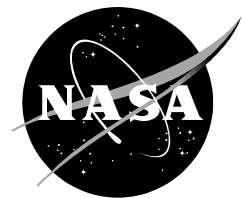


NASA/TM—2015—218427



# **A Formal Messaging Notation for Alaskan Aviation Data**

*Joseph L. Rios  
Ames Research Center  
Moffett Field, California*

---

**January 2015**

## NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

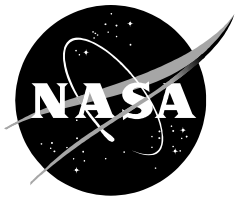
- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Phone the NASA STI Information Desk at 757-864-9658
- Write to:  
NASA STI Information Desk  
Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199

NASA/TM—2015—218427



# A Formal Messaging Notation for Alaskan Aviation Data

*Joseph L. Rios  
Ames Research Center  
Moffett Field, California*

National Aeronautics and  
Space Administration

*Ames Research Center  
Moffett Field, CA 94035-1000*

---

**January 2015**

## Acknowledgments

Thanks to Lev Walkin, the creator of the open source ASN.1 compiler `asn1c`. His compiler and support were invaluable throughout the process of defining this specification.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA Center for AeroSpace Information  
7115 Standard Drive  
Hanover, MD 21076-1320  
443-757-5802

National Technical Information Service  
5301 Shawnee Road  
Alexandria, VA 22312  
703-605-6000

This report is also available in electronic form at  
<http://ntrs.nasa.gov>

## Table of Contents

List of Figures .....	iv
List of Tables .....	v
Summary .....	1
1. Introduction .....	1
2. Background .....	2
2.1 Pilot Weather Reports .....	2
2.1.1 Turbulence Element .....	3
2.1.2 Air Temperature Element .....	3
2.1.3 Icing Element .....	4
2.1.4 Sky Condition Element .....	4
2.1.5 Wind Vector Element .....	4
2.1.6 Weather Element .....	4
2.1.7 Example PIREPs .....	4
2.2 Aviation Surface Weather Reports .....	5
2.2.1 METAR Wind .....	6
2.2.2 METAR Visibility .....	6
2.2.3 METAR Runway Visual Range .....	6
2.2.4 METAR Present Weather .....	7
2.3 Weather Polygons .....	7
3. Abstract Syntax Notation One .....	9
4. Data Descriptions .....	10
4.1 Encoding Locations .....	11
4.2 Encoding Time .....	11
4.3 ASN.1 PIREP Data .....	12
4.4 ASN.1 METAR Data .....	15
4.5 ASN.1 Description of Other Data .....	15
5. Concluding Remarks .....	17
6. References .....	18
Appendix A—ASN.1 Source Files .....	19
A.1 Overall Message Definition .....	20
A.2 Common Data Elements .....	22
A.3 METAR Messages .....	28
A.4 PIREP Messages .....	33
A.5 Image Messages .....	39
A.6 Emergency Messages .....	40
A.7 Route Messages .....	41
A.8 Weather Polygon Messages .....	43
A.9 Weather Cam Messages .....	44
A.10 System Messages .....	49
Appendix B—Encoding Examples .....	51
B.1 Weather Polygon Example .....	51
B.2 METAR Example .....	51
B.3 Weather Polygon Encoding .....	56

## List of Figures

Figure 1. METAR present weather values.....	7
Figure 2. Icing prediction from the Alaska Aviation Weather Unit. ....	8
Figure 3. Turbulence prediction from the Alaska Aviation Weather Unit. ....	8
Figure 4. Flight category prediction from the Alaska Aviation Weather Unit. ....	9
Figure B1. Graphical METAR data. ....	52
Figure B2. Detail of four METARs in northern Alaska. ....	52
Figure B3. A turbulence forecast for Alaska. ....	56
Figure B4. Turbulence forecast polygon over St. Lawrence Island redrawn in Google Earth. ....	57

## List of Tables

Table 1. Summary of PIREP weather phenomenon elements .....	3
Table 2. Subset of available METAR weather elements.....	6
Table 3. Comparison of PIREP encodings .....	13
Table 4. Comparison of METAR encodings .....	16





## Summary

Data exchange is an increasingly important aspect of the National Airspace System. While many data communication channels have become more capable of sending and receiving data at higher throughput rates, there is still a need to use communication channels efficiently with limited throughput. The limitation can be based on technological issues, financial considerations, or both. This paper provides a complete description of several important aviation weather data in Abstract Syntax Notation format. By doing so, data providers can take advantage of Abstract Syntax Notation's ability to encode data in a highly compressed format. When data such as pilot weather reports, surface weather observations, and various weather predictions are compressed in such a manner, it allows for the efficient use of throughput-limited communication channels. This paper provides details on the Abstract Syntax Notation One (ASN.1) implementation for Alaskan aviation data, and demonstrates its use on real-world aviation weather data samples as Alaska has sparse terrestrial data infrastructure and data are often sent via relatively costly satellite channels.

## 1. Introduction

Alaska is a unique environment. This is especially true in the world of aviation. Due to the lack of ground-based transportation infrastructure, there is a heavy dependence on aviation for the movement of people and goods throughout this massive state. When this dependence on aviation is coupled with extreme weather and mountainous terrain, there is a marked increase in aviation accidents in Alaska on a per-operation basis compared to the conterminous United States [1].

The need to increase aviation safety in Alaska has been well recognized. A report on the occupational dangers of flying in Alaska has been documented by the Centers for Disease Control and Prevention (CDC) [2]. A major engineering and research endeavor called "Capstone" was undertaken to evaluate the use of new technology to reduce accidents in Alaska [3]. In addition, the National Transportation Safety Board (NTSB) has recognized the general need for improved situational awareness for general aviation by issuing the challenge to "identify and communicate hazardous weather" as part of its "Most Wanted List" for 2014 [4].

With this background, NASA and the state of Alaska agreed to cooperate on fundamental research and development of a system aimed at providing general aviation pilots in Alaska improved situational awareness through the use of commercial off-the-shelf technologies. The project is named Traffic and Atmospheric Information for General Aviation (TAIGA). The high-level concept of the project is to create infrastructure, data retrieval, and processing, plus interface software using currently available data sources of interest and importance to general aviation pilots in Alaska. Pilots would receive the messages via a sensor device that includes at least a satellite receiver and potentially other sensors (global positioning system (GPS), Automatic Dependent Surveillance-Broadcast (ADS-B) in, altimeter, etc.) and then interact with the received data on a commercial mobile device, such as an iPad or Android-based tablet. Several aspects of this project are proceeding in parallel. The work presented here is concerned with the specification of the messages that are sent via satellite from a ground station to the pilot's receiving device.

Because the cost of sending such messages via satellite may be considerable, it is important to have efficient data encoding. The data need to be as complete as possible with minimal redundancies and non-essential information. This document discusses the various trade-offs proposed to strike such a balance between completeness and efficiency. Note that a major motivation for this document is to allow public comment on work that has been heretofore NASA-only research. The specification presented here is current as of publication of this document and is subject to significant revision once comments from stakeholders are received. In the future, there will be a formal release of the specification along with processes to keep the specification up to date.

The remainder of this document is organized as follows: Section 2 provides background on some of the data that are proposed to be communicated via this system, namely Pilot Weather Reports (Section 2.1), Aviation Surface Weather Reports (Section 2.2), and Generic Weather Polygons (Section 2.3). Then an introduction to Abstract Syntax Notation One (ASN.1), which is used to formally specify the messages within TAIGA broadcasts, is provided in Section 3. Next, a discussion of the encoding choices for the various message types is given in Section 4. Concluding remarks are provided in Section 5, followed by two important appendices. In Appendix A, the proposed ASN.1 specification for the messages is provided in its current state as of the publishing of this document. Finally, Appendix B provides concrete examples of the encoding.

## **2. Background**

This section provides background information relevant to the development of the data descriptions described later in this document. Specifically, a high-level view of the format and content of aviation weather data including Pilot Weather Reports (PIREPs), Aviation Routine Weather Reports (METARs), and various weather predictions and observations are provided.

### **2.1 Pilot Weather Reports**

Pilots rely on reports from other pilots to aid in flight planning and situational awareness during flight. The formal mechanism for reporting weather information by pilots is the Pilot Weather Report or PIREP [5]. These reports contain information about turbulence, wind, visibility, icing, and temperature, as well as information about the aircraft and location involved with the report. Typically, these PIREPs are solicited by Flight Service Stations (FSSs) or pilots offer them freely to FSSs. These PIREPs are then entered into a central database for access by other FSSs or users via various web interfaces. The PIREPs may then become part of the briefing given during pre-flight planning or perhaps en route when requested by a pilot via radio. PIREPs are not typically available to en route pilots in an automated fashion. They were originally designed to be easily transmitted via voice over the radio.

TABLE 1. SUMMARY OF PIREP WEATHER PHENOMENON ELEMENTS

Phenomenon	TEI	Examples
Turbulence	TB	/TB LGT 040, /TB MOD CHOP 220
Air Temperature	TA	/TA 08, /TA M08
Icing	IC	/IC LGT-MOD MX 085, /IC SEV CLR 035-062
Sky Condition	SK	/SK OVC-TOP085, /SK OVC015-TOP035
Wind Direction and Speed	WV	/WV 28080KT, /WV 010105KT
Visibility and Weather	WX	/WX FV20SM, /WX FV10SM HZ

Every PIREP includes the message type, time (TM), location (OV), altitude (FL), and aircraft type (TP) for the reporting aircraft. For this work, these elements are considered to describe the “header” for the PIREP. The message type is either routine (UA) or urgent (UUA). A PIREP is labeled urgent if it reports tornadoes, funnel clouds, waterspouts, severe or extreme turbulence, severe icing, hail, low-level wind shear, volcanic ash, or any other phenomenon considered hazardous by the flight services’ specialist. Time is reported in Universal Time, Coordinated (UTC) and represents when the reported phenomenon was encountered. The location is described in reference to a navigational aid or airport (or two such points in the case of route segments). The flight level is reported in hundreds of feet. Aircraft type is reported using the appropriate designator [6]. If the PIREP is reported by a pilot who has been certified as a “SKYSPOTTER” (one who has received specialized training), ‘/AWC’ is appended to the PIREP.

Each element, except the leading message type element, is preceded by a solidus (/), thus cleanly dividing the various elements into distinct fields. Each element is officially referred to as a Text Element Identifier (TEI).

The rest of the PIREP contains information about the weather phenomena being reported. Those phenomena are divided into one of six categories. A seventh “remarks” category describes additional details or phenomena not covered by the six categories. These categories are summarized in Table 1 and further detailed in the sections below.

**2.1.1 Turbulence Element**

The turbulence (TB) TEI has up to three components. The only required field is the intensity described as light, moderate, severe, or extreme. The abbreviations LGT, MOD, SEV, and EXTRM, respectively, are used to denote the intensities. These intensities may also be combined to show a range of intensity, such as LGT-MOD. If the pilot reports it, the PIREP will indicate if the turbulence is clear air turbulence (CAT), choppy (CHOP), or both. If the encounter with turbulence occurs at a different altitude than indicated in the /FL TEI, the /TB element may also include altitude information. For the two TB examples in Table 1, the first shows a report of light turbulence occurring at 4,000 feet, and the second shows moderate, choppy turbulence at 22,000 feet.

**2.1.2 Air Temperature Element**

Air temperature (TA) is reported in degrees Celsius. The value is always described with two digits. For negative values, an “M” is prepended.

### **2.1.3 Icing Element**

Icing (IC) is described with an intensity, type, and optional flight level. The intensity is indicated as TRACE, LGT, MOD, or SEV. The type is rime, clear, or mixed, indicated by RIME, CLEAR, or MX, respectively. When icing is reported, the pilot is also required to supply a TA report, however this requirement is often overlooked in practicality.

### **2.1.4 Sky Condition Element**

The Sky Condition (SK) TEI reports on cloud coverage. Descriptions of the cloud cover as broken, few, overcast, scattered, or sky clear are abbreviated as BKN, FEW, OVC, SCT, and SKC, respectively. In addition to this qualitative description, a range of flight levels (or just a top) is also provided.

### **2.1.5 Wind Vector Element**

The Wind Vector (WV) TEI describes both the direction and speed of the wind at the reported time and location. The first three digits are used for the magnetic direction (000-359), and the next two to three digits indicate the speed in knots. A zero is prepended to a wind speed less than 10 knots. Finally, the characters “KT” are appended to the element.

### **2.1.6 Weather Element**

The WX element describes flight visibility, flight weather, or both. Visibility, if reported, is provided in statute miles (SM) and is indicated with a preceding “FV” followed by a two-digit value followed by “SM.” If visibility is unrestricted, a value of 99 is indicated. For weather types, the Federal Aviation Administration (FAA) provides a set of abbreviations to describe 40 phenomena such as rain, drifting sand, shallow fog, and others. As an example, in Table 1 there is a report for haze (HZ). For precipitation phenomena, an intensity value may be indicated with a ‘-’ or ‘+’, describing light and heavy precipitation, respectively. For extreme weather phenomena, additional information may be required within the WX TEI or the remarks section. An altitude range may also be provided.

### **2.1.7 Example PIREPs**

To provide a complete idea of what all of these elements look like in an actual PIREP, four examples, each with a brief description, are provided below.

1. UA /OV ILI047045/TM 0013/FL210/TP SF34/TA M23/IC MOD RIME
2. UUA /OV OMN115095/TM 0027/FL360/TP B737/TB SEV
3. UA /OV FSD/TM 0236/FL100/TP PAT4/SK UNKN050-TOP067/TA M08/IC LGT MX
4. UA /OV PIR/TM 0107/FL110/TP C208/WX -SN/TA M10/IC NEG

The first PIREP is a routine PIREP (UA) reported at 47 degrees and 45 nmi from ILI (ILI047045) at 0013Z. The pilot was at flight level 210 in a Saab 340 turboprop (SF34). The pilot reported an air temperature of  $-23^{\circ}\text{C}$  (M23) and moderate rime icing (MOD RIME).

The second PIREP was urgent (UUA) reported at 115 degrees and 95 nmi from OMN (OMN115095) at 0027Z. The pilot was at flight level 360 in a Boeing 737. The report contains only one element: severe turbulence (TB SEV). Severe turbulence reports necessitate the tagging as urgent.

The third PIREP was reported over FSD at 0236Z while flying at flight level 100. From a Piper T-1040 (PAT4), the pilot reported a sky condition of “unknown” from flight levels 050 to 067 (UNKN050TOP067). The air temperature was  $-8^{\circ}\text{C}$  (M08), and there was light mixed (rime and clear) icing (LGT MX).

The fourth and final PIREP shown here was flying over PIR at 0107Z. The pilot’s flight level was 110 in a Cessna 208 Grand Caravan (C208) when there was a report of light snow (-SN). There was no icing (IC NEG) and the air temperature was  $-10^{\circ}\text{C}$  (M10).

## **2.2 Aviation Surface Weather Reports**

The Surface Weather Observation Program collects and disseminates Aviation Routine Weather Reports (METARs) and Aviation Selected Special Weather Reports (SPECIs) using a set of observation stations managed by the National Weather Service. The stations may be automated or human operated, but the key weather elements that are provided within a METAR or a SPECI are essentially the same and are guided by rigid and formal standards. This section provides an overview of these surface weather reports. The interested reader is directed to the Federal Meteorological Handbook [7] from which all the information in this section was derived.

Because the content of a METAR and a SPECI is essentially the same, the following references to METARs assume that both message types apply unless explicitly stated otherwise. Each METAR consists of exactly two major parts: a Body and a Remarks section. The Body contains up to 11 groups of data:

1. Type or report
2. Station identifier
3. Time stamp
4. Report modifier
5. Wind
6. Visibility
7. Runway visual range
8. Present weather
9. Sky condition
10. Temperature and dew point
11. Altimeter

The first four elements in the preceding list could be considered a header that provides information describing the type of report (METAR or SPECI), station identifier (for example, an airport identifier), the time and date of the report, and whether the report is fully automated or manually corrected. The remaining elements are descriptions of actual observed weather conditions. Any element may be omitted from any given METAR. A brief description of some of the weather elements is provided in the following sections, and a few examples are shown in Table 2 to help ground the discussion. The Remarks section has two categories.

TABLE 2. SUBSET OF AVAILABLE METAR WEATHER ELEMENTS

Phenomenon	Examples
Wind	04007KT, 20003KT
Visibility	10SM, 3/4SM
Runway Visual Range	R09R/1200V3000FT, R35L/P6000FT
Present Weather	-DZ, -RA BR

The first is “Automated, Manual, and Plain Language Remarks.” Typically these remarks relate to the information provided in the body by elaborating or further detailing the various weather phenomena. They may also provide more emergent messages such as volcanic activity/eruptions and funnel cloud activity. The second category includes “Additive and Maintenance” data. This includes additional meteorological data relating to precipitation (amount of precipitation at various timescales, snow depth, water equivalent of snow), cloud types, sunshine, hourly temperature, and dew point, among others. This second category also potentially contains information about the station itself, namely if it is in need of maintenance and the status of its sensors.

### 2.2.1 METAR Wind

The wind group describes the horizontal wind velocity at the station location. At a minimum, the wind group typically contains the direction and speed. In addition, any gusts, variations, shifts, and peak speeds may also be reported depending on conditions and the capabilities of the reporting station. Gusts are defined as a rapid fluctuation in wind speed with a variation from the reported speed of at least 10 knots. Peak wind speed is the maximum instantaneous wind speed since the last routine METAR. A wind shift is a change in direction of at least 45 degrees while maintaining wind speeds of at least 10 knots. A wind speed of less than 6 knots is considered variable, as is a wind speed of at least 6 knots with the wind varying by at least 60 degrees.

### 2.2.2 METAR Visibility

Visibility is provided in statute miles (SM). Automated stations provide visibility values in 1/4 SM steps up to 2 SM, then in 1/2 SM steps up to 3 SM, then in 1 SM steps up to 10 SM. Manual visibility reports may vary in 1/16 SM steps up to 3/8 of an SM and then in 1/8 SM steps up to 2 SM, then in 1/4 SM steps up to 3 SM, then in 1 SM steps up to 15, and then in 5 SM steps thereafter. A modifier ‘M’ may be prepended to indicate “less than” for the provided measurement, usually on the minimal reportable value. For example, for visibility less than 1/4 SM, an automated station would report “M 1/4.”

### 2.2.3 METAR Runway Visual Range

“The runway visual range (RVR) is the maximum distance at which the runway, or the specified lights or markers delineating it, can be seen from a position above a specified point on its center line” [7]. RVR is measured in feet. Up to 1,000 feet, the steps are 100 feet, then up to 3,000 feet a step size of 200 feet is used, and finally, values between 3,000 and 6,000 use a step size of 500 feet. The runway number is also included with the RVR, separated from the value with a solidus ‘/’. If the RVR is varying over the report period, the range of variation is reported using a ‘V’.

QUALIFIER		WEATHER PHENOMENA		
INTENSITY OR PROXIMITY 1	DESCRIPTOR 2	PRECIPITATION 3	OBSCURATION 4	OTHER 5
- Light	<b>MI</b> Shallow	<b>DZ</b> Drizzle	<b>BR</b> Mist	<b>PO</b> Well-Developed Dust/Sand Whirls
Moderate <sup>2</sup>	<b>PR</b> Partial	<b>RA</b> Rain	<b>FG</b> Fog	<b>SQ</b> Squalls
+ Heavy	<b>BC</b> Patches	<b>SN</b> Snow	<b>FU</b> Smoke	<b>VA</b> Volcanic Ash
<b>VC</b> In the Vicinity <sup>3</sup>	<b>DR</b> Low Drifting	<b>SG</b> Snow Grains	<b>DU</b> Widespread Dust	<b>FC</b> Funnel Cloud Tornado Waterspout <sup>4</sup>
	<b>BL</b> Blowing	<b>IC</b> Ice Crystals	<b>SA</b> Sand	<b>SS</b> Sandstorm
	<b>SH</b> Shower(s)	<b>PL</b> Ice Pellets	<b>HZ</b> Haze	<b>DS</b> Duststorm
	<b>TS</b> Thunderstorm	<b>GR</b> Hail	<b>PY</b> Spray	
	<b>FZ</b> Freezing	<b>GS</b> Small Hail and/or Snow Pellets		
		<b>UP</b> Unknown Precipitation		

1. The weather groups shall be constructed by considering columns 1 to 5 in the table above in sequence, i.e., intensity, followed by description, followed by weather phenomena, e.g., heavy rain shower(s) is coded as +SHRA  
2. To denote moderate intensity no entry or symbol is used.  
3. See paragraph 8.4.1.a.(2), 8.5, and 8.5.1 for vicinity definitions.  
4. Tornadoes and waterspouts shall be coded as +FC.

Figure 1. METAR present weather values [7].

### 2.2.4 METAR Present Weather

The present weather section describes “precipitation, obscurations, well-developed dust/sand whirls, squalls, tornadic activity, sandstorms, and dust storms. Present weather may be evaluated instrumentally, manually, or through a combination of instrumental and manual methods” [7]. The list of codes to successfully describe this collection of meteorological activities is quite long. The list of these activities is essentially the set of all legal combinations of the elements shown in Figure 1, which is cropped directly from the National Oceanic and Atmospheric Administration (NOAA) manual [7] regarding METARs.

### 2.3 Weather Polygons

Weather phenomena are typically described in relation to geographic locations. These locations are often times areas that are affected by the phenomena being described. As such, geographic polygons with associated data are a natural way to fully describe the weather phenomena.

Some examples from the Alaska Aviation Weather Unit are provided next. The first, Figure 2, is a graphic describing the forecast for icing over Alaska. The second, Figure 3, shows a similar graphic for the current turbulence prediction. Finally, Figure 4 provides the prediction of conditions in terms of flight rules: Instrument Flight Rules (IFR), Visual Flight Rules (VFR), or Marginal VFR (MVFR). Note that areas in Figure 4 are VFR except where designated as IFR or MVFR specifically.

The important thing to note from each of these images is that they can be described with a set of polygons and associated data. For example, the icing forecast would need the vertices (or appropriate approximations thereof) or each polygon along with the descriptor for each polygon. In the legends of Figures 2 and 3, “ISOL MOD” means “isolated moderate” and “MOD ISOL

SEV” means “moderate isolated severe.” The diagrams also include appropriate time information (issue time and valid times).

There are several weather phenomena (thunderstorm activity, volcanic activity, and many others) that fit this data description model of polygon-description-times.

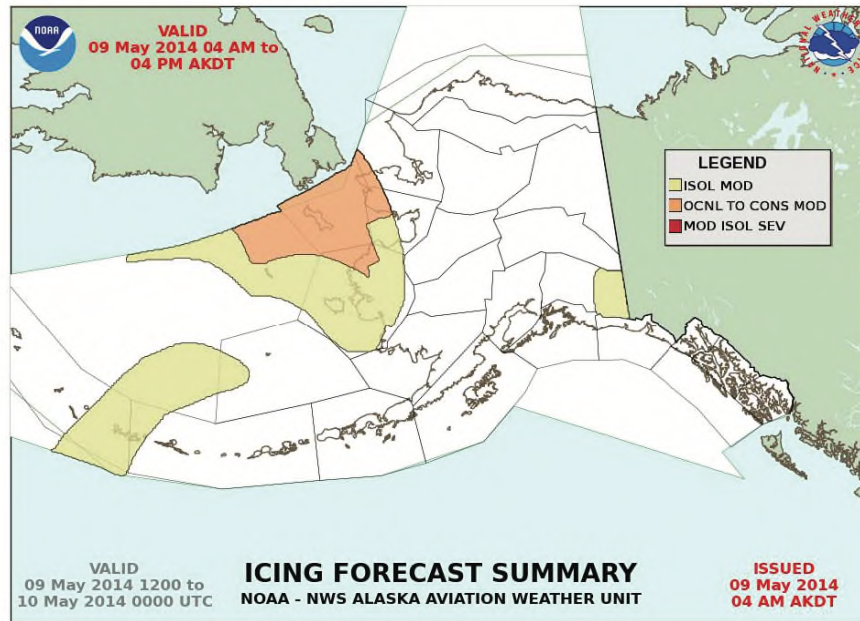


Figure 2. Icing prediction from the Alaska Aviation Weather Unit.

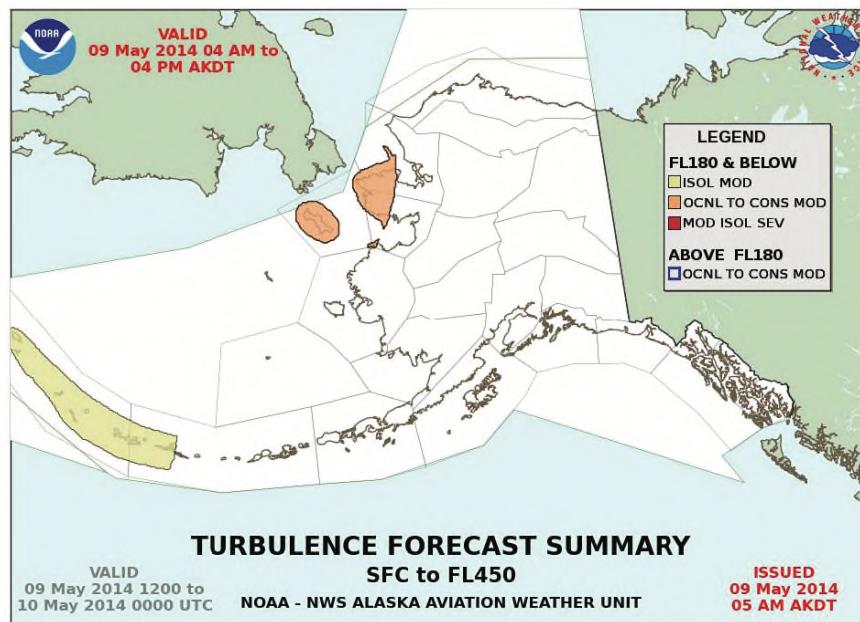


Figure 3. Turbulence prediction from the Alaska Aviation Weather Unit.



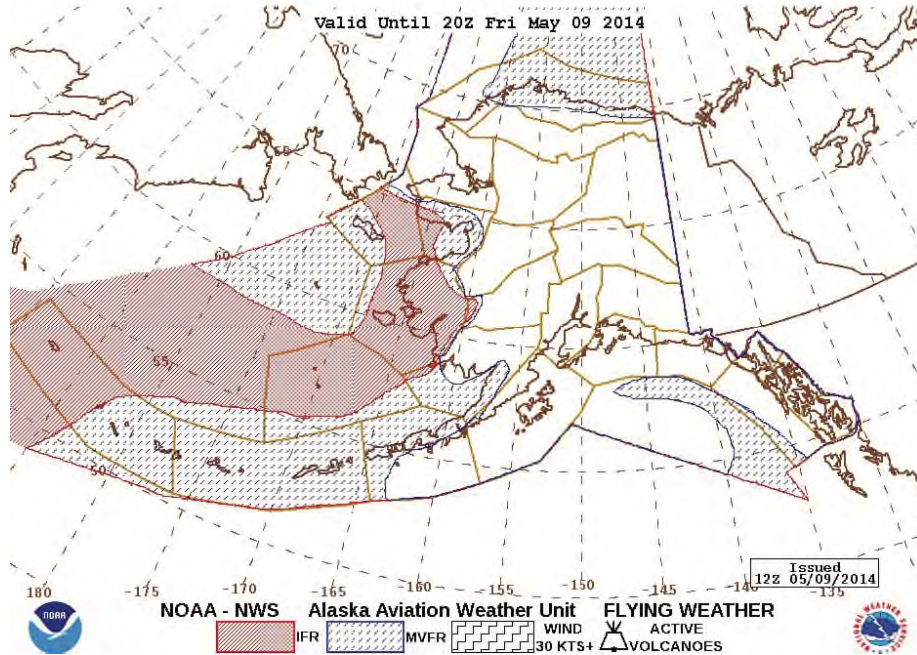


Figure 4. Flight category prediction from the Alaska Aviation Weather Unit.

Certain aviation elements can also be described with this pattern. For example, a Special Use Airspace could use this data description pattern as well.

### 3. Abstract Syntax Notation One

To fully describe Abstract Syntax Notation One (ASN.1) would require several chapters of a textbook. As such, the interested reader is directed to the works of Dubuisson [8] and Larmouth [9] for a more complete description. A brief description is provided here to ground the future discussion of these implementations.

ASN.1 is a notation system to formally describe data messages for exchange between computer systems. ASN.1 can be a means to completely describe a messaging protocol. Several widely adopted protocols are formally described with ASN.1 including Radio-Frequency Identification (RFID), Lightweight Directory Access Protocol (LDAP), and Aeronautical Telecommunication Network (ATN), among others. ASN.1 also provides a set of encoding rules to create well-defined instances of messages adhering to the protocol's ASN.1 specification. Simply stated, ASN.1 provides the rules for describing the content of messages as well as the specific rules for encoding those messages as bits, bytes, or strings, depending on the type of encoding desired.

One of the benefits of using ASN.1 is the fact that it frees the protocol developer and protocol implementer from designing the specification of the messaging application as well as the encodings for the implementation of the messaging application. Various encoding rules associated with ASN.1 include basic encoding rules (BER), XML encoding rules (XER), and packed encoding rules (PER), among others. BER is based on collections of triplets of the form (Type, Length, Value) that are ultimately encoded as octets (bytes). XER encodes the data as XML. PER is the most efficient in terms of message length as all data are put into binary form taking advantage of look-up tables for constrained data and potentially (in the unaligned version

of PER or “UPER”) ignoring the concept of bytes to write data values across byte boundaries without “wasted” bits within bytes. UPER will be used to efficiently encode the various data in this study.

To provide some additional context, the following example (modified from Oliver Dubuisson’s ASN.1 book [8]) may be illustrative.

```
Payment-method ::= CHOICE {
    check          NumericString (SIZE (15)),
    credit-card    Credit-card,
    cash           NULL }

Credit-card ::= SEQUENCE {
    type           Card-type,
    number         NumericString (SIZE (20)),
    code           INTEGER (0..999) OPTIONAL ,
    expiration-date UTCTime }

Card-type ::= ENUMERATED { mastercard(0), visa(1), eurocard(2), diners (3),
    american-express (4), other (5) }
```

This example describes a payment system. The top level element is a choice of payment methods (check, credit card, or cash). The “check” option only requires the number on the check as further data, and the cash option requires no further data. Note that the “credit card” option is described by another data structure because there are several data elements needed to describe it. If the Payment-method is a credit card, then the type of the card, the number on the card, the code on the back of the card, and the expiration date are all needed. The Card-type is an enumeration of options, the number is a string of 20 digits, and the code on the reverse is optional and must be between 0 and 999. It is through a set of such definitions that one formalizes a data description in ASN.1. With such a formalization, one is able to encode instances of such data elements using the various encoding schemes discussed above. A particular payment method by credit card could be encoded as an XML document (XER encoding rules for ASN.1), squeezed down to a bit string as tightly as possible (UPER encoding rules for ASN.1), or some other encoding as allowed within the ASN.1 specification.

## 4. Data Descriptions

The formal ASN.1 specifications for TAIGA are provided in Appendix A. This section discusses those specifications with an emphasis on the differences between TAIGA and existing descriptions of the same data. Of particular interest are the differences between the formal, operational data descriptions for PIREPs and METARs and why those differences exist in the TAIGA ASN.1 specification. In general, differences only exist for the sake of efficient transmission of the various data. If data compression was not the central concern, the most basic ASN.1 specification for the TAIGA data would be to encode, say, a PIREP as a string and just transmit the existing PIREP as that string without altering it. That naive approach would not allow for any compression of the original data. By understanding the redundancies and other qualities of the data, a specification was created that allows for highly efficient data encoding.

## 4.1 Encoding Locations

Location information is central to describing any weather or aviation phenomenon. In many aviation applications, locations are described by named infrastructure like airports or navigational aids (NAVAIDs). For example, a PIREP is described relative to such locations in terms of a radial (direction in degrees from true north) and a distance in nautical miles. So a PIREP reported 31 nautical miles due west (270 degrees) from Ted Stevens International Airport in Anchorage (ANC) would be described as ANC270031 in the PIREP. For METARs the location is indicated only by a station identifier. Transmitting just those data would require a client system to have access to all known named locations and their mapping to latitude-longitude (lat-lon) values for visualization on a modern user interface.

This approach to describing locations also varies significantly from the approach used to describe other aviation-related features, such as weather prediction polygons and the like. For example, to describe a turbulence forecast, typically there would be a set of vertices describing the polygon that will be affected. These vertices would be described as lat-lon pairs, with no reference to aviation infrastructure. The messaging scheme presented here attempts to unify location descriptions for more consistent data encoding and efficiency in message delivery.

The chosen approach is called “Geohashing” and it is a well-known method of encoding lat-lon pairs. The approach bisects the globe laterally and longitudinally a set number of times. Depending on which side of the bisection the target point lies, a 1 or 0 (also known as a ‘bit’) is used to designate that location. The more bisections that are done, the more accurate the geohash encoding and the longer the string of bits used in the encoding. After the desired accuracy is obtained, the bits for the horizontal and vertical bisections are interleaved. The reason for interleaving is that any bits removed from the tail of the string will leave the location information intact, but with less accuracy/resolution. The new interleaved string of bits is then grouped into sets of 5 bits each. Those 5 bits are mapped to an alphabet of 32 characters (recall that  $2^5 = 32$ ). The resulting character string is how the location data are ultimately encoded and transferred.

In Appendix A, Section A.2, the characters for this encoding are shown and the above description is provided again. In Appendix B, Section B.3, an example of encoded points of various resolutions is provided.

## 4.2 Encoding Time

Temporal information is also a key feature of most aviation data. As such, data related to time are present in nearly all the information that may be transmitted using the encodings described in this document. To make the encoding and transmission of temporal data as efficient as possible, a “reference time and offset” approach is taken with this system. For each message a reference time is provided in the header. This reference time is always in UTC. Rather than provide a date with each message, a reference day is provided. To provide a date, one needs to be able to encode 31 values for the day of the month, potentially 12 values for the month, and some value to represent the year. Even if the decision was to encode just the day of the month, that would require 5 bits of information, whereas to encode a choice of 7 days of the week requires only 3 bits. Because nearly all (potentially all) of the data is temporary in nature and does not have a shelf life over 24 hours, using the day of the week seems appropriate enough. If examples appear in the future where this type of encoding becomes problematic, the issue can be revisited.

Given the reference time (and day) provided in the header, all times that are used in any payload are described relative to this reference time. As such, the reference time needs to be the earliest time that might be needed by any data element across all the payloads in a given message. After that is determined, when a time is described, only the number of 10-minute steps (or ‘ticks’) since the reference time are provided. For example, if the reference time is 0340 UTC, and a PIREP is reported for 0412UTC, that second time would be encoded as three 10-minute ticks since the reference time. Note that this introduces some loss in precision and only a guarantee that any given time is within 5 minutes of its actual time. For aviation data (specifically aviation weather) information, this level of precision seems appropriate.

By using 10-minute ticks as the time description and using the assumption that 24 hours is the longest amount of time needed to encode for the data items in this schema, only 144 10-minute ticks are needed to cover that 24-hour period. This allows any time within any payload to be encoded with just 7 bits.

### **4.3 ASN.1 PIREP Data**

The initial phase of this work focused on developing a method for compressing the data described by PIREPs. This was done without the benefit of ASN.1. In that work [10], compression ratios under 0.20 were achieved, which means that only 20 percent of the original bits needed to encode the data were required in the new approach. By extending that work to include several other data types (METARs, weather polygons, etc.) under a common ASN.1 specification, a complete messaging system for general aviation can be developed.

As described in the original PIREP compression paper, the approach is a lossy compression approach when compared to a raw PIREP. This loss in information is designed to be minimal in impact to the end user (pilot) by maintaining the most important elements of the PIREP while working to minimize data redundancies. The differences that occur between raw PIREP and a PIREP encoded via the proposed ASN.1 specification are detailed in Table 3. The reasons behind these differences are discussed more qualitatively here.

The differences in what we refer to as the PIREP header information that is common to all PIREPs (location, time, flight level, and aircraft type) are as follows. The location is described as a latitude and longitude pair rather than a reference point together with an optional radial and distance. This frees any client from needing to have a database of reference points (airports and NAVAIDs) to calculate a precise location on a map. For interacting and reading data on an electronic map, this seems like a reasonable approach. Theoretically, if a database of reference points were available on the client device, a radial and distance from a reference fix could be calculated from a lat-lon pair. This would be useful if pilots wanted to describe the point verbally or see a textual description not overlaid on a base map. Like most of the time stamps in this proposed specification, the time for a given PIREP is actually an offset from a time provided in the main header for the broadcast message. The offset steps are in 10-minute increments, thus the accuracy of the time will be within 10 minutes of the actual PIREP report time. The flight level is provided in hundreds of feet as is the standard for describing flight levels, however, this specification limits this (and all) flight level description to a maximum of flight level 511. In aviation most aircraft typically fly below 50,000 feet, so this upper bound seems reasonable.

TABLE 3. COMPARISON OF PIREP ENCODINGS

Element	Federal Standard [5]	Difference in TAIGA Encoding
Type	Urgent or Routine	–
Location	NAVAID or airport identifier, with potential radial and distance or route segment of two such points	Latitude-Longitude pair with sub 0.1 km precision. No segments allowed.
Time	4-digit UTC time	Time accurate within 5 minutes.
Flight Level	Altitude in hundreds of feet, potential DURC (during climb) or DURD (during descent) modifier in remarks. Potentially UNKN.	Same, with flight level limit of 511 and no UNKN indicator. Suggest using value of “511” as indicator for “UNKN.”
Aircraft Type	Code for aircraft type [6].	Class of reporting aircraft within set of 8 classes.
Sky Condition	Cloud cover descriptor, base altitude, top, optional “sky clear” indicator, optional “in clouds” indicator (IMC noted in remarks section).	Same, but TAIGA limits flight levels of tops and bases to 511. Also, multiple layers must be reported as separate elements, which would be equivalent to having multiple ‘/SK’ elements in a regular PIREP.
Flight Visibility and Weather	Visibility reported in statute miles from 0 to 99. Weather type with intensity and optional altitude.	Same, but no optional altitude allowed. Multiple weather types need to be reported as separate elements.
Temperature	Air temperature using two digits.	Same, with limits of –60°C and 67°C.
Wind Vector	Wind direction and speed.	Same elements, but direction is one of 16 major compass rose directions and wind speed limited to 255 knots.
Turbulence	Intensity, duration, type (CAT or CHOP), and altitude of turbulence event.	Range of intensity (e.g., LGT-MOD) limited to neighboring values (e.g., cannot have LGT-SEV). Multiple reports in single PIREP need to be reported as separate elements. Altitude limited to flight level 511 and below. Cannot encode “ABV” or “BLO” indicators.
Icing	Intensity, type, and altitude of icing event.	Range of intensity (e.g., LGT-MOD) limited to neighboring values (e.g., cannot have LGT-SEV). Multiple reports in single PIREP need to be reported as separate elements. Altitude limited to flight level 511 and below. Cannot encode “ABV” or “BLO” indicators.
Remarks	Important phenomena and other information not within standard TEIs.	–

The aircraft type is much different in this specification than it is in the traditional PIREP. Instead of providing an exact aircraft type based on the FAA descriptions of aircraft [6], this approach classifies the aircraft into one of eight classes based on size. Initial, informal discussions with pilots indicate that this may be reasonable as the primary goal for knowing the aircraft type is to know the effect of a particular phenomenon (such as turbulence) on the aircraft. The savings in terms of data compactness are tremendous with this approach. To describe an aircraft as belonging to one of eight classes takes exactly three data bits. To describe an aircraft textually with up to 4 characters would take up to 28 bits, which is nearly an order of magnitude more data to be transmitted. As a concrete example, consider a Piper Super Cub. The code for that aircraft is “PA18” which takes 7 bits per character to transmit (28 bits total), but if the class of that aircraft is described as “very small,” which might have class value of “1,” it would require only 3 bits to transmit.

Now for the weather phenomena of the PIREP. First, Sky Condition is described with the same data as a traditional PIREP with the limitation on the flight level (511 is the maximum). The Sky Condition element may contain multiple layers in a traditional PIREP. To achieve that in the ASN.1 encoding, multiple Sky Condition elements would need to be included, one for each reported layer.

Flight visibility and weather are completely encoded in the ASN.1 scheme. The only difference, again, is that multiple phenomena need to be included as additional “WX” elements. Temperature is properly encoded as well, with the limitation that temperatures must lie between  $-60^{\circ}\text{C}$  and  $67^{\circ}\text{C}$ . This limitation allows for encoding the temperature in seven bits because the range is 127 degrees.

The wind vector element is correctly encoded with a slight loss in accuracy. In a traditional PIREP, the direction is encoded to the nearest 10 degrees, but in the ASN.1 scheme, the nearest of the 16 major compass rose directions is used. Thus, to translate between the two, the nearest compass rose direction to the provided PIREP value must be determined. This results in a maximum of 10 degrees of error, with an average error of just 5.6 degrees. When discussing wind directions, which are already somewhat uncertain, this error seems like it should be tolerable for aviation purposes. Pilots, in general, would like to know the basic direction of the wind, but high precision is not of great concern. By limiting to the 16 compass rose directions, the wind direction can be encoded in 4 bits, versus the 6 bits it would take to encode the 36 10-degree values normally used in PIREP reporting. This is a 50-percent savings in transmission cost for this particular datum.

Turbulence is mostly correctly encoded, but is again limited to altitudes below FL511. Also, there is no mechanism to indicate “ABV” or “BLO” when discussing altitude ranges. This can be overcome by using extreme values as indicators. For example, turbulence from FL020 and below can be indicated with a range from FL000 to FL020. The only other limitation in encoding a turbulence element is that ranges of intensity must be composed of only neighboring values. For instance, a range of light to moderate is acceptable, but the ASN.1 specification cannot encode a range of light to severe. This would be a rare event and would likely be well-served by using the more intense value anyway. The icing element has the same limitations as the turbulence element and no other limitations.

The remarks section can encode all values described in the FAA standard [5]. Specifically, it can handle low-level wind shear, funnel clouds, thunderstorms, lightning, electric discharge, clouds, and plain language description of other phenomena. As a rule, implementors of this system should be aware of the high cost of remarks that may have little value to pilots. Sending plain language messages should be a last resort as they will be the most costly messages and will quickly overwhelm the compression savings gained elsewhere in the encoding scheme. The important part for this discussion, however, is that remarks can always be sent as the original “RM” string whenever needed and can encode the important events (LLWS, FC, etc.) when needed.

#### **4.4 ASN.1 METAR Data**

The limitations in encoding standard METAR messages are similar to those seen in the PIREP encoding. It is hoped that these limitations are minor and will not have a significant effect on the utility of the messages. The differences include directionality limited to 16 compass rose directions, time accurate within 10 minutes, and a few visibility values being unencodable (2 1/2 SM, for example). The details of the differences are provided in Table 4.

#### **4.5 ASN.1 Description of Other Data**

There are other message types available that encode some less standardized data elements. For example, this message schema can send image data when necessary (see “ImageMessageDefinition” in the Section A.5). An “Emergency” message (Section A.6) can also be sent that simply includes a text string along with an issue time and an optional end time for the emergency.

A route message (Section A.7) is also defined such that general information about recommended routes can be broadcast. This utility lets pilots in a given area, flying between locations with relatively high operation counts, know the current recommended route as determined by the service provider. This message type is based on the FAA’s route advisory mechanism that controls flows through and around congested or capacity-limited areas [11]. It provides a reason for the route, the affected facilities, flights that should be included or excluded, and a set of routes (in the form of open polygons in this schema).

Weather polygons may also be described in this schema through the WeatherPolygon message (Section A.8). This message type provides for the times associated with a weather polygon (valid times and forecast times as appropriate) and includes a text string for remarks, the lat-lon points describing the polygon, and some other optional elements. This could be useful in describing forecasts from, say, the Alaska Aviation Weather Unit (icing, turbulence, IFR/VFR conditions), volcanic ash activity, observed precipitation, and any other general phenomena affecting (or planned to affect) a defined area.

TABLE 4. COMPARISON OF METAR ENCODINGS

Element	Federal Standard [7]	Difference in TAIGA Encoding
Type	METAR or SPECI	–
Station ID	4-letter code	Same, but limited to AK
Date and Time	2-digit date, hours, minutes	Day of week, time accurate within 5 minutes
Report Modifier	AUTO or COR or none	–
Wind	Direction within 10 degrees, speed, gust, variability.	Same elements, but direction is one of 16 major compass rose directions and wind speed limited to 255 knots.
Visibility	Fractional and/or integer values, “less than” allowed.	Strictly fractional or integer (e.g., 2 1/2 SM is impossible, largest value is 15, “less than” not allowed.
Runway Visual Range	Runway number, runway position, lowest value, highest value, “plus/minus” modifiers.	–
Present Weather	Up to three groups, which each may include intensity/proximity, descriptor, and phenomenon.	One group, otherwise same. TAIGA organizes combinations of descriptor and phenomenon into a comprehensive enumerated list. This list is believed to be complete, but requires verification.
Sky Condition	Automated stations report up to three layers. Each layer is either sky cover amount and height; vertical visibility; or clear.	Same, but TAIGA does not distinguish between SKC and CLR (manual versus automated stations).
Temperature and Dew Point	Two measures in degrees Celsius rounded to nearest integer.	Same, with limits of –60°C and 67°C.
Altimeter	Four digits representing tens, units, tenths, and hundredths of inches of mercury.	–
Remarks	Several potential fields related to more detailed or important data not covered in fields within the main body. Rather than list all the dozens of potential fields here, the reader is referred to the appropriate METAR documentation [7] and just the fields that have been encoded in this ASN.1 scheme are provided. Note that any field may be encoded, but with each added field, there is additional coding and data transmission overhead for all METAR data. For each field that is potentially encoded, there must be a bit transmitted with every METAR that indicates if that field is present.	Peak wind, wind shift, variable visibility, sector visibility, hourly temperatures, sea level pressure, and a maintenance flag are all currently encoded. Hourly temperatures are rounded to nearest degree.



Weather cameras are of particular interest to Alaskan pilots. The FAA's weather camera project produces live or near-live images of various important and/or remote locations of interest to pilots in Alaska. The WxCam message (Section A.9) in this schema allows for a synthesized description of the weather cam images, not the images themselves. The synthesized information could include visibility and cloud cover height. In the future, other items may be added. These data could be generated automatically based on the images themselves, or input manually by humans processing the images subjectively. These data could prove useful to pilots without access to the Internet to fetch these images themselves.

Finally, messages that might be used for system maintenance or testing can also be encoded (Section A.10). These might include data or code for updating clients that are active in the field. Likely, the first use of this message type would be to send test messages to clients.

## **5. Concluding Remarks**

The specification of aviation-related data into an Abstract Syntax Notation One format allows for efficient and standardized encodings. These encodings are ideal for the transmission of data, especially in throughput-limited scenarios. This work demonstrates the potential for using such a specification and subsequent encoding to reduce transmitted file sizes by up to 80 percent in the case of PIREPs with similar reductions for other data types. These results open the possibility of economically using traditionally expensive communication channels like satellite systems for increasing the situational awareness of pilots in remote locations. These pilots may not have access to any other means of information gathering at various points in their journey, so any additional data could prove useful (even vital), aiding pilots in making more informed decisions. The specification presented here will hopefully enter operational testing in Alaska.

## 6. References

1. Mode, N.A.; O'Connor, M.B.; Conway, G.A.; and Hill, R.D.: A Multifaceted Public Health Approach to Statewide Aviation Safety. *American Journal of Industrial Medicine*, vol. 55, no. 2, 2012, pp. 176–186.
2. O'Connor, M.; Lincoln, J.; and Conway, G.A.: Occupational Aviation Fatalities—Alaska, 2000–2010. *Centers for Disease Control and Prevention Morbidity and Mortality Weekly Report*, vol. 60, no. 25, July 2011, pp. 837–840.
3. Stapleton, D. and Cieplak, J.: Alaska's Capstone Program—Systems Engineering for Communication, Navigation and Surveillance. *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, vol. 1, Oct. 2004, pp. 1.D.2–1.1–6.
4. National Transportation Safety Board: NTSB Most Wanted List 2014—General Aviation: Identify and Communicate Hazardous Weather. <http://www.nts.gov/mostwanted>, accessed May 2014, 2014.
5. U.S. Department of Transportation, Flight Services – JO 7110.10V, Federal Aviation Administration, Washington, DC, July 2012.
6. U.S. Department of Transportation, Contractions – JO 7340.2C, Federal Aviation Administration, Washington, DC, Feb. 2012.
7. Office of the Federal Coordinator for Meteorological Services and Supporting Research, U.S. Department of Commerce, Washington, DC: *Federal Meteorological Handbook No. 1: Surface Weather Observations and Reports*, Sept. 2005.
8. Dubuisson, O.: *ASN.1: Communication Between Heterogeneous Systems*. OSS Nokolva, 2000.
9. Larmouth, J.: *ASN.1 Complete, Open Systems Solutions*, 1999.
10. Rios, J. and de la Cruz, M.: Compression of PIREPs for Throughput-Limited Transmission. *Aviation Technology, Integration, and Operations Conference, AIAA*, Aug. 2013.
11. U.S. Department of Transportation, Facility Operation and Administration – JO 7210.3Y, Federal Aviation Administration, Washington, DC, April 2014.

## **Appendix A**

### **ASN.1 Source Files**

This appendix provides copies of the ASN.1 source files described earlier in this document. These files are provided for reference purposes only and are current only at the time of publication. For the most recent versions of the ASN.1 source files, interested readers should contact the author. Each source file is presented on a new page.

## A.1 Overall Message Definition

```
-- File name:      TAIGAMessageDefinition.asn1
-- Author:        Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:  Aviation Systems Division,
--               NASA Ames Research Center,
--               Moffett Field, CA
-- Created:       Tue Nov 26 14:05:40 PST 2013

TAIGAMessageDefinition DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS ALL;
IMPORTS Pirep, AircraftType          FROM PirepMessageDefinition
       Image                        FROM ImageMessageDefinition
       System                       FROM SystemMessageDefinition
       WeatherPolygon              FROM WxPolyDefinition
       Emergency                   FROM EmergencyMessageDefinition
       Metar                       FROM MetarMessageDefinition
       RouteMessage                FROM RouteMessageDefinition
       WxCam                       FROM WxCamMessageDefinition
       TenMinuteTicks, Day, Time   FROM CommonTypeDefinition;

-----
--
--               TAIGAMessage
--
--   A message consists of a time, a day,
--   and a payload.  Each part
--   of the payload may vary in type.
-----

TAIGAMessage ::= SEQUENCE {
    reference-time    Time,
    day              Day,
    payload-sequence SEQUENCE (SIZE (0..3,...)) OF Payload
}

-- Keep the choices to a power of 2?
-- 8 choices seem good?
Payload ::= CHOICE {
    wxcam      PayloadWxCam,
    sytem      PayloadSystem,
    emergency  PayloadEmergency,
    image      PayloadImage,
    pirep      PayloadPirep,
    metar      PayloadMetar,
    wxpoly     PayloadWxPoly,
    route      PayloadRoute
}

PayloadWxCam ::= SEQUENCE (SIZE (0..31)) OF WxCam
PayloadSystem ::= SEQUENCE (SIZE (0..3)) OF System
PayloadEmergency ::= SEQUENCE (SIZE (0..3)) OF Emergency
PayloadImage ::= SEQUENCE (SIZE (0..3)) OF Image
```

```
PayloadPirep      ::= SEQUENCE (SIZE (0..31)) OF Pirep
PayloadMetar      ::= SEQUENCE (SIZE (0..31)) OF Metar
PayloadWxPoly     ::= SEQUENCE (SIZE (0..15)) OF WeatherPolygon
PayloadRoute      ::= SEQUENCE (SIZE (0..3)) OF RouteMessage
END
```

## A.2 Common Data Elements

```
-- File name:      TAIGAImageDefinition.asn1
-- Author:        Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:  Aviation Systems Division,
--               NASA Ames Research Center,
--               Moffett Field, CA
-- Created:       Wed Nov 27 09:54:51 PST 2013

CommonTypeDefinition DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS ALL;

-- A polygon in this system will have at most 63 points.
-- If more points are needed, then an additional Polygon
-- should be created. Note that there are no restrictions
-- on the concavity/convexity of the polygons. If a shape
-- requires a rounded edge, then that shape would need to
-- be approximated with one of these Polygons.
Polygon      ::= SEQUENCE (SIZE (LongPoly)) OF GeohashString
LongPoly     ::= INTEGER(0..63)

-- A GeohashString encodes a single point (lat/lon pair).
-- For a description of how geoahsing works and examples, please see:
-- http://en.wikipedia.org/wiki/Geohash
-- http://wiki.xkcd.com/geohashing/Main\_Page
-- http://openlocation.org/geohash/geohash-js/
-- In a nutshell, geohashing works by iteratively bisecting the globe,
-- alternating between vertical and horizontal cuts. Depending on
-- which side of each cut the point under consideration falls, you
-- encode a 0 or 1. As you string more of these together, you get
-- improving resolution on the point. For every 5 bits that are
-- encoded, you map that 5-bit string to one of 32 letters and then
-- encode groups of those letters together to ultimately describe the
-- point. For this project we've chosen to encode either 4, 5, or 7
-- letters for resolutions of 20km, 2.4km, or 0.076km respectively.
-- Note that these are the maximum errors that occur at the equator.
-- These errors decrease as you approach the poles.
GeohashString ::=
    IA5String (FROM("0123456789bcdefghjkmnpqrstuvwxyz")) (SIZE (4|5|7))
GeohashStringHigh ::=
    GeohashString (SIZE (7)) -- +/- 0.076km error at the equator
GeohashStringMed  ::=
    GeohashString (SIZE (5)) -- +/- 2.4km error at the equator
GeohashStringLow  ::=
    GeohashString (SIZE (4)) -- +/- 20km error at the equator

-- Time will be assumed to be UTC in all messages within
-- this system. These two integers should be more compact
-- than a formal UTCTime data element.
Time ::= SEQUENCE {
    hour    INTEGER(0..23),
    minute  INTEGER(0..59)
}

-- These are used to act as a positive offset to a provided
```

```

-- Time instance.  There are only 144 ten-minute timesteps
-- in 24 hours.
TenMinuteTicks ::= INTEGER(0..144)

-- All practical commercial and general aviation takes place
-- below 51,100 feet, so this limit seems reasonable.  In
-- aviation, flight levels are commonly described using the
-- altitude/100.  So "flight level 210" corresponds to an
-- altitude of 21,000 feet.
FlightLevel ::= INTEGER(0..511)

-- This will be used frequently in many contexts.
AltitudeLevelRange ::= SEQUENCE {
    baseAltitude    FlightLevel,
    ceilingAltitude FlightLevel
}

-- These are based on the standard cloud coverage values used
-- in METARs and PIREPs.
SkyCloudCover ::= ENUMERATED {
    none,
    bkn,
    few,
    ovc,
    sct,
    skc,
    not-supplied,
    vv
}

-- This is used in combination with the Time element
-- in lieu of a proper date in order to take care the
-- most common time interpretation issues (times near
-- midnight probably).
Day ::= ENUMERATED {
    sunday,
    monday,
    tuesday,
    wednesday,
    thursday,
    friday,
    saturday
}

-- We could use an integer between 0 and 359, but this
-- encoding uses fewer bits and provides an appropriate
-- resolution for describing wind direction since such
-- measurements are not extremely accurate in the first
-- place and pilots do not require high precision to make
-- informed decisions.  152 degrees versus 148 degrees
-- will not make a difference to most pilots.
WindDirection ::= ENUMERATED {
    n,
    e,
    w,
    s,

```

```

    ne,
    nw,
    se,
    sw,
    nne,
    nnw,
    sse,
    ssw,
    wsw,
    ese,
    ene,
    wnw
}

-- If we want or need a precise value (within 1 degree) for
-- Wind Direction, we can use this.  Currently (5/1/2014)
-- not used anywhere?
WindDirectionDegrees ::= INTEGER(0..359)

-- Wind speeds will not reach 255.  If they did
-- ever happen to surpass that value, cutting it
-- off at 255 will be plenty enough warning that
-- there probably shouldn't be any flying going
--
WindSpeed ::= INTEGER(0..255)

WindReport ::= SEQUENCE {
    direction WindDirection,
    speed      WindSpeed
}

DirectionOfMovement ::= SEQUENCE {
    fromDirection WindDirection,
    toDirection   WindDirection
}

-----
--
--          Station Identifier
--
-----

-- The station identifier should ultimately be an enumeration
-- of the possible stations?  The list is quite long (150+)
-- for AK and would be extremely long for the CONUS.
StationIdentifier ::= ENUMERATED {
    padk,
    pafm,
    pakp,
    panc,
    palh,
    pamr,
    pani,
    pant,
    panv,
    parc,

```



paak ,  
pabr ,  
paba ,  
pabe ,  
pabt ,  
palv ,  
pabv ,  
pabl ,  
patw ,  
palu ,  
paeh ,  
pacz ,  
parl ,  
palr ,  
pajc ,  
pacr ,  
pacd ,  
pacv ,  
pasc ,  
pade ,  
pabi ,  
padl ,  
paeg ,  
pasy ,  
paii ,  
paei ,  
pael ,  
paed ,  
paem ,  
pazk ,  
pafa ,  
pafk ,  
pafb ,  
pfyu ,  
pafn ,  
pagb ,  
paga ,  
pagm ,  
pagl ,  
pagk ,  
pags ,  
pahn ,  
pahz ,  
pahv ,  
paho ,  
paoh ,  
pahp ,  
pahs ,  
pahy ,  
pail ,  
paim ,  
pajn ,  
pafe ,  
pakv ,  
paen ,  
pakt ,

pakn ,  
kqrc ,  
pavl ,  
pakw ,  
padq ,  
paot ,  
pakk ,  
paku ,  
paml ,  
pamx ,  
pamc ,  
pain ,  
pamy ,  
pamm ,  
pamd ,  
pamh ,  
pabn ,  
pann ,  
paom ,  
paec ,  
paor ,  
pawp ,  
paqt ,  
poli ,  
paaq ,  
paxk ,  
papg ,  
papo ,  
ppiz ,  
paap ,  
palj ,  
papc ,  
paph ,  
pato ,  
papr ,  
paud ,  
papt ,  
pard ,  
pasm ,  
pasd ,  
pasa ,  
pask ,  
paso ,  
pawd ,  
pash ,  
pasi ,  
pagy ,  
pasw ,  
padt ,  
pasl ,  
palk ,  
pasx ,  
pasv ,  
papb ,  
pasn ,  
paer ,

```
pajv,  
pasp,  
patk,  
pata,  
patl,  
patc,  
patg,  
paum,  
paun,  
padu,  
pavd,  
pavw,  
pawi,  
paws,  
pawr,  
pauo,  
paiz,  
pawg,  
paya,  
...  
}
```

```
-- This is a range of 127 and will be encoded with 7 bits using  
-- the PER. This range covers all practical Celcius values for  
-- aviation purposes. If a temperture somehow needs to fall out  
-- of this range, it would be effectively the same as providing  
-- the extreme values. For example if there was a need to report  
-- a temperature of 72 degrees celcius, that is for all practical  
-- purposes the same as reporting 67 degrees celcius, i.e. it is  
-- pretty dang hot. Likewise for lower temps, if we needed to  
-- report -70 degrees celcius, it would be just as effective to  
-- report -60 for aviation purposes.  
Temperature ::= INTEGER(-60..67)
```

```
END
```

### A.3 METAR Messages

```
-- File name:      TAIGAMetarDefinition.asn1
-- Author:         Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:   Aviation Systems Division,
--                 NASA Ames Research Center,
--                 Moffett Field, CA
-- Created:        Mon Dec 02 15:41:18 PST 2013

MetarMessageDefinition DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS ALL;
IMPORTS Temperature, WindReport, TenMinuteTicks,
        WindDirection, SkyCloudCover, StationIdentifier,
        WindSpeed FROM CommonTypeDefinition
        PirepWeatherMetarCode FROM PirepMessageDefinition;

-----
--                                     --
--                 METAR                 --
--                                     --
--   The formatting of this message     --
--   type follows the standard formatting --
--   of METARs as described in:         --
--   Surface Weather Observations and   --
--   Reports by Dept of Commerce/NOAA   --
-----

Metar ::= SEQUENCE {
    metarType    MetarType,
    station      StationIdentifier,
    time         TenMinuteTicks, -- offset from the header time
    isAuto       BOOLEAN,
    isCorrected  BOOLEAN,
    wind         MetarWind,
    visibility   MetarVisibility,
    rvr          MetarRvr OPTIONAL,
    weather      MetarWeather OPTIONAL, -- not always present
    sky          MetarSkyConditions,
    temperature  MetarTemperature,
    altimeter    MetarAltimeter,
    remarks      MetarRemarks OPTIONAL
}

-----
--                                     --
--                 Metar Type           --
--                                     --
-----

MetarType ::= ENUMERATED {
    metar,
    speci
}

-----
--                                     --
```

```

--                               Wind                               --
--                               --                               --
--                               --                               --
-----
MetarWind ::= SEQUENCE {
    windReport    WindReport,
    gust          WindSpeed OPTIONAL,
    variable      MetarVariableWind OPTIONAL
}

MetarVariableWind ::= SEQUENCE {
    directionA    WindDirection,
    directionB    WindDirection
}

-----
--                               Visibility                       --
--                               --                               --
--                               --                               --
-----
-- If the visibility is fractional, then the value
-- member will represent binary fractions, otherwise,
-- it will represent a standard integer from 0 to 15.
-- Binary fraction place values would be:
--
-- 0.5  0.25  0.125  0.0625
--
-- So, the value of 3/4 would be represented
-- by 1100 (Integer value 12).
MetarVisibility ::= SEQUENCE {
    isFractional  BOOLEAN,
    value         INTEGER(0..15)
}

-----
--                               Runway Visual Range           --
--                               --                               --
--                               --                               --
-----
MetarRvr ::= SEQUENCE {
    runway        INTEGER(1..36),
    position      MetarRunwayPosition,
    range         MetarRunwayVisualRange
}

MetarRunwayPosition ::= ENUMERATED {
    left,
    right,
    mid,
    none
}

MetarRunwayVisualRange ::= SEQUENCE {

```

```

        valueA    MetarRvrValue ,
        valueB    MetarRvrValue OPTIONAL
    }

MetarRvrValue ::= SEQUENCE {
    modifier    MetarRvrValueModifier OPTIONAL,
    value       MetarRvrValueInt
}

MetarRvrValueModifier ::= ENUMERATED {
    m,
    p
}

-- This type of description for RVR can't be handled
-- well in ASN.1. Use enumeration instead.
--MetarRvrValueInt ::= INTEGER (600|700|800|900|
--    1000|1200|1400|1600|1800|2000|2200|2400|2600|
--    2800|3000|3500|4000|4500|5000)
MetarRvrValueInt ::= ENUMERATED {
    e600 ,
    e700 ,
    e800 ,
    e900 ,
    e1000 ,
    e1200 ,
    e1400 ,
    e1600 ,
    e1800 ,
    e2000 ,
    e2200 ,
    e2400 ,
    e2600 ,
    e2800 ,
    e3000 ,
    e3500 ,
    e4000 ,
    e4500 ,
    e5000
}

-----
--                                     --
--                               Weather                               --
--                                     --
-----

MetarWeather ::= SEQUENCE {
    intensityVicinity    MetarWxIntensityVicinity,
    weather               PirepWeatherMetarCode
}

MetarWxIntensityVicinity ::= ENUMERATED {
    plus,
    minus,
    vicinity,

```

```

    none
}

-----
--
--           Sky Conditions
--
--
-----

MetarSkyConditions ::= SEQUENCE (SIZE(0..3)) OF SkyConditionLayer

SkyConditionLayer ::= CHOICE {
    regular SkyConditionLayerRegular,
    vertical SkyConditionLayerVertical,
    clear    BOOLEAN
}

SkyConditionLayerRegular ::= SEQUENCE {
    skyCloudCover    SkyCloudCover,
    cloudCoverHeight CloudCoverHeight
}

SkyConditionLayerVertical ::= SEQUENCE {
    cloudCoverHeight CloudCoverHeight
}

CloudCoverHeight ::= CHOICE {
    upto5000    CcUpto5000,
    upto10000   CcUpto10000,
    over10000   CcOver10000
}

-- From FAA METAR documentation:
-- Table 14-6: Increments of Reportable Values of Sky Cover Height
-- Range of Heights (feet) Reportable Values (feet)
-- 5,000 or less To nearest 100
-- >5,000 but <=10,000 To nearest 500
-- Above 10,000 To nearest 1,000

CcUpto5000 ::= INTEGER(0..50) -- Multiply by 1
CcUpto10000 ::= INTEGER(11..20) -- Multiply by 5
CcOver10000 ::= INTEGER(11..42) -- Multiply by 10

-----
--
--           Temperature
--
--
-----

MetarTemperature ::= SEQUENCE {
    temperature Temperature,
    dewPoint      Temperature
}

-----
--
--

```

```

--          Altimeter          --
--          --                  --
--          -----
-- Value is multiplied by 100. Divide by 100 to get actual value.
MetarAltimeter ::= INTEGER(0..9999)

-----
--          Remarks          --
--          --                  --
--          -----
MetarRemarks ::= SEQUENCE {
    peakWind      MetarPeakWind OPTIONAL,
    windShift     MetarWindShift OPTIONAL,
    -- alternateViz MetarVisibility OPTIONAL,
    variableViz   MetarVariableVisibility OPTIONAL,
    sectorViz     MetarSectorVisibility OPTIONAL,
    hourlyTemp    MetarTemperature OPTIONAL,
    seaLevelPressure INTEGER(0..201) OPTIONAL,
    maintenance   BOOLEAN,
    ...
}

MetarPeakWind ::= SEQUENCE {
    windReport    WindReport,
    time          TenMinuteTicks
}

MetarWindShift ::= SEQUENCE {
    hour          INTEGER(0..23),
    minute        INTEGER(0..59),
    hasFropa      BOOLEAN
}

MetarVariableVisibility ::= SEQUENCE {
    visibilityA   MetarVisibility,
    visibilityB   MetarVisibility
}

MetarSectorVisibility ::= SEQUENCE {
    direction     WindDirection,
    visibility     MetarVisibility
}

END

```



## A.4 PIREP Messages

```
-- File name:      TAIGAPirepDefinition.asn1
-- Author:        Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:  Aviation Systems Division,
--               NASA Ames Research Center,
--               Moffett Field, CA
-- Created:       Wed Nov 27 09:51:09 PST 2013

PirepMessageDefinition DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS Pirep, PirepWeatherMetarCode, PirepSkyCover;
IMPORTS GeohashStringHigh, FlightLevel,
        AltitudeLevelRange, WindDirection,
        Temperature, WindReport, DirectionOfMovement,
        SkyCloudCover FROM CommonTypeDefinition;

-----
--
--               Pilot Weather Report
--               (PIREP)
--
-----

Pirep ::= SEQUENCE {
    isUUA          BOOLEAN,
    isAWC          BOOLEAN,
    flightLevel    FlightLevel,
    duringClimb    BOOLEAN,
    duringDescent  BOOLEAN,
    aircraft       AircraftType,
    timeOffset     INTEGER(0..31),
    position       GeohashStringHigh,
    elementSequence SEQUENCE (SIZE (0..31)) OF PirepElement
}

AircraftType ::= ENUMERATED {
    tiny,
    very-small,
    small,
    medium,
    medium-large,
    large,
    heavy,
    jumbo
}

PirepElement ::= CHOICE {
    turbulence PirepElementTurbulence,
    temperature PirepElementTemperature,
    icing       PirepElementIcing,
    sky         PirepElementSkyCondition,
    wind        PirepElementWindVector,
    weather     PirepElementWeather,
    remark      PirepElementRemark,
    reserved    PirepElementReserved
}
```

```

-----
--                                     --
--           Turbulence Element         --
--           /TB                        --
--                                     --
-----
PirepElementTurbulence ::= SEQUENCE {
    intensity    PirepTurbulenceIntensity,
    cat          BOOLEAN,
    chop         BOOLEAN,
    duration     PirepTurbulenceDuration,
    flightLevel AltitudeLevelRange OPTIONAL
}

PirepTurbulenceIntensity ::= ENUMERATED {
    lgt,
    lgt-mod,
    mod,
    mod-sev,
    sev,
    sev-extrm,
    extrm,
    neg
}

PirepTurbulenceDuration ::= ENUMERATED {
    cont,
    ocnl,
    intmt,
    none
}

-----
--                                     --
--           Air Temperature Element    --
--           /TA                        --
--                                     --
-----
PirepElementTemperature ::= SEQUENCE {
    temperature  Temperature
}

-----
--                                     --
--           Icing Element              --
--           /IC                        --
--                                     --
-----
PirepElementIcing ::= SEQUENCE {
    intensity    PirepIcingIntensity,
    type        PirepIcingType,
    flightLevel AltitudeLevelRange OPTIONAL
}

PirepIcingIntensity ::= ENUMERATED {

```

```

    trace,
    trace-lgt,
    lgt,
    lgt-mod,
    mod,
    mod-sev,
    sev,
    neg
}

PirepIcingType ::= ENUMERATED {
    none,
    clear,
    rime,
    mixed
}

-----
--                               --
--           Sky Condition Element           --
--           /SK                               --
--                               --
-----

PirepElementSkyCondition ::= SEQUENCE {
    skyClearAbove      BOOLEAN,
    inCloudsIMC       BOOLEAN,
    cloudCoverRange   PirepSkyCloudCoverRange,
    flightLevel       AltitudeLevelRange
}

PirepSkyCloudCoverRange ::= SEQUENCE {
    cloudCoverA       PirepSkyCloudCover,
    cloudCoverB       PirepSkyCloudCover
}

PirepSkyCloudCover ::= SkyCloudCover

-----
--                               --
--           Wind Vector Element           --
--           /WV                               --
--                               --
-----

PirepElementWindVector ::= WindReport

-----
--                               --
--           Weather Element           --
--           /WX                               --
--                               --
-----

PirepElementWeather ::= SEQUENCE {
    visibility   PirepWeatherVisibility OPTIONAL,
    phenomenon   PirepWeatherPhenomenon OPTIONAL
}

```

```

PirepWeatherVisibility ::= INTEGER(0..127)

PirepWeatherPhenomenon ::= SEQUENCE{
    intensity    PirepWeatherIntensity,
    metar-code   PirepWeatherMetarCode
}

PirepWeatherIntensity ::= ENUMERATED {
    plus,
    minus,
    none
}

PirepWeatherMetarCode ::= ENUMERATED {
    drsn,    -- Drifting snow
    blsn,    -- Blowing snow
    drdu,    -- Drifting dust
    drsa,    -- Drifting sand
    dz,      -- Drizzle
    fzdz,    -- Freezing drizzle
    du,      -- Dust
    bldu,    -- Blowing dust
    ds,      -- Duststorm
    fg,      -- Fog
    fzfg,    -- Freezing fog
    fzra,    -- Freezing rain
    fc,      -- Funnel cloud
    gr,      -- Hail (1/4" diameter or more)
    shgr,    -- Hail Shower
    hz,      -- Haze
    ic,      -- Ice crystals
    pl,      -- Ice pellets
    shpl,    -- Ice pellet showers
    br,      -- Mist (visibility 5/8sm or more)
    bcfg,    -- Patchy fog
    prfg,    -- Patchy fog on part of airport
    ra,      -- Rain
    shra,    -- Showers
    sa,      -- Sand
    blsa,    -- Blowing sand
    ss,      -- Sandstorms
    mifg,    -- Shallow fog
    shgs,    -- Small hail/snow pellets showers
    gs,      -- Small hail/show pellets
    fu,      -- Smoke
    sg,      -- Snow grains
    sn,      -- Snow
    shsn,    -- Snow showers
    py,      -- Spray
    sq,      -- Squalls
    ts,      -- Thunderstorm
    -- +fc,   Tornado/Waterspout can't be represented in ASN.1,
    --       but can be encoded using intensity '+' and fc
    up,      -- Unknown precipitation
    va,      -- Volcanic ash
    po,      -- Well-developed dust/sand whirils
}

```

```

    imc,      -- Instrument meteorological conditions
    vmc,      -- Visual meteorological conditions
    clr,      -- Clear
    tsra,
    tssn,
    tspl,
    tsgr,
    tsgr,
    ...
}

-----
--                                     --
--          Remark Element             --
--          /RM                         --
--                                     --
-----
PirepElementRemark ::= SEQUENCE {
    text          UTF8String OPTIONAL,
    llws          PirepLowLevelWindShear OPTIONAL,
    fc            PirepFunnelCloud OPTIONAL,
    ts            PirepThunderstorm OPTIONAL,
    lightning     PirepLightning OPTIONAL,
    discharge     AltitudeLevelRange OPTIONAL,
    clouds        UTF8String OPTIONAL,
    ...
}

PirepLowLevelWindShear ::= SEQUENCE {
    plus          BOOLEAN,
    minus         BOOLEAN,
    altitude      AltitudeLevelRange OPTIONAL,
    text          UTF8String OPTIONAL
}

PirepFunnelCloud ::= SEQUENCE {
    type          PirepFunnelCloudType,
    direction     DirectionOfMovement OPTIONAL
}

PirepFunnelCloudType ::= ENUMERATED {
    tornado,
    waterspout,
    funnelcloud
}

PirepThunderstorm ::= SEQUENCE {
    coverage      PirepThunderstormCoverage,
    description   PirepThunderstormDescription OPTIONAL,
    direction     DirectionOfMovement OPTIONAL
}

PirepThunderstormCoverage ::= ENUMERATED {
    isol,
    few,
    sct,

```

```

        nmrs
    }

    PirepThunderstormDescription ::= ENUMERATED {
        ln,
        bknl,
        sldln
    }

    PirepLightning ::= SEQUENCE {
        frequency    PirepLightningFrequency,
        ltgic        BOOLEAN OPTIONAL,
        ltgcc        BOOLEAN OPTIONAL,
        ltgcg        BOOLEAN OPTIONAL,
        ltgca        BOOLEAN OPTIONAL
    }

    PirepLightningFrequency ::= ENUMERATED {
        ocnl,
        frq,
        cons,
        not-reported
    }

-----
--                               --
--                               --
--                               --
--                               --
-----

    PirepElementReserved ::= SEQUENCE {
        ...
    }

    END

```

## A.5 Image Messages

```
-- File name:      TAIGAImageDefinition.asn1
-- Author:        Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:  Aviation Systems Division,
--               NASA Ames Research Center,
--               Moffett Field, CA
-- Created:       Wed Nov 27 12:48:56 PST 2013

ImageMessageDefintion DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS ALL;
IMPORTS WindDirectionDegrees,
        GeohashStringLow,
        GeohashStringMed,
        GeohashStringHigh FROM CommonTypeDefinition;

-----
--                               --
--               Image           --
--                               --
--   If bandwidth and ecomonics allow, --
--   this message type will provide a  --
--   means to transmit an image as a   --
--   string of bytes (OCTET STRING)    --
-----

Image ::= SEQUENCE {
    description UTF8String,
    image       OCTET STRING,
    location    ImageLocation OPTIONAL,
    direction   WindDirectionDegrees OPTIONAL,
    ...
}

ImageLocation ::= CHOICE {
    hiFidelityPt GeohashStringHigh,
    mdFidelityPt GeohashStringMed,
    loFidelityPt GeohashStringLow
}

END
```

## A.6 Emergency Messages

```
-- File name:      TAIGAEmergencyDefinition.asn1
-- Author:        Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:  Aviation Systems Division,
--               NASA Ames Research Center,
--               Moffett Field, CA
-- Created:       Mon Dec 02 15:11:47 PST 2013

EmergencyMessageDefinition DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS Emergency;
IMPORTS TenMinuteTicks FROM CommonTypeDefinition;

-----
--                               --
--               Emergency       --
--                               --
--   This message type could be used   --
--   for urgent of emergency messages  --
--   that do not fit any other category. --
-----

Emergency ::= SEQUENCE {
    message      UTF8String,
    issueTime    TenMinuteTicks,
    endingTime   TenMinuteTicks OPTIONAL,
    ...
}

END
```



## A.7 Route Messages

```
-- File name:      TAIGARouteDefinition.asn1
-- Author:        Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:  Aviation Systems Division,
--               NASA Ames Research Center,
--               Moffett Field, CA
-- Created:       Mon Mar 03 11:26:37 PST 2014

RouteMessageDefinition DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS RouteMessage;
IMPORTS AltitudeLevelRange, StationIdentifier, Polygon FROM CommonTypeDefinition;

-----
--
--               Route
--
--   If there were a need to transmit
--   a route definition to an aircraft
--   or group of aircraft, one could do
--   so with this message type. This is
--   roughly modeled upon the way the FAA
--   defines reroute advisories, but does
--   not have an immediate, current
--   analog in Alaska.
-----

RouteMessage ::= SEQUENCE {
    type                RouteType OPTIONAL,
    reason              RouteReason OPTIONAL,
    comment             UTF8String OPTIONAL,
    timeInclusion        RouteTimeType OPTIONAL,
    typeInclusion        RouteAircraftType OPTIONAL,
    depFacilities       SEQUENCE (SIZE (0..7)) OF StationIdentifier OPTIONAL,
    arrFacilities       SEQUENCE (SIZE (0..7)) OF StationIdentifier OPTIONAL,
    altInclusion         AltitudeLevelRange OPTIONAL,
    typeExclusion        RouteAircraftType OPTIONAL,
    altExclusion         AltitudeLevelRange OPTIONAL,
    route              SEQUENCE (SIZE (0..3)) OF Polygon
}

RouteType ::= ENUMERATED {
    recommended,
    required,
    informational,
    other
}

RouteReason ::= ENUMERATED {
    weather,
    equipment,
    military,
    other
}

RouteTimeType ::= ENUMERATED {
```

```
        etd,  
        eta,  
        fca,  
        other  
    }  
  
RouteAircraftType ::= ENUMERATED {  
    all-aircraft  
}  
  
END
```

## A.8 Weather Polygon Messages

```
-- File name:      TAIGAWxPolyDefinition.asn1
-- Author:        Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:  Aviation Systems Division,
--               NASA Ames Research Center,
--               Moffett Field, CA
-- Created:       Mon Dec 02 13:41:28 PST 2013

WxPolyDefinition DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS WeatherPolygon;
IMPORTS GeohashPointHighFidelity, GeohashPointLowFidelity,
        GeohashPointMedFidelity, TenMinuteTicks,
        AltitudeLevelRange, WindDirection,
        Polygon FROM CommonTypeDefinition;

-----
--
--               Weather Polygons
--
--   This message type allows for the
--   description of various weather
--   polygons.
-----

WeatherPolygon ::= SEQUENCE {
    type           WeatherPolygonType,
    title          UTF8String,
    polygon        Polygon,
    validTime      TenMinuteTicks,
    validTimeEnd   TenMinuteTicks OPTIONAL,
    forecastTime   TenMinuteTicks OPTIONAL,
    flightLevel    AltitudeLevelRange OPTIONAL,
    direction      WindDirection OPTIONAL,
    remarks        UTF8String OPTIONAL
}

-- These types are not codified anywhere else, but they represent
-- our best guess at the types of polygons that will be of interest
-- to users of this system. Note that this ENUMERATED type is
-- extensible and that any polygon can be described using the 'generic'
-- type with an appropriate remark in the WeatherPolygon data.
WeatherPolygonType ::= ENUMERATED {
    generic,
    turbulence,
    icing,
    precipitation,
    volcanic-ash,
    visibility,
    downslope-winds,
    ...
}

END
```

## A.9 Weather Cam Messages

```
-- File name:      TAIGAWxCamDefinition.asn1
-- Author:        Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:  Aviation Systems Division,
--               NASA Ames Research Center,
--               Moffett Field, CA
-- Created:       Tue Apr 01 11:40:39 PDT 2014

WxCamMessageDefinition DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS ALL;
IMPORTS WindDirection, GeohashStringHigh
        FROM CommonTypeDefinition
        MetarVisibility, CloudCoverHeight
        FROM MetarMessageDefinition;

-----
--                               --
--           Weather Cameras     --
--                               --
--   This message type allows for the --
--   description of data that might --
--   be available from weather cameras. --
-----

WxCam ::= SEQUENCE {
    id          WxCamIdentifier OPTIONAL, -- make this optional in case
                                                -- we're describing a non-standard
                                                -- location.

    position    GeohashStringHigh,
    direction   WindDirection,             -- the direction of the camera
    visibility   MetarVisibility,          -- this may be determined through
                                                -- analysis of the image.

    ceiling     CloudCoverHeight OPTIONAL,
    comment     UTF8String OPTIONAL,
    ...
}

-- These are all of the Alaska weather cam stations as of 10 April 2014.
WxCamIdentifier ::= ENUMERATED {
    akhiok,
    akun-island,
    allakaket,
    ambler,
    anaktuvuk-pass,
    anchor-point,
    anchorage,
    angoon,
    aniak,
    anvik,
    arctic-village,
    atqasuk,
    barrow,
    beaver,
    beluga,
    berners-bay,
```

bethel ,  
bettles ,  
birchwood ,  
black-rapids ,  
bradley-lake ,  
buckland ,  
cape-fanshaw ,  
cape-spencer ,  
cape-yakataga ,  
central ,  
chalkyitsik ,  
chandalar-shelf ,  
chefornak ,  
chevak ,  
chickaloon ,  
chignik-bay ,  
chignik-lagoon ,  
chignik-lake ,  
chilkat ,  
chistochina ,  
chitina ,  
clarks-point ,  
cold-bay ,  
coldfoot ,  
cooper-landing ,  
cordova ,  
craig ,  
crooked-creek ,  
deadhorse ,  
deering ,  
delta-junction ,  
dillingham ,  
dutch-ballyhoo ,  
dutch-haystack ,  
dutch-ndb ,  
eagle ,  
edna-bay ,  
eek ,  
egegik ,  
elim ,  
emmonak ,  
ester-dome ,  
false-pass ,  
fort-yukon ,  
galena ,  
gambell ,  
golovin ,  
grave-point ,  
grayling ,  
gulkana ,  
gustavus ,  
gustavus-dock ,  
haines ,  
hawk-inlet ,  
holy-cross ,  
homer ,

honolulu ,  
hoonah ,  
hooper-bay ,  
huslia ,  
hydaburg ,  
hyder ,  
igiugig ,  
iliamna ,  
isabel-pass ,  
johnstone-point ,  
johnstone-point-vor ,  
kake ,  
kalskag ,  
kaltag ,  
karluk ,  
kasaan ,  
kasigluk ,  
ketchikan ,  
kiana ,  
king-cove ,  
king-salmon ,  
kipnuk ,  
kivalina ,  
klawock ,  
knik ,  
knob-ridge ,  
kodiak ,  
kokhanok ,  
koliganek ,  
kotlik ,  
kotzebue ,  
koyuk ,  
kwethluk ,  
kwigillingok ,  
lake-clark-pass-east ,  
lake-clark-pass-rco ,  
lake-clark-pass-west ,  
larsen-bay ,  
lena-point ,  
level-island ,  
lime-village ,  
livengood ,  
manokotak ,  
marshall ,  
mcgrath ,  
mckinley-north ,  
mckinley-park ,  
mckinley-south ,  
mekoryuk ,  
mentasta ,  
merrill-pass-high ,  
merrill-pass-low ,  
metlakatla ,  
meyers-chuck ,  
middleton-island ,  
minchumina ,

minto ,  
misty-fjords ,  
moose-pass ,  
mountain-village ,  
nanwalek ,  
napakiak ,  
nelson-lagoon ,  
nenana ,  
new-stuyahok ,  
newtok ,  
nikiski ,  
nikolai ,  
noatak ,  
nome ,  
nondalton ,  
north-slope ,  
northway ,  
nuiqsut ,  
nulato ,  
nunapitchuk ,  
nyac ,  
old-harbor ,  
ouzinkie ,  
palmer ,  
pedersen-hill ,  
pedro-bay ,  
pelican ,  
perryville ,  
petersburg ,  
pilot-point ,  
platinum ,  
point-higgins ,  
point-hope ,  
point-lay ,  
port-alexander ,  
port-heiden ,  
port-lions ,  
portage-creek ,  
portage-glacier ,  
potato-point ,  
puntilla-lake ,  
quinhagak ,  
red-dog ,  
rohn ,  
ruby ,  
ruby-airport ,  
russian-mission ,  
savoonga ,  
scammon-bay ,  
selawik ,  
seward ,  
shageluk ,  
shaktoolik ,  
sheep-mountain ,  
shishmaref ,  
shungnak ,

```
sisters-island ,
sitka ,
skagway ,
skwentna ,
sleetmute ,
soldotna ,
st-marys ,
st-michael ,
st-paul ,
summit ,
tahneta-pass ,
takotna ,
taku-inlet ,
talkeetna ,
tanana ,
tazlina-tolsona ,
teller ,
tenakee-springs ,
thompson-pass ,
thorne-bay ,
togiak ,
tok ,
toksook-bay ,
trading-bay ,
tuluksak ,
tuntutuliak ,
uganik-bay ,
unalakleet ,
valdez ,
wainwright ,
wales ,
wasilla ,
white-mountain ,
whittier ,
willow ,
wrangell ,
yakutat ,
yukon-river-bridge ,
...
}
```

END



## A.10 System Messages

```
-- File name:      TAIGASystemDefinition.asn1
-- Author:        Joseph Rios, joseph.l.rios@nasa.gov
-- Organization:  Aviation Systems Division,
--               NASA Ames Research Center,
--               Moffett Field, CA
-- Created:       Thu Feb 13 16:39:35 PST 2014

SystemMessageDefinition DEFINITIONS AUTOMATIC TAGS ::= BEGIN

EXPORTS ALL;

-----
--                               --
--               System          --
--                               --
--   This message type amay be used for  --
--   system updates to software within the --
--   receiving system or testing the      --
--   system.                             --
-----

System ::= SEQUENCE {
    description      UTF8String,
    data             OCTET STRING
}

END
```



## Appendix B

### Encoding Examples

This appendix provides concrete encoding examples. The first is a pair of artificial weather polygons and the second is a set of real Alaskan METARs.

#### B.1 Weather Polygon Example

Because many weather products for Alaska are only provided in graphical or textual formats and not in particularly machine-readable formats, two artificial weather polygons that represent the type of data that might be encoded in this scheme have been created.

#### B.2 METAR Example

At the far northern edge of Alaska, there are four METAR stations located relatively close together. In an operational TAIGA system, the reports from these four stations might be bundled together into a single METAR update for pilots in the vicinity of those weather stations. In this example, the METARs for each of those stations are encoded using the ASN.1 scheme presented here.

Figure B1 summarizes all of the METAR data for the state of Alaska during the 2200Z hour on May 7th with an added highlight around the four aforementioned stations in the far north. Subsequently, Figure B2 shows an enlarged view of the four stations. To provide a sense of how METARs are currently encoded in text format, the text of those four particular METARs is shown below (for full interpretation, the reader is directed to NOAA's manual [7] regarding METARs.):

- PALU 072155Z AUTO 36003KT 10SM CLR M04/M06 A2993 RMK AO2 SLP135 T10411060
- PPIZ 072156Z AUTO 26006KT 10SM CLR M04/M08 A2993 RMK AO2 TSNO T10381077 SLP137 \$
- PAWI 072200Z AUTO 18008KT 10SM BKN010 BKN034 M03/M06 A2989 RMK AO2 TSNO
- PABR 072153Z 23010KT 10SM OVC007 M05/M06 A2989 RMK AO2 CIG 005V010 SLP124 T10501061

They are described in the ASN.1 specification detailed in Appendix A. Upon doing so, the data can be encoded in any of the available ASN.1 encoding schemes. For this example, the data was encoded using the XML encoding rules (XER) and the unaligned packed encoding rules

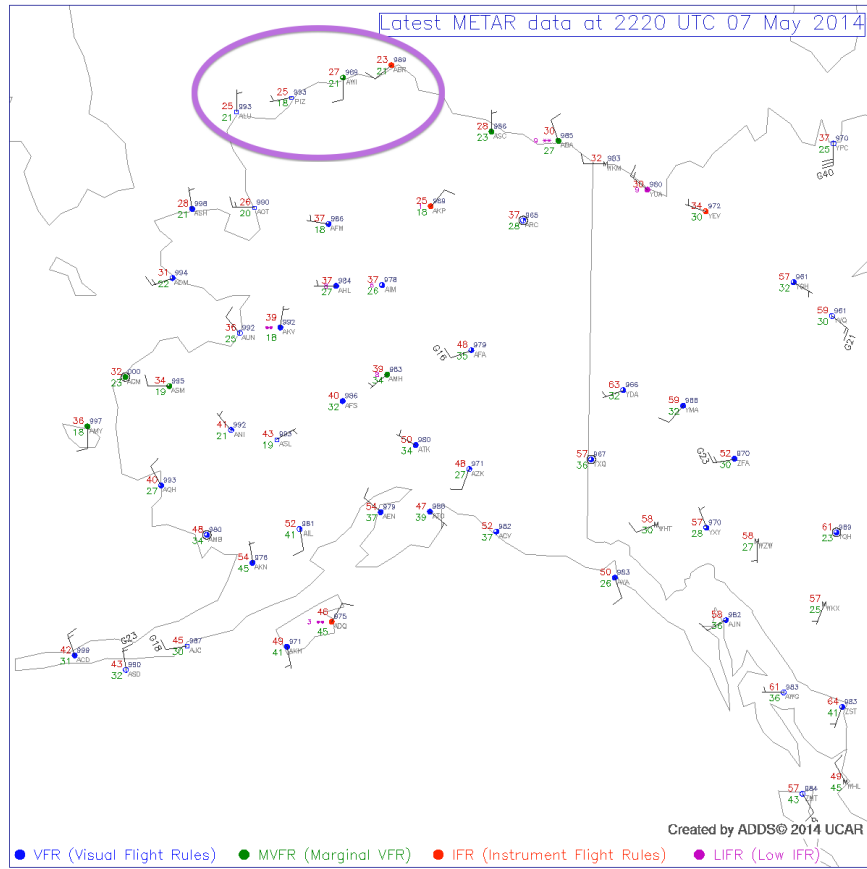


Figure B1. Graphical METAR data.

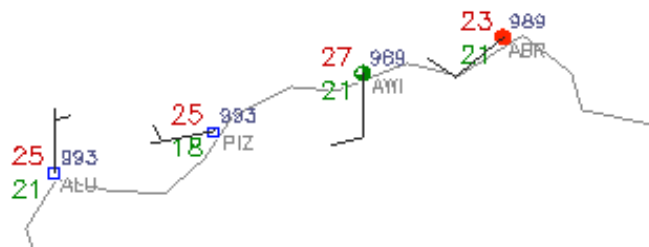


Figure B2. Detail of four METARs in northern Alaska.

(uPER). The XER allows for a human-readable format of the data. The uPER allows for a highly efficient machine-readable format.

The resulting XML is provided here in its entirety:

```
<TAIGAMessage>
  <reference-time>
    <hour>21</hour>
    <minute>50</minute>
  </reference-time>
  <day><wednesday/></day>
  <payload-sequence>
    <metar>
      <Metar>
        <metarType><metar/></metarType>
        <station><palu/></station>
        <time>0</time>
        <isAuto><true/></isAuto>
        <isCorrected><false/></isCorrected>
        <wind>
          <windReport>
            <direction><n/></direction>
            <speed>3</speed>
          </windReport>
        </wind>
        <visibility>
          <isFractional><false/></isFractional>
          <value>10</value>
        </visibility>
        <sky>
          <clear><true/></clear>
        </sky>
        <temperature>
          <temperature>-4</temperature>
          <dewPoint>-6</dewPoint>
        </temperature>
        <altimeter>2993</altimeter>
        <remarks>
          <hourlyTemp>
            <temperature>-4</temperature>
            <dewPoint>-6</dewPoint>
          </hourlyTemp>
          <seaLevelPressure>135</seaLevelPressure>
          <maintenance><false/></maintenance>
        </remarks>
      </Metar>
    <Metar>
      <metarType><metar/></metarType>
      <station><ppiz/></station>
      <time>0</time>
      <isAuto><true/></isAuto>
      <isCorrected><false/></isCorrected>
      <wind>
        <windReport>
          <direction><w/></direction>
          <speed>6</speed>
        </windReport>
      </wind>
    </Metar>
  </payload-sequence>
</TAIGAMessage>
```

```

        </windReport>
    </wind>
    <visibility>
        <isFractional><false/></isFractional>
        <value>10</value>
    </visibility>
    <sky>
        <clear><true/></clear>

    </sky>
    <temperature>
        <temperature>-4</temperature>
        <dewPoint>-8</dewPoint>
    </temperature>
    <altimeter>2993</altimeter>
    <remarks>
        <hourlyTemp>
            <temperature>-4</temperature>
            <dewPoint>-8</dewPoint>
        </hourlyTemp>
        <seaLevelPressure>137</seaLevelPressure>
        <maintenance><true/></maintenance>
    </remarks>
</Metar>
<Metar>
    <metarType><metar/></metarType>
    <station><pawi/></station>
    <time>1</time>
    <isAuto><true/></isAuto>
    <isCorrected><false/></isCorrected>
    <wind>
        <windReport>
            <direction><s/></direction>
            <speed>8</speed>
        </windReport>
    </wind>
    <visibility>
        <isFractional><false/></isFractional>
        <value>10</value>
    </visibility>
    <sky>
        <regular>
            <skyCloudCover><bkn/></skyCloudCover>
            <cloudCoverHeight>
                <upto5000>10</upto5000>
            </cloudCoverHeight>
        </regular>

        <regular>
            <skyCloudCover><bkn/></skyCloudCover>
            <cloudCoverHeight>
                <upto5000>34</upto5000>
            </cloudCoverHeight>
        </regular>

    </sky>

```

```

        <temperature>
            <temperature>-3</temperature>
            <dewPoint>-6</dewPoint>
        </temperature>
        <altimeter>2989</altimeter>
    </Metar>
    <Metar>
        <metarType><metar/></metarType>
        <station><pabr/></station>
        <time>0</time>
        <isAuto><false/></isAuto>
        <isCorrected><false/></isCorrected>
        <wind>
            <windReport>
                <direction><sw/></direction>
                <speed>10</speed>
            </windReport>
        </wind>
        <visibility>
            <isFractional><false/></isFractional>
            <value>10</value>
        </visibility>
        <sky>
            <regular>
                <skyCloudCover><ovc/></skyCloudCover>
                <cloudCoverHeight>
                    <upto5000>7</upto5000>
                </cloudCoverHeight>
            </regular>
        </sky>
        <temperature>
            <temperature>-5</temperature>
            <dewPoint>-6</dewPoint>
        </temperature>
        <altimeter>2989</altimeter>
        <remarks>
            <hourlyTemp>
                <temperature>-5</temperature>
                <dewPoint>-6</dewPoint>
            </hourlyTemp>
            <seaLevelPressure>124</seaLevelPressure>
            <maintenance><false/></maintenance>
        </remarks>
    </Metar>
</metar>

    </payload-sequence>
</TAIGAMessage>

```

Note that this is quite verbose and would not be appropriate for broadcasting over a throughput-limited communication channel. The size of this XML encoding is 6354 bytes. This could clearly be reduced by removing extraneous white space and potentially compressing the file with a standard compression algorithm or tool. Such a compression with gzip, for example, provides a file size of 657 bytes, which is nearly an order of magnitude improvement.

Encoding this example in the unaligned packed encoding rules results in a file of size 56 bytes. That is four METARs in the space normally needed to encode 56 (or potentially 64) characters. Note that the original, raw METARs are of lengths 74, 79, 72, and 84 characters, respectively. The uPER encoding offers a significant savings in data that would need to be transmitted. This is with only a minor loss in information. Specifically, some precision was lost in the time values (within 10 minutes), the tenths value in the hourly temperatures within the remarks, and potentially a few degrees in direction for each of the provided directions.

### B.3 Weather Polygon Encoding

Because many relevant weather products are not currently available in machine-readable formats, the example provided here was manually generated based on a real-world forecast provided as an image file. Section 2.3 provided an example of a turbulence forecast in Alaska (Fig. 3). That figure is provided once again here (Fig. B3) for reference purposes. Using this image, the polygon encompassing St. Lawrence Island was chosen to encode. A similar polygon was drawn using Google Earth (see Figure B4) to obtain lat-lon values. Using those lat-lon values, the appropriate geohash values were determined.

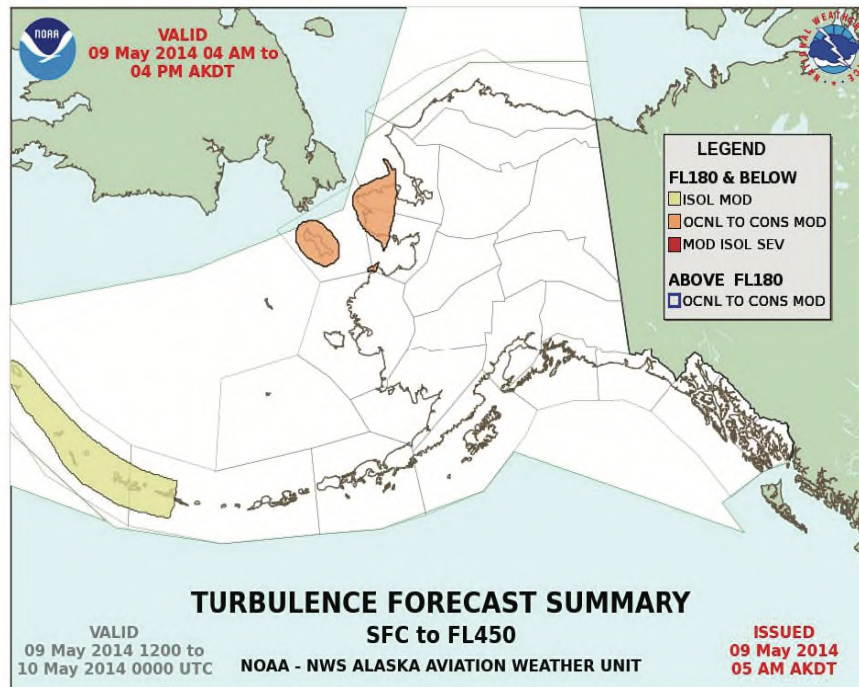


Figure B3. A turbulence forecast for Alaska.





Figure B4. Turbulence forecast polygon over St. Lawrence Island redrawn in Google Earth.

Equipped with the information provided on the original forecast image and the approximated latitude values for the polygon, the weather polygon can now be described in the new ASN.1 schema. The information from the image that is relevant includes the issue (or forecast) time (1300 UTC Wednesday), valid start time (1200 UTC Wednesday), valid end time (0000 UTC Thursday), altitudes, and predicted turbulence intensity (“OCNL TO CONS MOD”). Based on the set of times here, the reference time in the TAIGA message header should be 1200 UTC because it is the earliest time to be referenced in the message. All three times for this payload (1200 UTC, 1300 UTC, and 0000UTC the next day) will all be described in 10-minute steps from the reference time (0, 6, and 72, respectively). Upon encoding this information using the XML encoding rules (XER), the following XML document is produced.

```

<TAIGAMessage>
  <reference-time>
    <hour>12</hour>
    <minute>0</minute>
  </reference-time>
  <day><friday/></day>
  <payload-sequence>
    <wxpoly>
      <WeatherPolygon>
        <type><turbulence/></type>
        <title>OCNL TO CONS MOD</title>
        <polygon>
          <GeohashString>b5pt1qy</GeohashString>
        </polygon>
      </WeatherPolygon>
    </wxpoly>
  </payload-sequence>
</TAIGAMessage>

```

```

        <GeohashString>b70ppt1</GeohashString>
        <GeohashString>b7212p6</GeohashString>
        <GeohashString>b5q7tb5</GeohashString>
        <GeohashString>b5md2fv</GeohashString>
        <GeohashString>b5jxtrw</GeohashString>
    </polygon>
    <validTime>0</validTime>
    <validTimeEnd>72</validTimeEnd>
    <forecastTime>6</forecastTime>
    <flightLevel>
        <baseAltitude>0</baseAltitude>
        <ceilingAltitude>450</ceilingAltitude>
    </flightLevel>
</WeatherPolygon>
</wxpoly>

</payload-sequence>
</TAIGAMessage>

```

That XML data takes 1189 bytes to encode, which could be reduced as discussed in the previous section. When this same sample is encoded using the unaligned Packed Encoding Rules (uPER), the space required to store it is only 55 bytes. Note that these 55 bytes completely describe the weather polygon including its boundaries (6 vertices) and meteoroconditions associated with it. Also note that other polygons could have been included in this message, but the manual process is somewhat cumbersome. When such predictions become available in a machine-readable format in the future, conversion to this ASN.1 version can be better automated.