

U.S. Department

of Transportation

**Federal Aviation**

**Administration**

**JAVA Messaging Service Description Document**

**[PROGRAM NAME]**

**Enterprise Service Monitor (ESM)**

**Status Message Provider (ESMP) Service**

JAVA Messaging Service Description Document

[PROGRAM NAME] ESM Status Message Provider Service

Approval Signatures

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Organization** | **Signature** | **Date Signed** |
| AJM | Enterprise Programs |  |  |
|  |  |  |  |
|  |  |  |  |

JAVA Messaging Service Description Document

[PROGRAM NAME] ESM Status Message Provider Service

Revision Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Description** | **Revision****Date** | **Entered By** |
| A | Initial Release | October 30, 2016 | AJM-3122 |
| B | Template Update based on Schema updates | April 20, 2017 | AJM-3122 |
|  |  |  |  |

Table of Contents

[1 SCOPE 1](#_Toc480457292)

[1.1 Background 1](#_Toc480457293)

[1.2 Intended Use 1](#_Toc480457294)

[2 APPLICABLE DOCUMENTS 2](#_Toc480457295)

[2.1 Government Documents 2](#_Toc480457296)

[2.2 Non-Government Standards and Other Publications 2](#_Toc480457297)

[3 DEFINITIONS 4](#_Toc480457298)

[3.1 Terminology 4](#_Toc480457299)

[3.1.1 Client 4](#_Toc480457300)

[3.1.2 Service 4](#_Toc480457301)

[3.1.3 Service Interface Implementation 4](#_Toc480457302)

[3.1.4 NEMS 4](#_Toc480457303)

[3.2 Abbreviations and Acronyms 5](#_Toc480457304)

[4 Service Profile 7](#_Toc480457305)

[4.1 Service Provider 7](#_Toc480457306)

[4.1.1 Point of Contact 7](#_Toc480457307)

[4.2 Service Consumers 7](#_Toc480457308)

[4.3 Service Functionality 7](#_Toc480457309)

[4.4 Security 8](#_Toc480457310)

[4.5 Qualities of Service 8](#_Toc480457311)

[4.6 Service Policies 8](#_Toc480457312)

[4.7 Environmental Constraints 8](#_Toc480457313)

[5 Service Interfaces 9](#_Toc480457314)

[5.1 Interface 9](#_Toc480457315)

[5.2 Operations 9](#_Toc480457316)

[5.2.1 Processing Considerations 9](#_Toc480457317)

[5.3 Messages 9](#_Toc480457318)

[5.3.1 StatusEventMessage Format 9](#_Toc480457319)

[5.4 Exception Handling 11](#_Toc480457320)

[5.5 Data 11](#_Toc480457321)

[6 Service Implementation 12](#_Toc480457322)

[6.1 Bindings 12](#_Toc480457323)

[6.1.1 Transport Binding 12](#_Toc480457324)

[6.1.1.1 Data Protocol 13](#_Toc480457325)

[6.1.1.2 Message Protocol 13](#_Toc480457326)

[6.1.1.3 Transport Protocol 13](#_Toc480457327)

[6.1.1.4 Other Protocols 13](#_Toc480457328)

[6.2 End Points 13](#_Toc480457329)

[6.2.1 End Point [ENDPOINT] 13](#_Toc480457330)

[6.3 Subscription 13](#_Toc480457331)

List of Figures

Figure 1 - Subscribe Flow 15

List of Tables

[Table 1 : Quality of Services Parameters 8](#_Toc480457333)

[Table 2 : Environmental Constraints 8](#_Toc480457334)

[Table 3 : List of Published Messages 9](#_Toc480457335)

[Table 4 : StatusEventMessage Header 10](#_Toc480457336)

[Table 5 : StatusEventMessage Data Elements 10](#_Toc480457337)

[Table 6 : SituationCategory Description 11](#_Toc480457338)

[Table 7 : Simple Data Types 12](#_Toc480457339)

[Table 8 : Complex Data Types 12](#_Toc480457340)

# SCOPE

This JAVA Messaging Service Description Document (JMSDD) describes the [PROGRAM NAME] ESM Status Message Provider (ESMP) Service. The [PROGRAM NAME] ESMP service publishes status information for consumption by the Enterprise Service Monitor (ESM) capability via the National Airspace System (NAS) Enterprise Messaging Service (NEMS). This JMSDD was prepared in accordance with Federal Aviation Administration (FAA) Standard (STD) FAA-STD-073.

## Background

[PROGRAM BACKGROUND AS NEEDED]

## Intended Use

This JMSDD is intended to be used by the ESM capability to facilitate the integration of the [SERVICE NAME] service with the ESM monitoring capability. It may also be used by the FAA staff who may administer these services.

# APPLICABLE DOCUMENTS

## Government Documents

|  |
| --- |
| [FAA-STD-063] XML Namespaces, 1 May 2009. <http://www.tc.faa.gov/its/worldpac/standards/faa-std-063.pdf>  |
| [FAA-STD-064] Web Service Registration, 1 May 2009. <http://www.tc.faa.gov/its/worldpac/standards/faa-std-064.pdf>  |
| [FAA-STD-073] Preparation of Java Message Service Description Document, W3C Working Draft, 29 January 2014. <http://www.tc.faa.gov/its/worldpac/standards/faa-std-073.pdf>  |
| [FAA-STD-066] Web Service Taxonomies, 26 February 2010. <http://www.tc.faa.gov/its/worldpac/standards/faa-std-066.pdf>  |
| [FAA-STD-068] Preparation of Standards, 4 December 2009.<http://www.tc.faa.gov/its/worldpac/standards/faa-std-068.pdf>  |
| [NAS-IC-XXXXXXXXX] NAS Enterprise Messaging Service (NEMS) Asynchronous Messaging Interface Control Document (ICD), Rev A Draft, 20 July 2012. |
| Non-Government Standards and Other Publications |
| [W3C XML Recommendation] World Wide Web Consortium eXtensible Markup Language (XML) Version 1.9, Fifth edition, 26 Nov 2008. <http://www.w3.org/TR/2008/REC-xml-20081126/>[IEE 802.3] Information Technology – Telecommunication & Information Exchange between Systems – LAN/MAN – Specific Requirements – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2002.[WSDR] Web Services Description Requirements, W3C Working Draft, 28 October 2002.<http://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028/>[RFC 2119] Key words for Use in RFCs to Indicate Requirement Levels, Network WorkingGroup, March 1997. <http://www.rfc-editor.org/rfc/rfc2119.txt>[ISO/IEC 11179-1] Information Technology – Metadata Registries (MDR) – Part 1:Framework, 15 September 2004. <http://metadata-standards.org/11179/>[WS Glossary] Web Services Glossary, W3C Working Draft, 14 November 2002. <http://www.w3.org/TR/2002/WD-ws-gloss-20021114/>[WSA] Web Services Architecture, W3C Working Group Note, 11 February 2004.<http://www.w3.org/TR/ws-arch> |

[JAVA API] Java 2 Platform Enterprise Edition, v 5.0 API Specifications <http://docs.oracle.com/javaee/5/api/javax/jms/Message.html>

# DEFINITIONS

## Terminology

Within the SWIM service-oriented architecture (SOA) environment, there are key elements that must be defined to properly identify boundaries, functionality, and components. The key high‑level entities are identified in the list below:

* Client
* Service
* Service Interface Implementation
* NAS Enterprise Messaging Service (NEMS)

### Client

A client is an external entity that interacts with a service. A client makes a request of a service and receives a response from the service. The client may also request a subscription and receive messages when a service publishes information. A client may be a software system, software application, or another service. A client may be a NAS client or a non-NAS client. It should be noted that the terms “consumer” and “end-user” can be and are used interchangeably with “client”.

### Service

In the most general sense, a service is a set of functionalities that is performed upon demand based upon a defined interface. In the request-response paradigm, a defined request must be provided by the client that invokes the service, and the service returns a defined response to the client. In the publish-subscribe paradigm, a service publishes information for one or more clients to consume.

### Service Interface Implementation

The service interface implementation is the hardware and software that handles the interaction between the client and the rest of the service. The definition and the implementation are independent of each other. It is possible to maintain the same service interface definition and replace the implementation. The Publish/Subscribe service interface implementation is based on the software in the NEMS.

### NEMS

NEMS contains the consumer topics used to provide the appropriate information to a consumer of the Publish/Subscribe service. [PROGRAM NAME] ESMP Service will publish data onto queues on the NEMS. A consumer will connect to NEMS to retrieve data from the subscribed topic(s).

## Abbreviations and Acronyms

[ADD PROGRAM SPECFIC ABBREVIATIONS AND ACRONYMS AS NEEDED]

|  |  |
| --- | --- |
| APDSARTCCs | Airport Data ServiceAir Route Traffic Control Centers |
| ESM | Enterprise Service Monitor |
| FAAFNTBFTIGeNUS | Federal Aviation AdministrationFTI National Test BedFAA Telecommunications InfrastructureGeneral NAS User Services |
| ICAO | International Civil Aviation Organization |
| IP | Internet Protocol |
| ISO | International Standards Organization |
| JMSJMSDDMTBFNA | Java Messaging ServiceJava Messaging Service Description DocumentMean Time Between FailuresNot Applicable |
| NAS | National Airspace System |
| NDRB | NAS Data Release Board |
| NEMSNSRR | NAS Enterprise Messaging ServiceNAS Service Registry Repository  |
| PMR&D | Program ManagerResearch and Development |
| SOA | Service Oriented Architecture |
| SSDSSISTD | System Specification DocumentSensitive Security InformationStandard |
| SUI | Sensitive Unclassified Information |
| SWIM | System-Wide Information Management |
| TCP | Transmission Control Protocol |
| TRACONURL | Terminal Radar Approach Control facilityUniform Resource Locator |
| UTC | Universal Time Coordinated/ Coordinated Universal Time |
| WJHTC | William J. Hughes Technical Center |
| XML | eXtensible Markup Language |

# Service Profile

This section provides the information needed to discover and use the [PROGRAM NAME] ESMP service.

| **Service Profile** |
| --- |
| **Name** | [PROGRAM NAME] ESMP service  |
| **Description** | The [PROGRAM NAME] ESM Status Service provides status information for consumption by the Enterprise Service Monitor capability via the National Airspace System (NAS) Enterprise Messaging Service (NEMS). |
| **Namespace** | us:gov:dot:faa:atm:terminal:entities:status:v2-0  |
| **Version** | 1.0 |
| **Service category** | [Refer to urn:us:gov:dot:faa:taxonomies:service-category] |
| **Lifecycle stage** | [lifecycle] |
| **Service criticality** | Routine |

## Service Provider

The [PROGRAM NAME] ESMP service is provided by the [PROGRAM ORGANIZATION].

###  Point of Contact

| **Point of Contact** |
| --- |
| **Name** | POC NAME |
| **Organization** | POC ORGANIZATION |
| **Title** | POC TITLE |
| **Phone** | POC PHONE |
| **Email** | POC EMAIL |

##

## Service Consumers

The ESM capability is the only planned consumer of the [PROGRAM NAME] ESMP service.

## Service Functionality

[PROGRAM NAME] ESMP publishes status information for the ESM capability, providing situational awareness of the [PROGRAM NAME] program. The [PROGRAM NAME] ESMP service publishes the data listed below. For more information about individual message types, see section 5.3.

* Publishes Status Events (StatusEventMessage)

## Security

NEMS manages all security features for Publish/Subscribe services for authorized NAS and non-NAS consumers and NAS producers.

Access controls are supported through the use of username and password credentials supplied when establishing connections to NEMS interfaces. Username and password credentials are unique to each NEMS client and established during on-ramping.

The [PROGRAM NAME] ESMP service does not contain sensitive flight data.

## Qualities of Service

[PROGRAM NAME] ESMP Quality of Service parameters are listed in the following table.

Table 1 : Quality of Services Parameters

| **QoS Parameter** | **Value** | **Unit** | **Definition** | **Calculation** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Service Policies

No specific service policies are applied to the [PROGRAM NAME] ESMP service.

[PROGRAM NAME] ESMP data is determined to contain no sensitive information.

## Environmental Constraints

This service covers three NEMS operating environments:

1. Research & Development (R&D) NEMS at Melbourne, Florida
2. FAA Telecommunications Infrastructure (FTI) National Test Bed (FNTB) NEMS at William J. Hughes Technical Center (WJHTC)
3. NAS-OPS NEMS deployed to Air Route Traffic Control Centers (ARTCCs).

Table 2 : Environmental Constraints

|  |  |
| --- | --- |
| **Service Constraints** | FTI environment |
| **Deployed NEMS Environment** | R&D, FNTB and NAS-OPS |
| **Message Producer Type** | NAS application |
| **Record Type** | Live |

# Service Interfaces

The following sections provide detailed information about the types and content of messages that the [PROGRAM NAME] ESMP service publishes. The service is also described in terms of the interfaces that it communicates with.

## Interface

The [PROGRAM NAME] ESMP service follows the “point-to-point” messaging model and employs a single queue interface to NEMS, through which all status data can be subscribed to and received. NEMS follows the “Publish/Subscribe” messaging model whereby each client interested in subscribing to all [PROGRAM NAME] ESMP data establishes a connection to a custom data topic provided by NEMS and defined during on-ramping.

## Operations

The [PROGRAM NAME] ESMP service publishes status data for [STATUS SCOPE]. ESMP data is published in XML format.

### Processing Considerations

There are no special processing considerations for [PROGRAM NAME] ESMP data.

## Messages

The foundation for the external message format for [PROGRAM NAME] ESMP service is based upon the ESM Schema Description Document, which is derived from the Web Services Distributed Management (WSDM) standard. The following messages are published by [PROGRAM NAME] ESMP service via NEMS. Each message has a header and a payload. The header contains routing data used by NEMS to deliver messages to the correct subscriber. Note that the MsgType column defines the two character abbreviation that is used in the JMS header to delineate the message type.

Table 3 : List of Published Messages

| **Name** | **Definition** | **MsgType** |
| --- | --- | --- |
| StatusEventMessage | Provides a notification of the status of a monitored resource. This status may be that a significant change in the state of a monitored resource has occurred, or may simply be a heartbeat message. The exact situation that caused the message to be published is detailed in the SituationCategory element. | SE |

### StatusEventMessage Format

The following table lists the header for a StatusEventMessage. This message is published upon a significant change in the status of a monitored resource or on a periodic basis every X seconds.

Table 4 : StatusEventMessage Header

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Element** | **Description** | **Cardinality** | **Type** |
| [MSG HEADER ELEMENT] |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

The following table lists the data elements in the payload of a StatusEventMessage.

Table 5 : StatusEventMessage Data Elements

| **Data Element** | **Description** | **Cardinality** | **Type** |
| --- | --- | --- | --- |
| msgType | Defines the type of message | 1 | xsd:string |
| reportTime | UTC date and time of message generation | 1 | xsd:dateTime |
| eventID | Unique identifier for the message | 1 | xsd:string |
| reporterComponent | Identifier for the managed entity providing the status message | 1 | esm:ResourceIdType |
| sourceComponent | Identifier for the managed entity being reported on | 1 | esm:ResourceIdType |
| Current Operational State | The current operational state of the component being reported on | 1 | esm:StateType |
| Last Operational State Transition | Information about the last operational state transition (for example, a report on when component last transitioned from DOWN to UP) | 0 to 1 | esm:StateTransitionType |
| Business Service Operational Status | The current operational status of the business service | 1 | esm:StateType |
| Situation Report | Details regarding the current situation being reported on | 1 | esm:SituationType |
| Endpoint Report | Details on the subcomponents of the business service or source component, including status and metrics | 0 to 1 | esm:EndpointReportType |

**SituationCategory** categorizes the type of the situation that caused the event message. The permissible values represent the names of elements in the ESM namespace. The categories are listed in the order of precedence. In a case where there may be some ambiguity about which category to use, the higher precedent category SHOULD be used. The ordering of situation categories is based on empirical data showing relative importance of various types of events. The use of a higher precedent category permits more effective and timely correlation and analysis of events that may indicate the presence of a serious problem. This is a required element.

Table : SituationCategory Description

|  |  |
| --- | --- |
| Situation Name | Description |
| AvailabilitySituation | deals with situations regarding operational state and availability. |
| ConfigureSituation | deals with components identifying configuration changes. |
| StopSituation | deals with the shutdown process for a component |
| StartSituation | deals with the startup process of a component |
| ConnectSituation | deals with the situations related to aspects about a connection attempt from one component to another |
| ReportSituation | deals with situations that occur as a result of some setting or occurrence that causes the resource to report various types of data. Situations of this nature are considered informational only, and as such this situation category should not be used for any events that an operator should be alerted to. |

## Exception Handling

None

## Data

The following tables describe the data types defined in the [PROGRAM NAME] ESMP messages.

Specific formats required for any data types will be included in the types’ definition text. The detail for the obligation and maximum occurrences of the data types and elements is included in the schema definitions within the NSRR.

A type listed as *complex* means that the entity consists of one or more sub-entities, which are detailed in the entity’s definition.

Table 7 : Simple Data Types

| **Name** | **Definition** | **Permissible Values** | **Data Type** | **Format** |
| --- | --- | --- | --- | --- |
| dateTime | This type defines the date and time. It is a primitive data type. | Valid date and time | primitive | yyyy-mm-ddThh:mm:ss.sssZ |

Table 8 : Complex Data Types

| **Name** | **Definition** | **Permissible Values** | **Data Type** | **Format** | **Obligation** |
| --- | --- | --- | --- | --- | --- |
| ResourceIdType | Defines the attributes of a monitored resource |  | Complex |  | Required |
| StateType | Defines applicable states in accordance with the services state model | AvailableDegradedJeopardyFailedUnavailableMaintenanceUnknown | xsd:String |  | NA |
| StateTransitionType | Defines the state transition of the component |  | Complex |  | Optional |
| SituationType | Defines the type of the situation that caused the event message |  | Complex |  | Required |
| MetricType | Defines the metric  |  | Complex |  | Optional |
| EndpointType | Defines the endpoint component |  | Complex |  | Optional |
| EndpointReportType | Enables reporting of multiple endpoints |  | Complex |  | Optional |

# Service Implementation

The [PROGRAM NAME] ESMP is implemented as a JMS Producer service. All access points to the service are wholly contained within NEMS. The [PROGRAM NAME] ESMP service publishes data to NEMS via a JMS queue. Clients connect to a topic established by NEMS in order to receive [PROGRAM NAME] ESMP data. The endpoint for the subscribe [PROGRAM NAME] ESMP operation is described in the following sections.

## Bindings

### Transport Binding

All [PROGRAM NAME] ESMP messages are bound to the NEMS JMS interface through [ActiveMQ].

#### Data Protocol

For data serialization, all [PROGRAM NAME] ESMP data is published to NEMS using the Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C, November 2008, [http://www.w3.org/TR/2008/REC-xml-20081126/ [11](http://www.w3.org/TR/2008/REC-xml-20081126/%20%20%5B11)].

#### Message Protocol

The [PROGRAM NAME] ESMP uses the Java Messaging System (JMS) as described in Java 2 Platform Enterprise Edition, v 5.0, [http://docs.oracle.com/javaee/5/api/javax/jms [12](http://docs.oracle.com/javaee/5/api/javax/jms%20%5B12)] as a message-level protocol.

#### Transport Protocol

The [PROGRAM NAME] ESMP uses Transmission Control Protocol (TCP), RFC 675, Network Working Group, December 1974, [http://tools.ietf.org/html/rfc675 [15](http://tools.ietf.org/html/rfc675%20%5B15)] as a transport-level protocol.

#### Other Protocols

See the NEMS ICD for details about other protocols, e.g., application network and datalink protocol.

## End Points

The TCP/IP addresses are not included in this document for security reasons. Please refer to NEMS On-Ramping forms and section 4.1.1 for more details.

### End Point [ENDPOINT]

Associated Binding: [BINDING]

Network Address: Established by NEMS and provided to service consumer during on-ramping of a consumer agent.

## Subscription

The TCP/IP addresses are not included in this document for security reasons. Please refer to Subscribe Flow for more details.

An [PROGRAM NAME] ESMP client may subscribe to a NEMS specified topic to obtain published status messages. Individual subscriber topics will be identified as part of a client’s on-ramping process.

The message flow to publish a message is asynchronous and starts with the client requesting a connection and a subscription from NEMS. As data becomes available from the [PROGRAM NAME] ESMP, it is published to the NEMS. The NEMS stores the messages on the client’s registered topic. If the client is connected to its topic, the messages will be delivered from the client’s topic to the client. If the client is not connected to its topic, the message is purged from the topic after its time-to-live has expired. The subscription request (steps 1 and 2 below) is performed once for each desired subscription, and the messages are published to the client until the subscription is cancelled.

A detailed message flow for receiving a message is shown below:

1. The client connects to the NEMS and requests a subscription.
2. The NEMS accepts the connection, validates the client’s security, and replies with the status of the subscription. This is the NEMS service delivery point.
3. [PROGRAM NAME] ESMP service publishes the messages to the assigned queue on the NEMS.
4. The NEMS publishes the message to each of the topics associated with clients that have registered for the message.
5. Each client connected to the NEMS pulls the message out of its topic.

Data flow

Client connection flow

5. The Client consumes the message from the topic

2. NEMS accepts the connection

1. The Client connects to NEMS

Client

3. An XML message is published on the queue to NEMS.

4. NEMS publishes the message on the topic

NEMS

[PROGRAM NAME]

ESMP SERVICE

Figure - Subscribe Flow