ISATDATA PRO

MESSAGE DEFINITION FILE DESCRIPTION

VERSION 1.1A

PUBLICATION DATE: 25-FEB-2021

Contents

1: Introduction

1.1: Scope and purpose

This document provides an overview of the IsatData Pro (IDP) Message Definition File (MDF) concept of operation and file structure.

1.2: Intended audience

This document is intended for software developers, technical users, support and system engineering resources.

1.3: Referenced Documents

- *IsatData Pro Service Description*
- *IsatData Pro Messaging API Specification*
- *IsatData Pro Administrative API Specification*

2: IsatData Pro Service

2.1: Overview

IsatData Pro (IDP) is a global L-band satellite network service optimized for Machine-to-Machine (M2M) and Internet of Things (IoT) applications. IsatData Pro is intended for event-driven data collection and remote control, but also enables applications such as forms and text messaging through Human-Machine Interfaces. Typical applications include tracking, fleet management, security, remote monitoring, telematics and SCADA.

More information about the IsatData Pro service is available in the *IDP Service Description*.

IDP communicates using short messages over a store-and-forward system architecture, with a variable size from a few bytes up to several kilobytes. Typical applications for IDP use very small messages, on average less than 20 bytes. To maximize efficiency over the satellite network and minimize cost, IDP messages are often encoded using a binary field structure that allows data to be transported with single-bit granularity. The binary encoding is more difficult for many customer applications to parse, so Inmarsat's message gateway provides an option to convert/abstract those binary-mapped fields into tagged data structures, in both directions of communication. The conversion mechanism uses a codec file or Message Definition File (MDF).

The MDF is an XML file that defines the data structure for from-mobile and to-mobile messages. The codec algorithm may be manufacturer-specific, such that the IDP terminal sending and receiving those messages also includes a compatible codec representation. An MDF file can be associated to a Mailbox on the IDP Message Gateway System to enable codec translation.

More information about receiving and sending tagged message data structures is available in the *IDP Messaging API*.

2.2: Message Structure

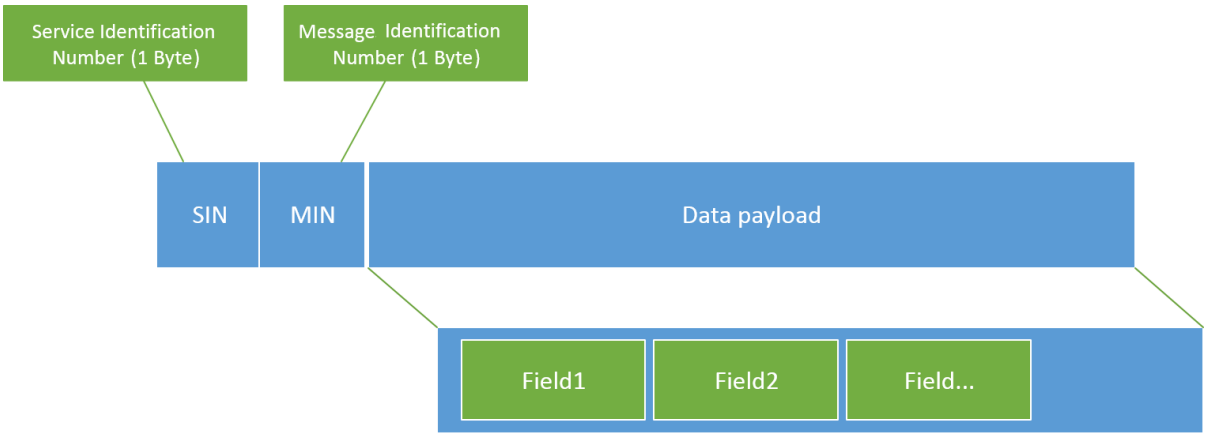


Figure 1. IsatData Pro Message Structure

The default codec algorithm relies on the use of the first two bytes of the IDP message payload, called Service Identifier Number (SIN) and Message Identified Number (MIN). The remaining payload consists of various data fields.

The SIN concept is that the service identifies a set of related functionality and operations, for example a tracking application. SIN 0-15 are reserved for system or manufacturer-specific use. Individual manufacturers may reserve additional SIN for their embedded programming environments or allow those to be used by customer applications. SIN is bidirectional in nature, shared by both from-mobile and to-mobile messages for the service in question.

The MIN concept provides for different data structures used by the service to distinguish specific operations of the service, for example a periodic location, alert notification or configuration command. MIN can be unidirectional or bidirectional, at the discretion of the embedded application designer.

Fields contain pieces of data that are defined by the application, for example latitude and longitude fields for location information.

2.3: MDF Support on the Inmarsat IDP Gateway

The Inmarsat IDP gateway supports by default the MDF for the modem messages (SIN 0 – 15). If encoding and decoding is required for other messages an additional Message Definition File can be uploaded to each mailbox.

Each mailbox can be associated with at most one Message Definition File. If multiple Message definition files need to be linked to the mailbox, they must be merged before uploading. See **Uploading the MDF to the Mailbox** on page 16 for more details.

2.4: Manufacturer Support

Inmarsat works with individual manufacturers to support various codec algorithms. At the time of publication of this document, Inmarsat supports the following codecs:

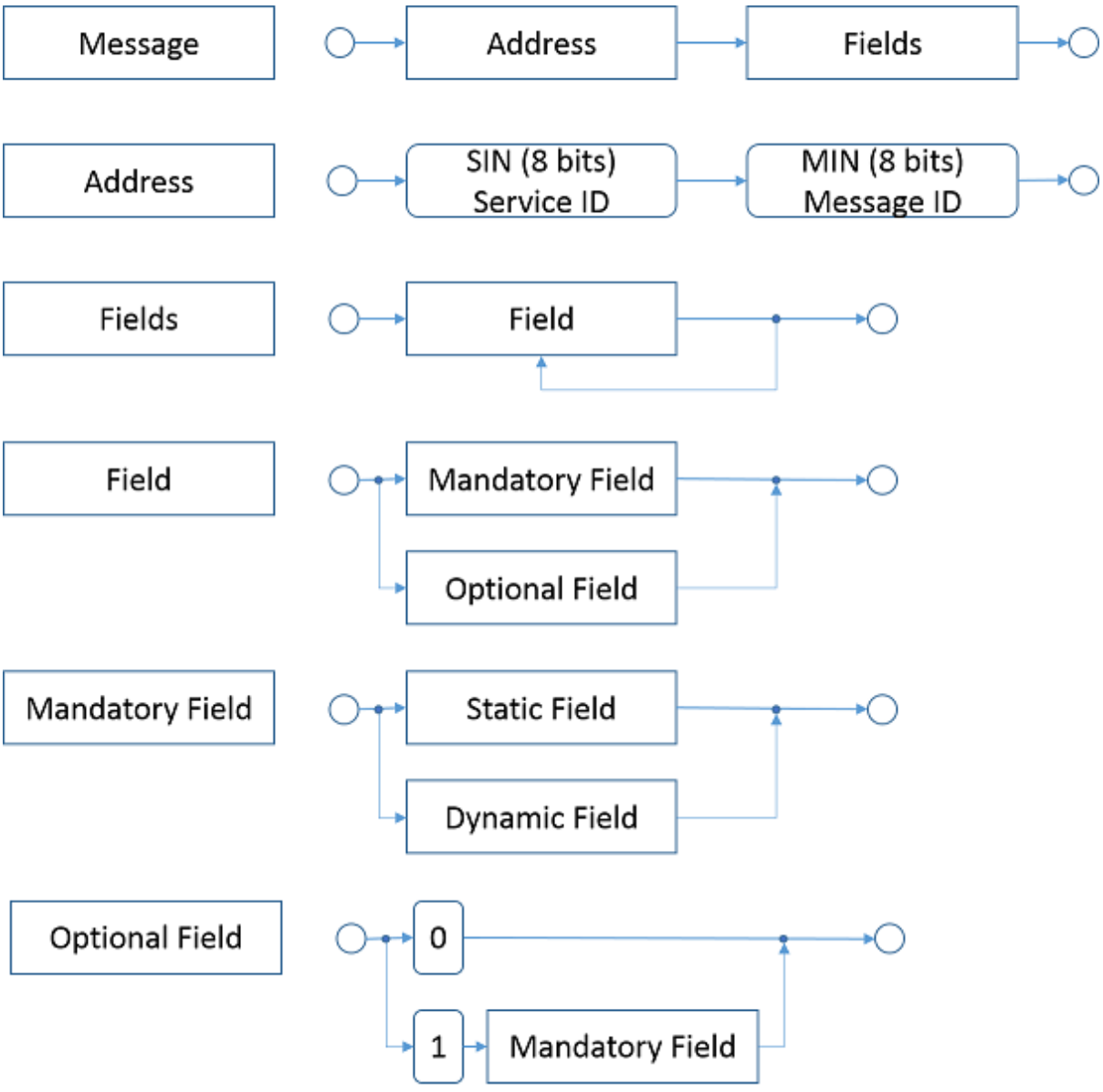
- ORBCOMM common message format

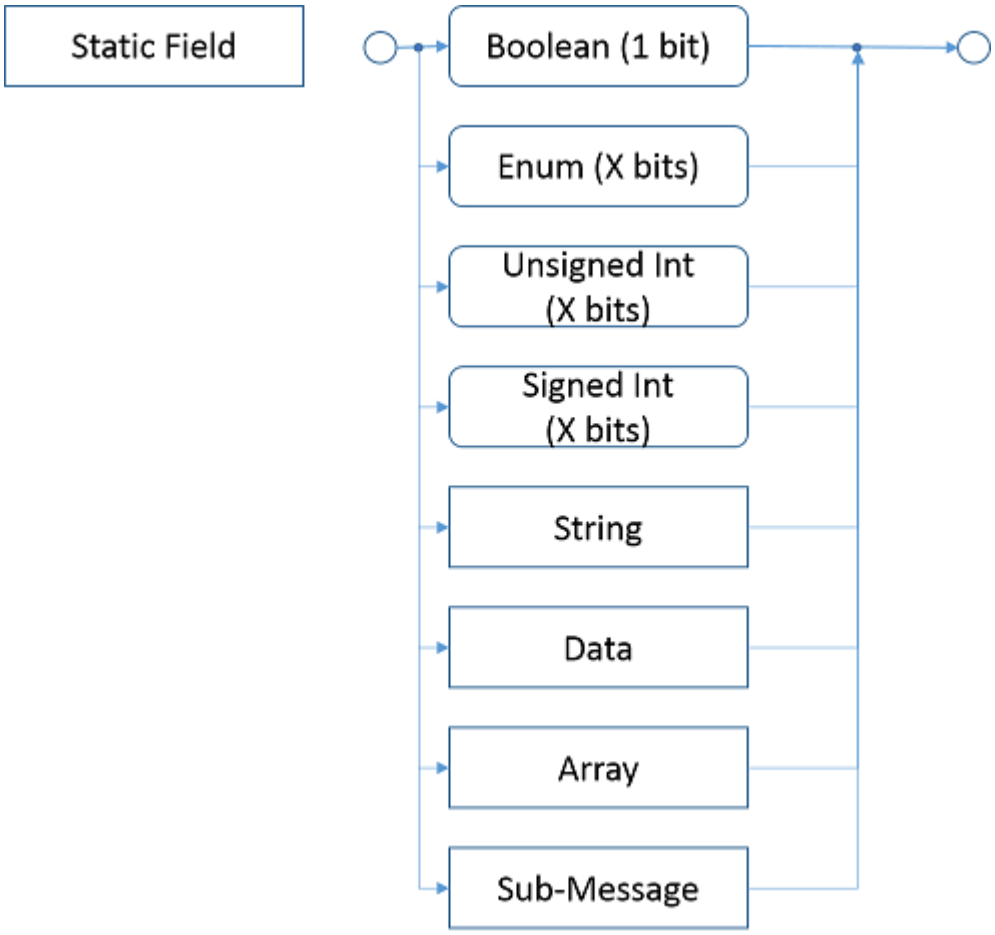
3: Codec Algorithms

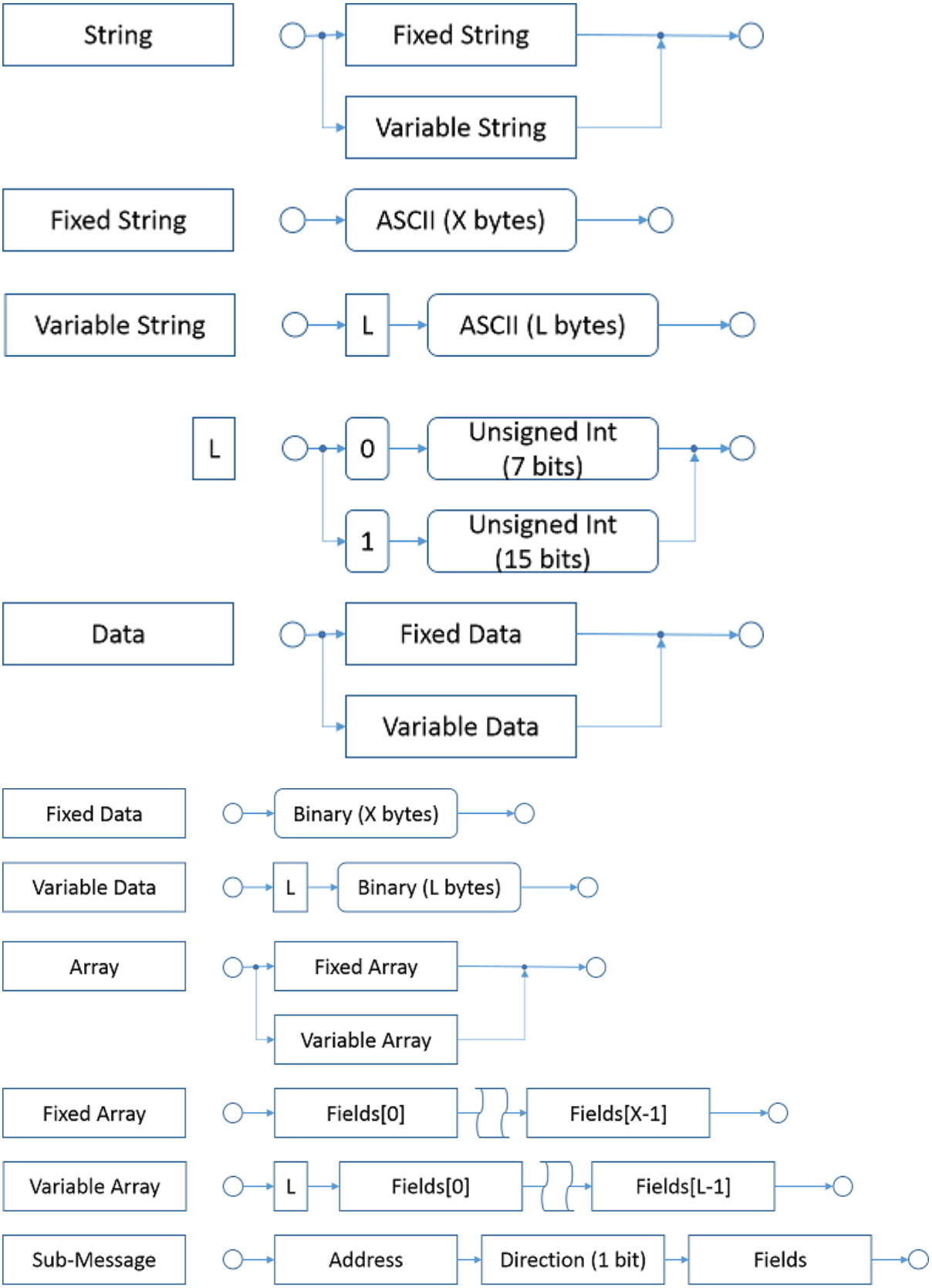
3.1: Common Message Formats

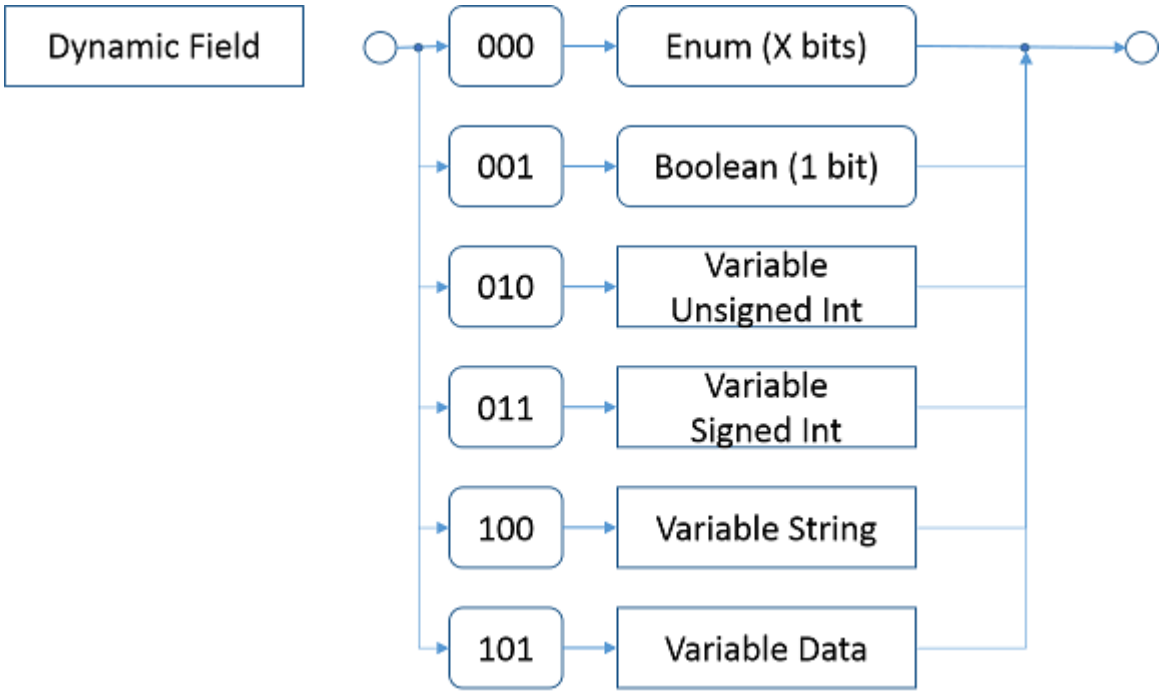
3.1.1: Concept Overview

ORBCOMM IDP terminals using the Lua Service Framework use a message encoding best described by a Backus-Naur syntax diagram. Each line defines a logical element, represented by a regular rectangle. Actual bits carried in the message payload are represented by rounded rectangles. The variable "X" denotes a size which is dictated by the message meta-data.

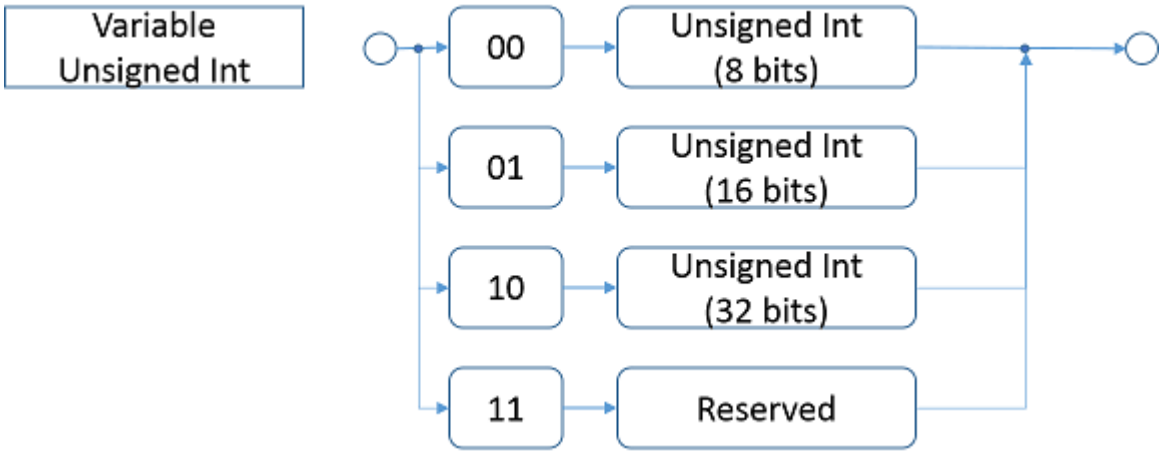


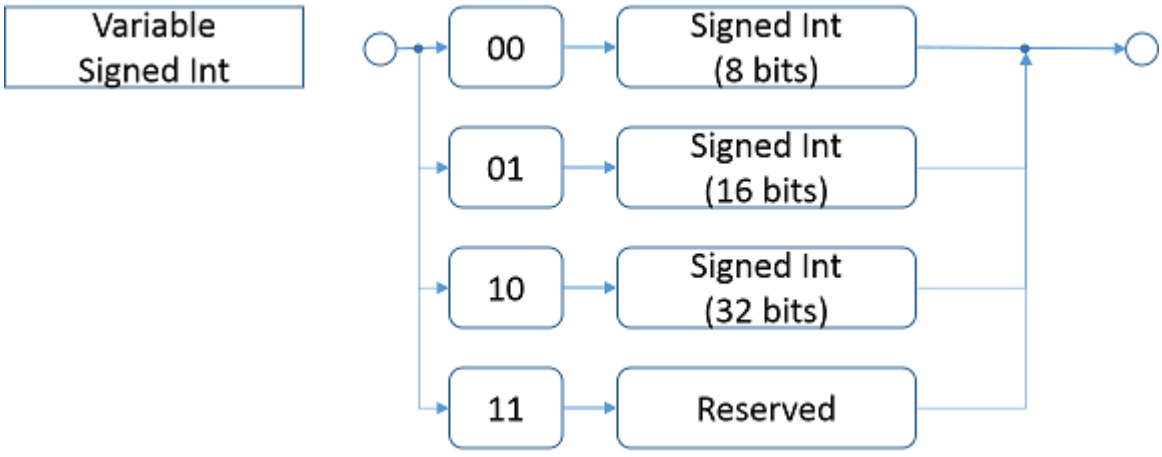



String
Fixed String
Variable String
Fixed String
ASCII (X bytes)
Variable String
L
ASCII (L bytes)
L
0
Unsigned Int (7 bits)
1
Unsigned Int (15 bits)
Data
Fixed Data
Variable Data
Fixed Data
Binary (X bytes)
Variable Data
L
Binary (L bytes)
Array
Fixed Array
Variable Array
Fixed Array
Fields[0]
Fields[X-1]
Variable Array
L
Fields[0]
Fields[L-1]
Sub-Message
Address
Direction (1 bit)
Fields



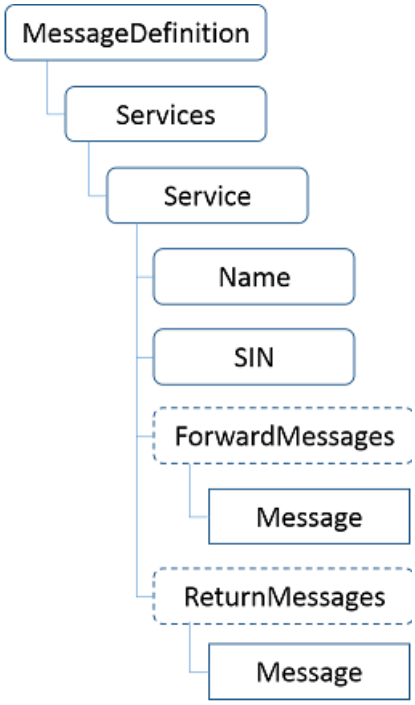
Note: A Property Field has the same structure as a dynamic field.

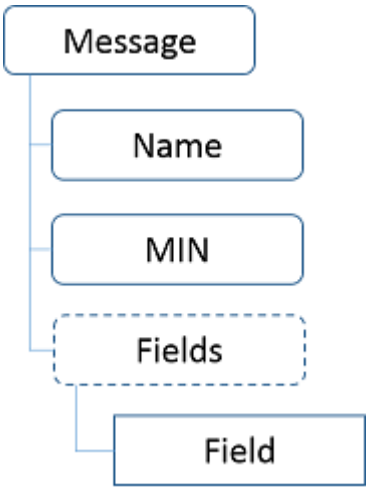


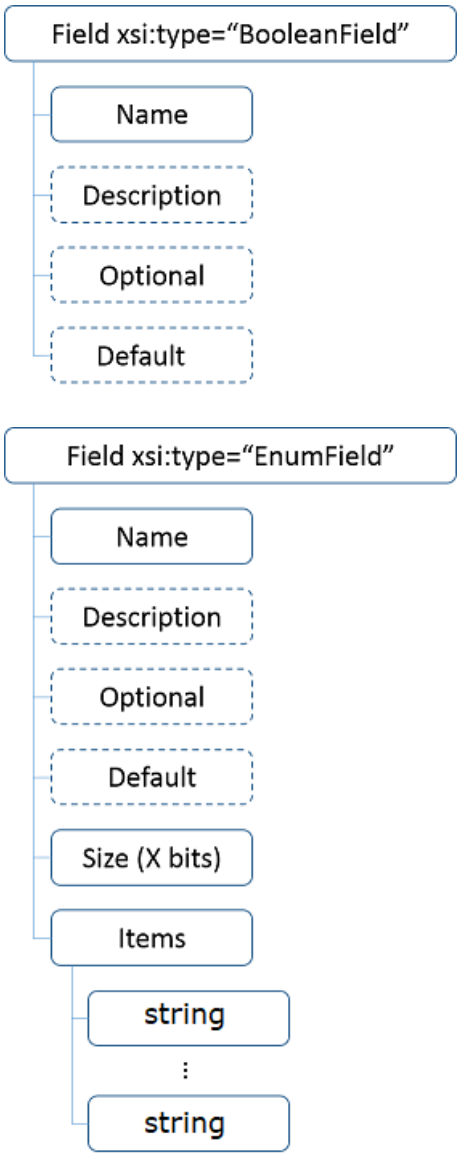


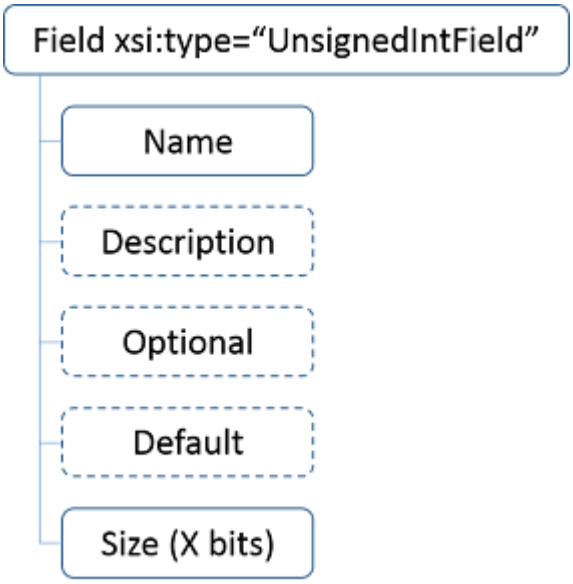
3.1.2: XML File Structure

The ORBCOMM IDP Developer Toolkit provides tools to generate a message definition file with file extension “idpmsg”. The IDP Toolkit generated file can be uploaded to a mailbox by your service provider or using the *IDP Administrative API* with appropriate privileges, upon request. The hierarchy of the XML file is detailed below. Solid boxes represent mandatory elements, while dashed lines indicate optional elements.

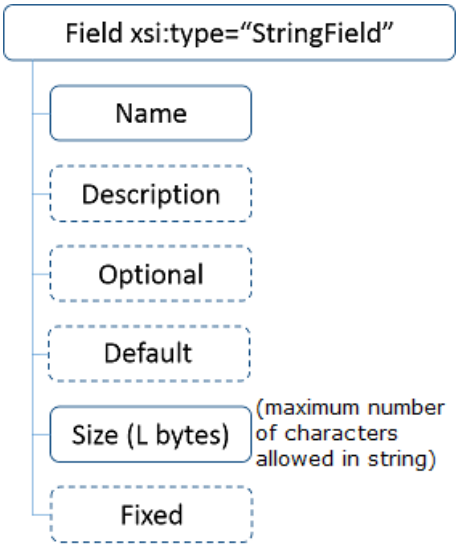


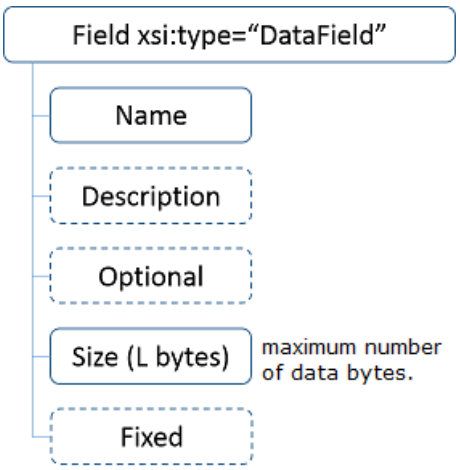


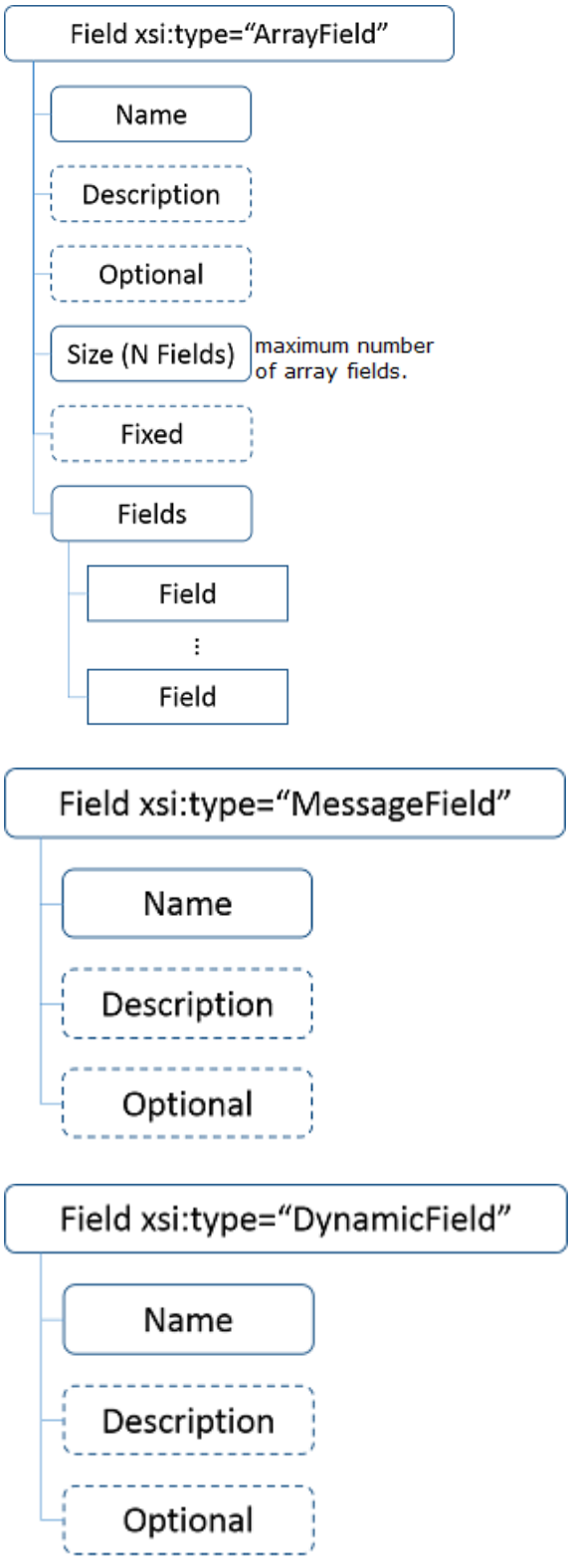


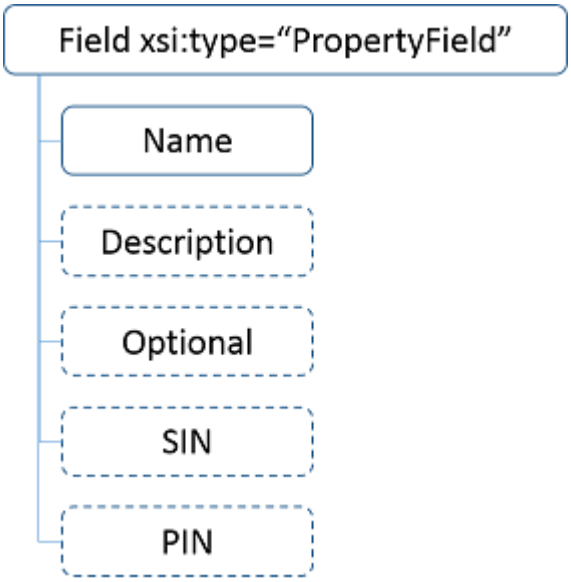


Note: SignedIntField has an identical definition.








Field xsi:type="PropertyField"
Name
Description
Optional
SIN
PIN

4: Message Definition Example for ORBCOMM Terminals

The examples used in this section are created for ORBCOMM terminals using the ORBCOMM development kit.

4.1: The Message definition

In below example of an MDF the forward and return messages are defined for the **EchoAgent**. If run on the terminal, the EchoAgent will respond to messages with SIN = 230. If It receives a message identified with MIN = 1, it will copy the text string, truncate if necessary and put it in a the return message.

The forward message `RequestEcho` (MIN 1) can carry a string of up to 9996 characters to be send to the terminal. The return message `RespondEcho` (MIN 1) can carry a string of up to 6396 characters.

```
<?xml version="1.0" encoding="utf-8"?>
<MessageDefinition xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Services>
    <Service>
      <Name>EchoAgent</Name>
      <SIN>230</SIN>
      <ForwardMessages>
        <Message>
          <Name>RequestEcho</Name>
          <MIN>1</MIN>
          <Fields>
            <Field xsi:type="StringField">
              <Name>longText</Name>
              <Size>9996</Size>
            </Field>
          </Fields>
        </Message>
      </ForwardMessages>
      <ReturnMessages>
        <Message>
          <Name>RespondEcho</Name>
          <MIN>1</MIN>
          <Fields>
            <Field xsi:type="StringField">
              <Name>shortText</Name>
              <Size>6396</Size>
            </Field>
          </Fields>
        </Message>
```

```
      </ReturnMessages>
    </Service>
  </Services>
</MessageDefinition>
```

4.2: Uploading the MDF to the Mailbox

The MDF can be uploaded to an IDP Mailbox using the IDP administration Portal or via the Administration API. The details can be found in the *IDP Administration Portal Manual* and the *IDP Administration API* description.

If multiple MDFs need to be linked to a mailbox, the MDFs must be merged into a single file before uploading. For instance if the Mailbox needs the MDF for the ORBCOMM core services (SIN 16 - SIN 127) and the application specific MDF, like the EchoAgent example above, both need to be combined.

Steps to take for merging MDFs `A.idpmsg` and `B.idpmsg`

1. Open both files in a text editor
2. Select and copy in `A.idpmsg` all the lines between the XML tags `<Services>` and `</Services>`. In *Figure 2* below, the relevant lines are coloured as an example.
3. Locate in `B.idpmsg` the xml tag `<Services>`
4. Copy the selected lines from `A.idpmsg` directly after the `<Services>` tag in `B.idpmsg`
5. Save the merged file and upload to the Mailbox.

Figure 2. Merging two data files: example

4.3: Submitting a forward message using the common Message Format

After uploading the Message Definition to the Mailbox, the Application can use the below JSON structured payload in the `Submit_Messages` API to submit a message to the terminals. The Message gateway will use the MDF to encode the payload into a stream of bytes.

```
POST /v1/idp/gateway/rest/submit_messages.json/ HTTP/1.1
Host: api.inmarsat.com
Content-Type: application/json
Cache-Control: no-cache

{
"accessID": "MB0002",
"password": "secret",
"messages": [
    {
        "DestinationID": "01234567SKY88AC",
        "UserMessageID": 111,
        "Payload":
```

```
      {
        "Name": "RequestEcho",
        "SIN": 230,
        "MIN": 1,
        "Fields": [
                  {
                     "Name": "longtext",
                     "Value": "Hello World"
                  }

      ]
      }
    }
]
}
```

Retrieving the return message using the common Message Format

In response to submitting the messages in the previous section, the terminals will copy the text sent (at most 6396 characters), and return it in the `ResponseEcho` message.

Upon retrieving the response message with the `Get_Return_Messages` the Message Gateway will use the MDF to decode the stream of bytes in to a JSON structure payload.

```
{
  "ErrorID": 0,
  "More": false,
  "NextStartUTC": "2016-06-28 11:34:51",
  "Messages": [
    {
      "ID": 23180,
      "MessageUTC": "2016-06-28 11:34:01",
      "ReceiveUTC": "2016-06-28 11:34:01",
      "SIN": 230,
      "MobileID": "01234567SKY88AC",
      "Payload": {
        "Name": "RespondEcho",
        "SIN": 230,
        "MIN": 1,
        "Fields": [
          {
            "Name": "shortText",
            "Value": ""
          }
        ]
      },
      "RegionName": "EMEARB7",
      "OTAMessageSize": 3
```

```
    }
  ],
  "NextStartID": -1
}
```

Appendix A: Acronyms/Glossary

Acronym	Definition
AMER	Americas
AORWSC	Atlantic Ocean Region West South Central
APAC	Asia Pacific
BB	bulletin board
Broadcast ID	unique identifier of a virtual terminal used for broadcast messaging
EMEA	Europe, Middle-East, Africa
Forward Channel	The satellite channel used to transport to-mobile traffic
Gateway Account	equivalent to a Mailbox
GPS	Global Positioning System
GNSS	Global Navigation Satellite System (GPS is one such system)
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over SSL
LES	Land earth station
Mailbox	logical entity on the Message Gateway System, which a customer application uses to send and receive messages to remote terminals
MIN	message identification number
Mobile ID	a unique identifier of an IsatData Pro terminal
Modem	The remote device that implements the IDP protocols (see also Terminal)
Partner	Inmarsat Partner, VAR or Distribution Partner
REST	Representational State Transfer (web service)
Return Channel	The satellite channel used to transport from-mobile traffic
SCADA	Supervisory Control and Data Acquisition
SIN	service identification number
SOAP	Simple Object Access Protocol (web service)
SP	Service Provider
SSL	Secure Sockets Layer
Segment	Depending on the size, Messages may be divided into several message segments before transfer via the satellite.
TCP/IP	A common Internet transport protocol
Terminal	A remote device utilising the IDP service. It consist of a Modem implementing the network interfaces and optionally additional functionality such as a scripting environment to process data received form the IDP network or before submitting it to the IDP network.
TLS	Transport Layer Security
UTC	Coordinated Universal Time