresponding with the alert type they are interested in consuming. A description of each feed is detailed below.

9.1 EAS ATOM

The ATOM Syndication Format provides a mechanism to query all alerts intended to activate EAS based on date and time. The EAS feed utilizes a three-step request protocol designed to meet the special requirements of broadcasters. This arrangement is designed for and intended to be used by TV and Radio broadcasters. Third party alert consumers interested in consuming EAS messages should utilize the Public Feed arrangement detailed in the next section. The following diagram illustrates the series of transactions that leverage ATOM

Figure 12: EAS-ATOM CAP Request Diagram



1.0 EAS CAP Alerts are posted from IPAWS-OPEN.

- A new ATOM Feed XML document is generated with all new EAS CAP Alert.
- Alert Metadata contained in the ATOM Feed XML document includes:
  - CAP Alert sent date
  - CAP Alert event code and state-level geocodes
  - Link to retrieve the CAP Alert

2.0 EAS CAP Requests from Broadcaster EAS Device

2.1 Broadcaster EAS Device polls IPAWS-OPEN CAP Alert-Feed Gateway for the <updated> date of the latest ATOM XML document.

2.2 Broadcaster EAS Device requests latest ATOM XML document.

2.3 Broadcaster EAS Device examines metadata and requests CAP Alert based on link extracted from ATOM XML document. A given CAP Alert is requested using the 12 digit Postedmsgid, which is referenced in the ATOM feed and in the ATOM XML document (previously this was the EASMSGID 7 digit)."

As of IPAWS-OPEN 4.02, a PIN in the HTTPS Request is not required. The feed will work with a valid PIN, an invalid PIN, or no PIN.

HTTPS Request: ATOM Feed <updated> Date and Time – This corresponds to the flow labelled (**2.1**) in the previous diagram. This is an optional step to provide a simple check to determine if the ATOM XML document needs to be downloaded. An EAS device can compare current <updated> timestamp with last time that the full ATOM XML document was downloaded to determine if the latest ATOM XML file needs to be requested. This mechanism serves to minimize bandwidth utilization for polling by EAS devices.

HTTPS Get Method call may include a PIN number, but it is not required.

The <hostname> for the CAP Alert-Feed in FEMA TDL (Staging) and Production (CE) are:

- EAS TDL: tdl.apps.fema.gov
- EAS Production: apps.fema.gov

An HTTP 403 error will be returned if attempting to retrieve an expired alert.

Example: Returned ATOM XML Document with empty <entry> Element

```
<?XML version="1.0" encoding="UTF-8"?>
<feed XMLns="http://www.w3.org/2005/Atom">
<title type="text">IPAWS EAS FEED</title>
<updated>2023-08-20T19:45:55.609Z</updated>
<id>https://apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/feed</id>
<entry/>
</feed>
```

HTTPS Request: EAS-Feed ATOM XML Document – This corresponds to the flow labelled (**2.2**) in the previous diagram. This step retrieves the latest ATOM XML document from CAP Alert-Feed service.

An ATOM XML Document with <entry> elements identifying available non-expired EAS CAP Alerts available for downloading will be returned.

Example: Returned ATOM XML with <entry> Elements

```
<?XML version="1.0" encoding="UTF-8"?>
<feed XMLns="http://www.w3.org/2005/Atom">
<title type="text">IPAWS EAS FEED</title>
<updated>2023-08-29T19:45:55.609Z</updated>
<id>https://apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/feed</id>
<entry>
<title>SVR</title>
<link href="https://apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/eas/10001"</link>
<id>https://apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/eas/10001</id>
<updated>2022-07-29T19:45:55.609Z</updated>
<category term="24"label="statefips"></category>
<category term="51"label="statefips"></category>
<category term="SVR" label="event"></category>
</entry>
<entry>
<title>TRW</title>
<link href="https://apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/eas/10002"</link>
<id>https://apps.fema.gov/IPAWSOPEN_EAS_SERVICE/rest/eas/10002</id>
<updated>2022-07-29T19:45:55.609Z</updated>
<category term="20"label="statefips"></></category>
<category term="40"label="statefips"></></category>
<category term="SVR" label="event"></category>
</entry>
</feed>
```

Table 108: ATOM XML Element Description

Element	Sub-Element	Description
<feed>		The <feed> element contains the following tags: <title>, <updated>, <id> and zero one or more <entry> elements.
	<title>	Human readable title for the feed.
	<updated>	Indicates the last time the feed was modified. Format is UTC ZULU time.
	<id>	Universally unique and permanent URI to identify the feed.
<entry>		The <entry> element contains the following tags: <title>, <link>, <id>, <updated>, <category> tags.
	<title>	Title for the entry which is populated from the CAP Alert <info> block <eventCode> element with a <valueName> of "**SAME**".
	<link>	URL to retrieve the entry EAS CAP Alert. The URL contains the retrieval identifier for the alert.
	<id>	Universally unique and permanent URI to identify the entry. The **<link>** element is universally unique and permanent, therefore the **<id>** has the same value as **<link>.**
	<updated>	Indicates the last time the entry was modified. Value from the CAP **<sent>** element.
	<category>	Specifies a category that entry belongs to and can have multiple **<category>** elements within a single **<entry>** element. The **<category>** element includes one required attribute **<term>** which identifies the category and one optional attribute **<label>** which identifies the term.

HTTPS Request: EAS CAP Alert – This corresponds to the flow labelled (**2.3**) in the previous diagram. This step retrieves an EAS CAP Alert based on filtering of <category> terms in the ATOM XML document, and then using the <link> element associated with a selected entry to retrieve the full CAP Alert. The following is an example: https://<hostname>/IPAWSOPEN_EAS_SERVICE/rest/eas/100001

The response includes a digitally Signed CAP Alert based on retrieval identifier included in the ATOM XML <link> element.

- In the example above, the number "100001" is the cap retrieval identifier (this is not the CAP <identifier> element).
- Format and content of the response XML will be in accordance with:
  - Common Alerting Protocol (CAP) Version 1.2 OASIS Standards.
  - CAP v1.2 USA Integrated Public Alert and Warning System Profile Version 1.0 Standards.

9.2 EAS

https://<hostname>/IPAWSOPEN_EAS_SERVICE/rest/eas/recent/<date/time>

This URL path returns alerts that are valid for EAS (500) dissemination.

This service does not follow the three-step protocol as in the EAS ATOM Feed described above.

9.3 NWR (NWEM)

https://<hostname>/IPAWSOPEN_EAS_SERVICE/rest/nwem/recent/<date/time>

This URL path returns alerts that are valid for NWR (NWEM - 400) dissemination.

NWS consumes alerts from this feed for NOAA Weather Radio broadcast and distribution of the alert in human consumable formats over some NWS dissemination systems such as NOAAPORT.

9.4 WEA

https://<hostname>/IPAWSOPEN_EAS_SERVICE/rest/PublicWEA/recent/<date/time>

This URL path returns alerts that are valid for WEA (600) dissemination.

These are CAP v1.2 Alerts and not the CMAC formatted messages that are sent to cell carriers via the C-Interface. There is no capability currently for alert consumer to obtain the CMAC message.

9.5 Public "IPAWS All Hazards Information Feed"

https://<hostname>/IPAWSOPEN_EAS_SERVICE/rest/public/recent/<date/time>

This URL path returns any alert (800/801) that successfully passes CAP v1.2 (200) and IPAWS Profile (300) processing regardless of dissemination path and BLOCKCHANNEL preferences.

9.6 Multiple CAP Alert Retrieval

The capability is available to retrieve multiple CAP Alerts based on date and time. The Alert requestor should submit a request that only includes the date and time from the last instance that Alerts were requested. With this mechanism an Alert requestor has the capability to directly query for Alerts without the need to request an ATOM "Update" or "Feed" XML document.

- A configurable maximum date/time limit is established to encourage Alert requestors to query often enough to limit the bandwidth utilization that could occur if the request covered a large time range.
- If the date/time limit is exceeded, only those Alerts within this limit will be returned (Default value is 30 minutes).

The following is an example: https://<hostname>/IPAWSOPEN_EAS_SERVICE/rest/eas/recent/<date/time>

The example above will retrieve all EAS Alerts posted since a given timestamp. Format will be ZULU time, which is the same used in the ATOM Feed XML.

If no Alerts are found for retrieval, the following text will be returned, which is an Alert object with no content.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><
ns1:alerts xmlns:ns1="http://gov.fema.ipaws.services/feed"></ns1:alerts>
```

The above response applies to both Multiple EAS CAP Alert Retrieval and Public, WEA and NWEM Message Retrieval.

9.7 AWS Simple Notification Services

IPAWS is piloting a new message delivery method utilizing the AWS Simple Notification Service (SNS), which provides a pub/sub model for end users to receive IPAWS-OPEN messages. Rather than the traditional method of accessing IPAWS feeds to poll for messages, subscribers to IPAWS SNS Topics will receive messages in real-time that qualify to be published to a given topic.

IPAWS maintains topics, which users can subscribe to and receive (publish) all qualified messages for that topic. Currently, the test topics of EAS, NWEM/NWR, and WEA for IPAWS Public messages are available. If no filters are requested, the default Public IPAWS All Hazards Information Feed is selected. Additional topics and features will be developed in the future. Available subscription methods include:

- Amazon SQS
- HTTP/HTTPS*
- Platform application endpoint*

* Subscription method is available, but has not been fully tested

We encourage interested parties to subscribe to an IPAWS SNS topic. There are several prerequisites based on the subscription method. Please contact IPAWS for additional detail.

10. Status Item Responses

The following table contains all status item detail that could be included in a "*post message*" response or a "*get message status*" request.

Table 109: Status Item Response Detail

Status Item	Error	Status Response	Status Remarks
10	N	Ack	CAP Alert was successfully transformed to a CMAS CMAC message and it was successfully disseminated to the specific mobile carrier identified in the message status response.
100	Y	invalid-federal-alert-gateway-id	These are possible error codes returned to IPAWS-OPEN when CMAS compliant CAP Alerts are transformed and disseminated to Mobile Carrier Gateways.
101	Y	protocol-version-not-supported	
102	Y	Server-error	
103	Y	Invalid-format	
104	Y	Invalid-element	
105	Y	missing-element	
106	Y	operation-not-allowed	
107	Y	operation-pre-empted	
108	Y	RMT-distribution-precluded	
198	Y	gateway-endpoint-invalid-response	This error code is returned when the response returned by a Wireless Carrier contains invalid XML response or empty response.
199	Y	Gateway-endpoint-is unreachable	CMAS Gateway cannot be reached or response time-out error.

200	N	Ack	Message validates properly to the CAP v1.2 schema
202	N	alert-signature-is-valid	CAP Exchange validation to indicate that digital signature is valid
207	Y	reference-element-missing	CAP Alert (Update, Cancel, Ack, Error) message missing <reference> element
208	Y	Alert-signature-is-not-valid	CAP Exchange validation to indicate that digital signature is not valid
209	Y	message-references-expired-alert	This error response code will be returned if a referenced CAP Alert included in an Update, Cancel, Ack or Error message is expired.
210	N	signer-not-authorized-for-broadcast-cog	This is an advisory response that the COG is not authorized to post to Broadcast COG and the posted CAP Alert <addresses> element includes the Broadcast COG. The Broadcast COG for the Alert will not be stored in metadata and not retrievable by all Alert Originators.
211	Y	restricted-alert-restriction-element-missing-or- empty	This error response code will be returned if the <restriction> element is not present with content when <scope> element is equal to "Restricted".
212	Y	private-alert-addresses-element-missing-or- empty	This error response code will be returned if the <addresses> element is not present with content when <scope> element is equal to "Private".
213	Y	invalid-value-in-element-circle	Additional validation to verify that Circle format is the paired values of a point and radius.
214	Y	geographic-area-outside-authorized-boundary	The <area> block <circle> and/or <polygon> element does not fall within the geographic area of authorized FIPS or Marine Zones.
215	Y	signer-not-authorized-for-event-code- CAPEXCH	This validation only applies if <eventCode> <valueName> element is equal to “SAME”. Alert Originator COG is not authorized for <eventCode> in COG Profile to disseminate CAP Alert Message.
216	Y	invalid-value-in-element-polygon	Polygon errors that may cause issues in downstream systems such as a collocated line segment with zero area, boundary crossover (flag pole) and duplicate vertices.
217	Y	invalid-format-dates	Invalid date format
218	Y	referenced-alert-not-found	Referenced CAP Alert (ACK/ERROR) message cannot be found in IPAWS-OPEN.
219	Y	signer-not-authorized-for-geocode-CAPEXCH	This validation only applies if <geocode> <valueName> element is equal to “SAME”. Alert Originator COG not authorized for <geocode> in COG Profile to disseminate CAP Alert Message.
220	Y	referenced-alert-previously-canceled	This error response will be returned if an alert is canceled more than once.
221	Y	invalid-CAPEXCHANGE-message	General – any error that results in invalid CAP Exchange message
222	Y	Restricted-characters-in-message	Element <identifier>, <sender> includes spaces, commas, or restricted characters (< and &)
223	Y	invalid-value-in-element-geocode	This validation only applies if <geocode><valueName> element is equal to “SAME”. 1) Element <geocode><value> is not 6 digits. 2) Element <geocode><value> contains an invalid value.
224	Y	reference-element-invalid	This response will be returned if the structure for <references> element is not in the "sender,identifier,sent" format. If format is correct but alert is not found IPAWS will return a 218 error.

Status Item	Error	Status Response	Status Remarks
225	N		This advisory response code will be returned if
300	**N**	**Ack**	**Message validates properly to the IPAWS Profile schema**
301	N	non-IPAWS-alert	1) Element <status> is not “Actual 2) Element <code> is not “IPAWSv1.0” 3) Missing info block if not a cancel message
302	Y	invalid-value-in-element-eventcode-and-category	Element <category> values different among info blocks. The <category> values must be identical in all info blocks.
303	Y	invalid-value-in-element-sent	Time in element <sent> is not within 5 minutes of current time
304	Y	invalid-value-in-element-expires	1) expires time < sent time
305	Y	invalid-value-in-element-geocode	1) <area> block is missing 2) Element <geocode> <value> is not 6 digits 3) Element <geocode> <valueName> is not “SAME” 4) Each info blocks must pass 1,2,3
306	Y	missing-required-element-expires	Element <expires> is missing or has an invalid format
311	Y	invalid-value-in-element-eventcode-or-missing-eventcode	Invalid Event Code Value for IPAWS Dissemination (Can be raised if multiple SAME Event Codes are included in the Alert)
312	Y	signer-not-authorized-for-value-EAS-ORG	This error code is returned if the value for the SAME Organization code in the CAP Alert does not match authorized Organization Code in the Alert Originator’s COG Profile.
313	Y	web-token-invalid	This error code is returned if the <web> element includes: 1. A missing http:// or https:// 2. A missing period 3. A space 4. A "?” 5. A length exceeding 2083 characters
314	Y	uri-token-invalid	This error code is returned if the <uri> element includes: 1. A missing http:// or https:// 2. A missing period 3. A space 4. A "?” 5. A length exceeding 2083 characters
400	**N**	**Ack**	**Valid NWR (NWEM) message disseminated**
401	N	message-not-disseminated-as-NWEM	Message not disseminated due to channel blocking or other associated error.
402	Y	no-parameter-EAS-ORG	Response Code to indicate that Alert validated for NWEM dissemination does not contain “EAS-ORG” parameter.
405	Y	invalid-value-mimeType	Element <resource> <mimeType> for broadcast content for delivery to the public does not “contain” one of the following values: 1) “audio/x-ipaws-audio-mp3" 2) "audio/x-paws-streaming-audio" 3) "video/x-ipaws-video" 4) "video/x-ipaws-streaming-video”
406	Y	invalid-value-in-element-expires	Element <expires> time more than 99.5 hours after sent time
407	Y	required-element-missing-description	Element <description> is empty/missing
412	Y	system-error-has-occurred-NWEM-service	Message/ internal dissemination system error
415	Y	signer-not-authorized-for-event-code-NWEM	Alert Originator not authorized for <eventCode> in COG Profile to disseminate NWEM Message.

Status Item	Error	Status Response	Status Remarks
419	Y	signer-not-authorized-for-geocode-NWEM	Alert Originator not authorized for <geocode> in COG Profile to disseminate NWEM Message.
421	N	exceeded-NWEM-geocode-max-limit	This advisory response code will be returned if a NWEM CAP Alert contains more than the geocode maximum limit for NWR, which is currently set at 31 geocodes.
500	**N**	**Ack**	**Valid EAS message disseminated**
501	N	message-not-disseminated-as-EAS	Message not disseminated due to channel blocking or other associated error.
502	Y	no-parameter-EAS-ORG	Response Code – to indicate that Alert validated for EAS dissemination does not contain “EAS-ORG” parameter.
505	Y	invalid-value-mimeType	Element <resource> <mimeType> for broadcast content for delivery to the public does not “contain” one of the following values: 1) “audio/x-ipaws-audio-mp3" 2) "audio/x-ipaws-streaming-audio" 3) "video/x-ipaws-video" 4) "video/x-ipaws-streaming-video
506	Y	invalid-value-in-element-expires	Element <expires> time more than 99.5 hours after sent time
507	Y	required-element-missing-description	Element <description> is empty/missing
508	N	invalid-value-character-limit	This advisory response code will be returned if the combined character count between the <instruction> and the <description> is greater than 1,800
512	Y	system-error-has-occurred-EAS-service	Message/ internal dissemination system error
515	Y	signer-not-authorized-for-event-code-EAS	Alert Originator not authorized for <eventCode> in COG Profile to disseminate EAS Message.
519	Y	signer-not-authorized-for-geocode-EAS	Alert Originator not authorized for <geocode> in COG Profile to disseminate EAS Message.
521	N	exceeded-EAS-geocode-max-limit’	This advisory response code will be returned if a EAS CAP Alert contains more than the geocode maximum limit for EAS, which is currently set at 31 geocodes.
600	**N**	**Ack**	**Valid CMAS (WEA) message disseminated**
601	N	message-not-disseminated-as-CMAS	Message not disseminated due to channel blocking or other associated error.
602	Y	invalid-value-in-element-responseType	Element <responseType> contains invalid CMAS values
603	Y	invalid-value-in-element-CMAMtext	1) CMAMtext value length > 90 2) CMAMlongtext value length > 360
604	Y	lacks-supported-language	Language - value other than 'en-US' or empty
605	Y	update-not-valid-for-CMAS-Alert-cancelled	If Alert Update message contains invalid CMAS values for <urgency>, <severity>, or <certainty> elements, then CMAC Cancel Message is disseminated
608	N	marine-zone-SAME-geocode-not- disseminated-CMAS	Interim Status Code to indicate a Marine Zone geocode was not disseminated to CMAS..
610	Y	CMAM-Text-is-missing-or-empty	CMAM-text is missing or empty
612	Y	system-error-has-occurred-CMAS-dissemination-service	Message/ internal dissemination system error

Status Item	Error	Status Response	Status Remarks
613	Y	invalid-value-in-element-expires	Expires time more than 24 hours after sent time
615	Y	signer-not-authorized-for-event-code-CMAS	Alert Originator not authorized for <eventCode> in COG Profile to disseminate CMAS Message.
616	N	advisory-unexpected-case-CMAMtext- parameter-name	CMAMtext parameter name has the incorrect case: (e.g., not exactly "CMAMtext")
619	Y	signer-not-authorized-for-geocode-CMAS	Alert Originator not authorized for <geocode> in COG Profile to disseminate CMAS Message.
620	N	referenced-alert-never-disseminated to CMAS	This advisory response code indicates that an Update or Cancel message disseminated as a WEA to Wireless Carriers includes an alert not previously disseminated as a WEA.
621	Y	invalid-characters-in-element-CMAMtext	This error response code will be returned if one of the following not-allowed characters was included in the CMAMtext parameter: "{", "}", "|", "\", "^", "~", "[", "]"
622	Y	invalid-mismatch-missing-or-empty-wea-handling-code	This error code response is returned if a CAP Alert contains an invalid or missing WEA-Handling Code parameter element (i.e., typo), or if the WEA-Handling parameter is a mismatch with acceptable values (i.e., using Public Safety handling code with CAE event code).
623	Y	exceeds-10-shapes-or-100-nodes	This error code response is returned if a CAP Alert <area> block either contains more than 10 shapes (polygons and/or circles) or over 100 total pair of coordinates across all included polygons.
626	Y	invalid-value-urgency-severity-or-certainty	This error code response is returned if CAP Alert does not contain valid values for <urgency>, <severity>, and <certainty> elements.
627	N	circle-polygon-coordinate-pairs-exceed-4- decimal-precision	This advisory code response is returned if the decimal precision for circle and polygon coordinate pairs exceed 4 decimal precision.
800	**N**	**Ack (Public Feeds)**	**Valid Non-EAS Public Message is disseminated**
801	N	message-not-disseminated-as-non-EAS-public	Message not disseminated, is an EAS message, or other associated error.
812	Y	system-error-has-occurred-public- dissemination-service	Message/ internal dissemination system error
815	Y	signer-not-authorized-for-event-code-PUBLIC	Alert Originator not authorized for <eventCode> in COG Profile to disseminate PUBLIC Message.
818	Y	referenced-PUBLIC-not-found	Referenced CAP Alert (Update) message cannot be found in IPAWS-OPEN
819	Y	signer-not-authorized-for-geocode-PUBLIC	Alert Originator not authorized for <geocode> in COG Profile to disseminate PUBLIC Message.

11. Appendix

Table 110: Acronym List

Acronym	Explanation
AA	Alerting Authority, used synonymously with Alert Originator (AO)
API	Application Programming Interface
AWS	Amazon Web Services
CAP	Common Alerting Protocol
CE	Cloud Environment
CDTE	Cloud Development & Test Environment
CMAC	Commercial Mobile Alert for C Interface
CMAS	Commercial Mobile Alerting System
CMSP	Commercial Mobile Service Provider
COG	Collaborative Operating Group
COTS	Commercial Off The Shelf
DBGF	Device-based Geofencing
DOM	Data Object Model
EAS	Emergency Alert System
GOTS	Government Off The Shelf
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IPAWS	Integrated Public Alert and Warning System Program
IPAWS-OPEN	Integrated Public Alert and Warning System - Open Platform for Emergency Networks
NOAA	National Oceanic and Atmospheric Administration
NWEM	Non-Weather Emergency Message (related to NWR)
NWR	NOAA Weather Radio All Hazards (related to NWEM)
NWS	National Weather Service
OASIS	Organization for the Advancement of Structured Information Standards
OPEN	Open Platform for Emergency Networks
PMID	Posted Message ID (POSTEDMSGID)
RDS	Relational Database Service
RFC	Requests For Comment (published by IETF)
SAME	Specific Area Message Encoding
SOAP	Simple Object Access Protocol
TDL	FEMA Test Development Lab (See ITE)
WEA	Wireless Emergency Alert (Related to CMAS)
WSDL	Web Services Description language

XML	Extensible Markup Language

IDG Revision History

Table 111: IDG Revision History

Version Number	Version Date	Summary of Changes
1.0	04/12/2010	Initial Version for DM-OPEN
2.0	11/01/2010	Updated Version for IPAWS-OPEN
2.01	11/12/2010	Added Sections for Request/Response Schema
3.00	11/30/2011	Updates CAP1.2 Aggregator and Related Services
3.01	01/20/2012	Updates for IPAWS-OPEN Release 3.01
3.01.02		Updates for CAP1.2 <resource><mimeType> Element
3.02	09/26/2012	Updates for New Features Delivered in Release 3.02
3.03		New Interface Features Delivered in Release 3.03
3.04	05/01/2013	Updates for New Features Delivered in Release 3.04
3.05	09/17/2013	Updates for New Features Delivered in Release 3.05
3.06	07/24/2014	Updates for New Features Delivered in Release 3.06
3.07	12/10/2014	Updates for New/Modified Features in Release 3.07
3.08	05/22/2015	Updates for New Features Delivered in Release 3.08
3.09	06/2/2016	Updates for New Features Delivered in Release 3.09
3.09.01	04/06/2017	Updates for New Features Delivered in Release 3.09.01
3.10	02/15/2019	Updates for New Features Delivered in Release 3.10
3.11	08/07/2020	Updates for New Features Delivered with Oracle 12c Upgrade and Release 3.11
4.0	12/07/2020	Transition to AWS Cloud Environment (CE)
4.01	06/29/2022	Updates for New Features Delivered in Release 4.01
4.01.01	11/23/2022	Minor corrections to content
4.02	02/15/2024	Reorganization of document. Update for new features delivered in Release 4.02
4.02.02	04/18/2024	Updates provides minor bug fixes and security updates
4.02.03	06/12/2024	Updates provides minor bug fixes and security updates
4.02.04	07/18/2024	Updates provides minor bug fixes and security updates
4.02.05	08/14.2024	Updates provides minor bug fixes and security updates