Traffic Flow Management System CDM Message Protocol Specification for the Traffic Flow Management-Modernization (TFM-M) Program

Final, Release 9, Version 2.5

Contract Number: DTFAWA-04-C-00045
CDRL: E05

November 19, 2012

Prepared for:
U.S. Federal Aviation Administration

Prepared by:
CSC
North American Public Sector – Civil Group
15245 Shady Grove Road
Rockville, MD 20850

Traffic Flow Management System CDM Message Protocol Specification for the Traffic Flow Management-Modernization (TFM-M) Program

Final, Release 9, Version 2.5

Contract Number: DTFAWA-04-C-00045
CDRL: E05

November 19, 2012

Prepared for:
U.S. Federal Aviation Administration

Prepared by:
CSC
North American Public Sector – Civil Group
15245 Shady Grove Road
Rockville, MD 20850

CSC/TFMM-10/1077
Final, Release 9, Version 2.5
November 19, 2012

CDM Message Protocol Specification
APPROVAL SIGNATURE PAGE

APPROVAL SIGNATURES

PARTICIPANT	NAME	DATE

CDM Message Protocol Specification

Date: November 19, 2012
Feature Described: CDM Message Protocol Specification
Document Version: V2.5
Remarks: Effective TFMS R9

Revision History		
Version	**Date**	**Description of Change**
1.0	5/24/2010	• The CDM Protocols defined in this document are commonly used by AOCNet, ARINC and FSM interfaces • To effectively maintain these protocols in the future releases, the entire text of CDM Message Protocol Specifications have been removed from the main body of the ICDs and consolidated in this document.
2.3	9/01/2010	• The revision number of this document was revised to 2.3 since it replaces the v2.2 file available on the FAA website • The following messages have been deleted since they are not implemented in TFMS: - M_FPPP_FLIGHT DATA PACKET - M_FPPP_FLIGHT_DATA_RESPONSE PACKET - M_DEL_ADL_AAR_GDP - M_DEL_ADL_COMP_PARAM - M_DEL_ADL_BLANK_PARAM - M_DEL_ADL_GS_PARAM - M_DEL_ADL_AFP_PARAM
2.4	01/10/2011	• Removed the restriction (“from 1 to 9999”) on the “Short Data” field to clarify that it is a 4 byte integer field within the following messages: - M_HB_REQ - M_HB_ACK - M_FLIGHT_DATA_PACKET - M_GDP_REQ - M_GDP_REPLY - M_EI_FLIGHT_DATA_PACKET - M_SS_DATA_PACKET - M_EDCT_REPLY - M_REQ_COMMAND - M_REQ_REPLY - M_ADL_DATA_ACK

Revision History		
Version	**Date**	**Description of Change**
2.5	11/19/2012	• Added a new section 1.2 to introduce a referenced document • Added a new section 1.3.4 to define CTOP session protocols • Revised all applicable sections to account for CTOP messages • Re-organized the overall document sections to improve the flow of material in the specification without altering the contents of the sections.

Table of Contents

1.1 Introduction 1
1.2 References 1
1.3 General CDM Session Protocols 1
1.3.1 Flight Data (FD) Message Protocols 1
1.3.2 Simplified Substitutions (SS) Protocols 3
1.3.3 ADL Session Protocols 5
1.3.4 Collaborative Trajectory Options Program (CTOP) Session Protocols 6
1.4 CDM Session Message Header 7
1.5 CDM Session Protocol Messages 9
1.5.1 M_ATMS_CONNECT [1] 11
1.5.2 M_ATMS_ACCEPT [2] 12
1.5.3 M_ATMS_REJECT [3] 12
1.5.4 M_DISCONNECT [4] 13
1.5.5 M_SHUTDOWN [5] 13
1.5.6 M_HB_REQ [10] 14
1.5.7 M_HB_ACK [11] 14
1.6 CDM Session Application Messages 16
1.6.1 M_FLIGHT_DATA_PACKET [101] 21
1.6.2 M_FLIGHT_DATA_REPLY [102] 22
1.6.3 M_SLOT_DATA [103] 22
1.6.4 M_GDP_REQ [104] 23
1.6.5 M_GDP_REPLY [105] 24
1.6.6 M_GDP_MESSAGE [106] 24
1.6.7 M_EI_FLIGHT_DATA_PACKET [111] 25
1.6.8 M_SS_DATA_PACKET [112] 26
1.6.9 M_REGISTER [201] 26
1.6.10 M_REGISTER_ACK [202] 27
1.6.11 M_UN_REGISTER [203] 28
1.6.12 M_UN_REGISTER_ACK [204] 29
1.6.13 M_START_ADL [205] 29
1.6.14 M_ADL_DATA [206] 30
1.6.15 M_END_ADL [207] 31
1.6.16 M_ELEMENT_DELETED [208] 31
1.6.17 M_EDCT_COMMAND [212] 32
1.6.18 M_EDCT_REPLY [213] 32
1.6.19 M_REQ_COMMAND [214] 33
1.6.20 M_REQ_REPLY [215] 33
1.6.21 M_ADL_DATA_ACK [218] 34
1.6.22 M_ADD_ADL_AAR [219] 35
1.6.23 M_ADD_ADL_ADR [220] 35
1.6.24 M_ADD_ADL_AAR_GDP [221] 36
1.6.25 M_ADD_ADL_GDP_PARAM [222] 36
1.6.26 M_ADD_ADL_COMP_PARAM [224] 37

1.6.27 M_ADD_ADL_BLANK_PARAM [225] 37
1.6.28 M_ADD_ADL_GS_PARAM [226] 38
1.6.29 M_DEL_ADL_AAR [227] 39
1.6.30 M_DEL_ADL_ADR [228] 39
1.6.31 M_DEL_ADL_GDP_PARAM [230] 39
1.6.32 M_UPDATE_ADL_REQ [235] 40
1.6.33 M_WEATHER_COMMAND [236] 41
1.6.34 M_WEATHER_REPLY [237] 41
1.6.35 M_AUTO_MONITOR_REQ [242] 42
1.6.36 M_AUTO_MONITOR_REPLY [243] 42
1.6.37 M_AUTO_MONITOR_MESSAGE [244] 43
1.6.38 M_ADD_ADL_AFP_PARAM [245] 43
1.6.39 CTOP Session Application Messages 44

List of Tables

Table 1-I. General Layout - CDM Session Messages 8
Table 1-II. CDM Session Protocol Message Summary 10
Table 1-III. M_ATMS_CONNECT Message 11
Table 1-IV. M_ATMS_ACCEPT Message 12
Table 1-V. M_ATMS_REJECT Message 12
Table 1-VI. M_DISCONNECT Message 13
Table 1-VII. M_SHUTDOWN Message 13
Table 1-VIII M_HB_REQ Message 14
Table 1-IX. M_HB_ACK Message 14
Table 1-X. CDM Session Application Message Summary 16
Table 1-XI. M_FLIGHT_DATA_PACKET Message 21
Table 1-XII. M_FLIGHT_DATA_REPLY Message 22
Table 1-XIII. M_SLOT_DATA Message 23
Table 1-XIV M_GDP_REQ Message 23
Table 1-XV. M_GDP_REPLY Message 24
Table 1-XVI. M_GDP_MESSAGE Message 25
Table 1-XVII. M_EI_FLIGHT_DATA_PACKET Message 25
Table 1-XVIII. M_SS_DATA_PACKET Message 26
Table 1-XIX. M_REGISTER Message 27
Table 1-XX. M_REGISTER_ACK Message 27
Table 1-XXI. M_UN_REGISTER Message 28
Table 1-XXII. M_UN_REGISTER_ACK Message Table 29
Table 1-XXIII. M_START_ADL Message 30
Table 1-XXIV. M_ADL_DATA Message 30
Table 1-XXV. M_END_ADL Message 31
Table 1-XXVI. M_ELEMENT_DELETED Message 31
Table 1-XXVII. M_EDCT_COMMAND Message 32
Table 1-XXVIII. M_EDCT_REPLY Message 32

Table 1-XXIX. M_REQ_COMMAND Message 33
Table 1-XXX. M_REQ_REPLY Message 34
Table 1-XXXI. M_ADL_DATA_ACK Message 34
Table 1-XXXII. M_ADD_ADL_AAR Message 35
Table 1-XXXIII. M_ADD_ADL_ADR Message 35
Table 1-XXXIV. M_ADD_ADL_AAR_GDP Message 36
Table 1-XXXV. M_ADD_ADL_GDP_PARAM Message 36
Table 1-XXXVI. M_ADD_ADL_COMP_PARAM Message 37
Table 1-XXXVII. M_ADD_ADL_BLANK_PARAM Message 37
Table 1-XXXVIII. M_ADD_ADL_GS_PARAM Message 38
Table 1-XXXIX. M_DEL_ADL_AAR Message 39
Table 1-XL. M_DEL_ADL_ADR Message 39
Table 1-XLI. M_DEL_ADL_GDP_PARAM Message 40
Table 1-XLII. M_UPDATE_ADL_REQ Message 40
Table 1-XLIII. M_WEATHER_COMMAND Message 41
Table 1-XLIV. M_WEATHER_REPLY Message 41
Table 1-XLV. M_AUTO_MONITOR REQ Message 42
Table 1-XLVI. M_AUTO_MONITOR_REPLY Message 42
Table 1-XLVII. M_AUTO_MONITOR_MESSAGE Message 43
Table 1-XLVIII. M_ADD_ADL_AFP_PARAM Message 44

1.1 Introduction

This specification details the application, connectivity and the protocols used by TFMS to exchange messages via the CDM participant (AOCNet, FSM, and ARINC) client interfaces. It covers the four types of CDM sessions:

- Flight Data (FD) Session
- Simple Substitution (SS) Session
- ADL Session
- CTOP Session

1.2 References

The following CTOP document is useful in understanding the contents of this document.

- Traffic Flow Management System (TFMS) Collaborative Trajectory Options Program (CTOP) Interface Control Document (ICD), Version 3.0, Final, CSC/TFMM-11/1246, November 19, 2012

1.3 General CDM Session Protocols

The inter-process communication between processes at the various sites (airline, FAA) is performed in "sessions" through dedicated TCP/IP sockets. In each session, the application running at the CDM Participant site is considered the **client** and the application running at TFMS is considered the **server**. The general approach is as follows:

1. A client process opens a socket connection to a server process using a well-known IP address and starts a session.
2. Data is exchanged between the client and server indefinitely.
3. Either the client or server terminates the session and closes the connection.

The session could range from a single message sent and reply returned or months of continuous data exchange. There are a number of sub-protocols that are also implemented, depending on the data to be transferred.

1.3.1 Flight Data (FD) Message Protocols

Flight Data Message protocols are used by AOCNet and ARINC interfaces to exchange flight data messages, responses and Unsolicited Messages.

For flight data messages, the session can be opened and closed simply by opening and closing the socket connection. In other words, the client just opens a socket and begins sending messages. Each message contains client information that is used by the server to validate the connection.

A client/server session may be established and maintained in two different ways depending on the needs of the client. The simplest session protocol is conducted through the following sequence of events:

- A client opens a TCP/IP socket connection to the server using a designated IP address.
- The server validates the IP address of the connecting client. If invalid, the server closes the connection. If valid, the session for that client has started.
- Flight data messages are sent from client to server. Each message includes client name and tag, as well as the body of the message text.
- The server validates the client name and tag when first flight data message is received. If there is any problem, the server closes the connection. Otherwise this and all subsequent messages are accepted.
- Optionally, replies are sent from server to the client IP or ARINC MQ address.
- When a client wants to terminate a session, it closes the socket connection.
- When a server wants to terminate a session, it closes the socket connection.

A client can, if desired, also use the following alternative method to establishing a Flight Data session:

- A client opens a TCP/IP socket connection to the server using a designated IP address.
- The server validates the IP address of the connecting client. If invalid, the server closes the connection.
- The client sends a connect message.
- The server validates the connect message, and if valid, sends an accept message to the client, thus starting the FD session. If not valid, the server sends a reject message.
- Once the session has been started, when the client wants to verify that the connection is still open, it sends a “keep-alive” message. The server responds with a “keep-alive” acknowledgement.
- When the server terminates, it sends a “shutdown” message before closing the socket.

Additional notes about flight data sessions:

- A NAS User client may open and close sessions whenever it wishes; that is, there is not a requirement to keep an open connection with the server at all times, although that is allowed.
- A client may have multiple open connections if so desired. However, each simultaneous connection must have a unique client tag. Multiple client tags must be assigned for this purpose.

- A client tag may only be associated with one active connection at a time.

Flight Data messages are transmitted to AOCNET Participants using the following protocol:

- Once the session is established, the client sends flight data messages and the server optionally sends replies. The client may choose how their messages are acknowledged, using these two choices:
    - Whether a reply is sent for every message, or only when errors occur in processing the message.
    - Whether the replies go back to the sending address or to an alternate ARINC MQ address. (These choices are controlled by the client through the use of keywords in the FD packet header, as in the current ARINC-only protocol.)
- The server generates replies in the following manner by default:
    - A reply is sent for every message.
    - The reply is sent to the client from whom the initial message was received.
- If a client does NOT wish to get an unconditional reply for a message, it indicates so by using the NOACK keyword in the FD packet header line. In this case replies are only sent when an error is encountered in processing the message contained in the FD packet.
- The use of the NOACK and ARINC address keywords are part of the FD packet header as described in earlier sections of this ICD.

1.3.2 Simplified Substitutions (SS) Protocols

Simplified Substitution protocols are used by AOCNET, ARINC and FSM interfaces. For Simplified Substitutions, the session operates similarly to Flight Data sessions, with one exception. With Simplified Substitution sessions, the first application message may be from the server to the client. That means that for this scenario, if a client is connecting to monitor the simplified sub messages sent by TFMS, the client must send a message after opening a socket to identify itself. For this scenario, the client must send the connect protocol message to establish a Simplified Substitution session. However, as described below, when the client intends to send the first application message, no connect protocol message is required.

A client/server session may be established and maintained in two different ways depending on the needs of the client. The simplest session protocol is conducted through the following sequence of events:

- A client opens a TCP/IP socket connection to the server using a designated IP address.

- The server validates the IP address of the connecting client. If invalid, the server closes the connection. If valid, the session for that client has started.
- The client sends a connect message which identifies the client by providing a client name and tag. The client name must be configured as a simplified sub (SS) client by TFMS.
- When a delay program is issued, the server sends the slot list out to any SS client currently connected and is also configured to receive such messages. Once a delay program is issued, SS packets may be sent from client to server. Each message includes the body of the message text in the buffer.
- Replies are sent from server to the client over the same socket that the message was received on.
- At any time, the SS client can request information about the current delay programs. Replies are sent back over the same socket.
- When the state of a delay program changes (substitutions turned off, substitutions turned on, GDP cancelled, etc.), a message is sent out to any currently connected SS clients.
- When an SS client wants to terminate a session, it closes the socket connection.
- When the server wants to terminate a session, it closes the socket connection.

A client can, if desired, also use the following alternative method to establish a Simplified Substitution session:

- A client opens a TCP/IP socket connection to the server using a designated IP address.
- The server validates the IP address of the connecting client. If invalid, the server closes the connection.
- The client sends a connect message.
- The server validates the connect message, and if valid, sends an accept message to the client, thus starting the SS session. If not valid, the server sends a reject message.
- Once the session has been established, when the client wants to verify that the connection is still open, it sends a “keep-alive” message. The server responds with a “keep-alive” acknowledgement.
- When the server terminates, it sends a “shutdown” message before closing the socket.

Additional notes about Simplified Substitution sessions:

- The simplified subs use the same port as the flight data message feed. Therefore, an SS client can send regular flight data messages as part of the same session as simplified subs, if desired.

- The CDM user may have multiple SS clients connected if so desired. These may share the same name, but each concurrent client must have a unique client tag.
- Simplified Substitution messages are transmitted to AOCNET Participants using the following protocol:
- Once the session is established, the client sends simplified subs messages and the server sends replies. The client has control over where the reply goes, but the reply is mandatory.
- The server generates replies in the following manner by default.
    - The reply is sent to the address from which the message was received.

1.3.3 ADL Session Protocols

ADL Session protocols are used by AOCNET and FSM interfaces. For ADL distribution, additional messages are used to manage sessions. The client sends an initial connect message identifying itself; the server uses this information to validate the connection. Additional messages are used to notify client/server of a shutdown (refer to Section 1.5 for details of the CDM Session Protocol messages).

An ADL client/server session is established and maintained through the following sequence of events:

- A client opens a socket connection to a server using a designated IP address.
- The client sends the appropriate connect message.
- The server validates the connect message, and if valid, sends an accept message to the client. If not valid, the server sends a reject message.
- The server generates a new ADL file.
- Various messages are sent between client and server to exchange ADL and GDP data.
- When the client wants to verify that the connection is still open, it sends a "keep-alive" message. The server responds with a "keep-alive" acknowledgement.
- When a client wants to terminate a session, it sends a disconnect message to the server and disconnects from the socket.
- When a server wants to terminate a session, it sends a shutdown message to the client(s) and closes the socket(s).
- The messages used to initiate, validate, and terminate sessions are referred to as "CDM Session Protocol messages".

ADL data files are transmitted from the TFMS to AOCNET Participants using the following protocol:

- A client first establishes a session using the CDM Session Protocol messages.
- When a client wants data for an element it sends a register message to the server.
- The server registers that client for that element and starts periodically shipping data files to the client. This involves file compression and encryption process as shown below:
  - The server compresses the file using GZIP.
  - The server encrypts the file using Blowfish.
  - The server sends the file to the client through the socket until the end of file is reached.
  - The client writes the data to a file.
  - When the data transmission is complete, the server notifies the client through the socket that the file is ready.
  - The client decrypts, decompresses, and reads the file.
- When a client no longer wants data for that element, it sends an un-register message.
- The server stops sending files to that client for that element.
- When a client shuts down it ends the session by sending a disconnect message to server prior to closing the socket connection.
- In the event that server shuts down, it closes its sessions by sending shutdown messages to all clients prior to closing the socket connections.

1.3.4 Collaborative Trajectory Options Program (CTOP) Session Protocols

Collaborative Trajectory Options Program session protocols are used by AOCNET interfaces, specifically by the Flight Operator System (FOS) clients. The CTOP data exchange is modeled after the existing CDM data exchange for flight data updates, Ground Delay Programs (GDPs), and Airspace Flow Programs (AFPs). The basic features of a CTOP session are as follows:

- The CTOP data exchange occurs through dedicated TCP/IP socket connections over private networks, either true private networks or virtual private networks (VPNs). This connection is solely for CTOP message exchange and is separate from any connections used for CDM, GDP, and AFP message exchange.
- The FOS acts as the client and the TFMS acts as the server. That is, the FOS initiates the socket connection.
- TFMS provides a single connection point with an IP address. It is the responsibility of the FOS to persistently attempt to connect, using the IP until a connection is made.
- The FOS determines whether to use a continuous socket connection or not (although a continuous connection is recommended for most users).

- TFMS sends dynamic update messages to an FOS only if that FOS has a connection open at the time the message is generated. If a socket is not open for that FOS, TFMS discards the message. There is no queuing and re-transmission of individual CTOP messages.
- The FOS is expected to use data requests to recover lost data as needed. The data requests allow the FOS to determine what CTOPs and FCAs are in place, what flights are impacted, and what their current trajectory assignments are.

The FOS is responsible for ensuring that its data is in synchronization with the TFMS. If a message is sent to TFMS, the FOS should consider that message as having been accepted by the TFMS if a positive reply is received. If not, it is the responsibility of the FOS to correct and re-transmit the message.

1.4 CDM Session Message Header

CDM Session messages consist of Protocol Messages and Application Messages. The Protocol Messages are used for establishing and maintaining sessions. Application Messages are used for sending/receiving application data. All CDM Session messages consist of a fixed length, 24-byte message header and an optional message body. The message header consists of six 4-byte integer fields.

Note – the six header fields are 4 byte integer fields which are **not** ASCII encoded, and are in network byte order. The header fields are shown below.

The FD, SS and ADL session headers contain items 1, 4, 5, 6, 7 & 9 below, whereas the header for the CTOP session messages are made up of items 1, 2, 3, 6, 8 & 10.

1. Message Type (FD, SS, ADL & CTOP)
2. Message Time (CTOP only)
3. Uncompressed Size (CTOP only)
4. Message Source (FD, SS & ADL only)
5. Message Destination (FD, SS & ADL only)
6. Client Tag (FD, SS, ADL & CTOP)
7. Short Data (FD, SS & ADL only)
8. Sequence Number (CTOP only)
9. Message Length (FD, SS & ADL only)
10. Data Buffer Length (CTOP only).

The message body consists of one optional field, Data Buffer, which is present when the Message Length or the Data Buffer length field is greater than zero. Table 1-I illustrates the general layout for each of the CDM Session Messages describing the fields that comprise the message header and message body.

Table 1-I. General Layout - CDM Session Messages

Field	Unit/Format	Description	Bytes
Message Type (All Session Types)	Binary	Integer value of the message type.	4
Message Time (CTOP only)	Binary	Time at which the message is generated in UNIX timestamp format (number of seconds since January 1, 1970). Not used for CDM messages	4
Uncompressed Size (CTOP only)	Binary	Size of the data buffer when uncompressed. Set to zero if there is no data buffer. If the data buffer is not compressed, the uncompressed size equals the Data Buffer Length.	4
Message Source Address (FD, SS & ADL only)	Binary	Encoded source identifier assigned by FAA. Not used for flight data messages.	4
Message Destination Address (FD, SS & ADL only)	Binary	Encoded destination identifier assigned by FAA. Not used for flight data messages.	4
Client Tag (All Session Types)	Binary	Used to identify one of a set of clients (e.g.,, if there is more than one client running at a site)	4
Short Data (FD, SS & ADL only)	Binary	32 bit integer containing message specific data. Included with every message.	4
Sequence Number (CTOP only)	Binary	A number assigned by FOS client to be used by TFMS to reply back to the sender	4
Message Length (FD, SS & ADL only)	Binary	Length of the data buffer (may be zero; maximum is 128 Kbytes).	4
Data Buffer Length (CTOP only)	Binary	A number assigned by FOS client to be used by TFMS to reply back to the sender	4
Data Buffer* (All Session Types)	*	An array of bytes that is 'Message Length' bytes long. It is only sent if the Length field in the header is non-zero. Format of the information in the buffer is message-specific.	0+ Variable
*Note –For all Protocol Messages, the Data Buffer contains Zero Bytes. If Message Length or the Data Buffer length is greater than zero, this field is included in the Application message.			

Deleted: bits of

Deleted: -

Deleted: as required

Deleted: Protocol

Deleted: See specific Protocol message below for applicable data entries for this field.

The Data Buffer field of a CDM Application message is used to convey specific CDM application data. For Flight Data and Simplified Substitution sessions, the Data Buffer field provides the message packet, which may consist of one or more applications messages. For CTOP messages, the data buffer provides the application messages in Extensible Markup Language (XML) format.

The message packet (contained in the Data Buffer field) includes a packet header followed by one or more application messages.

1.5 CDM Session Protocol Messages

As stated earlier, there are four CDM session types:

Deleted: three

1. Flight Data - used to exchange flight data messages (FC, FM, FX, and EI).
2. Simplified Substitution - used to exchange simplified subs messages (slot lists, simplified sub packets, bridging requests/replies, and EDCT data requests/replies).
3. ADL - utilized to obtain ADL data, to transmit GDP parameters, FSM Broadcast messages, and to provide access to various required commands.
4. CTOP – used to exchange CTOP FCA, Trajectory, TMI and Substitution messages.

Each of these session types uses a subset of the Protocol messages listed in Table 1-II to initiate, validate, and terminate sessions. Table 1-II provides an overall summary of the FD, SS, ADL and CTOP session protocol messages. It lists the message number of the Protocol message, the message name, the transfer path depicting the direction of the message flow, the type of session used by the particular Protocol message and the section reference in this specification where the messages are discussed in detail. Following the table, the session protocol messages are individually broken out and detailed in individual subsections.

Table 1-II. CDM Session Protocol Message Summary

Msg #*	Message Name	Transfer Path**	Session Type				Reference
			FD (AOCNet and ARINC only)	SS (AOCNet, ARINC, and FSM)	ADL (AOCNet and FSM Only)	CTOP (AOCNet only)	
1	M_ATMS_CONNECT	C → S	Yes***	Yes	Yes		1.5.1
	M_CTOP_CONNECTION_REQUEST	C → S				Yes	CTOP ICD- Section 4.1, CTOP ICD-Section.5.3.1
2	M_ATMS_ACCEPT	C ← S	Yes***	Yes	Yes		1.5.2
	M_CTOP_CONNECTION_ACCEPTED	C ← S				Yes	CTOP ICD- Section 4.1, CTOP ICD-Section.5.3.1
3	M_ATMS_REJECT	C ← S	Yes***	Yes	Yes		1.5.3
	M_CTOP_CONNECTION_REJECTED	C ← S				Yes	CTOP ICD- Section 4.1, CTOP ICD-, Section.5.3.1
4	M_DISCONNECT	C → S			Yes		1.5.4
5	M_SHUTDOWN	C ← S	Yes	Yes	Yes		1.5.5
	M_CTOP_SHUTDOWN	C ← S				Yes	CTOP ICD-Section 4.1, CTOP ICD-Section.5.3.1
10	M_HB_REQ	C ↔ S	Yes	Yes	Yes		1.5.6
	M_CTOP_HB_REQ	C → S				Yes	CTOP ICD-Section 4.1, CTOP ICD-Section.5.3.1
11	M_HB_ACK	C ← S	Yes	Yes	Yes		1.5.7
	M_CTOP_HB_REPLY	C ← S				Yes	CTOP ICD Section 4.1, CTOP ICD-Section.5.3.1

Deleted: Mgmt/App***
Deleted: Flight Data¶
Deleted: and ARINC
Deleted: Simpl Subs¶ (All three I/Fs)
Deleted: ****
Deleted: Mgmt
Deleted: ****
Deleted: Mgmt
Deleted: ****
Deleted: Mgmt
Deleted: Mgmt
Deleted: Mgmt
Deleted: 10
Deleted: HB_REQ
Deleted: →
Deleted: Yes
Deleted: Yes
Deleted: Mgmt
Deleted: 11
Deleted: ACK
Deleted: ←
Deleted: Mgmt
Deleted: 101
Deleted: M_FLIGHT_DATA_PACKET
Deleted: App
Deleted: 102
Deleted: FLIGHT_DATA_REPLY
Deleted: App
Deleted: 103 ...
Deleted: GDP
Deleted: Yes
Deleted: App

Msg #*	Message Name	Transfer Path**	Session Type				Reference	
			FD (AOCNet and ARINC only)	SS (AOCNet, ARINC, and FSM)	ADL (AOCNet and FSM Only)	CTOP (AOCNet only)		

* Note – the following message numbers are not allocated: 6 through 9

** Note – TFMS is "Server", designated as "S". AOCNET, ARINC, and FSM are Clients, designated as "C" in the Transfer Path column.

*** Note – Usage of the M_ATMS_CONNECT message is optional. The M_ATMS_ACCEPT and M_ATMS_REJECT messages are only used, as appropriate, in response to the optional M_ATMS_CONNECT message.

Deleted: Mgmt/App***

Deleted: Flight Data¶

Deleted: and ARINC

Deleted: Simpl Subs¶
(All three I/Fs)

1.5.1 M_ATMS_CONNECT [1]

This session management message is used to tell a server at the TFMS TPC that a new client process is coming on-line. The message identifies where the client is running. The server validates the request. If it is valid, the session starts. The server notifies the client whether the connection was accepted or rejected. If there is more than one client at this site, the client tag field must contain different invocation numbers for each client. This message is optional for a Flight Data session, but is required for ADL and Simplified Subs sessions.

Table 1-III. M_ATMS_CONNECT Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **1**, indicating M_ATMS_CONNECT	4
Message Source Address	Binary	Encoded source identifier (one of two entries): 1. **103** - if FSM Server 2. **0** - if Flight Data or Simplified Subs clients	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static Entry - **0**	4
Message Length	Binary	Static Entry - **0**	4
Data Buffer	---	Not applicable to message	0

1.5.2 M_ATMS_ACCEPT [2]

This session management message from the server tells the client that the socket connection has been set up correctly and is accepted. The message is in response to an M_ATMS_CONNECT message. The Short Data field should be zero, and there is no message body.

Table 1-IV. M_ATMS_ACCEPT Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **2**, indicating M_ATMS_ACCEPT	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static Entry - **0**	4
Message Length	Binary	Static Entry - **0**	4
Data Buffer	---	Not applicable to message	0

1.5.3 M_ATMS_REJECT [3]

This session management message from the server tells the client that the connection has not been accepted (i.e.,, is rejected). The message is in response to an M_ATMS_CONNECT message. The Short Data field contains an error code explaining the cause of the failure.

Table 1-V. M_ATMS_REJECT Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **3**, indicating M_ATMS_REJECT	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Error code (one of these selections): • 1 - Unknown message source • 2 - Invalid password • 3 - Server processing currently not available	4

Field	Unit/Format	Description	Bytes
		• 4 - Client with same invocation number already running Other entries are possible in the future	
Message Length	Binary	Static Entry - **0**	4
Data Buffer	---	Not applicable to message	0

1.5.4 M_DISCONNECT [4]

This session management message is sent from the client to the server to indicate that the client process is shutting down. The Short Data field is zero and there is no message body.

Table 1-VI. M_DISCONNECT Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **4**, indicating M_DISCONNECT	4
Message Source Address	Binary	Encoded source identifier • **103**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static Entry - **0**	4
Message Length	Binary	Static Entry - **0**	4
Data Buffer	---	Not applicable to message	0

1.5.5 M_SHUTDOWN [5]

This session management message is sent from a server to a client to indicate that the server process is shutting down. The Short Data field is zero and there is no message body. This gives the client the chance to shut down gracefully.

Table 1-VII. M_SHUTDOWN Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **5**, indicating M_SHUTDOWN	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned	4

Field	Unit/Format	Description	Bytes
		by FAA	
Short Data	Binary	Static Entry - **0**	4
Message Length	Binary	Static Entry - **0**	4
Data Buffer	---	Not applicable to message	0

1.5.6 M_HB_REQ [10]

This session management message is sent from a client to the server to request a reply that confirms that the connection is still alive.

Table 1-VIII M_HB_REQ Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **10** indicating M_HB_REQ	4
Message Source Address	Binary	Encoded source identifier (one of two entries): • **103** - if FSM Server • **0** - if Flight data or simplified subs client	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client **	4
Message Length	**0**	Static Entry - **0**	4
Data Buffer	---	Not applicable to message	0
**Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message			

Deleted: (1 to 9999)

1.5.7 M_HB_ACK [11]

This session management message is sent from the server to a client as the reply to a M_HB_REQ request. It confirms that the connection is alive.

Table 1-IX. M_HB_ACK Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **11** indicating M_HB_ACK	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination	Binary	Encoded destination identifier (one of	4

Field	Unit/Format	Description	Bytes
Address		two entries): • **103** - if FSM Server • **0** - if Flight Data or Simplified Subs client	
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number supplied by the client in the M_HB_REQ **	4
Message Length	Binary	Static Entry - **0**	4
Data Buffer	---	Not applicable to message	0
**Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message			

Deleted: (1 to 9999)

1.6 CDM Session Application Messages

This section defines the CDM application messages for the FD, SS, ADL and the CTOP sessions. Table 1-X lists all the application messages pertaining to the various sessions.

Table 1-X. CDM Session Application Message Summary

Msg #*	Message Name	Transfer Path**	Session Type				Reference
			FD (AOCNet and ARINC only)	SS (AOCNet, ARINC, and FSM)	ADL (AOCNet and FSM Only)	CTOP (AOCNet only)	
101	M_FLIGHT_DATA_PACKET	C → S	Yes				1.6.1
102	M_FLIGHT_DATA_REPLY	C ← S	Yes	Yes			1.6.2
103	M_SLOT_DATA	C ← S		Yes			1.6.3
104	M_GDP_REQ	C → S		Yes			1.6.4
105	M_GDP_REPLY	C ← S		Yes			1.6.5
106	M_GDP_MESSAGE	C ← S		Yes			1.6.6
111	M_EI_FLIGHT_DATA_PACKET	C → S	Yes				1.6.7
112	M_SS_DATA_PACKET	C → S		Yes			1.6.8
201	M_REGISTER	C → S			Yes		1.6.9
202	M_REGISTER_ACK	C ← S			Yes		1.6.10
203	M_UN_REGISTER	C → S			Yes		1.6.11
204	M_UN_REGISTER_ACK	C ← S			Yes		1.6.12
205	M_START_ADL	C ← S			Yes		1.6.13
206	M_ADL_DATA	C ← S			Yes		1.6.14
207	M_END_ADL	C ← S			Yes		1.6.15
208	M_ELEMENT_DELETED	C ← S			Yes		1.6.16
212	M_EDCT_COMMAND	C → S			Yes		1.6.17
213	M_EDCT_REPLY	C ← S			Yes		1.6.18
214	M_REQ_COMMAND	C → S			Yes		1.6.19
215	M_REQ_REPLY	C ← S			Yes		1.6.20
218	M_ADL_DATA_ACK	C ← S			Yes		1.6.21
219	M_ADD_ADL_AAR	C → S			Yes		1.6.22
220	M_ADD_ADL_ADR	C → S			Yes		1.6.23

Msg #*	Message Name	Transfer Path**	Session Type				Reference
			FD (AOCNet and ARINC only)	SS (AOCNet, ARINC, and FSM)	ADL (AOCNet and FSM Only)	CTOP (AOCNet only)	
221	M_ADD_ADL_AAR_GDP	C → S			Yes		1.6.24
222	M_ADD_ADL_GDP_PARAM	C → S			Yes		1.6.25
224	M_ADD_ADL_COMP_PARAM	C → S			Yes		1.6.26
225	M_ADD_ADL_BLANK_PARAM	C → S			Yes		1.6.27
226	M_ADD_ADL_GS_PARAM	C → S			Yes		1.6.28
227	M_DEL_ADL_AAR	C → S			Yes		1.6.29
228	M_DEL_ADL_ADR	C → S			Yes		1.6.30
230	M_DEL_ADL_GDP_PARAM	C → S			Yes		1.6.31
235	M_UPDATE_ADL_REQ	C → S			Yes		1.6.32
236	M_WEATHER_COMMAND	C → S			Yes		1.6.33
237	M_WEATHER_REPLY	C ← S			Yes		1.6.34
242	M_AUTO_MONITOR_REQ	C → S			Yes		1.6.35
243	M_AUTO_MONITOR_REPLY	C ← S			Yes		1.6.36
244	M_AUTO_MONITOR_MESSAGE	C → S			Yes		1.6.37
245	M_ADD_ADL_AFP_PARAM	C → S			Yes		1.6.38
300	M_CTOP_FCA	C ← S				Yes	CTOP ICD-Section 4.2, CTOP ICD-Section 5.3.2
301	M_CTOP_FCA_LIST_REQUEST	C → S				Yes	CTOP ICD-Section 4.2, CTOP ICD-Section 5.3.2
302	M_CTOP_FCA_LIST_REPLY	C ← S				Yes	CTOP ICD-Section 4.2, CTOP ICD-Section 5.3.2
303	M_CTOP_FCA_RE-SYNCH_REQUEST	C → S				Yes	CTOP ICD-Section 4.2, CTOP ICD-Section 5.3.2

Msg #*	Message Name	Transfer Path**	Session Type				Reference
			FD (AOCNet and ARINC only)	SS (AOCNet, ARINC, and FSM)	ADL (AOCNet and FSM Only)	CTOP (AOCNet only)	
304	M_CTOP_FCA_RE-SYNCH_REPLY	C ← S				Yes	CTOP ICD-Section 4.2, CTOP ICD-Section 5.3.2
305	M_CTOP_FCA_FLIGHT_LIST_REQUEST	C → S				Yes	CTOP ICD-Section 4.2, CTOP ICD-Section 5.3.2
306	M_CTOP_FCA_FLIGHT_LIST_REPLY	C ← S				Yes	CTOP ICD-Section 4.2, CTOP ICD-Section 5.3.2
307	M_CTOP_FCA_REQUEST_ERROR	C ← S				Yes	CTOP ICD-Section 4.2, CTOP ICD-Section 5.3.2
308	M_CTOP_FCA_DELETE	C ← S				Yes	CTOP ICD-Section 4.2, CTOP ICD-Section 5.3.2
320	M_CTOP_TOS_MESSAGE	C → S				Yes	CTOP ICD-Section 4.3, CTOP ICD-Section 5.3.3
321	M_CTOP_TOS_REPLY	C ← S				Yes	CTOP ICD - Section 4.3, CTOP ICD-Section 5.3.3
322	M_CTOP_TOS_ERROR	C ← S				Yes	CTOP ICD-Section 4.3, CTOP ICD-Section 5.3.3
323	M_CTOP_TOS_RE-SYNCH_REQUEST	C → S				Yes	CTOP ICD-Section 4.3, CTOP ICD,- Section 5.3.3
324	M_CTOP_TOS_RE-SYNCH_REPLY	C ← S				Yes	CTOP ICD-Section 4.3, CTOP ICD-Section 5.3.3

Msg #*	Message Name	Transfer Path**	Session Type				Reference
			FD (AOCNet and ARINC only)	SS (AOCNet, ARINC, and FSM)	ADL (AOCNet and FSM Only)	CTOP (AOCNet only)	
325	M_CTOP_TOS_RE-SYNCH_ERROR	C ← S				Yes	CTOP ICD-Section 4.3, CTOP ICD-Section 5.3.3
330	M_CTOP_TMI	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
331	M_CTOP_TRAJECTORY_ASSIGNMENT	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
332	M_CTOP_POP-UP	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
333	M_CTOP_DROP-OUT	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
334	M_CTOP_LIST_REQUEST	C → S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
335	M_CTOP_LIST_REPLY	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
336	M_CTOP_RE-SYNCH_REQUEST	C → S				Yes	CTOP ICD, Section 4.4- CTOP ICD-Section 5.3.4
337	M_CTOP_RE-SYNCH_REPLY	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
338	M_CTOP_RE-SYNCH_ERROR	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4

Msg #*	Message Name	Transfer Path**	Session Type				Reference
			FD (AOCNet and ARINC only)	SS (AOCNet, ARINC, and FSM)	ADL (AOCNet and FSM Only)	CTOP (AOCNet only)	
339	M_CTOP_TRAJECTORY_ASSIGNMENT_RE-SYNCH_REQUEST	C → S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
340	M_CTOP_TRAJECTORY_ASSIGNMENT_RE-SYNCH_REPLY	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
341	M_CTOP_TRAJECTORY_ASSIGNMENT_RE-SYNCH_ERROR	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
342	M_CTOP_CANCEL	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
343	M_CTOP_FLIGHT_PLAN_AMENDMENT	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
344	M_CTOP_SUSPEND	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
345	M_CTOP_RESUME	C ← S				Yes	CTOP ICD-Section 4.4, CTOP ICD-Section 5.3.4
350	M_CTOP_SUBSTITUTION_MESSAGE	C → S				Yes	CTOP ICD-Section 4.5, CTOP ICD-Section 5.3.5
351	M_CTOP_SUBSTITUTION_REPLY	C ← S				Yes	CTOP ICD-Section 4.5, CTOP ICD-Section 5.3.5
352	M_CTOP_SUBSTITUTION_ERROR	C ← S				Yes	CTOP ICD-Section 4.5, CTOP ICD-Section 5.3.5

Msg #*	Message Name	Transfer Path**	Session Type				Reference
			FD (AOCNet and ARINC only)	SS (AOCNet, ARINC, and FSM)	ADL (AOCNet and FSM Only)	CTOP (AOCNet only)	
* Note – CDM Session Application message numbers begin at 101. The following message numbers are not allocated: 107 through 110, 113 through 200, 209 through 211, 216 through 217, 223, 229, 231through 234, 238 through 241, 246 through 299, 309 through 319, 326 through 329, and 346 through 349 ** Note – TFMS is "Server", designated as "S". AOCNET, ARINC, and FSM are Clients, designated as "C" in the Transfer Path column.							

1.6.1 M_FLIGHT_DATA_PACKET [101]

This message from a client to the server sends flight data messages from the airline. The data buffer contains the packet formatted as: packet header followed by a variable number of messages terminated by line feeds. The Short Data field contains a number, which is a 32 bit integer value.

Table 1-XI. M_FLIGHT_DATA_PACKET Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **101** indicating M_FLIGHT_DATA_PACKET	4
Message Source Address	Binary	Encoded source identifier • **0 – FSM Server** • **105 – FSM Client**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client **	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Message packet containing a header line and a variable number of FD application messages.	*
*Note – The message packet header identifies the packet as an FD packet. The length of the FD packet (in bytes), including the packet header is specified by the Message Length field. **Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message			

Deleted: (1 to 9999)

1.6.2 M_FLIGHT_DATA_REPLY [102]

This message from the server to a client sends the replies to the flight data messages that had been sent from the airline (sent in an FD packet) or simplified substitution messages (sent in an SS packet). The reply uses the same packet type as the message being replied to, FD or SS. The data buffer contains an acknowledgment line followed by a variable number of error messages, when needed. The acknowledgement option exists for the packet header of this reply; namely, "NOACK" (i.e.,, if no "good" acknowledgment is desired), then an optional ARINC MQ address for errors. M_FLIGHT_DATA_REPLY message components are not terminated by line feeds, as in the ARINC MQ protocol, but sent as separate strings in the data buffer. The Short Data field contains the number from the Short Data in the flight data message to which the server is replying.

Table 1-XII. M_FLIGHT_DATA_REPLY Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **102** indicating M_FLIGHT_DATA_REPLY	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **0 - FSM Server** • **105 – FSM Client**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number supplied by the client in the M_FLIGHT_DATA_PACKET message or the M_SS_DATA_PACKET **	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Message packet containing an acknowledgment and a variable number of error messages, as appropriate.	*

*Note – The message packet header identifies the packet as an FD or SS packet, depending on the context of the data exchange (Flight Data session or Simplified Subs session). The length of the FD or SS packet (in bytes), including the packet header is specified by the Message Length field.

**Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message

1.6.3 M_SLOT_DATA [103]

This message from the server to the client sends slot lists (as part of a delay program) from the server to the airline. The data buffer contains the slot list formatted as a stream of ASCII as specified in the simplified subs requirements documents.

Table 1-XIII. M_SLOT_DATA Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **103** indicating M_SLOT_DATA	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static Entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	A slot list formatted according to the simplified subs definitions.	*
*Note – The message packet header identifies the packet as an SS packet. The length of the SS packet (in bytes), including the packet header is specified by the Message Length field.			

1.6.4 M_GDP_REQ [104]

This message from the client to the server sends either EDCT report request(s) or a bridging request. The data buffer contains the request formatted as a stream of ASCII rather than as a packet as is done for other SS session messages. The Short Data field contains a number, which is a 32 bit integer value.

Table 1-XIV M_GDP_REQ Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **104** indicating M_GDP_REQ	4
Message Source Address	Binary	Encoded source identifier • **0 - FSM Server** • **105 – FSM Client**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client.**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	A request.*	*
*Note – Only EDCT report request(s) or a Bridging request is allowed in the Data Buffer field and is sent without a header. The length of the request(s), in bytes, is specified by the Message Length field. The application messages allowed are the Report Requests: EDCT LIST, EDCT SUB SHOW, EDCT SLIST, EDCT UNASSIGNED SLOTS or the Bridging Requests: EDCT BRIDGING ON or EDCT			

Deleted: (1 to 9999)

Field	Unit/Format	Description	Bytes
BRIDGING OFF. **Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message			

1.6.5 M_GDP_REPLY [105]

This message from the server to the client sends the response to a GDP request. Namely, the requested EDCT report or confirmation of the requested bridging action is sent from the server to client. The data buffer contains the message formatted as a stream of ASCII.

Table 1-XV. M_GDP_REPLY Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **105** indicating M_GDP_REPLY	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **0 - FSM Server** • **105 – FSM Client**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number supplied by the client in the M_GDP_REQ message.**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	A reply message.*	*
*Note – Only a single message is allowed in the Data Buffer field. The length of the message (in bytes) is specified by the Message Length field. The single allowed message is a Reply to an EDCT Report Request or a reply to a Bridging Request that was received in the M_GDP_REQ message (in the form of: EDCT LIST, EDCT SUB SHOW, EDCT SLIST, EDCT UNASSIGNED SLOTS, EDCT BRIDGING ON, or EDCT BRIDGING OFF). **Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message			

Deleted: (1 to 9999)

1.6.6 M_GDP_MESSAGE [106]

This message from the server to the client sends unsolicited messages (associated with delay program adjustments) from the server to the airline. The data buffer contains a single unsolicited message formatted as a stream of ASCII as specified in the simplified subs requirements documents.

Table 1-XVI. M_GDP_MESSAGE Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **106** indicating M_GDP_MESSAGE	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **0 - FSM Server** • **105 – FSM Client**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static Entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	One Unsolicited message.	*
*Note – Only a single unsolicited message is allowed in the Data Buffer field. The length of the message (in bytes) is specified by the Message Length field.			

1.6.7 M_EI_FLIGHT_DATA_PACKET [111]

This message from a client to the server sends early intent messages from the airline to the server. The data buffer contains the EI packet header followed by one early intent flight plan.

Table 1-XVII. M_EI_FLIGHT_DATA_PACKET Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **111** indicating M_EI_FLIGHT_DATA_ PACKET	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client **	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	A message packet containing a header line and one early intent message as defined in the early intent protocol definitions.	*

Deleted: (1 to 9999)

Field	Unit/Format	Description	Bytes
*Note – The message packet header identifies the packet as an EI packet. The length of the EI packet (in bytes), including the packet header is specified by the Message Length field. **Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message			

1.6.8 M_SS_DATA_PACKET [112]

This message from a client to the server sends simplified subs and slot credit sub messages from the airline to the server. The data buffer contains the SS packet header followed by a variable number of messages terminated by line feeds. The Short Data field contains a number, which is a 32 bit integer value. All SS session messages sent from a client must be responded to. The M_FLIGHT_DATA_REPLY message is used for replies to the M_SS_DATA_PACKET message.

Table 1-XVIII. M_SS_DATA_PACKET Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **112** indicating M_SS_DATA_PACKET	4
Message Source Address	Binary	Encoded source identifier • **0 - FSM Server** • **105 – FSM Client**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client **	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	A message packet containing a header line and a variable number of messages as defined in the simplified subs and slot credit subs definitions.	*
*Note – The message packet header identifies the packet as an SS packet. The length of the SS packet (in bytes), including the packet header is specified by the Message Length field. **Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message			

Deleted: (1 to 9999)

1.6.9 M_REGISTER [201]

This message from the client tells the server that data for a particular element is desired. The Short Data field tells the server what type of data is being requested. The message body (data buffer) must contain the name (identifier) of the element being requested and the pathname where the files go. The pathname specified is used by the server when sending data to clients in the M_START_ADL message.

Table 1-XIX. M_REGISTER Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **201** indicating M_REGISTER	4
Message Source Address	Binary	Encoded source identifier • **103**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	ADL Type requested - one of the following entries: • **0** - Airport ADL with Arrivals Only • **1** – Airport ADL with Departures Only • **2** - Airport Arrivals and Departures • **3** - FEA ADL • **7** - FCA ADL	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	text	Client name, Control Element name, client pathname, and client version; fields are space delimited	*

- Note-The total number of bytes (including spaces) of the client name, Control Element name (i.e.,, airport, FEA, or FCA), client pathname, and client version. Note-Since fields in the data buffer are space delimited, spaces are not allowed within each value. Due to this a path with spaces is not allowed.
- Note-FSM does not make actual use of the path. A place holder path is included in this message and is then stripped by FSM from reply messages.

1.6.10 M_REGISTER_ACK [202]

This message from the server to the client tells client whether the registration was accepted or not. The Short Data field is 0 if the registration was accepted, or contains an error code if the registration was not accepted. The data buffer contains the name of the Control Element (i.e.,, airport, FEA, or FCA) for which the registration was requested.

Table 1-XX. M_REGISTER_ACK Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **202** indicating M_REGISTER_ACK	4
Message Source	Binary	Encoded source identifier	4

Field	Unit/Format	Description	Bytes
Address		• **0**	
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	One of the following entries: • **0** - Accepted • **1** - Unknown Control Element • **2** - Control Element already registered • **3** - Maximum number of Control Elements exceeded	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	Laa[a]	Control Element name	3-4

1.6.11 M_UN_REGISTER [203]

This message from the client tells the server to stop providing collected data for a particular Control Element (airport, FEA, or FCA) for this client. The Short Data field tells the server what type of data is to be Un-Registered. The message body (i.e.,, Data Buffer field) should contain the Control Element name, or the keyword "ALL". If "ALL" is specified, then the ADL Data Distributor should stop providing data for all Control Elements which have been previously registered by this client.

If a client is shutting down, it should send an un-register for all Control Elements for which data is currently being provided prior to sending a M_DISCONNECT message.

Table 1-XXI. M_UN_REGISTER Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **203** indicating M_UN_REGISTER	4
Message Source Address	Binary	Encoded source identifier • **103**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	ADL Type being unregistered - one of the following entries: • **0** - Airport ADL with Arrivals Only	4

Field	Unit/Format	Description	Bytes
		• **1** – Airport ADL with Departures Only • **2** - Airport Arrivals and Departures • **3** - FEA ADL • **7** - FCA ADL	
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	Laa[a]	Control Element name or "ALL"	3-4

1.6.12 M_UN_REGISTER_ACK [204]

This message from the server to the client tells the client whether the un-registration was accepted or rejected. The Short Data field is 0 if the un-registration was accepted, or contains an error code is the un-registration was not accepted. The data buffer contains the name of the Control Element for which the un-registration was requested.

Table 1-XXII. M_UN_REGISTER_ACK Message Table

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **204** indicating M_UN_REGISTER_ACK	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	One of the following entries: • **0** - Accepted • **1** - Control Element not registered • **2** - No Control Elements registered	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	Laa[a]	Control Element name or "ALL"	3-4

1.6.13 M_START_ADL [205]

This message from the server to client tells the client that the server is beginning to send a new ADL file. The data buffer contains the pathname of the ADL file. The client opens the file and waits for data packets, which it writes to the file. The client also starts a sequence counter for the file packets. The Short Data field contains the first sequence number that is always one.

Table 1-XXIII. M_START_ADL Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **205** indicating M_START_ADL	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Packet Sequence Number, always **1** (Start of ADL packets)	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Pathname of file	*
* Note – The pathname of the ADL file being sent is the pathname of the file on the client's site. The pathname includes the ADL file name. The length (in bytes) of the pathname is specified in the field Message Length.			

1.6.14 M_ADL_DATA [206]

This message is used to send a piece of an ADL file from the server to the client. The data buffer contains the ADL data. The client extracts the ADL data from the buffer and write the ADL data to the file that is already open (the pathname, including the file name was specified in the M_START_ADL message). The Short Data field contains the packet sequence number. The client checks the sequence number, notifying the server if there is an error.

Table 1-XXIV. M_ADL_DATA Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **206** indicating M_ADL_DATA	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Packet Sequence Number	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	ADL Data	*
* Note – The Data Buffer contains an ADL data packet, which can be as long as necessary to provide a complete ADL.			

1.6.15 M_END_ADL [207]

This message from the server to client tells the client that the ADL file download is complete. The data buffer contains the pathname of the file. The Short Data field contains the last sequence number. The client checks the received ADL file and if all is well, closes the file.

Table 1-XXV. M_END_ADL Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **207** indicating M_END_ADL	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Packet Sequence Number	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Pathname of file	*
* Note – The pathname of the ADL file is the pathname of the file on the client's site. The pathname includes the ADL file name. The length (in bytes) of the pathname is specified in the field Message Length.			

1.6.16 M_ELEMENT_DELETED [208]

This message from the server to client tells the client that a particular Control Element which that client is monitoring has been deleted. This indicates to the client that no further data for this Control Element is to be transmitted.

Table 1-XXVI. M_ELEMENT_DELETED Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **208** indicating M_ELEMENT_DELETED	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Control Element Number: • **3** – FEA • **7** – FCA	4

Field	Unit/Format	Description	Bytes
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Control Element Name	*
* Note – The Control Element, for which this delete notification is being issued, is limited to FEAs and FCAs. The Control Element name, therefore, is the name of the deleted FEA or FCA.			

1.6.17 M_EDCT_COMMAND [212]

This message from the client to the server is an M_EDCT_COMMAND. The message body contains the full text of the command.

Table 1-XXVII. M_EDCT_COMMAND Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **212** indicating M_EDCT_COMMAND	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Any single EDCT Command.	*
* Note - The full text of the EDCT command is in ASCII format. The length (in bytes) of the command is specified by Message Length. ** Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message.			

1.6.18 M_EDCT_REPLY [213]

This message from the server to the client is the reply generated in response to the EDCT command received by the server. The message body contains the full text of the reply.

Table 1-XXVIII. M_EDCT_REPLY Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **213** indicating M_EDCT_REPLY	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination	Binary	Encoded destination identifier	4

Field	Unit/Format	Description	Bytes
Address		• **105**	
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client in the M_EDCT_COMMAND message **	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Text of reply	*
* Note – The reply to the EDCT command is in ASCII format. The length of the reply is specified by Message Length. ** Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message.			

Deleted: (1 to 9999)

1.6.19 M_REQ_COMMAND [214]

This message from the client to the server is an EDCT command line request. The message body contains the full text of the command.

Table 1-XXIX. M_REQ_COMMAND Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **214** indicating M_REQ_COMMAND	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **1**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client **	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Request	*
* Note – The format of the command line EDCT command is in ASCII format with a length specified by Message Length. ** Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message.			

Deleted: (1 to 9999)

1.6.20 M_REQ_REPLY [215]

This message from the server to the client is the reply in response to an EDCT command line request. The message body contains the full text of the reply.

Table 1-XXX. M_REQ_REPLY Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **215** indicating M_REQ_REPLY	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **105**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number supplied by the client in the M_REQ_COMMAND message **	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Text of reply	*
* Note – The reply to the EDCT command line request is in ASCII format with a length specified by Message Length. ** Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message.			

Deleted: (1 to 9999)

1.6.21 M_ADL_DATA_ACK [218]

This message from the server to the client tells the client that the previous GDP data message was received. The Short Data field equals the value of the Short Data field of the GDP message. GDP data received by the server is included in subsequent ADL downloads from server to client.

Table 1-XXXI. M_ADL_DATA_ACK Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **218** indicating M_ADL_DATA_ACK	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **103**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number supplied by the client in the previous GDP message **	4
Message Length	Binary	Static entry - **0**	4
Data Buffer	---	Not applicable to this message	0
** Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message.			

Deleted: (1 to 9999)

1.6.22 M_ADD_ADL_AAR [219]

This message from a client to the server submits an updated AAR for a Control Element (airport, FEA, or FCA). The data received by the server included in subsequent ADL downloads from server to client.

Table 1-XXXII. M_ADD_ADL_AAR Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **219** indicating M_ADD_ADL_AAR	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Name of the Control Element and the new rates (i.e.,, new AAR values for the specified Control Element).	*
* Note – The Data Buffer contains identification of the particular Control Element and the new AAR values for that Control Element.			

1.6.23 M_ADD_ADL_ADR [220]

This message from a client to the server submits an updated ADR for an airport. The data received by the server is included in subsequent ADL downloads from server to client.

Table 1-XXXIII. M_ADD_ADL_ADR Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **220** indicating M_ADD_ADL_ADR	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Name of the airport and the new rates	*

Field	Unit/Format	Description	Bytes
* Note – The Data Buffer contains identification of the particular airport and the new ADR values for that airport.			

1.6.24 M_ADD_ADL_AAR_GDP [221]

This message from a client to the server submits the AAR associated with a delay program (GDP or AFP) for a Control Element. This message must be sent prior to message M_ADD_ADL_GDP_PARAM. The delay program data received by the server is included in subsequent ADL downloads from server to client.

Table 1-XXXIV. M_ADD_ADL_AAR_GDP Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **221** indicating M_ADD_ADL_AAR_GDP	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Name of the Control Element and the new rate	*
* Note – The Data Buffer contains identification of the particular Control Element and the delay program related AAR value for that Control Element.			

1.6.25 M_ADD_ADL_GDP_PARAM [222]

This message from a client to the server submits the parameters for a GDP. The GDP data received by the server is included in subsequent ADL downloads from server to client.

Table 1-XXXV. M_ADD_ADL_GDP_PARAM Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **222** indicating M_ADD_ADL_GDP_PARAM	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned	4

Field	Unit/Format	Description	Bytes
		by FAA	
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Name of the airport and the associated GDP parameters	*
* Note – The GDP data is in ASCII format and is then included in the "GDP_PARAMS" block of subsequent ADLs for the identified airport.			

1.6.26 M_ADD_ADL_COMP_PARAM [224]

This message from a client to the server submits the parameters for a compression of a previously issued GDP or AFP. The compression data received by the server is included in subsequent ADL downloads from server to client.

Table 1-XXXVI. M_ADD_ADL_COMP_PARAM Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **224** indicating M_ADD_ADL_COMP_PARAM	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Name of the Control Element and the compression parameters	*
* Note – The compression data is in ASCII format and is then included in the "COMP_PARAMS" block of subsequent ADLs for the identified Control Element.			

1.6.27 M_ADD_ADL_BLANK_PARAM [225]

This message from a client to the server submits the parameters for a blanket program for a previously issued GDP or AFP. The blanket program data received by the server is included in subsequent ADL downloads from server to client.

Table 1-XXXVII. M_ADD_ADL_BLANK_PARAM Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **225** indicating M_ADD_ADL_BLANK_ PARAM	4

Field	Unit/Format	Description	Bytes
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Name of the Control Element and the blanket program parameters	*
* Note – The blanket program data is in ASCII format and is then included in the "BKT_PARAMS" block of subsequent ADLs for the identified Control Element.			

1.6.28 M_ADD_ADL_GS_PARAM [226]

This message from a client to the server submits the parameters for a ground stop. The ground stop data received by the server is included in subsequent ADL downloads from server to client.

Table 1-XXXVIII. M_ADD_ADL_GS_PARAM Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **226** indicating M_ADD_ADL_GS_PARAM	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Name of the airport and the ground stop parameters	*
* Note – The ground stop data is in ASCII format and is then included in the "GS_PARAMS" block of subsequent ADLs for the identified airport.			

1.6.29 M_DEL_ADL_AAR [227]

This message from a client to the server deletes the AAR for a Control Element.

Table 1-XXXIX. M_DEL_ADL_AAR Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **227** indicating M_DEL_ADL_AAR	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	Laa[a][a][a]	Name of the Control Element	3-4

1.6.30 M_DEL_ADL_ADR [228]

This message from a client to the server deletes the ADR for an airport.

Table 1-XL. M_DEL_ADL_ADR Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **228** indicating M_DEL_ADL_ADR	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	Laa[a]	Name of the airport	3-4

1.6.31 M_DEL_ADL_GDP_PARAM [230]

This message from a client to the server deletes all parameters for a GDP including GS, BLANK and COMP for an airport.

Table 1-XLI. M_DEL_ADL_GDP_PARAM Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **230** indicating M_DEL_ADL_GDP_PARAM	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	Laa[a]	Name of the airport	3-4

1.6.32 M_UPDATE_ADL_REQ [235]

This message from a client to the server requests an updated ADL for a Control Element.

Table 1-XLII. M_UPDATE_ADL_REQ Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **235** indicating M_UPDATE_ADL_REQ	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client*	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	Laa[a][a][a]	Name of the Control Element	3-4
*Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message.			

1.6.33 M_WEATHER_COMMAND [236]

This message from a client to the server requests a Meteorological Aviation Report (METAR) and Terminal Aerodrome Forecast (TAF) report for an airport. The requested METAR and TAF are included in subsequent ADLs downloaded from server to client as well as provided in an explicit reply.

Table 1-XLIII. M_WEATHER_COMMAND Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **236** indicating M_WEATHER_COMMAND	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number provided by the client*	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	Laa[a]	Name of the airport	3-4
*Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message.			

1.6.34 M_WEATHER_REPLY [237]

This message from a server to the client contains the METAR and TAF report that was requested for an airport.

Table 1-XLIV. M_WEATHER_REPLY Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **237** indicating M_WEATHER_REPLY	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **105**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	A number supplied by the client in the corresponding M_WEATHER_COMMAND (as provided in the previous message) **	4
Message Length	Binary	Total number of bytes in data buffer	4

Field	Unit/Format	Description	Bytes
Data Buffer	*	Text of reply	*
* Note – The requested METAR and TAF are in ASCII format. **Note – The number is generated by the client by whatever algorithm the client desires to use and is not used by the TFMS software other than to return it to the client in a related reply message.			

1.6.35 M_AUTO_MONITOR_REQ [242]

This message from a client to the server requests an Auto Monitor report (i.e.,, an FSM Broadcast message request). This report contains the current traffic management initiatives that are in place, are proposed, or have been purged and the current FEAs and FCAs available for monitoring with FSM.

Table 1-XLV. M_AUTO_MONITOR REQ Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **242** indicating M_AUTO_MONITOR_REQ	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **105**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Static Entry - **0**	4
Data Buffer	---	Not applicable to message	0

1.6.36 M_AUTO_MONITOR_REPLY [243]

This message from the server to the client sends the Auto Monitor report to the client that made the request.

Table 1-XLVI. M_AUTO_MONITOR_REPLY Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **243** indicating M_AUTO_MONITOR_REPLY	4
Message Source Address	Binary	Encoded source identifier • **105**	4
Message Destination Address	Binary	Encoded destination identifier • **0**	4
Client Tag	Binary	Client identification number assigned by FAA	4

Field	Unit/Format	Description	Bytes
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Text of Auto Monitor report	*
* Note – The data buffer contains the Auto Monitor report formatted as a stream of ASCII.			

1.6.37 M_AUTO_MONITOR_MESSAGE [244]

This message from the server to the client sends the Auto Monitor report to any connected client. The form of the Auto Monitor report is sent unsolicited at a regular time interval defined by a system parameter, as well as in response to the following event:

- When a TMI is processed
- When an FEA/FCA that is tagged for monitoring is created, modified, or removed

Table 1-XLVII. M_AUTO_MONITOR_MESSAGE Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **244** indicating M_AUTO_MONITOR _MESSAGE	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **105**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Text of Auto Monitor report	*
* Note – The data buffer contains the Auto Monitor report formatted as a stream of ASCII.			

1.6.38 M_ADD_ADL_AFP_PARAM [245]

This message from a client to the server submits the parameters for an AFP. The AFP parameters are included in subsequent downloads of the ADL from server to client.

Table 1-XLVIII. M_ADD_ADL_AFP_PARAM Message

Field	Unit/Format	Description	Bytes
Message Type	Binary	Coded static entry **245** indicating M_ADD_ADL_AFP_PARAM	4
Message Source Address	Binary	Encoded source identifier • **0**	4
Message Destination Address	Binary	Encoded destination identifier • **105**	4
Client Tag	Binary	Client identification number assigned by FAA	4
Short Data	Binary	Static entry - **0**	4
Message Length	Binary	Total number of bytes in data buffer	4
Data Buffer	*	Name of the Control Element and associated AFP parameters	*
* Note – The AFP data is in ASCII format and is then included in the "GDP_PARAMS" block of subsequent ADLs for the identified Control Element.			

1.6.39 CTOP Session Application Messages

The CTOP session protocol messages are identified in Table 1-II. Table 1-X lists the CTOP Application Messages. The CTOP protocols and the associated application messages are covered in their entirety in the TFMS CTOP ICD referenced in Section 1.2.