

Commercial in Confidence

FATM Message Translator

Message Translator ICD and API

Contract Number: **TBD**

Prepared for:

Prepared By:

Flight ATM Systems Ltd.



Document Number
FATM-TRANS02

Rev
A0.01

Page
1/19

Filename: FATM AFTN Translator ICD and API.doc

Paper size: A4

All information contained in this document remains the sole and exclusive property of Flight ATM Systems Ltd. No part of it may be copied, or disclosed by the recipient to third persons, without the prior written consent of Flight ATM Systems Ltd.; nor shall it be used for any purpose other than in connection with an agreement or proposed agreement with Flight ATM Systems Ltd. Registered in the United Kingdom. registration number 562 5816:

Referenced Documents

No.	Identification	Rev Date	Edition	Name	Source
[1]	FATM-TRANS01	21 Sep 2015	V2.0	FATM Message Translator AFTN ATS/OLDI Message Translator	FATM

Table 1 - Referenced Documents

Table of Contents

1	Document Overview	4
1.1	Identification	4
1.2	Purpose.....	4
1.3	DOCUMENT OVERVIEW.....	4
2	Interface Design	5
2.1	Interface Identification and Diagrams.....	5
2.2	INTERFACE PROTOCOL.....	7
2.2.1	Message Input	7
2.2.2	Message Output.....	9
2.2.3	Heartbeat Message Input.....	11
2.3	AOMT Service MANAGEMENT	12
2.3.1	Preconditions.....	12
2.3.2	ZIP Archive Contents.....	12
2.3.3	AOMT Service Management.....	14
2.3.4	ADEXP Configuration Files	15
2.3.5	OLDI Configuration Files.....	15
3	REQUIREMENTS TRACEABILITY.....	16
4	Appendix A – Acronyms	17

List of Figures

Figure 1 -	AOMT Interface Overview	6
Figure 2 -	TCP Start up Invocation.....	14
Figure 3 -	UDP Start up Invocation	15
Figure 4 -	TCP Log File Output Example.....	15
Figure 5 -	UDP Log File Example	15

List of Figures

Table 1 -	Referenced Documents.....	2
Table 2 -	Message Input Protocol	7
Table 3 -	Message Output Protocol.....	10
Table 4 -	Heartbeat Message Protocol	11
Table 5 -	Software Delivery Content	13
Table 6 -	Acronyms.....	18

1 Document Overview

1.1 Identification

Program: FATM Message Translator

Document Name: Message Translator ICD and API

Document Number: FATM-TRANS02

Revision: A0.01

Revision Date: Monday, 21 September 2015

Document Owner: Peter Venton – Flight ATM Systems Ltd.

File Name: FATM AFTN Translator ICD and API.doc

1.2 Purpose

This document provides a description of the Flight ATM Systems AFTN ATS/OLDI Message Translator, (AOMT) interface and API that is able to translate:

- ATS ICAO format messages into ADEXP format;
- ATS ADEXP format messages into ICAO format;
- OLDI ICAO format messages into ADEXP format;
- OLDI ADEXP format messages into ICAO format;
- Pre-FPL 2012 version messages into Post-FPL 2012 format (irrespective if message format is OLDI or ADEXP);
- Post-FPL 2012 version messages into Pre-FPL 2012 format (irrespective if message format is OLDI or ADEXP);

The AOMT is provided as a network service using either UDP or TCP/IP to provide a service containing the AOMT software.

The AOMT receives messages with a target version and format instructions and returns the translated message. The format instruction, version instruction and the message text are sent as a text string using the IA5 character set.

1.3 DOCUMENT OVERVIEW

This document specifies the interface design of the AOMT in order that the AOMT is able to perform the translation processing as required by a calling system. The message protocol and formats are described to sufficient detail that the interface can be implemented by a system wishing to use the AOMT service.

2 Interface Design

The interface design uses an application protocol based on either UDP or TCP protocols that can send and receive messages to/from the AOMT.

Three UDP ports for UDP communication will be provided for command, response and AOMT status.

- The command port is used for communication with AOMT.
- The response port is used for sending the translation response.
- The status port is used for a cyclic status message.

The same message structure is used for all messages. The AOMT will output message counters in the status message.

All three ports are defined as start up parameters.

The following restrictions apply to this design:

- The AOMT only listens for UDP messages from localhost::commandPort.
- The AOMT sends the result back to localhost:responsePort
- The AOMT send status message to localhost:statusPort

2.1 Interface Identification and Diagrams

The input data consists of the following items:

- A message to be translated;
- A target message format instruction;
- A target message version instruction;

The message to be translated is a text string of IA5 characters representing any of the following messages:

- ATS Message in the ICAO format;
- OLDI Message in the ICAO format;
- ATS Message in the ADEXP format;
- OLDI Message in the ADEXP format;
- ETFMS Message in the ADEXP format;
- CADF Message in the ADEXP format;

The data input and output are shown in figure 1. For a complete description of the AOMT functionality refer to the documents **[Error! Reference source not found.]**.

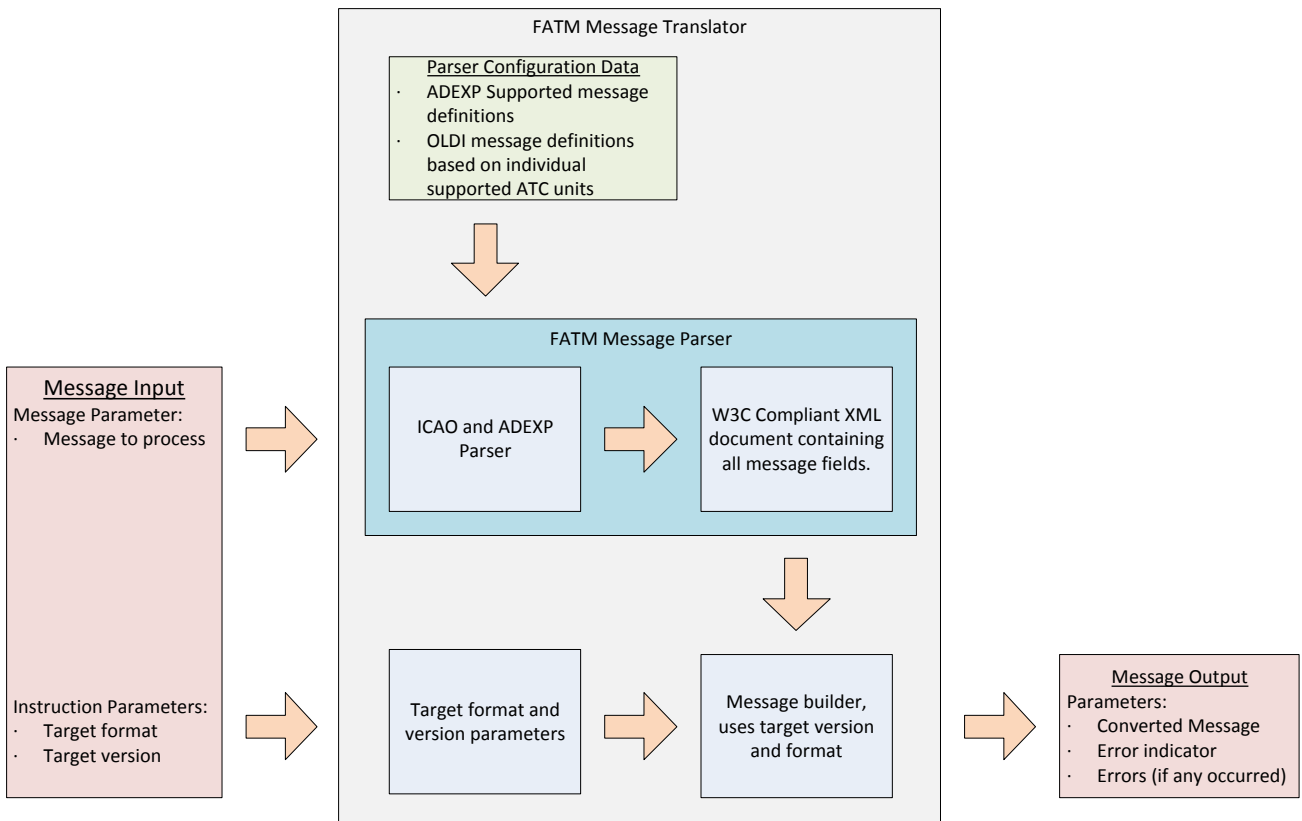


Figure 1 - AOMT Interface Overview

2.2 INTERFACE PROTOCOL

The interface protocol is defined in this section. The input and output data structures are described in the sections that follow. Messages are received and sent asynchronously. A message is output by the AOMT in response to a message input.

2.2.1 Message Input

When data is input to the AOMT, the message protocol must conform to the structure described in Table 2 - below.

Data Item	Length (bytes)	Value	Comment
Start Message Indicator	1	ASCII 001 ₁₀	The ASCII SOH character
Channel Name	Variable	Any printable ASCII character apart from the hyphen (ASCII 045 ₁₀)	A channel identifier indicating the ATMGW channel that is requesting translation of this message.
Separator	1	ASCII 045 ₁₀	The ASCII hyphen character '-' used to separate the channel name from the target format and version instruction.
Target format instruction	4 or 5	ICAO or ADEXP	Instruction used to specify the output format of a message being input.
Separator	1	ASCII 045 ₁₀	The ASCII hyphen character '-' used to separate the target format and version instruction.
Target version instruction	8 or 9	PRE_2012 or POST_2012	Instruction used to specify the target output FPL 2012 version of a message being processed.
Separator	1	ASCII 045 ₁₀	The ASCII hyphen character '-' used to separate the target version instruction and the node name.
Node Name	Variable	Any printable ASCII character apart from the hyphen (ASCII 045 ₁₀)	A channel node name identifying the ATMGW node that a translated message will be sent to.
Instruction/message separator	1	ASCII 002 ₁₀	The ASCII STX character.
Message to Translate	Variable	Any IA5 string of characters	The OLDI, ATS, ETFMS or CADF message to translate, this must include the AFTN header if the message is NOT an OLDI message. OLDI messages have no header and therefore the header can be omitted.
End of message indicator	1	ASCII 003 ₁₀	The ASCII ETX character

Table 2 - Message Input Protocol

2.2.1.1 Message Input Example

```
<SOH>CHANNEL_NAME-ICAO-PRE_2012-NODE_NAME<STX>FF LTAAZFZX LTBBZFZX  
LTAYZGZX  
261133 LTFCZPZX  
(FPL-AUA201-IS  
-F70/M-SRWY/S  
-LOWW0500A  
-N0410F350 SITNI1B SITNI UL856 NEGRA  
-LSZH0104  
-DOF/010723 RVR/200 ORGN/RPL)  
<ETX>
```


2.2.2 Message Output

When data is output from the AOMT, the message protocol conforms to the structure described in Table 3 - below. Character encoding for the message output is UTF8, with the exception that the text related to the message body itself is comprised only of characters in the IA5 character set.

Data Item	Length (bytes)	Value	Comment	Compulsory
Start Message Indicator	1	ASCII 001 ₁₀	The ASCII SOH character	Yes
Channel Name	Variable	Any printable ASCII character apart from the hyphen (ASCII 045 ₁₀)	A channel identifier indicating the ATMGW channel that is requesting translation of this message.	Source Channel number
Separator	1	ASCII 045 ₁₀	The ASCII hyphen character '-' used to separate the channel name and target format and version instruction.	Yes
Target format instruction	4 or 5	ICAO or ADEXP	Instruction used to specify the output format of a message being input. This is the same as input to the translator.	Yes
Separator	1	ASCII 045 ₁₀	The ASCII hyphen character '-' used to separate the target format and version instruction.	Yes
Target version instruction	8 or 9	PRE_2012 or POST_2012	Instruction used to specify the output FPL 2012 version of a message being input. This is the same as input to the translator.	Yes
Separator	1	ASCII 045 ₁₀	The ASCII hyphen character '-' used to separate the target version instruction and the node name.	
Node Name	Variable	Any printable ASCII character apart from the hyphen (ASCII 045 ₁₀)	A channel node name identifying the ATMGW node that a translated message will be sent to.	
Instruction/message separator	1	ASCII 002 ₁₀	The ASCII STX character	Yes
Translated message	Variable	Any IA5 string of characters	The translated OLDI, ATS, ETFMS or CADF message.	Yes
Error Message Separator	1	ASCII 010 ₁₀	The ASCII NL character	No
Error field indicator	6	ERRORS	The first line output if there are one or more errors to include in the output.	No
Newline indicator	1	ASCII 010 ₁₀	The ASCII NL character	No

Data Item	Length (bytes)	Value	Comment	Compulsory
Start error index	1 to 4	ASCII 048 ₁₀ To ASCII 057 ₁₀	An index to the start of the erroneous data to which the error pertains with respect to the start of the field containing the error.	No
Index separator	1	ASCII 044 ₁₀	The ASCII comma character	No
End error index	1 to 4	ASCII 048 ₁₀ To ASCII 057 ₁₀	An index to the end of the erroneous data to which the error pertains with respect to the start of the field containing the error.	No
Index separator	1	ASCII 044 ₁₀	The ASCII comma character	No
Field index	1 to 4	ASCII 048 ₁₀ To ASCII 057 ₁₀	The start index to the field containing the erroneous data with respect to the start of the message.	No
Index separator	1	ASCII 044 ₁₀	The ASCII comma character	No
Field identifier	Variable	Any IA5 string of characters	The ICAO field identifier or the ADEXP field name to which this error applies.	No
Index separator	1	ASCII 044 ₁₀	The ASCII comma character	No
Error Message	Variable	Any IA5 string of characters	An error message reported by either the translator or the message parser. These optional fields (including the previous item 'Newline indicator') can occur as many times as there are errors reported by the AOMT.	No
End of message indicator	1	ASCII 003 ₁₀	The ASCII ETX character	Yes

Table 3 - Message Output Protocol

2.2.2.1 Message Output Example

```

<SOH>CHANNEL_NAME-ICAO-PRE_2012-NODE_NAME<STX> FF LTAAZFZX LTBBZFZX
LTAYZGZX
261133 LTF CZPX
(FPL-AUA201-IS
-F70/M-SRWY/S
-LOWW0500A
-N0410F350 SITNI1B SITNI UL856 NEGRA
-LSZH0104
-DOF/010723 RVR/200 ORGN/RPL)
<NL>ERRORS<NL>
0,9,163,F13,FIELD 'LOWW0500A' TOO LONG
<ETX>

```

2.2.3 Heartbeat Message Input

The following table describes the heartbeat messages.

Data Item	Length (bytes)	Value	Comment
Start Message Indicator	1	ASCII 001 ₁₀	The ASCII SOH character
Heartbeat message identifier	6	STATUS	Label to identify this message as a heartbeat message.
Instruction/message separator	1	ASCII 002 ₁₀	The ASCII STX character
Number of messages received by the AOMT	1 to 'n'	ICAO or ADEXP	The number of messages received by the AOMT; the integer value can be any length depending on the number of received messages.
End of message indicator	1	ASCII 003 ₁₀	The ASCII ETX character

Table 4 - Heartbeat Message Protocol

2.2.3.1 Message Input Example

```
<SOH>STATUS<STX>12345<ETX>
```

2.3 AOMT Service MANAGEMENT

The AOMT is distributed as a ZIP archive. It contains the structure required to run the software. The AOMT application can be run on either windows or linux.

2.3.1 Preconditions

The application requires a Java 8 run time environment, either on the path or or defined by the JAVA_HOME environment variable.

The application requires write permission to the log directory.

2.3.2 ZIP Archive Contents

The AOMT service is delivered as zip file named:

aomt-[version].zip

The [version] contains the software delivery version number.

When unzipped, a number of directories are created containing the following files that together constitute the software delivery as shown in the following table.

Directory	Content Description	Content
aomt\bin	Start up scripts	aomt.sh aomt.bat
aomt\lib	Java™ library files (*.jar) – third party libraries	commons-beanutils-1.7.0.jar commons-beanutils-core-1.8.0.jar commons-cli-1.2.jar commons-collections-3.2.1.jar commons-configuration-1.6.jar commons-digester-1.8.jar commons-lang-2.4.jar commons-logging-1.1.1.jar dom4j-1.6.1.jar icu4j-2.6.1.jar jaxen-1.1.1.jar jdom-1.0.jar xalan-2.6.0.jar xercesImpl-2.6.2.jar xml-apis-2.0.2.jar xmlParserAPIs-2.6.2.jar xom-1.0.jar
	Java™ library files (*.jar) – Flight ATM System Ltd. libraries	fatm-adexp-configuration.jar fatm-adexp.jar fatm-aomt.jar fatm-base.jar fatm-documents.jar fatm-parser-common.jar fatm-parser.jar

Directory	Content Description	Content
aomt\config\adexp	ADEXP configuration files	ADEXP_auxilliary_terms.xml ADEXP_primary_fields.xml ADEXP_subfields.xml ADEXP_Supported_Messages.xml GlobalErrorText.xml
aomt\config\oldi	OLDI configuration files	AdjUnitDefault.properties
aomt\config\aomt	AOMT configuration files	aomt.properties
aomt\docs	Documentation	IDD SSS SRS
aomt\etc	<pre># Logging handlers = java.util.logging.FileHandler, java.util.logging.ConsoleHandler .level = INFO # File Logging java.util.logging.FileHandler.formatter = java.util.logging.SimpleFormatter java.util.logging.FileHandler.level = INFO # Console Logging java.util.logging.ConsoleHandler.level = WARNING</pre>	logging.properties
aomt\log	AOMT log files containing system status information.	aomt.log (created by the AOMT when running).

Table 5 - Software Delivery Content

2.3.3 AOMT Service Management

There are a number of parameters to set that specify the IP and port number of the service. The AOMT service runs as a server and expects connections from the system requiring the translation service.

2.3.3.1 Prerequisites

The application requires a java 1.8 runtime to be on the path or defined by the environment variable JAVA_HOME. An Oracle JRE is recommended.

2.3.3.2 Starting the AOMT Service

The AOMT can be started in either a TCP or UDP mode by invoking the start up scripts with appropriate command line parameters to invoke the AOMT with the required network protocol.

2.3.3.2.1 Start in TCP Mode

The AOMT TCP start up command synopsis is as follows:

```
aomt TCP [port number]
```

To start the application in TCP mode invoke the start up script with the following command:

```
...\aomt\bin\aomt TCP 9990
```

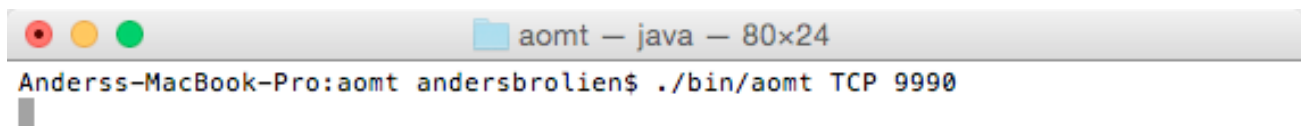


Figure 2 - TCP Start up Invocation

The AOMT waits for connections on the specified port.

2.3.3.2.2 Start in UDP Mode

The AOMT UDP start up command synopsis is as follows:

```
aomt UDP [command port number][response port number][status port number][Reply IP Address][Heartbeat Cyclic Rate]
```

Start the AOMT by invoking the start UP script with the following parameters:

- Parameter 1 – Specify UDP as the mode;
- Parameter 2 – Command port number – A port number on which the input message and translation instruction are sent, the localhost::commandPort;
- Parameter 3 – Response port number – A port number on which the output message and translation instruction are received, the localhost::responsePort;
- Parameter 4 – Status port number – A port number on which the heartbeat status messages are sent, the localhost::statusPort;
- Parameter 5 – (optional) – The IP address to send the reply to;
- Parameter 6 (optional) – Specifies the cyclic period for sending the also an optional fifth argument allowing configuration of the status message transmission rate. The fifth argument is the number of seconds between status message trasmission. If this is not provided AOMT will use a default value of 10 seconds between status message transmission.

The following is an example for the UDP start up command.

```
...\aomt\bin\aomt UDP 9990 9991 9992 20
```



Figure 3 - UDP Start up Invocation

Once the AOMT has started in UDP mode is is ready to receive message input with an associated command on the UDP localhost::commandPort. Translated messages are returned on the UDP localhost::responsePort along with any heartbeat messages.

2.3.3.3 Log File Output

The AOMT software writes status messages to a log file located in the path shown below. The log file output includes output from the translator and parser software.

...\aomt\log\aomt.log

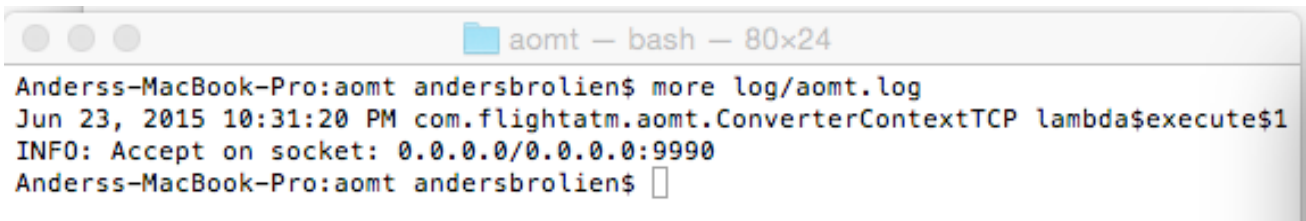


Figure 4 - TCP Log File Output Example

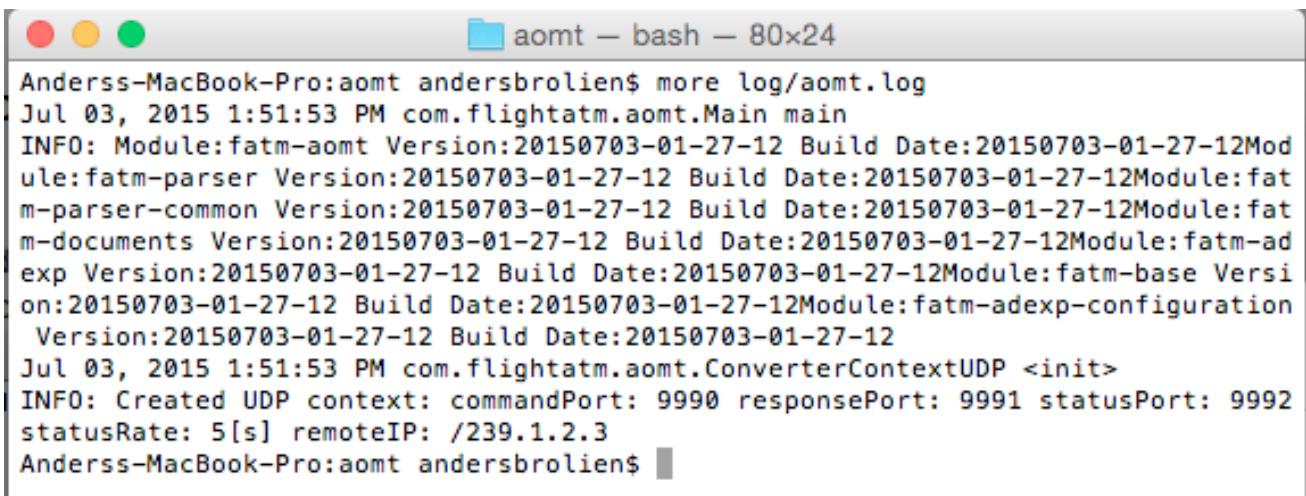


Figure 5 - UDP Log File Example

2.3.4 ADEXP Configuration Files

The ADEXP parser XML configuration files are located in the following directory:

...\aomt\config\adexp*.xml

2.3.5 OLDI Configuration Files

The OLDI parser property configuration file is located in the following directory:

...\aomt\config\oldi*.properties

3 REQUIREMENTS TRACEABILITY

None as of producing this document;

4 Appendix A – Acronyms

	Acronym	Description
A		
	ACCS	Air Defence Command and Control System (NATO)
	ADEXP	ATS Data Exchange Presentation
	AOMT	ATS and OLDI Message Translator
	ASCII	American Standard Code for Information Interchange
	ATM	Air Traffic Management
	ATMGW	ATM Gateway
	ATS	Air Traffic Service
B		
C		
	CADF	Centralised Airspace Data Function
	CDRL	Contract Deliverables Requirements List
	CODE	ICAO F18 subfield, specifies the aircraft code
	CSCI	Computer Software Configuration Item
D		
	DID	Data Item Description
	DOF	ICAO F18 sub-field, Date of Flight
E		
	ETFMS	Enhanced Tactical Flow Management System
	ETX	End of Text, ASCII escape character
F		
	FPL	Flight Plan Message (ICAO)
G		
H		
I		
	IA5	International Alphabet 5
	ICAO	International Civil Aviation Organisation
	IDD	Interface Design Document
	IP	Internet Protocol
	IT	Information Technology
J		
	JRE	Java Runtime Environment
L		
M		
	MFG	Manufacturing
	MIL	Military
N		

	Acronym	Description
	NATO	North Atlantic Treaty Organization (Brussels, Belgium)
	NL	Newline, ASCII Character
O		
	OLDI	On-Line Data Interchange
	ORGN	ICAO F18 sub-field, contains AFTN originator address
P		
R		
	REV	Revision Message (OLDI)
	RPL	Repetitive Flight Plan
	RVR	Runway Visual Range
S		
	SOH	Start of Header, ASCII escape character
	SRS	System Requirements Specification
	SSS	System Segment Specification
	STD	Standard
	STX	Start of Text, ASCII escape character
T		
	TBP	To Be Provided
	TCP	Transmission Control Protocol
U		
	UTF8	Unicode Transformation Format (8-Bit)
W		
X		
	XML	Extensible Mark up Language
Y		

Table 6 - Acronyms

LAST PAGE OF DOCUMENT