Report no. ASDI-FD-001

Aircraft Situation Display To Industry:

Functional Description and Interface Control Document

Version 4.0

4 August 2000

Volpe Center
Automation Applications Division, DTS-56
55 Broadway St.
Cambridge, MA 02142

Foreword

The Aircraft Situation Display to Industry (ASDI) Subsystem of the Enhanced Traffic Management System (ETMS) allows real-time air traffic data to be disseminated to members of the aviation industry. This document provides a functional description of the ASDI subsystem and gives the interface information that a vendor would need to interface with this subsystem or that a user would need to interpret the ASDI data.

Version 4.0 differs from Version 3.0, dated 5 March 1999, in the following ways.

- Previous versions of this document included information to help vendors and users make the transition from the previous version of this feed, which was known as the ATA feed, to the current version. The old feed was discontinued in 1999, so this information is omitted since it is no longer useful.
- The special treatment that must now be given to London data is documented.
- Additional software tips for vendors are included.
- The documentation on the RT message is rewritten.
- The description of altitudes in a TZ message has been improved.
- The title has been slightly changed.
- A report number has been added.
- Editorial changes have been made to improve clarity.

This document and the work it describes are sponsored by the Federal Aviation Administration under the guidance of Steve Alvania of AUA-740. This document was written by Mike Francis and Maureen Hogan of RLM Software and by Rick Oiesen of the Volpe Center.

Contents

FOREWORD ..... **II**

1. INTRODUCTION **1**

1.1 BACKGROUND AND TERMINOLOGY 1
1.2 CONTACTS 1
1.3 RELATED DOCUMENTS 1

2. FUNCTIONAL OVERVIEW **3**

3. CONTENT OF THE ASDI DATA STREAM SENT TO THE VENDORS **4**

3.1 FILTERING MESSAGES OUT OF THE ASDI DATA STREAM 4
3.2 REMOVING FIELDS FROM THE ALLOWED ASDI MESSAGES 5
3.3 FRAMING THE MESSAGES 5
3.4 FACILITY IDENTIFIERS 6

4. SOFTWARE INTERFACE, HARDWARE INTERFACE, AND SECURITY **8**

4.1 SOFTWARE INTERFACE 8
4.2 HARDWARE INTERFACE 9
4.3 SECURITY 11
4.4 NBAA VENDOR CODE OF CONDUCT 12
4.5 HANDLING THE LONDON DATA 13

5. NAS MESSAGE FORMATS **15**

5.1 INTRODUCTION 15
5.2 NAS MESSAGE SYNTAX SYMBOLS EXPLAINED 15
5.3 NAS AF MESSAGE 16
5.4 NAS AZ MESSAGE 17
5.5 NAS DZ MESSAGE 18
5.6 NAS FZ MESSAGE 19
5.7 NAS RZ MESSAGE 21
5.8 NAS TZ MESSAGE 22
5.9 NAS UZ MESSAGE 24

6. ETMS MESSAGE FORMATS **25**

6.1 RT MESSAGE 25
6.2 TO MESSAGE 27
6.3 HB MESSAGE 28

7. RT DECODING ALGORITHM **29**

SAMPLE RT DUMP SOFTWARE *29*

APPENDIX A: SAMPLE MESSAGES **A-1**

APPENDIX B: NAS FIELD FORMATS **B-1**

APPENDIX C: ETMS FIELD FORMATS **C-1**

1. Introduction

1.1 Background and Terminology

In 1992, The Federal Aviation Administration (FAA) initiated a Cooperative Research Data Agreement with the Air Transport Association (ATA). This agreement provided airlines and other aviation-related organizations with access to real-time air traffic data from the National Airspace System (NAS). This enabled airlines to use this more accurate positional information in their planning and decision making.

This original data feed, which was called the ATA feed (sometimes also called the Sprint feed), was decommissioned in 1999, and it has been replaced with a re-engineered feed known as the Aircraft Situation Display to Industry (ASDI) feed, which became operational on 2 June 1998. This document defines the characteristics of the ASDI feed.

An organization that installs a communications line and a router at Volpe so that it can receive the ASDI feed directly will be termed a *vendor*. An organization that uses the data is called a *user*. It is expected that a typical vendor will re-sell the data to users, though it might be the case that a vendor is also a user.

The purpose of this document is to provide vendors and users with the information that they need to receive and interpret the ASDI feed. If a user purchases the feed from a vendor, it might be that this user will require additional information from the vendor.

1.2 Contacts

This document refers to an FAA contact and a Volpe contact. As this is written, the FAA contact is Tim Grovac, ATT-220, (703) 904-4402, grovac@faa.gov. The Volpe contact is Rick Oiesen, DTS-56, (617) 494-2309, oiesen@volpe.dot.gov. Generally, a vendor should get in touch with the FAA contact to get information about the memorandum of agreement that entitles the vendor to get the feed. After a vendor has signed that memorandum of agreement, it should get in touch with the Volpe contact to learn how to get connected, to ask technical questions about the feed, or to find out more about the documents listed in the next section.

1.3 Related Documents

National Airspace System En Route Configuration Management Document, Computer Program Functional Specifications: Remote Outputs, NAS-MD-315, November 20, 1997, National En Route System Division, AOS-300, Federal Aviation Administration Technical Center, Atlantic City International Airport, New Jersey 08405. This NAS specification describes all the messages that the Host computer can generate. The ETMS receives all messages destined for the Central Flow Automation Facility (CFAF).

National Airspace System En Route Configuration Management Document, Computer Program Functional Specifications: Route Conversion and Posting, NAS-MD-312, 16 August 1993, National En Route System Division, AOS-300, Federal Aviation Administration Technical Center, Atlantic City International Airport, New Jersey 08405. This NAS specification describes the syntax of the field 10 (route of flight) that is used on the AF, FZ, and UZ messages.

National Airspace System En Route Configuration Management Document, Computer Program Functional Specifications: Message Entry and Checking, NAS-MD-311, 20 November, 1997, National En Route System Division, AOS-300, Federal Aviation Administration Technical Center, Atlantic City International Airport, New Jersey 08405. This NAS specification describes the syntax of each field of a NAS message.

System Requirements Document For the Aircraft Situation Display to Industry Subsystem of the Enhanced Traffic Management System, Version 1.0, Volpe Center, 7 October 1997. This is the requirements specification for the ASDI subsystem of the ETMS.

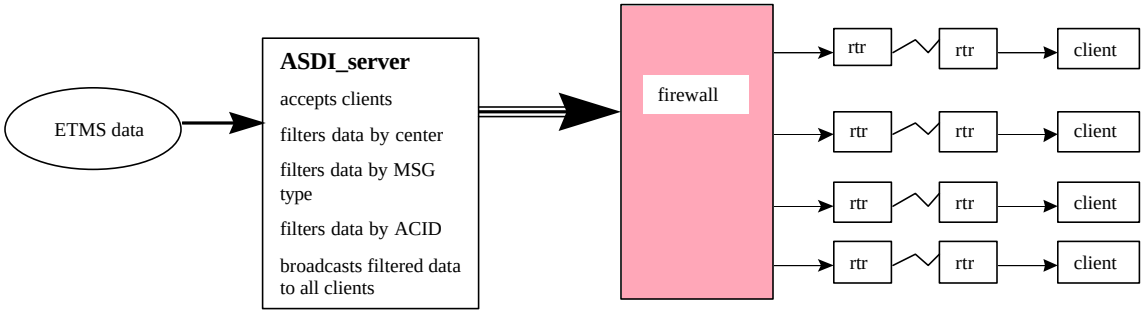
2. Functional Overview

The ASDI Subsystem is based around client-server architecture. The ASDI Server runs at the Volpe Center, which is the ETMS hubsite. This server registers with ETMS to receive the flow of raw data from ETMS; it filters out sensitive flights and message types before sending the data through a firewall to the registered clients. A client registers with the ASDI Server to receive the ASDI data; each client receives exactly the same data (except that some clients will not receive the data from London). Only vendors are allowed to register directly with the Volpe Center; other users get data not directly from Volpe but from the vendors.

The firewall controls the IP addresses from which clients are allowed to register. TCP/IP is used for all communications between the firewall and the vendors. The general flow of data is shown in the diagram below. Each jagged line represents a communications line between a vendor router on the left, which is located at the ETMS hubsite, and a vendor router on the right, which is located on the vendor's premises. Each vendor is responsible for its routers as well as the communications line between them.

Not shown is how the vendor forwards the data to any users that get the data from that vendor; this is not the responsibility of the Volpe Center and is not covered in this document.

Not shown in this diagram is that, for purposes of reliability, there are two ASDI servers and two firewalls.



3. Content of the ASDI Data Stream Sent to the Vendors

3.1 Filtering Messages Out of the ASDI Data Stream

The ASDI Server registers with the ETMS Nas.dist process, which provides the ASDI server with the following messages.

- All of the NAS messages received by ETMS.
- All of the TO messages generated by ETMS.
- All of the RT messages generated by ETMS.

Before sending these messages over the ASDI feed, the ASDI server filters the messages in various ways.

- Sensitive message types are filtered out. The only message types currently allowed into the ASDI data stream are AF, AZ, DZ, FZ, RZ, TZ, UZ, RT, TO, and HB messages; see Sections 5 and 6 for an explanation of these messages. Examples of message types that are filtered out are the BZ and FS messages.
- Messages on sensitive flights are filtered out. For example, messages on military flights are not allowed into the ASDI data stream. The specific filtering rules are given later in this section.
- Messages from certain facilities are filtered out. For example, prior to March of 1998, messages from Canadian facilities were filtered out. Currently messages are included from facilities in the United States, Canada, and the United Kingdom; see Section 3.4 for a list of the facilities that send data to ETMS.

The Host computer at a center will sometimes for efficiency pack as many as seven TZ messages into a single message. The ASDI server splits such a composite message into individual TZ messages before sending them over the ASDI feed.

The following are the specific rules that are used to decide which messages are filtered from the ASDI feed.

1. Check to see if the message is an allowed type. If not, e.g., the message is a BZ message, then the message is discarded; it does not go out in the ASDI feed. If it is valid, it is passed to the next step.

2. Look at the code in the message that shows the facility that generated the message. If the code is not in the configuration file that specifies legal facilities, then discard this message; it does not go out in the ASDI feed. If the code is in this file, then pass it to the next step. (Note: There are two feeds--one that contains London data and one that does not. This configuration file controls for each feed whether London data is included.)

3. Look at the aircraft call sign. If it is in the File of Forbidden Call Signs, then discard it; it does not go out in the ASDI feed. If it is not in this file, then pass it to the next step. (The File of Forbidden Call Signs, which is maintained by the Air Traffic Control System Command Center, contains call signs of non-military flights that are considered to be sensitive.)

4. Check to see if the call sign starts with 'N' followed by a digit followed by a digit or letter. That is, check to see if the first three characters of the call sign have the format 'Ndd' or ‘Ndl', where 'd' stands for digit and 'l' for letter. If so, this is considered to be a GA flight; it goes out in the ASDI feed. If not, then this flight is passed to the next step.

5. Check to see if the call sign is exactly 5 letters, the first two of which are ‘CF’ or ‘CG’. If so, this is considered to be a Canadian GA flight; it goes out in the ASDI feed. If not, then this flight is passed to the next step.

6. Check to see if the call sign starts with 'LN' followed by a digit followed by a digit or letter. That is, check to see if the first four characters of the call sign have the format 'LNdd' or ‘LNdl', where 'd' stands for digit and 'l' for letter. If so, this is considered to be a lifeguard flight; it goes out in the ASDI feed. If not, then this flight is passed to the next step.

7. Check to see if the call sign starts with 'TN' followed by a digit followed by a digit or letter. That is, check to see if the first four characters of the call sign have the format 'TNdd' or ‘TNdl', where 'd' stands for digit and 'l' for letter. If so, this is considered to be an air taxi flight; it goes out in the ASDI feed. If not, then this flight is passed to the next step.

8. Check to see if the call sign is three letters followed by a digit. If not, the flight is considered to be military and is discarded; it does not go out in the ASDI feed. If so, this is considered to be a commercial flight; pass it to the next step.

9. Check the first three letters of the call sign to see if they represent an airline for which messages are to be sent in the feed. (The Radiotelephony file, which is maintained by the Air Traffic Control System Command Center, specifies the airlines whose messages are to be included in the feed.) If so, this message is sent out in the ASDI feed. If not, discard this flight; it does not go out in the ASDI feed..

3.2 Removing Fields From the Allowed ASDI Messages

If a message passes the rules given in the previous section and is eligible to be sent out over the ASDI feed, the ASDI Server removes field 11 (remarks), if present, from this message before it is sent out.

3.3 Framing the Messages

Once a NAS or ETMS message has passed the filter and has had fields removed if necessary, that message is ready to be sent over the ASDI feed. Each ASDI message consists of five components.

1. Sequence Number (4 bytes)
2. Date-time stamp (8 bytes)
3. Facility identifier (4 bytes)
4. NAS or ETMS message (variable)
5. Line feed (1 byte)

The first component of a message is a sequence number. This sequence number is stored internally by the ASDI Server as a sixteen bit integer. For purposes of readability, however, this is transmitted as four printable, hexadecimal characters. For example, a value of 17 would be shown as ‘0011’. The first message sent at initialization will have a sequence of ‘0000’. The value after ‘FFFF’ will be ‘0001’. The value ‘0000’ is reserved as an initialization indicator. That is, if the ASDI Server at the hubsite needs to be restarted, the first message will have the sequence number ‘0000’. The recipient will then know that there is a discontinuity in the data stream and can estimate the magnitude of the loss.

The second component of a message is the date-time stamp, which is an eight-byte field in ddhhmmss format; the old feed only has an hhmmss time stamp. The date and time are in Coordinated Universal Time (UTC, formerly known as Greenwich Mean Time). The time given is the time assigned to the message when it was received by ETMS from the data source. For example, if the data came from a Host processor at an ARTCC, it is the time at the local site when the message was received by a local ETMS process.

The third component of a message is a four-byte facility identifier, which replaces the one-byte facility code in the old feed. This field is right justified and filled with leading blanks if necessary. The next section gives the facility identifiers that are being used.

The fourth component of a message is the NAS or ETMS message proper. This message will have a variable length. See Sections 5 and 6 for a explanation of each of the allowed messages.

The fifth component of a message is a line feed character (0x0A), which marks the end of the message.

A sample message is

123F23152309KZAUTZ XYZ123/456 400 300 4217N/12345W

123F is the sequence number.

23152309 is the date-time stamp, indicating that this message was generated on the 23rd of the month at the 15th hour and the 23rd minute and the 9th second.

KZAU indicates that the Chicago center generated this message.

The rest of this message is the data in a TZ message. Not shown is the line feed character that ends the message. Appendix A contains many sample messages.

In summary, when the ASDI server receives a message from ETMS, it does the following.

- It filters out any messages that are not allowed into the feed.
- It removes any fields that are not allowed into the feed.
- It adds a header to the message.
- It sends the filtered, framed message to every client that is registered.

3.4 Facility Identifiers

The following are the facility identifiers in use. The first column contains the one character code used in the old feed, and the second column is the four character identifier used in the new feed. Following the identifier is a brief description of where the facility is located. Note that 'X' and 'x' do not represent a NAS facility; rather, these codes indicate messages that are generated by ETMS, i.e., RT and TO messages. The new feed represents each facility with a four character identifier, which is right justified and filled with leading blanks if necessary. These IDs, especially the TRACON and foreign IDs, are tentative. In some cases, facilities that provide multiple data sources will have the same facility identifier (i.e., ZAN EARTS and ZAN OCS). ODAPS stands for the Oceanic Display and Planning System, which provides NAS messages other than TZs on oceanic flights; position updates on oceanic flights are in the TO message.

```
A KZAB      Albuquerque, New Mexico
B KZBW      Boston, Massachusetts
C KZOB      Oberlin, Ohio (Cleveland)
D KZDV      Denver, Colorado
E PZAN      Anchorage, Alaska
F KZFW      Dallas/Fort Worth, Texas
G KZAU      Aurora, Illinois (Chicago)
H KZHU      Houston, Texas
I KZID      Indianapolis, Indiana
J KZJX      Jacksonville, Florida
K KZKC      Kansas City, Missouri
L KZLA      Los Angeles, California
M KZME      Memphis, Tennessee
N KZNY      New York, New York
```

```
O KZOA       Oakland, California
P KZMP       Minneapolis, Minnesota
Q PZHN       Honolulu, Hawaii  (TZ only)
R KZMA       Miami, Florida
S KZSE       Seattle, Washington
T KZTL       Atlanta, Georgia
U KZLC       Salt Lake City, Utah
V PZLB       American Samoa    (No data)
W KZDC       Washington, District of Columbia
X ETMS       RT message generated by ETMS
Y TZSU       San Juan, Puerto Rico
d KDEN       Denver TRACON (TZ only)
f KDFW       Dallas/Fort Worth TRACON (TZ only)
g KC93       Chicago TRACON (TZ only)
n KN90       New York TRACON (TZ only)
o KOOA       ODAPS Oakland (AF, AZ, DZ, FZ, RZ, UZ only)
v KONY       ODAPS New York (AF, AZ, DZ, FZ, RZ, UZ only)
x ETMS       TO message generated by ETMS
1 KSCT       Southern California TRACON (TZ only)
0 YOWT       CANADA OTHER
2 CCZX       GANDER
3 CCZM       MONCTON
4 CCZU       MONTREAL
5 CCZY       TORONTO
6 CCZW       WINNIPEG
7 CCZE       EDMONTON
8 CCZV       VANCOUVER
< LLON       LONDON
>  ZPA       PACIFIC/AUSTRALIA (No data)
^  ZSA       SOUTH AMERICA     (No data)
```

4. Software Interface, Hardware Interface, and Security

4.1 Software Interface

The interface between the ASDI server running at the ETMS hubsite and the client software running on the vendor's premises is a standard TCP/IP socket. The redundant ASDI servers each listen separately on assigned port numbers for clients that request to be registered to receive the data. When the vendor signs the memorandum of agreement with the FAA, it will be given four IP addresses and port numbers that it can use to connect of the ASDI servers. (Actually, there are four IP addresses for the servers that provide the feed that contains London data for those vendors that qualify to receive the London data, and there are four IP addresses for the servers that provide the feed that does not contain the London data.) The IP addresses that the vendor will use must be provided to the Volpe Center so that Volpe can configure to allow clients from those addresses to connect to the appropriate servers. For sample code that shows how to write the client software, see W. R. Stevens, *Unix Network Programming*, Prentice Hall, Englewood Cliffs, NJ, 1990. See especially Chapter 6, section 6.6, with example programs on pp. 284-286. The surrounding sections also contain useful technical details.

An ASDI server sends identical data to all connected clients. The ASDI server buffers messages for each connected client. The ASDI server will discard data if the client's buffer overflows. Any write to a client that results in an error condition will cause the ASDI server to terminate the connection. If the connection is terminated, the client must re-establish a connection if it is to receive data. No data will be buffered for a disconnected client. A 'port filled' or 'no room in port' condition is not considered to be an error; if either of these conditions is returned when the ASDI server attempts to write to a socket, then this data is buffered, and later an attempt is made to re-transmit.

The ASDI server uses non-blocking write operations to its clients. The ASDI server sends all messages in sequence. The standard interpretation of any missing messages is that they resulted from (1) a buffer overflow forced by the client's not reading the messages fast enough or (2) the client's communication link not being able to accept the ASDI data feed fast enough.

After a vendor opens a socket, it is given 60 seconds to send a vendor name and password to the ASDI server. The message that contains the vendor name and password is called a registration message; once the ASDI server has received a valid registration message, the client is said to be registered. No data is sent until the ASDI server receives a valid registration message. If 60 seconds after the socket connection is made the ASDI server has not received a legal registration message, the ASDI server will close this socket. The table below specifies the format of the registration message, which contains a vendor name and password. Each element of this message may be delimited by one or more spaces. Spaces directly after an = sign are discarded and are not considered part of the vendor name or password.

Registration Message Format

ID	*Keyword specifying id field.*
=	*Field assignment character.*
<vendor name string>	*1 - 80 character alphanumeric string. Chosen by the vendor. May include embedded spaces.*
,	*Field separator character.*
PASSWORD	*Keyword specifying password field.*
=	*Field assignment character.*
<vendor password string>	*1 – 12 character alphanumeric string. Chosen by the vendor.*

Following is an example of a registration message:

ID = I AM A VENDOR , PASSWORD = mypassword

A vendor can choose any number of vendor name/password pairs; the vendors are encouraged to have back-up pairs. They should be communicated to the Volpe contact. It is required that the vendor name include in some way the name of the vendor; this will make it easier for Volpe support personnel to monitor the system and deal with problems.

A vendor will be allowed to connect once to each ASDI server with each password. It is expected that under normal conditions a vendor will have not more than two active connections to the ASDI servers, where one is used for production and one for testing; it is expected that additional connections will be made only in exceptional cases, e.g., if extra testing is needed.

The vendor name and password are needed when a vendor connects to Volpe to get the ASDI feed. How a user connects to a vendor to get the feed is beyond the scope of this document; a user should contact its vendor to learn how to connect to receive the feed from the vendor.

Software Tips for Vendors

Volpe will give a vendor four IP addresses to connect to; these four addresses provide redundancy. Volpe uses two ASDI servers and has two firewalls. By providing four addresses, Volpe can allow vendors to continue to get data if there is a failure of an ASDI server, failure of a firewall at Volpe, or even failure of one ASDI server and one firewall at Volpe. Volpe strongly recommends that a vendor write its client software so that, if it is unable to connect and get data from one of the IP addresses, that it automatically search through all four addresses until it makes a connection. By cycling through the four addresses in this way, a vendor can minimize down time.

Volpe's assumption is that if there is at least one IP address providing data, the vendors will be able to find it. Therefore, if Volpe needs to take down the servers to do maintenance, Volpe will take down one at a time. While the first server is down, vendors can connect to the second; when the first is back in service, the second will be taken down. In situations like this, no planned outage will be announced since it is assumed that the vendor clients can deal with it.

In short, it is Volpe's responsibility to make sure that at least one of the four addresses is providing data; it is the vendor's responsibility to find that address and connect to it. (As a matter of fact, experience shows that all four addresses are operating normally much more than 99 percent of the time.)

Volpe strongly recommends that vendor software discard any unknown messages types. There has been discussion of adding to the message types that are included in the feed. Volpe is required to give at least 60 days notice of any change such as this. Nevertheless, if a vendor's client software discards unknown messages, the vendor can avoid having to develop software against the clock.

4.2 Hardware Interface

This section contains information that a vendor needs in order to physically interface with the ASDI subsystem and to receive ASDI data.

The vendor is responsible for any communication lines to Volpe and for placing a router at Volpe. This router is connected to an Ethernet switch that is in turn connected to the Volpe firewall. Volpe is responsible for the Ethernet switch and for the cable that connects the Ethernet switch to the vendor's router. That is, the line of demarcation between what Volpe is responsible for and what the vendor is responsible for is the port on the vendor's router. Volpe will not touch the vendor's communication lines or his router; the vendor will not be allowed to touch anything in the Volpe operations room except its communications line and router.

Hardware Requirements

There are three hardware requirements that the vendor must satisfy in order to receive ASDI data.

1. The vendor shall install a communication line to Volpe.

2. The vendor shall provide a router with a 10Base-T port that can be installed in a nineteen-inch rack.
3. The vendor install a regular phone line and modem to be used for remote diagnosis of problems with the router and for remote reconfiguration of the router.

Volpe provides rack space, power, and an uninterruptible power supply.

Capacity of the Communications Line

Since the vendor must choose and pay for the communication line between Volpe and the vendor's headquarters, a standard question asked by vendors is: How much bandwidth does this line need? Unfortunately, it is not possible to give an entirely satisfactory answer to this question. The following, however, can be said.

1. As this is written in August of 2000, a 56Kbps line can handle the load for one feed, assuming that the router has extra memory for buffering and uses compression, but not for two feeds.
2. It will sometimes be the case, however, that a vendor will want both a production feed and a test feed. For example, when a change is being made to the feed, sometimes Volpe will make the new feed available on a test basis so that the vendors can test with it before it becomes the production feed. A prudent vendor would want to have enough capacity to connect for both a production feed and a test feed.
3. A larger capacity might be needed depending on what use the vendor plans to make of the line; it is possible that this line might also be used for other data such as Airport Demand Lists.
4. The FAA has plans to increase the amount of data in the feed, so eventually a 56 Kbps line will be insufficient to accommodate the data. For example, TZ messages (which are NAS position update messages) are currently received once every four minutes from centers in the U.S.. The FAA plans to increase the frequency of these messages so that they are received once every minute. This will increase the required bandwidth perhaps by as much as 250 percent. No definite timetable exists for when this extra data will start flowing, but some think that it might start flowing as early as the fall of 2000; it seems likely that this data will be flowing by the summer of 2001.

In summary, even though a 56Kbps line will handle the load today, a prudent vendor will want to be ready for the expected increase in the quantity of data with a line of 256Kbps or even 512Kbps. It should be stressed, however, that the decision on what bandwidth to buy is entirely the responsibility of the vendor; the remarks above are designed only to provide some facts to the vendor who is facing this decision.

Dealing with Equipment Failures

If there is an equipment failure, Volpe will, in cooperation with the vendor, diagnose the failure to determine if the problem lies with the equipment for which Volpe is responsible or with the equipment for which the vendor is responsible. If the problem lies with the equipment for which the vendor is responsible, then Volpe's role ends; it is the responsibility of the vendor to deal with the problem in whatever manner the vendor prefers. If the problem lies with the equipment for which Volpe is responsible, then Volpe is obligated to fix the problem.

Summary of What Needs to Be Done If a Vendor is to Receive the Feed

The steps that must be gone through if a vendor is to connect to an ASDI Server and get the feed are as follows.

1. The vendor must sign both the ASDI memorandum of agreement and also the NBAA Vendor Code of Conduct (see the next section) and return them to the FAA point of contact.
2. The FAA point of contact must inform the Volpe contact that a vendor has signed the documents required in the first step.
3. The vendor must put in a communications line to Volpe and install a router at Volpe. These installations should be coordinated with the Volpe point of contact.
4. The vendor must provide the Volpe contact with the IP address(es) from which it will be connecting to the ASDI Servers.
5. The vendor must provide the Volpe contact with the vendor name/password pairs that it will use.
6. The Volpe contact must provide the vendor with the IP addresses/ports on which the ASDI servers listen for connections.

If a vendor wants to receive the London data, it must go through additional steps spelled out in Section 4.4.

4.3 Security

One of the general requirements of the of the ASDI program is that security be maintained. There are two phases involved in maintaining security.

- Vendors and airlines only engage in permitted activities.
- Vendors, airlines, the Command Center, and Volpe take measures to detect and repel any prohibited activities.

The primary purpose of this section is to spell out which activities are permitted and which are prohibited so that vendors and airlines will not be in doubt about what is permitted. The secondary purpose is to state who is responsible when an prohibited action takes place.

Since ASDI is closely connected with the Collaborative Decision Making (CDM) Program, this section covers both.

Overview

The general principle is: **Any activity that is not explicitly permitted is prohibited.** As this document is written, only the following activities are permitted, and these are only permitted from an approved IP address.

- A CDM participating airline is only permitted to:
  - Register with the Aggregate Demand List (ADL) server at Volpe to receive ADLs.
  - Send CDM flight data to Volpe.
  - Register with the Surface Movement Advisor (SMA) server at Volpe to receive SMA data.
  - Browse the web sites at the Volpe Center and the Command Center.
  - Ping IP addresses at the Volpe Center and at the Air Traffic Control System Command Center.
- A vendor is only permitted to:
  - Register with the ASDI server at Volpe to receive ASDI data.
  - Ping IP addresses at the Volpe Center and at the Air Traffic Control System Command Center.

For purposes of testing and development the Command Center, Volpe, and Metron are permitted to engage in any of these activities. Other parties, e.g., airlines that do not participate in CDM or organizations that only receive ASDI data from a vendor, are prohibited from all activity, unless this activity is approved in writing by the Volpe contact.

It is expected that this list of permitted activities will change over time. For example, it might be the case that different levels of access for the web sites will be defined; that is, some data might be placed on the Command Center and Volpe web sites that vendors will be permitted to browse. Any changes in the list of permitted activities will be documented.

Detailed Guidelines

The following guidelines flesh out the general principle above by pointing out some of the prohibited activities and by specifying who is responsible for a violation of these guidelines.

1. **If a prohibited activity is launched from an organization's IP domain, then that organization is responsible for that violation.**

   The various organizations that are relevant include the ASDI vendors, the participating CDM airlines, the Command Center, and Volpe. For example, if a disgruntled employee engages in a prohibited activity from an

authorized IP address, then that person's employer, whether vendor, airline, Command Center, or Volpe, is responsible for this violation. For another example, if a hacker uses the Internet to break into a vendor's network and then engages in prohibited activity, then this vendor is responsible for this attack.

2. **A vendor shall only allow Volpe or the Command Center to be accessed by parties operating from an approved IP address.**

   That is, the vendor is responsible for providing network security.

3. **Any attempt to access an IP address other than those that are authorized shall be prohibited.**

   At the moment, the only authorized IP addresses are those of the ASDI Servers, the ADL Servers, the SMA server, the Flight Data Front End, the Volpe web site, and the Command Center web site. (ASDI and ADLs are both accessed through the Volpe firewall.) In addition, there might be IP addresses that the FAA and Volpe do not see, e.g., the IP address that an airline uses to register with a vendor to receive ASDI data.

4. **Any attempt to access a port other than a prescribed, published port shall be prohibited.**

   For example, if someone tries to access an unused port on the ASDI Server, this will be construed to be an attack.

5. **IP spoofing shall be prohibited.**

   That is, it is prohibited for one party to attempt to use another's IP address.

6. **No vendor is allowed to access the routers of any other vendor (unless permission has been given in writing, with a copy provided to the Volpe contact).**

   Since the various vendor routers at Volpe are all connected through an Ethernet switch, conceivably one vendor could attempt to access the router of another vendor. Such an attempt is prohibited, unless it has been authorized in writing.

7. **An organization shall only capture or read data that it is explicitly authorized to look at. In particular, a vendor shall not read, store, or make copies of Aggregate Demand Lists (ADLs).**

   A vendor is prohibited from using a packet sniffer to capture data that it is not authorized to access.

8. **A password is associated with a set of IP addresses. Use of that password from some other IP address is prohibited.**

   This means that it is not allowed for one vendor to use another's password.

Summary

These guidelines are not exhaustive in that they do not spell out all of the prohibited activities; they only deal with the more important or the more likely prohibited activities. Again, the general principle is: **Any activity that is not explicitly permitted is prohibited.**

4.4 NBAA Vendor Code of Conduct

It was mentioned in Section 3.1 that the Air Traffic Control System Command Center maintains a File of Forbidden Call signs. If a call sign is in this file, Volpe will not include in the ASDI feed any messages that contain data on this call sign. While this mechanism is sufficient for some cases, it is not always desirable. For example, a corporation might like to see its aircraft while no one else can see them. To achieve this, the National Business

Aviation Association (NBAA) developed the idea of keeping a list of sensitive call signs. Volpe will include messages containing data on these sensitive call signs in the ASDI feed A vendor is not allowed to send these call signs to any user, and in particular is not allowed to show these call signs on its displays or in its databases, unless the vendor has written permission from the aircraft operator to show the call signs.

Before a vendor is allowed to receive the ASDI feed, it is required to sign the NBAA Vendor Code of Conduct, which contains four provisions.

1. Vendors acknowledge that the ASDI information is intended to be used for improving flight management and to increase the safety and efficiency of aviation operations.

2. On a monthly basis, vendors will obtain an electronic list of aircraft registration numbers from NBAA. These numbers represent those aircraft for which owners have requested privacy protection in both real time and historical formats. This list will be referred to as the Blocked Aircraft Registration Request (BARR) database.

3. Vendors will not display, on their decision support displays or in historical formats, any registration information associated with aircraft contained in the BARR database unless they have written permission from the registered owner of the aircraft. After written permission has been obtained, the vendor and the operator of the relevant aircraft must agree upon the distribution of the BARR data.

4. Vendors will ensure that reasonable steps are taken to prevent disclosure of information associated with aircraft contained in the BARR database.

4.5 Handling the London Data

British flight data is sent to the ETMS hubsite at the Volpe Center; this data is called the London data. The British authorities have placed on the FAA the requirement that the distribution of this data be restricted effective 1 December 1999. Volpe operates two ASDI servers that provide a feed that contains the London data, and it also operates two ASDI servers that provide a feed from which the London data has been removed. This section discusses this requirement and its implications for ASDI vendors.

What Users Can Get the London Data?

The British require that a user only receive the London data if it satisfies one or both of the following conditions.

- It is an air carrier. This includes cargo carriers.
- It owns and operates aircraft in Europe, and, therefore, pays landing feed and air traffic control feeds in Europe.

If a user satisfies at least one of these conditions, it is said to be an *approved recipient* of the London data. The requirement imposed by the British, then, can be stated as saying that only approved recipients can receive the data. If a user does not fulfill at least one of these two conditions, then that user is not allowed to receive any of the London data.

What Vendors Can Get the London Data?

By default, a vendor does not get the London data. That is, unless a vendor has been given explicit permission to get the London data, Volpe will only allow that vendor to register to get the feed that does not contain the London data. If a vendor wants to get approval to register for the feed that contains the London data, it must complete three steps.

1. The vendor must certify in writing to the FAA contact (with a copy to the Volpe contact) that it has at least one user as a client that is an approved recipient of the London data.

2. The vendor must submit to the FAA contact (with a copy to the Volpe contact) a plan that spells out how it will handle the London data so that it can guarantee that the London data will only go to approved recipients. The vendor must receive approval of this plan from the FAA contact.
3. The vendor must certify to the FAA contact (with a copy to the Volpe contact) that it has implemented this approved plan.

Once a vendor has completed these three steps, Volpe will allow it to register to receive the London data. If a vendor later decides that it would like to change its plan for dealing with the London data, it can submit a revised plan to the FAA. Once the FAA approves a revised plan, it then becomes the plan that the vendor is required to follow.

If at any time the FAA discovers that a vendor is sending London data to an organization that is not an approved recipient, then the FAA has the option of immediately cutting off that vendor's access to the feed that contains London data; that vendor would then only be allowed to access the feed that does not contain the London data.

How Can the London Data be Identified?

Each message on the ASDI feed has four characters that indicate the facility that generated the message. These four characters are in bytes 13-16 (where the first byte is numbered 1). If these four bytes contain 'LLON', then this messages is from London. If these four characters are anything else, then this message is not from London.

5. NAS Message Formats

5.1 Introduction

There are two classes of messages that are included in the ASDI feed. The first class is NAS messages provided by the FAA. These are described below using material extracted from section three of the FAA document, *Computer Program Functional Specifications: Remote Outputs*, Model A4el.2, NAS-MD-315, 20 November 1997. The second class is messages that ETMS or the ASDI server generates, namely RT, TO, and HB messages. These are described in detail in Section 6.

For sample messages, see Appendix A; these can be used to test software that is designed to handle the ASDI feed. For the formats of the fields in the NAS messages, see Appendix B below.

All references are to FAA NAS documents NAS-MD-315, NAS-MD-311 and NAS-MD-312.

The following messages were taken from the NAS-MD-315 (chapter 3) document with minor editing for this document. Note that the NAS-MD-315 document is the definitive resource for all NAS message formats and should be consulted whenever there is any question regarding their content.

5.2 NAS Message Syntax Symbols explained

The following syntax rules are used for field specifiers:
() brackets an optional field
a is defined as an alphanumeric, i.e., any upper case letter or digit
d is defined as a digit
L is defined as an upper case letter only

For example:

La (a) (a) (a) (a) (a) specifies an upper case letter followed by at least one alphanumeric. As many as five optional alphanumerics may follow.

a (/) a (a) (a) (a) (a) (a) (a) (a) (a) (a) specifies a minimum of two alphanumeric characters and a maximum of eleven, with an optional slash between the first two.

(L) dddd specifies a four to five byte field. The last four bytes must be digits. If there are five bytes, the first byte must be an upper case letter.

5.3 NAS AF Message

The Flow Control Amendment Information (AF) message provides revised flight plan data whenever a flight plan is amended.

A recent Host change is the sending of an AF message with a field 02 (aircraft identifier) amendment. Previously the Host sent an RZ/FZ sequence. This change may not be implemented at all Hosts simultaneously so both contexts may be received from different facilities.

Another recent Host change is the sending of an AF message with a field 07 (proposed departure time) amendment. Previously the Host sent an RZ/FZ sequence. This change may not be implemented at all Hosts simultaneously so both contexts may be received from different facilities.

AF Field Format

Field No.	**Function**	**Description**	**Bytes**
0l	Message Type	2 characters - AF	2
	Field Separator	1 space	1
02	Aircraft Ident.	2-11 characters La(a) (a) (a) (a) (a) (/dda) Proposed - AID/CID Active - AID only	2-11
	Field Separator	1 space	1
26	Departure Point	2-12 characters aa(a) (a) (a)(a)(a) (a) (a) (a) (a) (a)	2-12
	Field Separator	1 space	1
27	Destination	2-12 characters aa(a) (a) (a)(a)(a) (a) (a) (a) (a) (a)	2-12
	Field Separator	1 space	1
12	Field Reference	2 character logical field number (d)d	1-2
	Field Separator	1 space	1
17**	Amendment Data	Data required for field modification	
	Field Separator	1 space	1
	Additional Amendment Information*		

*Field reference number and amendment data fields are repeated for each required field amendment.

**Fields 06 and 07 will be formatted as follows:

a. For an active flight plan, field 06 will be the present position fix in latitude/longitude format and field 07 will be the time at the present position fix with an 'E' prefix

b. For a proposed flight plan, field 06 will be as amended.

5.4 NAS AZ Message

Flow Control Arrival Information (AZ) messages are used to provide the ETMS with arrival data for all eligible arriving flights.

AZ Field Format

Field No.	Function	Description	Bytes
0l	Message Type	2 characters - AZ	2
	Field Separator	1 space	1
02	Aircraft Ident.	2-7 characters La(a)(a)(a)(a)(a)	2-7
	Field Separator	1 space	1
26	Departure Point	2-12 characters aa(a) (a) (a)(a)(a) (a) (a) (a) (a) (a)	2-12
	Field Separator	1 space	1
27	Destination	2-12 characters aa(a) (a) (a)(a)(a)(a)(a)(a) (a) (a)	2-12
	Field Separator	1 space	1
28	Arrival Time	4-5 characters (L)dddd	4-5
	Field Separator	1 space	1

Note: Field 28 may have an optional prefix of ‘A’ to indicate that the time is actual or ‘E’ to indicate that it is estimated.

5.5 NAS DZ Message

A Departure message (DZ) is transmitted for all eligible initially activated flight plans when the activation message is not from an adjacent NAS.

DZ Field Format

Field No.	Function	Description	Bytes
0l	Message Type	2 characters - DZ	2
	Field Separator	1 space	1
02	Aircraft Ident.	6-11 characters La(a)(a) (a) (a) (a)/dda	6-11
	Field Separator	1 space	1
03	Aircraft Data	Number of aircraft in the flight and/or heavy jet indicator ((d)(d) or (d)(a) or (a)) separated by a slash with 2 to 4 characters of aircraft type (aa(a)(a)) followed by an optional slash and a letter (L).	2-9
	Field Separator	1 space	1
26	Departure Point	2-12 characters aa(a)(a) (a) (a) (a)(a) (a) (a) (a) (a)	2-12
	Field Separator	1 space	1
07	Actual Departure Time	5 characters including the prefix D (Departure), E (Active) Ldddd	5
	Field Separator	1 space	1
27	Destination	2-12 characters aa(a)(a)(a)(a)(a)(a)(a)(a)(a)(a)	2-12
	Field Separator	1 space	1
28	ETA	4 characters representing the Estimated Time of Arrival at destination in hours and minutes (ETA supplied only if ETE was available) - dddd	4
	Field Separator	1 space	1

5.6 NAS FZ Message

Flight Plan Information (FZ) Messages are used to provide the ETMS with flight plan data for eligible flight plans.

A recent Host change is the sending of an AF message with a field 02 (aircraft identifier) amendment. Previously the Host sent an RZ/FZ sequence. This change may not be implemented at all Hosts simultaneously so both contexts may be received from different facilities.

Another recent Host change is the sending of an AF message with a field 07 (proposed departure time) amendment. Previously the Host sent an RZ/FZ sequence. This change may not be implemented at all Host simultaneously so both contexts may be received from different facilities. The maximum message size is less than 440 bytes.

FZ Field Format

Field No.	Function	Description	Bytes
0l	Message Type	2 characters - FZ	2
	Field Separator	1 space	1
02	Aircraft Ident.	2-11 characters La(a)(a)(a)(a)(a)(/dda) Proposed FP - ACID/CID, Active FP - ACID only	2-11
	Field Separator	1 space	1
03	Aircraft Data	Number of aircraft in the flight and/or heavy jet indicator ((d)(d) or (d)(a) or (a)) separated by a slash with 2 to 4 characters of aircraft type (aa(a)(a)) followed by an optional slash and a letter (L).	2-9
	Field Separator	1 space	1
05	Speed	2-4 characters One of the following: a. True Air dd(d)(d) b. MACH Speed Lddd c. Classified Speed SC d. Ground Speed ddd	2-4
	Field Separator	1 space	1
06*	Coordination Fix	2-12 alphanumerics aa(a)(a)(a)(/)(a)(a)(a)(a)(a)(a)	2-12
	Field Separator	1 space	
07*	Coordination Time	5 characters Ldddd D, E or P followed by a 4 digit time	5
	Field Separator	1 space	1
08	Assigned Altitude (Active)	2 - 7 characters One of the following: a. (d)dd - altitude b. (d)ddB(d)dd - a block of altitudes	2-7
	Field Separator	1 space	1
09	Requested Altitude (Proposed)	2-7 characters One of the following: a. (d)dd - altitude b. (d)ddB(d)dd - a block of altitudes	2-7
	Field Separator	1 space	1
10	Route	Refer to NAS-MD-312 for details or the examples in Appendix A of this document	variable
	Field Separator	1 space	

*Fields 06 and 07 will be formatted as follows:

a. When field 07, coordinate time, is transmitted as a 'P' or 'D' time, then fields 06 and 07 (the 4 digit time) will be transmitted as filed.
b. When field 07 is transmitted as an 'E' time, then field 06 will be the present position fix in latitude/longitude format and the 4 digit time of field 07 will be the time at the present position fix.

5.7 NAS RZ Message

Flow Control Cancellation (RZ) messages are used to provide the ETMS with cancellation data for all eligible flight plans.

A recent Host change is the sending of an AF message with a field 02 (aircraft identifier) amendment. Previously the Host sent an RZ/FZ sequence. This change may not be implemented at all Hosts simultaneously so both contexts may be received from different facilities,

Another recent Host change is the sending of an AF message with a field 07 (proposed departure time) amendment. Previously the Host sent an RZ/FZ sequence. This change may not be implemented at all Hosts simultaneously so both contexts may be received from different facilities.

RZ Field Format

Field No	Function	Description	Bytes
01	Message Type	2 characters - RZ	2
	Field Separator	1 space	1
02	Aircraft Ident.	2-11 characters La(a)(a)(a)(a)(a)(/dda) Proposed FP - AID/CID Active - AID only	2-11
	Field Separator	1 space	1
26	Departure Point	2-12 characters - aa(a)(a)(a)(a)(a)(a)(a)(a)(a)(a)	2-12
	Field Separator	1 space	1
27	Destination	2-12 characters - aa(a)(a)(a)(a)(a)(a)(a)(a)(a)(a)	2-12
	Field Separator	1 space	1

5.8 NAS TZ Message

Flow Control Track/Flight Data Block Information (TZ) messages are used to provide a position update along with other information used in a data block. TZ messages are transmitted to ETMS on a cyclic basis on all flat-tracked and non-tentative free-tracked eligible flight plans. The current frequency is one message every one to five minutes. The interval between reports is set by the sending facility.

TZ message transmissions on eligible flight plans will be terminated when a handoff to an adjacent center is accepted or the current clock time is greater than the outbound boundary crossing time, whichever occurs first. TZ message transmissions will be terminated from the non-ARTCC facilities under rules set locally at each facility.

There is no way to differentiate TZ messages generated by the Host and those sent through the Host by a directly connected ARTS facility. This means that currently it is only possible to see facility identifiers for five TRACONS: New York, Chicago, Denver, Dallas-Ft. Worth, and Southern California. Messages from all other TRACONS have a facility ID for the sending facility (usually an ARTCC) that the messages pass through on the way to ETMS.

TZ Field Format

Field No.	Function	Description	Bytes
01	Message Type	2 characters - TZ	2
	Field Separator	1 space	1
02*	Aircraft Ident.	6-11 characters La(a)(a)(a)(a)(a)/dda	6-11
	Field Separator	1 space	1
05	Ground Speed	3 digits, ddd, if not available ddd will be 3 zeros	3
	Field Separator	1 space	1
08	Altitude	One of the following: a. (d)dd – This is the aircraft's assigned altitude. This is reported if the mode C altitude reported by the aircraft's transponder is less than 200 feet from its assigned altitude. b. (d)ddT – This is an altitude that an aircraft has been cleared to; it is proceeding toward this altitude but has not yet reached it. It might be given a series of temporary altitudes. c. (d)ddB(d)dd – The aircraft is assigned a block of altitudes; the first given is the lower altitude and the second is the upper, where the upper and lower limits are allowed. d. (d)ddC – This is the mode C altitude reported by the aircraft's transponder. This is reported if the mode C altitude is more than 200 feet from its assigned altitude. e. OTP/(d)dd – This is an on-top clearance. The flight can, within certain limits, be anywhere at or above the specified altitude.	2-7
	Field Separator	1 space	1
23	Track Position Coordinates	12 characters, latitude/longitude in degree minutes ddddL/dddddL	12
	Field Separator	1 space	1

*Field 02 can have one of four formats

La(a)(a)(a)(a)(a)	no CID, typically foreign data sources
La(a)(a)(a)(a)(a)/000	There is no Host correlated flight plan, this is from an EARTS or an ARTS.
La(a)(a)(a)(a)(a)/FFF	There is no Host correlated flight plan, data is from an ARTS facility only.
La(a)(a)(a)(a)(a)/dda	The typical situation.

5.9 NAS UZ Message

Flow Control Update Information (UZ) messages are used to provide current flight plan information on active eligible flights that enter an ARTCC. Speaking loosely, then, a UZ is an ARTCC boundary crossing message. The UZ message is transmitted by the receiving NAS En Route Center when the flight plan is eligible for ETMS message transmission. The maximum message size is less than 440 bytes.

UZ Field Format

Field No.	Function	Description	Bytes	
0l	Message Type	2 characters - UZ		2
	Field Separator	1 space	1	
02	Aircraft Ident.	2-7 characters ACID La(a)(a)(a)(a)(a)	2-7	
	Field Separator	1 space	1	
03	Aircraft Data	Number of aircraft in the flight and/or heavy jet indicator ((d)(d) or (d)(a) or (a)) separated by a slash with 2 to 4 characters of aircraft type (aa(a)(a)) followed by an optional slash and a letter (L).	2-9	
	Field Separator	1 space	1	
05	Speed	2-4 characters, true air speed, mach speed, classified speed dd(d)(d), Lddd, SC		2-4
	Field Separator	1 space	1	
06	Boundary Crossing Point Inbound	9-12 alphanumerics latitude/longitude of boundary crossing point dddd(L)/(d)dddd(L)	9-12	
	Field Separator	1 space	1	
07	Calculated Inbound Boundary Crossing Time	5 characters Edddd	5	
	Field Separator	1 space	1	
08	Altitude	One of the following: a. (d)dd - altitude b. (d)ddB(d)dd - a block of altitudes	2-7	
	Field Separator	1 space	1	
10	Route	Refer to NAS-MD-312 for details or the examples in Appendix A of this document	variable	
	Field Separator	1 space	1	

6. ETMS Message Formats

6.1 RT Message

The purpose of the RT messages is to provide data from ETMS that is not other wise available. For example, an RT message will contain ETMS's current prediction of the wheels-up and wheels-down times for a flight, where these predictions are based on all the information available to ETMS. ETMS generates an RT message for a flight under a variety of circumstances, with the most common being the receipt of an FS, FZ, or UZ message on that flight. (An FS message is an internal message that ETMS generates when a flight in the Official Airline Guide is loaded into the active ETMS databases; this typically happens fifteen hours before the flight is scheduled to depart.)

A RT message has various information stored within its compressed structure. Since this data is used on an ASCII communications link and is quite large, a binary to ASCII conversion algorithm was used for certain fields. The following is the field specifiers for this data. The comments on certain lines referring to 1 byte format, 2 byte format and 3 byte format refer to the binary encoding. Section 7 contains algorithms that map these values into integers. If there is no reference to special formatting, then the field is an unformatted byte stream.

Since this software was originally coded for HP/APOLLO workstations using the DOMAIN/OS operating system, all times are whole number of seconds since January 1, 1980. The RT message will continue to use this format even though ETMS has now changed to the UNIX platform. Latitude and longitude positions are expressed as signed values where the values are degrees times sixty plus minutes.

Following is a table specifying the data types of the flight index, waypoints, sectors, fixes, airways, centers and route fields.

Data Type	Format
Flight index	32 bit unsigned integer
Waypoints	Array of 2*3 byte format (6 bytes per array entry)
Sectors	Array of 6 byte entries
Fixes	Array of 6 byte entries
Airways	Array of 6 byte entries
Centers	Array of 1 byte entries
Route	Variable length byte stream

Appendix A contains examples of RT messages. Section 7 includes sample code for decoding RT messages. Appendix C contains explanations of some of the fields in the RT message.

The following is the RT record structure:

```
01-02   RT
03-03   SPACE
04-10   ACID (BLANK FILLED)
11-13   CID  (BLANK IF NONE)
14-15   two spaces
16-21   ARRIVAL FIX
22-24   DEPARTURE DAY                    (3 BYTE FORMAT)
25-27   EDT                              (3 BYTE FORMAT)
28-30   CDT                              (3 BYTE FORMAT)
31-33   ETA                              (3 BYTE FORMAT)
34-36   CTA                              (3 BYTE FORMAT)
37-39   ARRIVAL FIX TIME                 (3 BYTE FORMAT)
40-40   FLIGHT_STATUS
41-41   AC_PHYSICAL_CLASS
42-42   AC_USER_CLASS
43-44   WAYPOINTS count                  (2 BYTE FORMAT)
45-45   SECTORS count                    (1 BYTE FORMAT)
46-46   FIXES count                      (1 BYTE FORMAT)
47-47   AIRWAYS count                    (1 BYTE FORMAT)
```

```
 48-48   CENTERS count                       (1 BYTE FORMAT)
 49-50   # bytes in route                    (2 BYTE FORMAT)
 51-53   high-order 16 bits of flight index  (3 BYTE FORMAT)
 54-56   low-order 16 bits of flight index   (3 BYTE FORMAT)
 57-59   OGTD                                (3 BYTE FORMAT)
 60-62   OGTA                                (3 BYTE FORMAT)
 63-66   departure airport
 67-70   arrival airport
 71-71   departure center
 72-72   message type                        (1 BYTE FORMAT)
 73-???  array of waypoints for count above
         array of sectors   for count above
         array of fixes     for count above
         array of airways   for count above
         array of centers   for count above
         field 10 - route field    for count above
one character Line feed
```

Note that the structure is rigid up to byte 72. Byte 73 onward contains the information specified by the counts in bytes 43-50. There may not be any data or there may be hundreds of bytes of data in this variant portion of the line. The line like all other lines is terminated by a line feed (hex 0A).

The binary data in a RT record is stored using a six bit mapping scheme. Refer to the sample decoding routine in Section 7 for details.

6.2 TO Message

A TO message is generated by the ETMS when Oceanic position reports are received via the ARINC network. The ETMS software translates these messages into TO messages

All data in the TO record structure is ASCII data. The following describes the fields of the TO record structure. Each component is delimited by a space. Appendix C contains explanations of some of the fields in this message.

Field	Data Type	Required (R)/ Optional (O)
1	TO (Message Type)	R
2	ACID (NAS field 2)	R
3	CALCULATED SPEED	R
4	REPORTED POSITION REPORT	* R
5	1ST PLANNED POSITION REPORT	* O
6	2ND PLANNED POSITION REPORT	* O
7	DEPARTURE AIRPORT	R (‘-‘ if not known)
8	ARRIVAL AIRPORT	R (‘-‘ if not known)

* A position report consists of three, space delimited ASCII strings representing, time, altitude and lat./lon at position.

The following is a sample TO message, followed by a description of each field.

TO AFR4572 528 30/0359 350 2800N/05000W 30/0523 350 3700N/04000W 30/0603 350 4130N/03500W KEWR-

AFR4572	Aircraft identifier, refer to Appendix B for further details
528	Calculated speed. The speed is computed from the two reported points and the specified times at those points, this is a very inaccurate computation and should be used after applying a validity test. This is always a three digit number. The units are nautical miles per hour for this calculated ground speed.
30/0359 350 2800N/05000W	Reported Position Report Field 1: Time is the two digit day of the month, a slash, the two digits of hour and two digits of minute. All information is presented in UTC time. This date is the thirtieth of the month at 0359UTC. Field 2: Current Altitude, always three digits. Field 3: Lat/lon
30/0523 350 3700N/04000W	Optional first planned reporting point
30/0603 350 4130N/03500W	Optional second planned reporting point
KEWR	Departure airport - Newark
-	Arrival airport – unknown

6.3 HB Message

The ASDI server generates a heartbeat message to all connected clients every 10 seconds. This message is sent so that clients will know that the connection to the server is intact. This message is composed of the same 5 components that comprise all messages sent by the ASDI server. The Facility Identifier field is filled with blanks and the message field consists of the two characters 'HB'. The message is terminated by a line feed. A sample message follows:

123F23152309 HB

where the sequence number is **123F**, the time stamp is **23152309**, the facility identifier field is blank, and the message type field is **HB**. An invisible line feed character follows the message type field.

7. RT Decoding Algorithm

The following is a sample of C code that illustrates algorithms that decode an RT message. This is included so that industry developers can see how to handle this ETMS message. This is sample code only and is not guaranteed in any way.

SAMPLE RT DUMP SOFTWARE

The following routine converts three eight bit characters into one sixteen bit word using a restrictive character set. A sixteen bit value had originally been split into three pieces, each one nominally six bits long (3 * 6 = 18 not 16). This routine rebuilds the sixteen bit value. This routine can be used to decode all 3-byte format fields.

```
void map_from_3_bytes ( char *text, short *i){    *i = ((eight_to_six[(short)
text[0]]-1)*62 +
                     (eight_to_six[(short)text[1]]-1))*62 +
                     (eight_to_six[(short)text[2]]-1);
}
```

The following routine converts two eight bit characters in the restricted character set into a sixteen bit integer. This routine can be used to decode all 2-byte format fields.

```
void unmap_word(char *text, short *i){    *i = (eight_to_six[(short)text[0]] -
1)*62 +
         (eight_to_six[(short)text[1]] -1)
}
```

The following routine converts an eight bit byte in the restricted character set into a sixteen bit integer. This routine can be used to decode all 1-byte format fields.

```
 void unmap_byte (char *text, short *i){    *i = eight_to_six[(short)text[0]]
-1;}
```

The following routine displays the lat/lon.

```
#define ABS(X) ( (X) < 0 ? -(X) : (X) )

 void display_way (short x, short y )
{
    short lat, lon;
    char  clat, clon;
    short i1, i2, i3, i4;

    lat = x;
    lon = y;

    if (lon > 180*60)
      lon = lon -360 * 60;

    if (lat < 0)
    {
        clat = 'S';
      lat = ABS(lat);
    } else
      clat = 'N';

    if (lon < 0)
    {
      clon = 'E';
      lon = ABS(lon);

    } else
      clon = 'W';

    i1 = lat/60;
    i2 = lat%60;
    i3 = lon/60;
    i4 = lon%60;
    printf("(%02d%02d%c/%03d%02d%c) ", i1, i2, clat, i3, i4, clon);
}
```

The following routine decodes the flight_index portion of a RT record.

```
void decode_flight_index (char  *text, unsigned long *index)
{
    short int2[2];

    map_from_3_bytes ((char *) &text[0], &int2[0]);
    map_from_3_bytes ((char *) &text[3], &int2[1]);
    *index = (unsigned long) int2[0];
    *index = ((unsigned long) *index << 16) | ((unsigned long) int2[1]
&0xffff);
```

This is the character mapping table to convert eight bits into a six bit character set.

```
static short eight_to_six[] = {
/*00 ?   ?   ?   ?   ?   ?   ?   ?   ?   ?  'LF'  ?   ?   ?   ?   ? */
     58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58,  58, 58, 58, 58, 58,

/*16 ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ? */
     58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58,

/*32 ' ' '!' '"' '#' '$' '%' '&' ''' '(' ')' '*' '+' ',' '-' '.' '/' */
      1, 43, 44, 40, 58, 46, 47, 48, 49, 50, 42, 41, 51, 52, 39, 38,

/*48 '0' '1' '2' '3' '4' '5' '6' '7' '8' '9' ':' ';' '<' '=' '>' '?'*/
      2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 53, 54, 55, 56, 57, 58,

/*64 '@' 'A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' 'L' 'M' 'N' 'O'*/
     59, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,

/*80 'P' 'Q' 'R' 'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z' '[' '\' ']' '^' '_'*/
     27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 60, 58, 61, 58, 58,

/*96 ''' 'a' 'b' 'c' 'd' 'e' 'f' 'g' 'h' 'i' 'j' 'k' 'l' 'm' 'n' 'o'*/
     58, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,

/*102'p' 'q' 'r' 's' 't' 'u' 'v' 'w' 'x' 'y' 'z' '{' '|' '}' '~' del*/
     27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 62, 45, 63, 58, 58,

/*118 ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ?   ? */
      58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58,
      58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58,
      58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58,
      58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58,
      58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58,
      58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58,
      58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58,
      58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58
};
```

This is the inverse table, it maps the restrictive six bit character set back into the eight bit set

```
static short six_to_eight[] = { /* 00    'LF'    ' '    '0'    '1'    '2'
'3'  */           10,      32,     48,     49,     50,     51,

/* 06    '4'     '5'    '6'    '7'    '8'    '9'  */
          52,     53,    54,    55,    56,    57,

/* 12    'A'     'B'    'C'    'D'    'E'    'F'  */
          65,     66,    67,    68,    69,    70,

/* 18    'G'    'H'    'I'    'J'    'K'    'L'   */
          71,    72,    73,    74,    75,    76,

/* 24    'M'    'N'    'O'    'P'    'Q'    'R'   */
          77,    78,    79,    80,    81,    82,
```

```
/* 30    'S'    'T'    'U'    'V'    'W'    'X'   */
          83,    84,    85,    86,    87,    88,

/* 36    'Y'    'Z'    '/'    '.'    '#'    '+'   */
          89,    90,    47,    46,    35,    43,

/* 42    '*'    '!'    '"'    '|'    '%'    '&'   */
          42,    33,    34,   124,    37,    38,

/* 48    '''    '('    ')'    ','    '-'    ':'   */
          39,    40,    41,    44,    45,    58,

/* 54    ';'    '<'    '='    '{'    '}'    '@'   */
          59,    60,    61,    62,    63,    64,

/* 60    '['    ']'    ' '    ' '                 */
          91,    93,   123,   125 };
```

The following C program is given a line of RT data and produces a formatted output.

```
void parse_rt( char *text){ short        n_waypoints,n_sectors; short
n_fixes,n_airways,n_centers; short        route_size;
 unsigned long   flight_index;
 short      way_x[90];
 short      way_y[90];
 int        offset;
 short      i,j,k;

   /*** ACID & CID ***/
   printf("\nacid: %7.7s  cid: %3.3s  ", (char *)&text[3], (char *)&text[10]);

   /*** ARRIVAL FIX ***/
   printf("arrival_fix: %6.6s  ", (char *)&text[15]);

   /*** DEPARTURE DAY ***/
   map_from_3_bytes ((char *)&text[21], &i);
   if (i == -1)
       printf ( "departure day: NONE\n");
   else
       printf ( "departure day: %d\n", i);


   /*** EDT ***/
   map_from_3_bytes ((char *)&text[24], &i);
   j = i / 60;
   k = i % 60;
   if (i == -1)
      printf ( "   edt: NONE  ");
   else
      printf ( "   edt: %02d:%02d  ", j,k);

   /*** CDT ***/
   map_from_3_bytes ((char *)&text[27], &i);
   j = i / 60;
   k = i % 60;
   if ( i == -1)
      printf ( "cdt: NONE  ");
   else
      printf ( "cdt: %02d:%02d  ", j,k);

   /*** ETA ***/
   map_from_3_bytes ((char *)&text[30], &i);
   j = i / 60;
   k = i % 60;
   if (i == -1)
      printf ( "eta: NONE  ");
   else
      printf ( "eta: %02d:%02d  ", j,k);

   /*** CTA ***/
   map_from_3_bytes ((char *)&text[33], &i);
   j = i / 60;
   k = i % 60;
```

```
   if (i == -1)
      printf ( "cta: NONE  ");
   else
      printf ( "cta: %02d:%02d  ", j,k);

   /*** ARRIVAL FIX TIME ***/
   map_from_3_bytes ((char *)&text[36], &i);
   j = i / 60;
   k = i % 60;
   if( i == -1)
      printf (" arrival_fix_time: NONE\n");
   else
      printf (" arrival_fix_time: %02d:%02d\n", j,k);

   /*** FLIGHT_STATUS, AC_PHYSICAL_CLASS & AC_USER_CLASS ***/
   printf("   flight_status: %1.1s  ac_physical_class: %1.1s  ac_user_class:
%1.1s\n",(char *)&text[39], (char *)&text[40], (char *)&text[41]);


   /*** #WAYPOINTS, #SECTORS, #FIXES, #AIRWAYS, #CENTERS & ROUTE SIZE ***/
   unmap_word((char *)&text[42], &n_waypoints);
   unmap_byte((char *)&text[44], &n_sectors);
   unmap_byte((char *)&text[45], &n_fixes);
   unmap_byte((char *)&text[46], &n_airways);
   unmap_byte((char *)&text[47], &n_centers);
   unmap_word((char *)&text[48], &route_size);

   printf("   #waypoints: %d  #sectors: %d  #fixes: %d  #airways: %d
#centers: %d  route_size: %d\n", n_waypoints, n_sectors, n_fixes, n_airways,
n_centers, route_size);


   /*** FLIGHT INDEX ***/
   decode_flight_index ((char *)&text[50], &flight_index);
   printf("   flight_index: %lu  ", flight_index);

   /*** OGTD ***/
   map_from_3_bytes ((char *)&text[56], &i);
   j = i / 60;
   k = i % 60;
   if (i == -1)
      printf ( "ogtd: NONE  ");
   else
      printf ( "ogtd: %02d:%02d  ", j,k);
```

```
   /*** OGTA ***/   map_from_3_bytes ((char *)&text[59], &i);   j = i / 60;
k = i % 60;
   if (i == -1)
      printf ( "ogta: NONE\n");
   else
      printf ( "ogta: %02d:%02d\n", j,k);

   /*** DEPARTURE AIRPORT & ARRIVAL AIRPORT ***/
   printf("   departure_airport: %4.4s  arrival_airport: %4.4s\n",
      (char *)&text[62], (char *)&text[66]);

   /*** DEPARTURE CENTER & MESSAGE TYPE ***/
   unmap_byte ((char *)&text[71], &i);
   printf("   departure_center: %1.1s  message_type: %d\n",
(char *)&text[70], i);


   /*** END OF FIXED PART OF MESSAGE, NOW PROCESS VARIABLE PART ***/
   offset = 72;

   /*** WAYPOINTS ***/
   printf ("%-15s","   waypoints:");
   for (j = 0; j < n_waypoints; ++j ) {
       map_from_3_bytes((char *)&text[offset], &way_x[j]);
       offset = offset + 3;
       map_from_3_bytes((char *)&text[offset], &way_y[j]);
       offset = offset +3;
   }
   for( i = 1; i <= n_waypoints; ++i) {
       display_way (way_x[i-1], way_y[i-1]);
       if ( (i%6 == 0) && (n_waypoints > i) )
           printf ("\n               ");
   }

   /*** SECTORS ***/
   printf ("\n%-15s", "   sectors: ");
   for( i = 1; i <= n_sectors; i++) {
       printf("%6.6s  ", (char *)&text[offset]);
       offset = offset + 6;
       if ((i%10 == 0) && (n_sectors > i)  )
           printf("\n               ");
   }


   /*** FIXES ***/
   printf ("\n%-15s", "   fixes: ");
   for (i = 1; i <= n_fixes; ++i) {
       printf ("%6.6s  ", (char *)&text[offset]);
       offset = offset + 6;
       if ( (i%10 ==  0) && (n_fixes > i) )
           printf("\n               ");
   }
```

```
   /*** AIRWAYS ***/   printf("\n%-15s", "   airways: ");   for (i = 1; i <=
n_airways; ++i) {       printf("%6.6s  ", (char *) &text[offset]);
offset = offset + 6;       if ( (i%10 ==  0) && (n_airways > i) )
           printf("\n               ");
   }


   /*** CENTERS ***/
   printf ("\n%-15s", "   centers: ");
   for (j = 0; j < n_centers; ++j) {
       printf ("%1.1s ", (char *)&text[offset]);
       offset = offset + 1;
   }


   /*** ROUTE ***/
   printf ("\n%-15s", "   route: ");
   printf ("%0.*s\n", route_size, (char *)&text[offset]);

}  /* parser_rt */
```

Appendix A: Sample Messages

The messages are in the following format. The first message has been exploded for clarity.

0000	Four digit hexadecimal sequence number
23194739	Eight digit timestamp
KZJX	Facility code
TZ	two character message type followed by the body of the message which is terminated by line feed.

Messages too long to display on one line are indented on the second and subsequent lines. Blank lines denote gaps in the sequence numbers.

```
000023194739KZJXTZ N811PJ/889 190 071 3000N/08111W
000123201925KZJXTZ AMT101/406 346 410 3142N/08339W
000223201925KZJXTZ DAL189/066 369 370 2911N/08056W
000323201925KZJXTZ DAL974/253 567 410 2819N/08624W
000423201925KZJXTZ N33TD/318 163 070 3015N/08230W
000523201925KZJXTZ FCI530/684 209 070 2944N/08337W
000623201925KZJXTZ DAL910/553 557 370 3121N/08622W
000723201925KZJXTZ N509QC/775 390 330 3050N/08141W
000823201925KZJXDZ N30549/704 C210/A AMG D2019 ISM 2143
000923201926KZJXTZ DAL2453/226 311 350 3308N/07948W

002123201933KZIDAF TRS175 CAK ATL 06 3940N/08124W 07 E2020 10
      CAK./.CTW..JPU..ODF.MACEY2.ATL/2133
002223201936KZIDAZ N655JG LOU 4I3 2021
002323201405KZHNTZ NWA925/924 540 360 2345N/15600W
002423201410KZHNTZ AAH275/087 350 121C 2112N/15813W
002523201410KZHNTZ AMT671/946 230 026T 2052N/15622W
002623201410KZHNTZ N310FX/665 390 380 2403N/15618W
002723194734KZSETZ JAL85/FFF 145 028 4720N/11922W
002823194737KZSETZ ASA346/925 459 330 4258N/12305W
002923194737KZSETZ N163Z/533 173 140 4444N/12143W
002A23194737KZSETZ DAL1625/927 367 130T 4526N/12114W
002B23194737KZSETZ SWA1212/847 382 240 4134N/12216W
002C23194737KZSETZ ASA437/940 108 030C 4541N/12249W
002D23194737KZSETZ ASA458/396 525 316C 4209N/12026W
002E23194738KZSETZ ROA625/475 503 330 4310N/12040W
002F23194738KZSETZ ASA309/819 362 100T 4504N/12240W
003023194738KZSETZ DAL271/027 208 130T 4541N/12235W
003123194738KZSETZ ASA551/446 437 220 4425N/12215W
003223194738KZSETZ DAL231/653 387 350 4548N/11433W
003323194740KZSETZ EJA838/233 000 000 4535N/12237W
003423194740KZSETZ QXE2955/821 150 018 4538N/12244W
003523194740KZSETZ N2163N/FFF 160 040 4532N/12302W
003623194741KZSETZ DAL271/027 210 052 4541N/12235W
003723194741KZSETZ DAL1063/118 290 099 4533N/12212W

004A23201941KZMEDZ FLG650/814 T/SF34/I MEM D2020 GSP 2225
004B23201411TZSUTZ USA1844/000 420 310 2059N/06641W
004C23201416TZSUTZ AAL1574/000 530 330 2034N/06845W
004D23201416TZSUTZ EGF417/000 240 100 1826N/06446W
004E23201416TZSUTZ LIA574/000 300 136C 1809N/06353W
```

```
004F23201416TZSUTZ N528AC/000 310 310C 1905N/06636W
005023201408KZSEAF N9640Y PAE PAE 06 4808N/12224W 07 E2014 10 PAE./.PAE320014..PAE
005123194757KDENTZ CGMJY/276 155 120 4024N/10444W
005223194757KDENTZ AWI083/486 222 121 3946N/10501W
005323201936KZABAZ ASH6451 FHU PHX 2013
005423201942KZABTZ AWE557/248 520 301C 3208N/11003W
005523201942KZABTZ COA1403/583 388 140T 3209N/11022W
005623201942KZABTZ AAL1890/176 508 330 3310N/11213W
005723201942KZABTZ N575SF/647 528 410 3454N/10158W
005823201942KZABTZ DAL1062/614 533 370 3327N/11106W
005923201942KZABTZ COA1946/846 464 330 3508N/11155W
005A23210229    HB
005B23201409KZKCUZ AAL1580 T/B722/G 0461 3714N/09011W E2019 330
      MSY./.SQS.J35.STL.STL349.MAGOO..BDF.BDF3.ORD/2115
005C23201409KZKCDZ N506E/771 T/C560/G IXD D2014 SUS 2054
005D23201412KZKCUZ NWA407 T/DC9/A 0438 3958N/09624W E2022 350
      MSP./.OVR.J21.ICT.J21.IRW.UKW4.DFW/2127
005E23201416KZKCTZ AAL494/202 469 330 3806N/10011W
005F23201416KZKCTZ N38LB/516 456 370 3558N/10124W
006023201416KZKCTZ TWA210/515 311 260T 3741N/08947W
006123201416KZKCTZ N551AC/736 443 430 3949N/09136W
006223201416KZKCTZ TWA208/757 342 150T 3904N/08919W
006323201416KZKCTZ AAL2022/247 423 330 3744N/09116W
006423201416KZKCTZ N91KA/635 240 100 3812N/09234W
006523201416KZKCTZ COA1874/432 431 330 4019N/09409W

006A23201416KZKCTZ AAL1006/272 438 330 4011N/09338W
006B23194753KZKCTZ NWA430/141 495 370 3916N/09007W
006C23194753KZKCTZ TWA432/031 493 270 3851N/09545W
006D23194753KZKCTZ TWA1315/600 412 310 3840N/09602W
006E23194753KZKCTZ TWA842/849 479 370 3948N/09443W
006F23194753KZKCTZ TWA216/182 205 110T 3856N/09045W
007023194753KZKCTZ LOF332/344 167 100 3841N/09024W
007123194753KZKCTZ VGD406/281 000 330 3921N/09441W
007223194754KZKCAF AWE350 PHX DTW 06 3859N/08923W 07 E1948 10
      PHX./.COWES235033..VHP..FWA.MIZAR3.DTW/2042
007323194756KZKCTZ N414AN/172 181 090 3737N/09349W
007423194802KZKCAF EXR712 SGF CPS 08 040
007523194812KZLCAF COA1041 IAH SEA 08 280
007623201412KZABTZ UAL2352/260 451 120T 3432N/11250W
007723201412KZABTZ N8416K/844 140 110B130 3511N/10344W
007823201412KZABTZ AAL1382/461 514 370 3257N/11224W
007923201413KZABTZ SWA294/468 435 350 3513N/11118W
007A23201413KZABTZ N91902/215 175 120 3442N/10336W
007B23201413KZABTZ AAL1679/881 412 310 3443N/10723W
007C23201413KZABTZ SWA751/480 000 310 3057N/10117W
007D23201413KZABTZ AWE223/237 226 060C 3324N/11210W

009523201945KZNYTZ USA256/546 180 020 3951N/07519W
009623201946KZNYTZ N137TA/890 260 056 3936N/07513W
009723201946KZNYTZ N68Z/FFF 000 001 3953N/07514W
009823201946KZNYUZ N92AE T/G4/E 0300 4125N/07435W E2020 170
      ROC*./.WEARD.V489.COATE..MMU
009923201946KZNYTZ ASH5796/19A 290 090 3924N/07600W
```

```
009A23201946KZNYFZ NKS409/018 T/DC9/A 0443 LGA P2215 310
      LGA..WHITE.J209.SBY.J79.KATZN.J193.WEAVR.J121.CHS.J79.OMN.BITHO7.MLB/0229
009B23201946KZNYTZ USA23/000 230 023 3950N/07506W
009C23201946KZNYTZ N9752C/FFF 100 018 4005N/07504W
009D23201946KZNYTZ GAA277/142 240 040 3933N/07453W
009E23201408CCZXTZ AZA618 436 350 5211N/05640W
009F23201409CCZXTZ CDN45 454 350 4805N/06041W
00A023201419CCZXTZ AZA610 436 351 4609N/05539W
00A123201421CCZXTZ COA19 469 350 5246N/05800W
00A223194803KZJXUZ BAW2037 B/B744/W M086 3420N/07837W E1949 390
      STEAM./.ORF.J121.CHS.J79.OMN.BITHO7.MCO
00A323194803KZJXAF N11541 HEF PMP 08 080
00A423194803KZJXTZ N647JM/831 289 147 3315N/08106W
00A523194805KZJXAF N305CJ SDL OMN 06 3010N/08327W 07 E1948 10
      SDL./.CTY331040..CARRA..OVI4..OVH5..OMN..OMN
00A623194805KZJXUZ N8047R BE36/G 0188 2843N/08233W E1948 100
      X16..BRNUM..3237/08526..AUO/2117
00A723194807KZJXUZ USA1277 T/B734/F 0431 3420N/07837W E1952 310
      PHL./.ILM295029.J121.CHS.J79.OMN.BITHO7.MCO/2108
00A823194808KZJXAF N533KC ABY 42J 08 050
00A923194808KZJXTZ N10XA/FFF 106 029 2903N/08053W
00AA23194808KZJXTZ N301EL/289 273 055 2938N/08113W
00AB23201414KC93TZ UAL1537/925 157 036 4206N/08803W
00AC23201414KC93TZ AAL2244/469 289 102 4142N/08827W
00AD23201414KC93TZ AMT820/733 266 052 4129N/08735W
00AE23201414KC93TZ AWI554/516 130 019 4203N/08759W
00AF23201420KC93TZ AIC123/181 184 044 4159N/08810W
00B023194714KZNYAF PDT3395 YYZ BWI 08 150
00B123194718KZNYAZ USA462 JAX PHL E1937
00B223194723KZNYTZ N4633S/396 121 040 4028N/07517W
00B323194724KZNYTZ MES2801/462 245 053 4205N/07709W
00B423201357CCZMTZ AAQ244 269 220 4815N/06707W

00B923201424CCZMTZ VKG097 474 370 4916N/06518W
00BA23194811LLONTZ BRY175/000 248 180 5553N/00328W
00BB23194811LLONTZ EXS17/000 165 070 5434N/00552W
00BC23194816LLONTZ BMA1536/000 406 261 5552N/00248E
00BD23194816LLONTZ AYR20/000 201 052 5504N/00518W
00BE23194816LLONTZ BRY58A/000 242 178 5417N/00216W
00BF23201935KZTLUZ USA585 T/F100/E 0428 3641N/08324W E2022 290
      IND./.DR..PSK263125..HMV.SHINE5.CLT
00C023201938KZTLAF N14VF PRX RDU 08 330
00C123201940KZTLTZ N97FB/412 390 350 3451N/08527W
00C223201940KZTLTZ LN83LJ/263 387 430 3550N/08407W
00C323201940KZTLTZ N202DS/507 182 040 3334N/08311W
00C423201940KZTLTZ N979C/406 361 198C 3336N/08631W
00C523201940KZTLTZ N50KF/091 158 060 3344N/08215W
00C623201940KZTLTZ DAL317/888 544 240T 3424N/08619W
00C723201940KZTLTZ N448T/505 209 250C 3342N/08659W

00D523201941KZTLTZ USA585/230 480 290 3653N/08344W
00D623201941KZTLTZ DAL838/027 539 330 3452N/08738W
00D723201943KZTLAZ DAL619 LGA ATL 2011
```

```
00D823201943KZTLUZ DAL61 H/B763/W M082 3649N/08253W E2027 390
      EDDH./.DR..ODF.MACEY2.ATL
00D923201944KZTLAZ AAL1772 DFW ATL 2012
00DA23201945KZTLTZ N4117P/627 138 046 3235N/08335W
00DB23201420KZHUUZ AAL1399 T/F100/E 0434 3131N/09345W E2018 310
      ORD./.LIT.J180.TBEND..DAS.DAS3.IAH/2047
00DC23201428KZHUUZ N2702K C340/I 0190 3116N/08725W E2014 150 AL15..BHM/2054
00DD23201420KZAUTZ N22SF/735 115 015 4029N/08900W
00DE23194812KZJXAF DAL1129 CVG MIA 08 390
00DF23194814KZJXTZ N6166J/FFF 107 020 3212N/08118W
00E023194814KZJXDZ EXR210/268 C210/U ORL D1948 SGJ 2028
00E123194816KZJXFZ COM297/673 T/CARJ/F 0464 TLH P2045 270
      TLH..SZW.J41.PIE.CYY3.MIA/0100
00E223194817KZJXTZ N9226V/884 121 079 3152N/08051W
00E323201938TZSUTZ AAL783/000 470 234C 1953N/06614W
00E423201938TZSUTZ CMM480/000 460 350 1730N/06545W
00E523201938TZSUTZ EGF545/000 230 200 1733N/06435W
00E623201938TZSUTZ NWA1717/000 460 241C 1939N/06714W
00E723201943TZSUTZ AMT8468/000 440 310 1637N/06322W
00E823201943TZSUTZ EGF452/000 230 068 1827N/06502W
00E923201943TZSUTZ EGF466/000 230 160 1830N/06705W
00EA23194815KDENTZ UAL2761/836 314 114 3947N/10454W
00EB23194821KDENTZ N600GP/739 201 067 3936N/10451W
00EC23194821KDENTZ FFT574/653 247 113 3939N/10445W
00ED23201944KC93TZ UAL1131/209 195 030 4206N/08801W
00EE23201944KC93TZ UAL8187/866 233 048 4201N/08748W
00EF23201944KC93TZ UAL1127/350 251 039 4212N/08800W
00F023201950KC93TZ UAL421/991 185 079 4221N/08818W
00F123201950KC93TZ EWW336/405 167 016 4159N/08759W

010723194817KZKCTZ N89KM/064 000 410 3651N/08935W
010823194818KZKCTZ N443PE/544 000 180 3704N/08937W
010923194818KZKCTZ N74YV/720 291 230 3644N/10240W
010A23194818KZKCUZ TWA279 T/B722/A 0461 3722N/08940W E1954 280
      MIA./.CGI.QBALL4.STL/2021
010B23194627ETMSRT AGU503 000        0,) I6G20 JFG20G20APC A28 1 W  01M? ID
      JPTJSJTFFJY8 G;00< G;00; G<00" G<00V G=00L G>00A G'0 ? H30 1 G, {. GR {4 GI
      ]:ZSUSJ ZSUST ZSU02 SJU   ISLLA FAJAR TOURO MALIE STT   GUYRO SLUGO PJM
      Y*TJSJ.RTE2.STT.A638.PJM..TFFJ/0100
010C23194537ETMSRT DAL126 156        0,) L-G20 SZG20G20FJC C771618  01!3 L- SZJFK
      LEMDN4 #J08O #F07* #[04( +:01: |A ,P |V (N &R .| 'P S1 &R IL "3 BZ *X 79 +B
      4R #A 2RZNYJF ZBW32 ZBW31 ZBW18 ZBW17 QMWH  QXTO  QXBA  BETTE RIFLE ACK
      WHALE BANCS HIDRA STG   ZMR   J62   J79   NB329
      <KJFK.BETTE2.BETTE..ACK..WHALE.N37A.BANCS..4600N/05000W..4800N/04000W..4900N
      /03000W..4800N/02000W..HIDRA..STG.UA33.ZMR.ZMR1C.LEMD/0612

011723194831KZLARZ N74V AJO OXR
011823201033KZDVDZ DAL214/139 L/B752/E EGE D2010 ATL 2237
011923201005CCZETZ DAL55 409 310 4843N/11342W
011A23201006CCZETZ CGMYS 105 090 5105N/11400W
011B23201008CCZETZ FKL100 279 250 5341N/11504W

012423201030CCZETZ ANA001 499 350 5920N/11551W
012523201032CCZETZ UAL801 492 310 5656N/10709W
```

```
012623201034CCZETZ GLR215 255 229 5611N/11729W
012723201035CCZETZ UAL883 514 310 5654N/11151W
012823201036CCZETZ KLM605 474 390 5200N/10959W
012923201042KZMPAZ NWA1071 MDW MSP E1949
012A23201046KZMPUZ AAL302 T/MD80/G 0438 4021N/09430W E2020 370
      DFW./.DWINE266011..MCW.KASPR2.MSP/2106
012B23201048KZMPAZ N9106V MSN ATW E2010
012C23210239    HB

015A23201838ETMSRT SKW430 052  VTU   0,) ING20 JYG20 JEATT A4511 Z  0A(- IZ J*MRY
      LAX O6 YO0=? Y00=| XM0<{ WO0;> WR0;C W00;C V@0:@ V@0:: V?0:+ V?0:T
      V;0:ZZOA10 ZLA15 ZLA14 ZLA13 ZLALA RZS   VTU   SADDE BAYST SMO   KEGGS J501
      V107  OLMRY./.SNS167029..RZS.SADDE6.LAX/2056
015B23201841ETMSRT SKW235 426        0,) I*G20 JAG20G20ATT 12  0 I  0A(( I+ J9LAX
      SAN L3 V;0:Z U!0-OZLALA ZLACS ZLANK LLAX.LAXL16.SAN/0023

015D23201949KZNYAZ BTA3629 ROC EWR A2020
015E23201950KZNYAZ TWA901 LPPT KJFK A2020
015F23201950KZNYAZ USA1106 MSY LGA A2020
016023201950KZNYAF SWA581 PVD BWI 08 180
016123201950KZNYTZ N2586Q/FFF 131 031 4204N/07707W
016223201951KZNYAF N44EL/024 TEB MCO 06 TEB 10
      TEB..WHITE.J209.SBY.J79.KATZN.J193..J121.CHS.J79.OMN.BITHO7.MCO/0206
016323201953KZNYTZ N928GF/480 263 039 4030N/07514W
016423201957KZNYAF BLR609 UNV IAD 06 4052N/07748W 07 E2020 10
      UNV./.FQM248054..HAR.V162.ROBRT..AML..IAD

19F23194826KZMAFZ COA1090/572 T/MD80/A 0446 PBI P2030 330
      PBI..BLUFI.A699.STIFF.BR65V.ADOOR.AR7.PANAL.AR3.CLB.AR1.ILM.J40.TYI..HPW.J19
      1.PXT.RBV1.EWR/0210
01A023194826KZMATZ N999DF/122 199 030 2634N/08000W
01A123201010ETMSRT FDX1954K02  MRLIN 0,, 7CG20 9{G20 9)SJF KBI630A    62 7C 9{EWR
      FLL NC #M08( .|08; .W091 .109" /60A4 ZG0A; Y(0BI YB0BN X"0B+ XD0C2 V=0DP
      U:0ER TJ0FX S+0F/ RL0FU PA0D? O:0E4 O+0EB OY0ED OK0ES OE0EYZNYJF ZNY68 ZDC58
      ZDC54 ZDC34 ZDC35 ZJX47 ZJX68 ZJX58 ZMA02 ZMA20 ZMAMI WHITE CYN   VILLS SBY
      LEESA KATZN CVI   WEAVR ISO   JMACK CHS   MILIE BEENO OMN   TRIPL DUBBL
      MRLIN TRITN WISCI J209  J37   J79   J193  J121  J174  J103
      NWJREWR..WHITE.J209.SBY.J79.KATZN.J193.WEAVR.J121.CHS.J79.OMN.MRLIN4.FLL/103
      8
01A223200951ETMSRT AAL366 659  LIZZI 0,) IVG20 K9G20 J,AJC N9N32 U  075T I. KFORD
      LGA G3 +#0M3 +P0KW +Q0JR +Q0I| +P0H; +A0FX +40EW #]0DJ #?0C@ #=0C. #,0B=
      #%0BL #"0BB #Z0A! #T0AI #R0A9 #P0A0 #L09% #F09/ #A09K #7098 #E08! #J08.
      #S08TZAUOR ZAU81 ZAU82 ZAU85 ZOB47 ZOB49 ZOB59 ZNY75 ZNY39 ZNYJF GIJ   PLAIN
      GERBS MIZAR IDEAS CETUS CXR   TEESY MIGET WEVEL ETG   SWISS WATSO MIP
      MARRC BILEY VIBES FJC   LIZZI BEUTY HARLM DREMS APPLE PROUD J146  J554  V433
      V6    GCNORD..GIJ.J146.ETG.MIP2.LGA/0141
01A323200954ETMSRT FLG610 106        0,) IVG20 K5G20G20ATT 66431 /  0 <B I. KBMEM
      LFT M3 W?0O4 VN0OM US0OF TH0OK S*0P8 SU0PL SD0Q0ZMEME ZME12 ZME66 ZHU30
      ZHU34 ZHUBT ZHULF SQS   JAN   MCB   CAZZZ BTR   V9    V11   V194  V70
      MHMEM..SQS..JAN..MCB.V194.BTR..LFT/0145
01A423200915ETMSRT GFT9149903  MRLIN 0,) IMG20 I>G20 I*ATC 624 0 U  06'N IP
      I[MYGFFLL R9 O"0D9 O*0DE O!0DS O+0EB OY0ED OK0ES OE0EYZMA21 ZMAPB ZMAMI ZFP
      MUNRO MRLIN TRITN WISCI RFPO.BR68V.MRLIN.MRLIN4.FLL/0030
```

```
01B823201044KZOAAF ASA339 SJC SEA 06 3828N/12203W 07 E2012 10
      SJC.LOUPE9.RBL..BTG.OLM2.SEA/2138

01CD23194834KZJXAF N928CD PBI OMA 06 2759N/08201W 07 E1951 10
      PBI./.LAL..3033/08422..SZW..4118/09553..OMA
01CE23194836KZJXUZ N16AS C650/G 0470 2806N/08205W E1948 390
      OPF./.LAL..CTY..SZW..IIU.DECEE3.IND/2139
01CF23194836KZKCTZ N717U/041 179 000 3627N/09305W

01F123210249    HB
01F223201952KZHNTZ AAH61/432 420 240 2112N/15745W
01F323201952KZHNTZ UAL45/396 520 350 2205N/15422W

7023201047KZAUDZ MES2980/262 T/SF34/A SBN D2010 DTW 2053
027123201047KZAURZ N398AC/251 MSN AGC
027223201047KZAUFZ N398AC/251 LJ55/G 0440 MSN P2040 410
      MSN..BAE..4121N/08210W..DJB..AGC/0115
027323201049KZAUUZ USA123 T/B733/F 0432 4039N/09235W E2019 310
      CLT./.DNV..LBF.SAYGE1.DEN/2203
027423201957KZLAFZ N493JX/181 BE9L/A 0240 BFL P2130 100
      BFL..EHF.V23.LIN.V108.SGD..APC/0120
027523201957KZLADZ SWA761/320 T/B733/I SAN D2020 ELP 2137
027623201958KZLAAF SWA260 LAX AUS 06 3343N/11712W 07 E2020 10
      LAX./.PERIS..ELP.J183.LLO.SLIMM1.AUS/2226
027723201958KZLADZ UAL2678/185 T/B733/F LAX D2020 TUS 2122
027823201959KZLAFZ SKW7926/329 T/E120/A 0284 LAX P2105 120
      LAX.GMN2.GMN.ARVIN1.BFL/0024
027923201427KZOATZ SWA259/351 401 310 3722N/11724W
027A23201427KZOATZ AAR214/727 520 370 3941N/12439W

028823201959KDENTZ UAL1429/036 301 161 4004N/10402W
028923201431KZDCAF DAL778 ATL EWR 08 290
028A23210259    HB
028B23201430KZKCAF SWA1006 PHX MCI 06 3638N/10227W 07 E2015 10
      PHX./.3638N/10227W..OTTER.JHAWK2.MCI
028C23201435KZKCAZ N406LM MTN ICT 2008
028D23201436KZKCTZ N6601V/648 090 074 3729N/09332W
028E23202001KZHUUZ SWA31 T/B733/I 0425 3110N/09639W E2024 370
      DAL.JPOOL8.CLL.STRUK8.HOU/2047
028F23202008KZHUTZ SWA192/394 144 140T 3021N/09746W
```

Appendix B: NAS Field Formats

The following are the formats for each NAS field. This information was extracted from Appendix E of NAS-MD-311. This information was included to add further details to the tables of NAS messages. If any questions should arise, please refer to the referenced NAS document for complete information.

The two digit number on the left of each page is the NAS field number as was specified within each NAS message in this document.

01 MESSAGE TYPE

Ex.
1. TZ
2. FZ

Message type designator LL

02 FLIGHT IDENTIFICATION

Ex.
1. AAL123
2. UAL51/051
3. AAL123/000
4. For ETMS
t he following are the allowable combinations for Field 02:

FZ	DZ	RZ	AZ	AF	UZ	TZ
abc	abc	abc	a	abc	a	abc
a		a		a		

a. Aircraft Identification -La(a)(a) Aircraft identification filed in a flight plan
b. The element separator / (slash)
c. Computer Identification- dda Computer Identification

Element d is the computer-assigned identification. A ddL format will normally be used only when all numbers of the ddd format are in use. The letters I and 0 will not be used.

03 AIRCRAFT DATA

Ex.
1. 3/B52/u
2. B707/T
3. T6
4. H/B747/A

a. Number of aircraft and/or Heavy Jet Indicator (d)(d) or (d)(a) or (a)
b. separator / (slash)
c. Type of Aircraft aa(a)(a)
d. Separator / (slash)
e. Airborne Equipment Qualifier (a)

Notes:
1. Elements (a,b) and (d,e) are optional in combination
2. The heavy jet indicator, if present, must immediately precede the element separator (b) as in example 4.
3. When the heavy jet indicator is present, the maximum number of aircraft in a flight is 9 (military only).
4. The first character of the aircraft type must be a letter.

05 SPEED

Ex.
1. 75

2. 565
3. M085 (MACH Speed)

a. True Air dd(d)(d) File true airspeed in Knots
   Lddd Filed MACH Speed
   SC Classified speed
b. Ground Speed ddd Aircraft's Current Tracking Speed

Notes:
1. The four digits of filed true airspeed in knots cannot exceed 3700.
2. MACH Speed cannot have a value higher than M500.

06 COORDINATION FIX

Ex.
1. DBN
2. 3500N/0400W
3. ABY349016

a. Coordination fix designator aa(a)(a)(a)(/)(a)(a)(a)(a)(a)(a) A fix mutually agreed upon by the facilities concerned for the coordination of center control transfer.

Notes:
1. When the flight plan represents an airfile, originating within this center area, this field contains the airfile point.
2. The fix is expressed in one of the conventions specified for fixes in field 10.

07 COORDINATION TIME

Ex.
1. P1021
2. E1523

a. Type of time One of the following:
   E Estimated time
   P Proposed Departure
   D Actual departure
b. Time dddd Coordinated Universal Time associated with the coordination fix

Notes:
1. P or D times are associated only with the departure fix
2. Both fields must be specified, i.e Lddd is the actual format

08 ASSIGNED ALTITUDE

Ex.
1. 330
2. 450B510

a. Altitude (d)dd
b. Block of altitudes (d)ddB(d)dd
c. Interim Altitude (d)ddT
d. Reported Mode C altitudes when it is not within Altitude Conformance Limit (ALCT) feet of the assigned altitude (d)ddC

Notes:
1. Altitude is expressed in hundreds of feet
2. The lower altitude is expressed first in block altitudes (i.e. before the B)

09 REQUESTED ALTITUDE

Ex.
1. 330
2. 450B510

a. Altitude (d)dd

b. Block of altitudes (d)ddB(d)dd

Notes:
1. Altitude is expressed in hundreds of feet
2. The lower altitude is expressed first in block altitudes (i.e. before the B)

10 ROUTE DATA (See NAS-MD-312 or examples in Appendix A)

11 REMARKS/GENERAL - not currently available

12 FIELD REFERENCE Ex: 11
a. number (d)d

Notes:
1. Used to specify which field is being modified
2. Must have a value between 01 and 11.

17 AMENDMENT OR CORRECTION DATA
Notes:
1. Used only in AF messages
2. Must follow a field 12
3. The format of this field is the format as specified in field 12
4. i.e. if field 12 contains an 07, then this field contains field 07 data and must then conform to all the rules for field 07.

23 TRACK POSITION VELOCITY COMPONENTS
a. Latitude/longitude in degrees/minutes ddddL/dddddL

26 DEPARTURE POINT Ex: KJFK
a. aa(a)(a)(a)(a)(a)(a)(a)(a)(a)(a)

27 DESTINATION Ex: EGLL
a. aa(a)(a)(a)(a)(a)(a)(a)(a)(a)(a)

28 ETA or ARRIVAL TIME Ex:
1. 0130
2. A0500
3. E2100

a. ETA dddd
b. Arrival Time (L)dddd (A - Arrival, E – Estimated)

Appendix C: ETMS Field Formats

The following are the formats for ETMS specified fields. These fields are used in constructing the RT and TO messages that are described in this document. The first six fields in this table are all runway times, i.e., wheels-up or wheels-down times; these are not gate times.

EDT	Estimated Departure Time as calculated by ETMS
ETA	Estimated Arrival Time as calculated by ETMS
CDT	Controlled Departure Time – NAS Field 92. Assigned after a Ground Delay Program has been put in effect
CTA	Controlled Arrival Time –ETMS calculated value, Assigned after a Ground Delay Program has been put in effect
OGTD	Original Departure Time – Initial departure time received by ETMS
OGTA	Original Arrival Time – Initial arrival time estimate received by ETMS
AC_PHYSICAL_CLASS	Aircraft Physical Class. One of following values. If unknown, field contains blank. P – Piston T – Turbo J – Jet
AC_USER_CLASS	Aircraft User Class. One of the following values. If unknown, field contains blank. T – Air Taxi F – Cargo C – Commercial G – General Aviation M – Military
FLIGHT STATUS	Flight Status. One of the following values. If unknown, field contains blank. N – None S – Scheduled L – Controlled F – Filed A – Active R – Ascending C – Cruising D – Descending T – Completed X – Cancelled M – Decontrolled

	E - Error
FLIGHT INDEX	Similar to CID of NAS Field 2. Value assigned by ETMS for flight identification purposes.
CALCULATED SPEED	Speed computed from the two reported points and the specified times at those points,. this is an inaccurate computation and should be used after applying a validity test. This is always a three digit number. The units are nautical miles per hour for this calculated ground speed
POSITION REPORT	Field 1: Time is the two digit day of the month, a slash, the two digits of hour and two digits of minute. All information is presented in UTC time. Field 2: Current Altitude, always three digits. Field 3: Lat/lon
ARRIVAL /DEPARTURE AIRPORT	LLLL or '-' if not known
DEPARTURE CENTER	This is the ARTCC from which the flight departs. The one character codes given in Section 3.4 are used.
MESSAGE TYPE	This field indicates the type of message that caused this RT message to be generated. The code is: 1 – AF 4 – DZ 5 – FZ 7 – UZ 9 – TZ 10 – FA 13 – FS 15 – EDCT 16 – TO (Oceanic update) 19 – CONTROL CANCEL 30 – GROUND STOP