AOC AIR-GROUND DATA AND MESSAGE EXCHANGE FORMAT

ARINC SPECIFICATION 633

PUBLISHED: June 30, 2007

AN ARINC DOCUMENT

Prepared by
AIRLINES ELECTRONIC ENGINEERING COMMITTEE
Published by
AERONAUTICAL RADIO, INC.
2551 RIVA ROAD, ANNAPOLIS, MARYLAND 21401-7435

DISCLAIMER

ARINC SPECIFICATION 633

AOC AIR-GROUND DATA AND MESSAGE EXCHANGE FORMAT

Published: June 30, 2007

Prepared by the AEEC

Specification 633 Adopted by the Airlines Electronic Engineering Executive Committee February 20, 2007

A description of the changes introduced by each supplement is included on Goldenrod paper at the end of this document.

FOREWORD

Aeronautical Radio, Inc., the AEEC, and ARINC Standards

Aeronautical Radio, Inc. (ARINC) was incorporated in 1929 by four fledgling airlines in the United States as a privately-owned company dedicated to serving the communications needs of the air transport industry. Today, the major U.S. airlines remain the Company's principal shareholders. Other shareholders include a number of non-U.S. airlines and other aircraft operators.

ARINC sponsors aviation industry committees and participates in related industry activities that benefit aviation at large by providing technical leadership and guidance and frequency management. These activities directly support airline goals: promote safety, efficiency, regularity, and cost-effectiveness in aircraft operations.

The Airlines Electronic Engineering Committee (AEEC) is an international body of airline technical professionals that leads the development of technical standards for airborne electronic equipment-including avionics and in-flight entertainment equipment-used in commercial, military, and business aviation. The AEEC establishes consensus-based, voluntary form, fit, function, and interface standards that are published by ARINC and are known as ARINC Standards. The use of ARINC Standards results in substantial benefits to airlines by allowing avionics interchangeability and commonality and reducing avionics cost by promoting competition.

There are three classes of ARINC Standards:

a) ARINC Characteristics – Define the form, fit, function, and interfaces of avionics and other airline electronic equipment. ARINC Characteristics indicate to prospective manufacturers of airline electronic equipment the considered and coordinated opinion of the airline technical community concerning the requisites of new equipment including standardized physical and electrical characteristics to foster interchangeability and competition.

b) ARINC Specifications – Are principally used to define either the physical packaging or mounting of avionics equipment, data communication standards, or a high-level computer language.

c) ARINC Reports – Provide guidelines or general information found by the airlines to be good practices, often related to avionics maintenance and support.

The release of an ARINC Standard does not obligate any airline or ARINC to purchase equipment so described, nor does it establish or indicate recognition or the existence of an operational requirement for such equipment, nor does it constitute endorsement of any manufacturer's product designed or built to meet the ARINC Standard.

In order to facilitate the continuous product improvement of this ARINC Standard, two items are included in the back of this volume:

An Errata Report solicits any corrections to the text or diagrams in this ARINC Standard.

An ARINC IA Project Initiation/Modification (APIM) form solicits any recommendations for addition of substantive material to this volume which would be the subject of a new Supplement.

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 Purpose of Document

The purpose of this specification is to support the exchange of certain Aeronautical Operational Control (AOC) air-ground and ground-ground messages. These messages are defined in this specification, apart from those defined in ARINC Specification 620, because they have unique qualities. Like the messages defined in ARINC Specification 620, their usage necessitates a single definition. Further, the messages have at least one of the following characteristics:

- The message is defined by an airframe manufacturer, EFB or avionics vender such that it is not modifiable by the airline and does not fit into an existing ARINC Specification e.g., ARINC 622 (FANS), ARINC 623 (ATS) or ARINC 702 (FMS).
- The distribution of the message is outside the control of a single airline, for example, messages shared by multiple parties such as de-icing services shared in common among a group of airlines from a single supplier.

COMMENTARY

Airlines may choose to define AOC applications in this specification that have safety implications. In the USA, the FAA may classify the results of the delivery of such a message as constituting more than a “minor hazard” should certain data contained in the message become corrupted. In this case, the principles specified in RTCA DO-296 need to be observed by the implementer. During the development of this specification some assessment of the safety impact of specified applications was performed, However, the airlines choosing to implement communication services defined in this specification will need to consult their local authorities and may have to perform a hazard analysis for certification of such applications.

This specification provides guidance for the encoding of the messages to be transmitted over the traditional ACARS air-ground links (VHF, SATCOM, and HF). The communications network technology onboard aircraft is evolving to include Ethernet and TCP/IP based networks. Air-ground links that support these commercial protocols have been available for some time. The configuration and equipage of cockpits is expanding to accept data through these commercial media. It is the intent of this specification to support multiple methods of transmission: e.g. traditional character-oriented ACARS air-ground links and commercial network based environments that utilize IP routing protocol and typically UDP or TCP Transport protocol.

When using TCP/IP based ground-ground and air-ground links, this specification encourages the use of XML for message coding. Therefore, for most applications, ACARS and XML message definitions (Schemas) have been designed.

1.2 Scope

The scope of this specification covers all the provisions necessary to enable communication between airborne and ground applications.

Various types of communication can take place between aircraft and ground:

1. ACARS system traditionally allows exchange of messages. These messages can be exchanged asynchronously or in a synchronous way (see note below).
2. Most EFB type applications also use messages to communicate (e.g. eFF, WBA).
3. EFB applications will need to be able to use other types of communications. (transactional communications, such as database access, web access….) that do not involve individual messages exchanges.

This specification addresses the first 2 types of communications. When the need for standardization arise, other types of exchanges than message exchanges will be added.

1.0 INTRODUCTION

Depending on the type of exchanges, asynchronous or synchronous message exchanges are addressed in this document.

Note:

‘Asynchronous ‘ communications means that there is no need for the end applications to be active and ready to communicate during the communication. Typically, an application can send a message to a ground message server, and the receiving application can retrieve it when it gets activated.

‘Synchronous’ communications is used here to depict exchanges where both end applications are active and communicating, allowing real bidirectional exchanges.

Transport technologies (especially TCP/IP based networks) are evolving rapidly; therefore the present document segregates application level functionalities from transport functionalities.

The document covers the air-ground communication function that can be functionally split into (see figure below):

- Applicative communications (the communication part of the application)
  - Message Structure, content and encoding (ACARS and/or XML depending on the application).
  - Minimum requirements on message processing / Dynamic aspects.
  - The objective here is to define ONLY those features necessary for interoperability – the document focuses on data description (XML schemas), but also specifies minimum communication services. The way the data are processed / operated is not defined in this specification.
- Transport (over ACARS, IP and physical media)
  - This specification defines how the messages are transferred using ACARS, IP networks, and physical media, i.e. USB stick or CD/DVD:
    - Data packaging (e.g. ARINC 665 for IP/Physical media)
    - Transport of data (e.g. HTTPs over TCP/IP, or ACARS)
- Security services: Encryption and compaction provisions (provisions only at the time of this writing)

  Note: Some definitions to complement text above.

- Designates the way a message is organized and what data it contains in an abstract language (before encoding for transmission) – in this specification, the message structure and content definition is done directly through definition of message encoding.
- Data Encoding designates the way the data are represented (ASCII, fixed, variable length…) – for example, there is a specific encoding in ACARS, and in XML.
- Message encoding designates the way a message is represented (in terms of structure and data) and data inside these messages are separated, and encoded, before transmission (ASCII, fixed, variable length…) – for example, there is a specific encoding for a given message in ACARS, and in XML.
- Packaging designates the way the data (or files) are encapsulated and associated to other data for transport (e.g. Packaging a given file or message in an ARINC 633 signed load).

Listing operational requirements or specifying the applications that lead to these message formats is outside the scope of this document. The messages are standardized, but not the functionality that necessitates them. This is not a functional spec. However, some guidelines about how the standardized AOC messages may be used are provided. This is done in two ways:

1. Some definition tables contain a column called “Meaning” or “Purpose” explaining the operational or functional context in which the message, element or parameter could generally be used.
2. Some definition tables contain a column called “Example” and some paragraphs contain textual examples that give guidance as to which values could appear under specific conditions in a message, element or parameter.

It should be noted that using messages compliant to this specification may not be sufficient to ensure that a complex air-/ground systems runs properly.

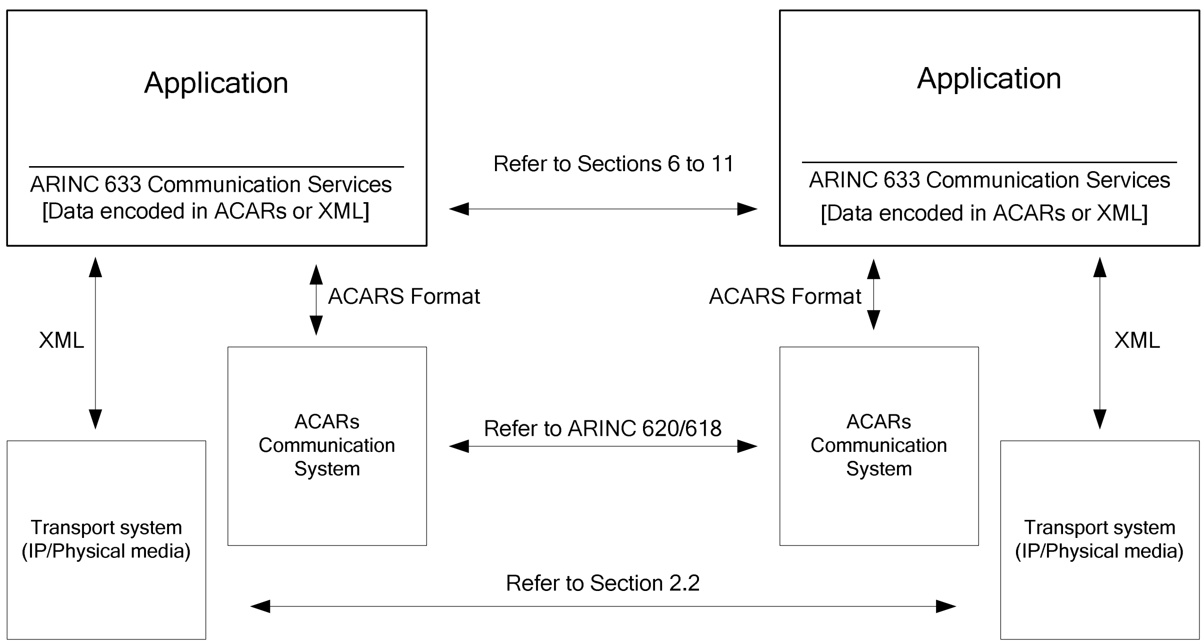


Figure 1.2-1 – Overview of ARINC 633

1.3 Document Organization

This specification is composed of 2 deliverables:

- A text (electronic or paper) document, constituting the body of the specification.
- Electronic schema, an electronic attachment to this document, containing all XML data and message definitions.

1.3.1 "Text" Document

This document is organized as follows:

Section 1	Introductory Material
Section 2	Gives main rules and constraints for definition of messages to be transported over ACARS , IP, or physical media. Also specifies how these messages are transported (what protocols..) over each medium (ACARS, IP, Physical media)
Section 3	ACARS security – at the time of this writing, this section is informative only (security for ACARS is not defined yet)
Section 4	This section defines how the messages will be structured (especially the headers and trailers content are defined), for ACARS and for XML (structures are different).
Section 5	Section 5 defines common elements, i.e. communication services / messages that can be used by various applications, but also data structures that can be used in the messages of various applications.
From Section 6 to last section	Define the applicative communication services as well as message contents, on an application per application basis.
Appendix A	Glossary and acronyms
Appendix B	Transportation of ARINC 633 messages over ACARS – examples of ACARS messages
Appendix C	ACARS messages routing
Appendix D	Aircraft Parameters list

It should be noted that XML schemas are not defined in this document. The XML schema are in the electronic portion of this specification. See Section 1.3.2.

1.3.2 "Electronic" Document

This electronic attachment can be obtained from ARINC Web site (www.arinc.com).

The electronic document only contains XML specification of the messages, as well as examples. It complements the "text"document and has precedence over it for schemas definition.

The electronic document is structured as follows: (see readme file in the zip archive)

- It is a ZIP file, with following name: "ARINC633_AOC.zip".
- Future versions of this electronic file will follow ARINC specification naming convention, i.e. "ARINC633_AOC-1.zip".
- The zip file contains folders, and files (schemas and text files) as follows:
    - A 'readme' file.
    - A change.xls file keeping track of all global changes.
    - 3 main folders:
        - Schema – contains all applicable definition (.xsd files and .xls files describing the change history).
        - Examples – contains example xml files.
        - Doc – contains schema in html format (easy to navigate using a web browser).

    Note: The definitions to be applied are the ones embedded in the 'schema' folder.

    - 'schema' contains:
        - m633header defines mandatory and supplementary headers.
        - m633common defines data types common to several applications.

- CommonData defines the content of data potentially used by multiple applications (e.g. flight plan, notam.).
- 1 folder per application defines the messages using one xsd file per application type (EFF, FUEL...).

As application schemas and instance examples may reference schemas from the “CommonData”, “m633common”, and “m633headers” folders by relative paths, filenames, directory and document names and location should not be changed.

XML Schemas and Instance examples can be viewed and edited with every Editor. However, there are several tools (many of them freeware) available which makes manipulating them much easier.

1.4 Related Documents

1.4.1 Relationship of this Document to Other ARINC Documents

A list of other ARINC documents that contain provisions that are related to the provisions defined in this specification include:

ARINC Specification 618: *Air/Ground Character-Oriented Protocol*

ARINC Specification 619: *ACARS® Protocols for Avionics End Systems*

ARINC Specification 620: *Data Link Ground System Standard and Interface*

ARINC Specification 622: *ATS Data Link Applications Over ACARS Air-Ground Network*

When avionics systems and subsystems are designed to use the capabilities provided by an ARINC Standard, designers should reference the most recent version (modified by Supplement) of the standard.

COMMENTARY

Eurocae has codified Air Traffic Service (ATS) messages in ED-85A, ED-89A, and ED-106A. Although the restrictions that apply to ATS message are more severe than those that apply to the AOC messages define herein, it may be edifying to use these documents as reference material in establishing a context in which critical communications should be conducted.

1.4.2 Relationship to Internet Documents

The Institute of Electrical and Electronic Engineers (IEEE) develops standards for a number of commercial industries, including the telecommunications industry and the computer industry. Some of these standards are shared with the American National Standards Institute (ANSI). On occasion, these standards are endorsed as international standards under the International Standards Organization (ISO) and the International Electrotechnical Committee (IEC) standardization program. The Ethernet network protocols utilized in a commercial environment are defined by IEEE 802.3. Specifically, IEEE Std 802.3, 2000 Edition, Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific Requirements – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method And Physical Layer Specifications.

The Internet Engineering Task Force has created Request for Comment (RFC) documents that serve as the commercial standards. Table 1.4.1 lists those that may serve as a convenient resource.

Table 1.4.1 – Useful RFC References

RFC	Title
IETF RFC 768	User Datagram Protocol, Aug-28-1980 (Status: STANDARD)
IETF RFC 793	Transmission Control Protocol, Sep-1981 (Status: STANDARD)
IETF RFC 791	Internet Protocol, Sep-01-1981, (Status: STANDARD)
IETF RFC 959	File Transfer Protocol, Oct-01-1985
	(Updated by RFC2228, RFC2640) (Status: STANDARD)
IETF RFC 1122	Requirements for Internet hosts - communication layers, Oct-01-1989,
	(Status: STANDARD)
IETF RFC 1123	Requirements for Internet hosts - application and support, Oct-01-1989,
	(Updated by RFC2181) (Status: STANDARD)
IETF RFC 1155	Structure and identification of management information for TCP/IP-based internets, May 1990. (Status: STANDARD)
IETF RFC 2616	RFC2616 -- Hypertext Transfer Protocol -- HTTP/1.1 http://www.ietf.org/rfc/rfc2616.txt
IETF RFC 2246	RFC2246 -- The TLS Protocol -- Version 1.0 http://www.ietf.org/rfc/rfc2246.txt

Other documents

Document ref	Title
	Extensible Markup Language (XML) 1.0 (Third Edition) W3C Recommendation 04 February 2004

COMMENTARY

RFCs are never modified or changed, only superseded. The list of which RFCs are current is published in an RFC identified as Standard 1. Standard 1 is contained within the most current RFC. Knowing which RFC contains Standard 1 identifies a well-known and invariant set of Internet Standards that are being used at that point in time (i.e., the current configuration of the Internet).

1.5 Document Precedence

This document incorporates commercial standards by reference. In the case of a conflict between this specification and the applicable ISO, IETF, or IEEE standards, it should be assumed that the deviation is intentional and that this specification should have precedence.

1.6 Regulatory Approval

Compliance with this specification, in and of itself, will NOT ensure regulatory approval. Implementers are urged to obtain all necessary information necessary for regulatory approval and work in close coordination with the appropriate regulatory authorities to gain certification as applicable.

1.7 Compliancy

In order to be ARINC 633 application compliant:

- A system should have implemented all services specified for that application as described in the corresponding chapter of this specification. Note that applications often contain optional services, messages or parameters. In this case, an arinc 633 compliant application should allow the user or administrator to inhibit or enable these optional elements, as long as implementing these optional elements is technically feasible in a given environment (e.g. aircraft design).
- A CMU-like system does not need to use the XML-based message definitions, even when IP networks are used (e.g. In the case of ACARS over IP protocol).
- An EFB-like system does not need to use the ACARS message definitions.
- An airborne system should be able to communicate electronically with ground systems, either through a physical media or through a wireless connection.
- A ground system should be able to communicate electronically with airborne systems or ground systems either through a physical media, through a wireless connection or through a wired connection.
- As a minimum, an airborne system should support at least the transport layer for physical media or the transport layer for wireless IP connections as defined herein, or the airborne system should use ACARS.
- A ground system should support the transport layers for all paths it offers.
- A ground system should support acars based message definitions, if they have been defined for the air-ground link.

In order to be 633 security compliant:

- A system should be able to execute all security features described in this specification for the transport path(s) it offers. A user or system administrator may or may not be able to switch off these features and/or replace them by a non-standardized security feature.

Requirements for compliance with security features for physical transport, IP transport, or ACARS will be added in a future supplement to this specification.

2.0 Message Encoding and Transport

2.1 Message Encoding Rules

2.1.1 ACARS/AOA

This section summarizes the major constraints imposed by ACARS protocols for encoding the messages.

ACARS is a character oriented protocol. Therefore, the following constraint applies:

- Character set restricted to ISO#5 per ARINC 618 Attachment 3. Note that an application may restrict itself to the VHF-Baudot-subset as shown in ARINC 618 Attachment 3.

ACARS messages are limited in size. Therefore, the following constraint applies:

- Message envelope can only contain one content element.
- Message length is limited to 16 ARINC 618 blocks. After subtracting sublabel, a maximum of three additional supplemental addresses and a 4 byte end-to-end CRC, messages should contain a maximum of 3411 uplink application text chars and a maximum of 3267 downlink application text chars.

ACARS message encoding rules:

- The number of parameters is fixed at a pre-determined number.
- Parameters can be defined as fixed length or variable length.
- Parameters can be defined as optional or mandatory. An optional parameter is a parameter that may be omitted, even if it has been defined as fixed length.
- Optional and mandatory parameters of variable length should always be followed by a comma "," delimiter.
- Optional parameters of fixed length should always be followed by a comma "," delimiter.
- Mandatory parameters of fixed length may immediately be followed by the next parameter, if especially defined that way in their ACARS message definition section, otherwise, they are also to be followed by a comma (i.e., ",") delimiter.
- The parameters of the messages will always be transmitted in the order shown in the element format tables (fixed order).
- Missing (optional) parameters are indicated by consecutive commas (i.e., “,,”).

ACARS message encoding rules for legacy systems:

- For compatibility with legacy systems <CR> and <LF> should not appear outside the “FreeText” parameter of uplinks that end in a CMU. <CR> and <LF> may appear anywhere in messages that are forwarded to other end-systems, e.g. EFBs.
- The Generic Message Envelope contains a Free Text parameter that can contain any ISO#5 characters, including <CR>, <LF>, but excluding “,” and “<” or “>” characters.

COMMENTARY

These additional rules are needed for compatibility with legacy systems.

2.1.2 ACARS Over IP

When it is intended to use the ACARS over IP protocol, Section 2.1 message encoding rules should be applied.

COMMENTARY

There is no industry standard for ACARS over IP. This section will be completed when need arise.

2.1.3 XML

This section defines the main rules and conventions applied for creation of XML schemas. It is intended to be used when defining an ARINC 633 XML schema.

2.1.3.1 General: Schema Use

XML messages and data defined in this specification apply XML version 1.0 as defined in “Extensible Markup Language (XML) 1.0 (Third Edition) W3C Recommendation 04 February 2004”

- Schemas are used versus Document Type Definition (DTD)
- When creating a Schema, define the following:
    - Default Element names: qualified
    - Default Attributes names: unqualified
    - Define a target namespace
    - Define an abbreviation for the target namespace

2.1.3.2 Typing & Grouping

- XML Schemas allow to create types for elements
    - A type is suffixed with the "Type" string
    - Example: "MyElementType" is the type for <MyElement/>
- XML Schemas allow to create groups of elements
    - A group is suffixed with the ".Grp" string
    - Example: subFolder.Grp
- XML Schemas allow to create attribute groups
    - An attribute group is suffixed with the ".AttGrp" string
    - Example: document.AttGrp

2.1.3.3 XML Data Naming Rules

Table 2.1.3.3-1 – XML Element Naming Convention

Convention	Examples
If it is a commonly known acronym, all uppercase should be used	<FOB>, </STD>
If it is not an abbreviation: First letter in uppercase, then first letter of all following words in uppercase	<ThisIsANewElement>, </TaxiFuel>
If it is a mixture of both: Abbreviation is uppercase	<ATCFlightPlan>, <ICAOAirportCode>

Table 2.1.3.3-2 – XML Attribute Naming Convention

Convention	Examples
same as for XML Elements, but first letter of attributes should always start in lowercase	<element thisIsAnAttribute="myAttribute"/>

COMMENTARY

ACARS Parameters should generally be named like XML attributes. If a corresponding XML schema exists, in which an ACARS parameter appears as an element, it should still follow the XML attribute naming convention.

2.2 Transport System

This section defines how the messages from applications can be transported over the various available communication systems.

There are three ways today for transferring data to/from aircraft:

1. ACARS communication system
   - Using VHF/VDL2 AOA, Satcom, HFDL media
2. TCP/IP communication system
   - Using e.g. Wifi technologies, Satcom media, Cellular technologies
3. Physical media
   - USB memory stick, CD/DVD

This section addresses all 3 systems.

ACARS system has a very low throughput and therefore imposes strong limitations on the way the data are encoded (see section 2.1) but also on the way they are packaged for transmission. Seen the low throughput available, the overhead for data encoding is designed to be minimized on ACARS (refer to section 4 for details)(this precludes the use of XML).

On the other side, TCP/IP and physical media such as USB memory stick and DVD/CD have less limitation in terms of data volume, but due to the wide usage of IT technologies, there are other issues such as security.

Therefore, specific means for transport of data over these media are defined in this section.

Moreover, it has been decided, for those messages sent over ACARS, to define an optimized specific language and encoding scheme. Another language has been defined for the other media: XML.

However, applying one encoding scheme should not constrain on the medium to be used: for example, it should be possible to send ACARS encoded messages over IP.

If an airborne end system (CMU or EFB) uses UDP or TCP over IP only, then it is expected to use XML for message encoding. If an airborne end system uses ACARS or ACARS over IP, then it may use ACARS for message encoding. However, no general rule can be defined: please refer to the appropriate sections for recommended formats for air-ground and ground-ground exchanges.

2.2.1 Transport Over ACARS Communication System

Messages from application should be sent encoded as specified in the appropriate section of this document (depending on the application: from section 6 to 11).

See appendix B for illustration of how ARINC 633 ACARS messages are structured and the relationship with the other ACARS protocol layers.

2.2.2 Transport Over TCP/IP and Physical Media

New applications are now developed to exchange data with the ground using TCP/IP networks, and physical media such as USB memory stick and DVD/CD.

ACARS format is less and less used. For these new communication systems, security is becoming a bigger issue seen the fact that airborne systems are using COTS and communication protocols widespread, and are therefore more subject to security threats.

COMMENTARY

> Current edition of this specification includes preliminary specification on transport technologies – it defines minimum requirements to be satisfied.

Future work will consist of defining the complete communication stack for full interoperability. System designers, suppliers, and users are thus advised to follow the development of future editions of this specification.

Although it does not specify any security mechanism above the transport services (e.g., HTTPs), implementors may chose to implement security functions between applications and HTTP as well as for transport of files over physical media.

2.2.2.2 Transmission over TCP/IP Networks

2.2.2.2.1 General

Transport of ARINC 633 messages to/from the aircraft is managed by Ground and aircraft communication services. These services support the air-ground communication stack.

The air-ground TCP/IP communication system provides inter-networked communication from aircraft applications to the ground applications located in the airline information system.

The airborne communication function and ground communication function should support HTTPS (HTTP 1.1 over TLS 1.0) to exchange messages and files.

The airborne communication function should behave as a client in HTTPS sessions – this client should poll regularly the server for presence of message to this aircraft.

The ground communication function should behave as a server in HTTPS sessions and manage a repository of messages/files to/from aircraft.

The airborne communication server should initiate all exchanges. GET and POST commands should be supported.

In the future, figures and examples will be added to illustrate how ARINC 633 messages are transferred to the aircraft using TCP/IP/HTTPs.

COMMENTARY

> Today only one technology is required – it is possible that in future editions, several transport technologies be allowed. In this case, the ground communication products should support all of them, while the aircraft systems should select at least one among the list of possible options.

2.3 Physical Media Transportation Rules

The purpose of this section is to standardize how flight related information should be organized, formatted and stored, before it is brought aboard the aircraft or from the aircraft to the ground by a physical media. Due to the characteristics of physical media, it should be noted that:

- No online connection between ground and airborne application available. Therefore, the sender typically does not know whether and when the receiver has received the information correctly.

- The transport may take some time. The information transported over physical media should be "self contained" and not of a "request/response" type.
- The files which reside on physical media may be signed and/or encrypted.

The following physical media may be supported:

- CD
- DVD
- USB memory stick

2.3.1 File System, Naming Structure and Documents Organization

The file system depends on the peripheral used (USB stick, DVD..). See following sections for details.

The Character set to be used for naming directories and filenames should be the following: see parameter Type F in Table 4.1.2.

COMMENTARY

It was decided not to allow the use of lower case character due to the discrepancies between various OS/file systems.

Maximum filename (including the path) length should be 120 characters. Maximum filename should be 64 characters.

This 64 characters limitation for filename is due to Joliet CD system limitation.

2.3.1.1 Folders and Files Structure

Various options are available for transferring files over physical media: activating security or not.

If no security is used, the folders should be present in the root directory of the physical media used.

The names of the folders can be found in the corresponding application chapters.

Data from multiple applications on a single physical device is allowed. If in this case authentication is used, each data package is treated separately.

Alternate solutions for folders and file structures are allowed if security is used.

2.3.2 Media Specificities

2.3.2.1 CD

Any type of CD should be accepted by the receiving system.

The ISO9660 file system and the IEEE standard Rock Ridge extension should be used. Alternately, the Joliet extension may be used.

2.3.2.2 DVD

Any type of DVD should be accepted by the receiving system.

It is recommended to use UDF.

2.3.2.3 PC-Card, USB Stick and Similar

The file system should be FAT32.

2.3.3 Rules for Import/Export of Files

The system should prohibit Autorun, however Autodetect should be allowed.

It is recommended that receiving systems do not erase automatically files read on a physical medium. However, if supported by the physical media, erasing files may be possible as an administrator or user option.

If the airborne application, receiving a data package on a physical media, requires to write on the media, this will be mentioned in the chapter in this specification that describes the application.

3.0 Acars Message Security and Compression

This section introduces security and compression provisions that will be completed in a future supplement – this section is provided for information only.

The aeronautical datalink communications environment is vulnerable to attacks by unauthorized entities that may access or modify AOC messages, the consequence of which may be to jeopardize the safety and integrity of aircraft operations or to expose sensitive information. The primary targets of exploitation are the air-ground data links, which are open RF channels, and the ground-ground networks, which may include both private and public communication networks.

Airlines and aircraft operators recognize the need to protect aeronautical datalink messages to ensure safety of flight, to maintain competitive advantage, to reduce operational liability, and to comply with legislative and/or regulatory requirements for protecting information linked to individuals. Consequently, airlines have indicated their desire for the capability to protect the ARINC 633 messages defined herein from unauthorized disclosure and modification. This chapter sets forth, through reference, the provisions available for the airlines to achieve this objective.

3.1 Airline Operations Considerations

The primary goal of ACARS message security is to provide protection appropriate for AOC messages, with consideration of the systems involved (e.g., legacy issues), end-users needs, and existing airline operations.

Until recently the aircraft has operated autonomously from the airline ground data communications network infrastructure. Datalink connections to the aircraft were made over dedicated, private ACARS links (VHF, HF, and SATCOM) provided by a datalink communication service provider (DSP). Other commercial links provided dedicated passenger services (e.g., telephone, television, and more recently, access to the Internet). However, airlines are investigating new methods (links) to communicate with the aircraft which will increase the use of shared services and further blur the distinctions among onboard functions.

The scope of this current specification is limited to paths supporting ACARS messages. However, as airlines take advantage of new communication links to provide connectivity to the aircraft, security services must also be extended to protect AOC message exchanges via these links.

From an airline operations perspective, it is also important to note that information security solutions are not simply cryptographic algorithms, but rather they include an inter-related set of management, operational and technical security controls, where:

- Management controls focus on processes that are performed by an airline to maintain aircraft information system security to an acceptable level of risk. Examples include specification of a security policy and configuration management of information systems in accordance with the specified security policy.
- Operational controls focus on processes that are performed by people. Examples include training of airline personnel in the proper operation of information security systems and periodic security maintenance (e.g., updating keys and certificates in accordance with airline security policy).
- Technical controls include those mechanisms that are implemented primarily in hardware, software, and firmware. Examples include encryption, message authentication, and data integrity algorithms.

In general, an operational control has one or more associated management controls, and a technical control has one or more associated operational and management controls. Investment in supporting management and operational controls is necessary to achieve effective implementation, deployment, operation, and maintenance of technical controls.

3.2 ACARS Message Security

In May 2005 in response to the security needs identified by the AOC Standardization Subcommittee, the AEEC Data Link Systems (DLK) Subcommittee proposed a work effort to develop ARINC Project Paper 8xx, ACARS Message Security (AMS) Standard. Development of the AMS Standard will be guided by **ARINC Specification 811:** *Commercial Aircraft Information Security Concepts of Operation and Process Framework*, which describes a risk-based information security process that facilitates identification of appropriate cost-effective security controls to reduce risk to an acceptable level.

The AMS Standard addresses confidentiality, data integrity, and message authentication security services, as well as data compression techniques to reduce the transmitted message size and to minimize the effects of security overhead. Application of the AMS Standard provides protection of messages exchanged via the existing ACARS communications infra-structure, independent of the underlying datalink sub-network (e.g., VHF, HF, or SATCOM).

The AMS standard includes policy-based security which permits each airline to apply its organizational security policies regarding protection of AOC messages. Airlines select the messages that require protection and select the security services which may be applied individually or cooperatively (e.g., authentication only or authentication with encryption). It is important to note that all security services are supported by a common underlying key management infrastructure, which is discussed in Section 3.3; consequently, once one security service is implemented by an airline, additional security services are achieved with minimal incremental effort.

The following sections summarize the security services that are included in the AMS standard and that airlines may choose to apply to protect any of the AOC messages defined in this specification.

3.2.1 Data Confidentiality

The AMS data confidentiality (i.e., encryption) security service provides assurance that AOC message content is protected from disclosure to unauthorized entities during transmission. The AMS standard supports multiple encryption algorithms which permits airlines to select the algorithm and level of protection that is consistent with its organizational security policies regarding protection of AOC messages.

3.2.2 Data Integrity and Message Authentication

Data integrity and message authentication security services are treated as a composite security service. The cryptographic mechanisms (e.g., digital signatures and message authentication codes) that permit a recipient to establish that data was generated by an authorized entity also provide assurance that data was not modified. In other words, if data is authentic, then it cannot have been modified; if data was modified, then it cannot be authentic.

3.2.2.1 Data Integrity

The AMS data integrity security service provides both content and sequence integrity.

Content integrity provides assurance that AOC messages are protected from unauthorized modification subsequent to the creation and transmission by an authorized source. Unauthorized modification includes insertion, substitution, or deletion of message content, whether accidental (e.g., errors caused by a noisy transmission channel) or intentional (e.g., errors introduced by an unauthorized entity).

Sequence integrity provides assurance that AOC messages are protected from unauthorized re-sequencing subsequent to the creation and transmission by an authorized source. Unauthorized re-sequencing includes message re-ordering, re-transmission, insertion, or deletion whether accidental (e.g., sequencing error caused by the network) or intentional (e.g., message replay by an unauthorized entity).

COMMENTARY

Non-cryptographic integrity check methods (e.g., CRC) offer protection against accidental modification; however, it is possible for an unauthorized entity to record a message, modify its contents, compute a valid integrity check, and then re-introduce the modified message into the communication system. To counter the potential for intentional modification, the AMS data integrity security service uses strong cryptographic integrity check methods (e.g., message authentication code algorithm) and a cryptographic key that is known only to authorized message senders and recipients.

3.2.2.2 Message Authentication

The AMS message authentication (also known as data origin authentication) security service mitigates spoofing1 threats by providing assurance that the source of AOC messages is as claimed. In other words, message authentication provides aircraft crew or an automated aircraft information system with confidence that messages originate from an authorized source, such as airline operations. Conversely, it provides a dispatcher or automated ground information system with confidence that messages originate from an authorized aircraft information system. This security service is achieved using a combination of digital signatures and message authentication codes.

3.2.3 Data Compression

COMMENTARY

Although data compression is not a security service, coordination with the application of security services is necessary to achieve effective data compression results.

The AMS standard also includes provision for data compression, which is applied to an AOC message prior to the application of any of the security services described in the previous sections. Data compression reduces the size of the original message, which in turn reduces the size of a protected message. As in the case of encryption algorithms, the AMS standard also supports multiple compression algorithms, which permits algorithm selection based on the message type (e.g., character-oriented versus bit-oriented). Note that for any given compression algorithm, the compression ratio is highly dependent on the size of the original message and the content (e.g., larger messages and/or messages with a degree of redundancy compress better than smaller messages and/or messages with random content).

3.3 Public Key Infrastructure

The AMS security services rely on a Public Key Infrastructure (PKI) to provide life-cycle management (i.e., assignment, generation, distribution, installation, maintenance, termination) of private keys, and public key certificates which are necessary for operation of digital signature and key agreement mechanisms.

The PKI employs a trusted entity, known as a Certificate Authority (CA), to issue public key certificates to the aircraft and ground information systems that implement ACARS message security. The CA digitally signs each public key certificate, which binds the identity (and any related attributes) of a unique aircraft or ground information system with its public key. Prior to performing a cryptographic operation that uses the public key contained in a certificate, an aircraft or ground information system first validates the CA digital signature included with a certificate to ensure its authenticity (i.e., the public key

---

[1] Spoofing occurs when a receiver accepts a message that appears to have originated from an authorized source.

included in the certificate is owned by the identified aircraft or ground information system, and it has not been modified during storage or distribution).

COMMENTARY

As recommended in ARINC Report 666, AMS may use the commercial PKI services of Certipath LLC and/or its founding partners – ARINC, ExoStar, and SITA – which compete in the aerospace marketplace to offer digital certificate services that are consistent with certificate policies standardized by the Air Transport Association (ATA) Digital Certificate Working Group (DCWG).

Certipath is a newly organized limited liability corporation (LLC) whose purpose is to design, implement, maintain, and market a secure public key infrastructure bridge, which is expected to be operational in late 2005. Further information is available at *http://www.certipath.com/*.

Optionally, PKI services may be available from other commercial vendors and from airline-internal information technology organizations (i.e., existing airline-operated PKI and CA).

The AEEC Aircraft Information Security (SEC) Subcommittee has identified a significant opportunity for the aeronautical industry to adopt common certificate policies and common certificate formats. These actions will allow multiple system vendors to develop open standard solutions that facilitate compatibility and interoperability among aircraft information systems. SEC efforts in this area are in progress.

4.0 Message Structure

This section defines common rules for structuring, composing and encoding the messages. Three technologies are available for transmission of these messages: ACARS, IP, and physical media. It has been decided, for those messages sent over ACARS, to define an optimized specific language and encoding scheme. Another language has been defined for the other media: XML.

However, applying one encoding scheme should not constrain the medium to be used. For example, it should be possible to send ACARS encoded messages over IP.

If an airborne end system (CMU or EFB) uses UDP or TCP over IP only, then it is expected to use XML for message encoding. If an airborne end system uses ACARS or ACARS over IP, then it may use ACARS for message encoding. However, no general rule can be defined: please refer to the appropriate sections for recommended formats for air-ground and ground-ground exchanges.

Section 4.1 specifies the meta definitions, i.e. the vocabulary that we will use to define parameter types.

Section 4.2 specifies the standard parameter types that are used in message definitions.

Section 4.3 defines the generic element structure (messages are composed of elements).

Section 4.4 defines the generic message structure for ACARS and XML formats. All messages defined in this document (in the following sections) are compliant to this generic message structure.

4.1 Meta-Definitions

Parameters are defined using the following conventions:

Table 4.1.1 – Parameter Length

Code	Range	Example
"integer"	A number directly preceding a meta character defines how many times the meta character should appear and therefore defines how long that part of the parameter is. Note that e.g. 3A has the same meaning as AAA	3A
("min"-"max")	Value range between minimum and maximum value	(3-5)A
(0-n)	Optional parameter of variable length	(0-n)C
V	Variable (1 - max length shown)	

Table 4.1.2 – Parameter Format

Code	Range
A	A..Z
B	Boolean (0, 1)
C	(A..Z)+(0..9)+(.)+(-)+(sp)
DD, DDD	Degrees (00..90) or (000..180)
E	(A..Z)+(a..z)
F	(A..Z)+(0..9)+(.)+(-)+(_)
<hh> or HH	Hours (00..23)
I	A single ISO #5 character between hex 20 and hex 7E, except "," or "/"
J	Hex (0..9) + (A..F)
<mm> or MM	Minutes (00..59)
N	0..9
SS	Seconds (00..59)
T	Tenth of Minutes (0..9)
X	(A..Z)+(0..9)
Q	Sign (+,-)
Y	Compass Direction (N, S)
Z	Compass Direction (E, W)
[]	Optional content. May or may not appear within a parameter

Table 4.1.3 – Other Meta Descriptions

Entry	Interpretation
Underlined characters or characters in "apostrophes"	Constants
<lower case character>	Description of variable text to be displayed or printed

4.2 Parameters Definitions

4.2.1 General

- A parameter is an object that contains data.
- A parameter can be defined directly by the use of meta definitions in chapter 4.1 or indirectly by the use of existing parameter type definitions.
- For the definition of physical values, a parameter type definition should use units typically used in aviation. If multiple units are in use, metric units should be used for communications. Applications may deviate from this general rule.
- Parameter type definitions which are useful for multiple applications are defined in chapter 4.2.2.
- A parameter is referenced by its name. When naming parameters, it is suggested to concatenate the words that build the parameter name and capitalize the first letter of each word, e.g. DepartureAirport.

COMMENTARY

The following conversion rules between metric and non metric units are deemed acceptable – this will have to be checked with the appropriate authorities when operational approval is needed:

- kg ==> lb 1 / 0.45359237
- lb ==> kg 1 x 0.45359237
- l ==> usg 1 x 0.2641721
- usg ==> l 1 x 3.785412
- m ==> in 1 x 39.37008
- in ==> m 1 x 0.0254

The structure of the parameter is defined by the parameter type definition. All parameter types are simple one dimensional character strings of fixed or variable length and defined structure. Parameter content which does not conform to its parameter type is considered invalid data.

COMMENTARY

All mass parameters in this specification have been named as weights, because the term weight is more commonly used than the term mass.

4.2.2 Parameter Type Definitions

The Parameter Type Definition Table defines parameters, which may appear in multiple applications.

Table 4.2.2-1 – Standard ACARS Parameter Type Definition Table

Parameter Type	Parameter Definition	Description	Content Example
dateTime	YYMMDD HHMMSS	full UTC date format (Year, Month, Day, Hour, Minute, Second)	050217235959
date	YYMMDD	date format (Year, Month, Day) 2 digits each	050217
day	DD	day, always in 2 digits	01, 17
time	HHMMSS	UTC time format (Hours, Minutes, Seconds), 2 digits each	235959, 000000
timeMin	HHMM	abbreviated UTC time format (Hours, Minutes), 2 digits each	2359, 1234
boolean	B	1= yes or normal, 0= no, degraded or broken Empty or any other number means not defined	0
string	(0-n)I	any unstructured string	AEEC633
name	(0-50)E	up to 50 alpha characters	Smith
integer	(1-9)N	Integer value between 0 and 999999999.	123

Standard XML Type Definitions:

Note: Standard ACARS and standard XML types may be coded differently, e.g. W3C defines "datetime" as:

"YYYY-MM-DDTHH:MM:SS-0x:00", while in ACARS "dateTime" is defined in this document as "YYMMDDHHMMSS".

Table 4.2.2-2 – Specific ACARS Parameter Type Definition Table

Parameter Type	Parameter Definition	Description	Content Example
flightIdType	(1-10)C	Flight Identifier= ATC call sign –. This parameter is the parameter used when filing the flight plan to the ATC. On the aircraft, it is the parameter entered on the FMS by the crew. This parameter is most of the time compliant to ICAO Doc 4444 Appendix 2 field 7 but there may be local regulations allowing different formats for their national flights.	KLM5K, AFR5556
registType	(5-7)C	Aircraft registration in official format (preferably without leading periods). Checks from receiving systems should be tolerant to the presence of leading periods. Note: hyphens are often used in the official aircraft registration.	N712UA, .F-GGEG D-AIPA, F-WWOW
flightNumType	(1-10)C	Flight number= commercial flight number generally compliant to IATA Flight No	LH012T, BA12345
mass-ttType	(3-4)N	mass in tenth of tons	890
mass-kgType	(1-6)N	mass in kilograms	89015
volume-ltType	(1-6)N	volume in liters	0, 135998
center-of-gravityType	NNN	center of gravity values in tenth of percent of mean aerodynamic chord	324
density-mType	NNN	metric density: 1000 * mass in kilogram / volume in liters	802
provider-idType	AAA	provider ID, e.g. the international standardized fuel supplier codes	SHL
nameType	(0-50)E	up to 50 alpha characters	Smith
labelType	XX	ACARS label	RF
Char3Type	XXX	can be used for service or message ID	SUB
ICAOType	XXXX	ICAO 4-character airport code	LFBO
IATAType	XXX	IATA 3 characters code	TLS
aircraftType	(2-4)X	ICAO A/C type (as per document 4444 appendix 2 (see ICAO doc using the aircraft designator as specified in ICAO 8643) – if no such designator has been assigned, insert ZZZZ.)	B734 (for B737-400)
aircraftSubType	(0-4)C	Additional A/L specific aircraft type definition	

Specific XML Type Definitions: refer to m633common.xsd file in ARINC 633 AOC.zip electronic document.

This file is located in ‘m633common’ sub-directory of schema in ‘ARINC633_AOC.zip’ document.

4.3 Generic Element Structure

4.3.1 General

- An element is an object within a message which forms a logical block.
- Elements consisting of parameters that are not part of the message envelope are called content elements.
- For IP messages, elements are defined using XML Schema version 1.0 syntax
- For ACARS messages, elements are defined using Message Definition Tables.
- An element or an ACARS message is referenced by its name. See the definition of element names below

Example of a XML-attribute coded content element containing 8 out of 13 parameters:

```
<SSSMMM orderFlag="1" acknowledgeFlag="1" finalFlag="1"
actualFlag="1" serviceAirport="" intoPlaneService="BPC"
serviceFlight="" blockFuel="87" taxiFuel="200" tripFuel="3450"
remaining="" aircraftMassUnitDisplay=" " limitMass=""
limitType=""/>
```

Example of the same element, coded for ACARS (no header is shown here):

```
1,1,1,1,,BPC,,87,200,3450,,,
```

4.3.2 Element names

- In an IP environment, the element name appears before and after the contents of the element, as it is usual for XML. Refer to section 4.4.3 for details.
- In ACARS, the element name appears as part of the message envelope before the contents of the element. See Section 4.4.2.
- Elements are referenced by an alphanumeric name. In ACARS, element name consists in serviceId followed by MessageId and version number as specified in Section 4.4.2 (e.g. FDASUB01). In XML, the ServiceId followed by message Id is used to name the element – version number is defined as an attribute of the element.

For some messages, the application needs to be formally identified. The application is identified using the SMI it supports (3 characters). When a given application supports multiple labels, then one SMI supported by this application will be selected as an identifier.

See the following table for list of element and application identifiers.

Table 4.3.2.1 – Application IDs And Element Name Formats

Number of Chars	Parameter Name	Format / parameter type	Example	Note
3	ApplicationID	Char3Type	JRF	
3	ServiceID	Char3Type	FPA	
3	MessageID	Char3Type	SUB	
2	VersionNb	NN	01	

Notes

1. The term "Application" is used for a group of functions (called services) which are typically closely related to each other and which originate in one end-system and end in one end-system. Application Identifiers needs to be unique within this document. For ACARS, the SMI is used as Application Identifier. An end-system like the CMU AOC can consist of multiple applications.
2. A "Service" consists of a flow of one or more messages to fulfill a given function. The service definition consists of element structure definitions and dynamic information giving guidelines about the conditions when the messages are generated and what they trigger when received by the sink. Service Identifiers need to be unique within each application
3. The Messages Identifier (MessageID) identifies the purpose of a message or element within a service. Message Identifiers need to be unique within each service.

4. The **Version Number** identifies the version of the message. All messages used within a service should have the same version number (VersionNb).

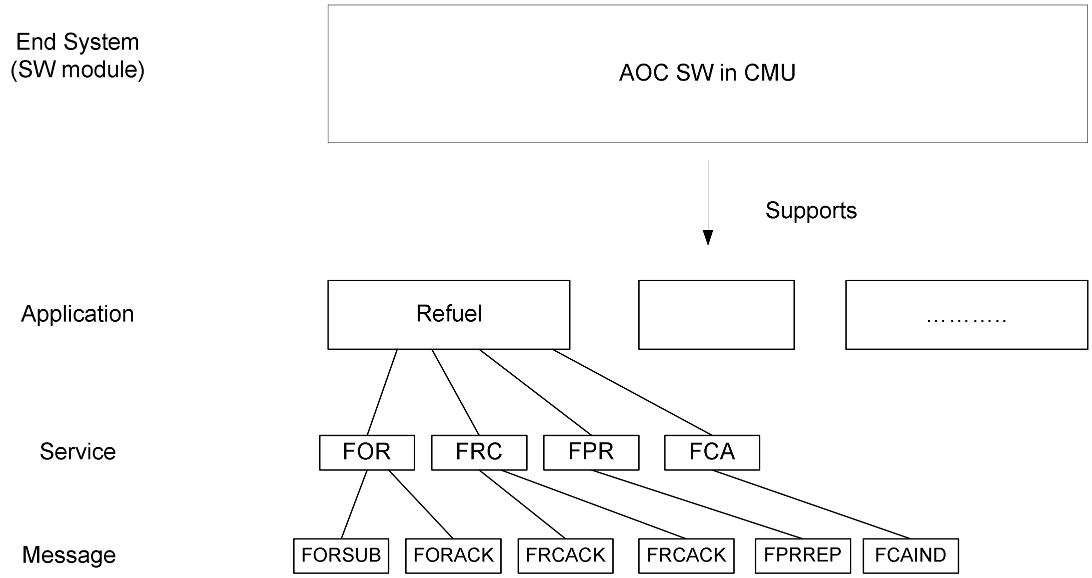


Figure 2.1.1-1 – Illustration of Application-Service-Message Concept

The following table gives some guidelines when defining Message Identifiers:

Table 4.3.2.2 – Message Identifier Table

Message Identifier	Description
IND	Element used to indicate an event
COM	Element used to command an action
REQ	Element used to request parameters
REP	Element used to report parameters
SUB	Element used to submit (insert) parameters
ACK	Element used to acknowledge parameters

4.4 Generic Message Structure

4.4.1 General

- A message is an object which can be transmitted over a communication channel independently from other objects.
- An ARINC 633 Message is a message which follows the message definitions of this document. It consists of one message envelope containing one message header, null or one content elements and one trailer. It can be either coded using ACARS or XML conventions.
- An ACARS message, in the context of this document, is an ARINC 633 message contained in an ARINC 618/620 envelope. An ARINC 633 ACARS message can only contain null or one content element. In an ACARS environment, an "element definition" is a message definition, except that it does not include the 633 message envelope.
- An XML message, in the context of this document, is an ARINC 633 message coded using XML conventions and typically sent over an IP protocol stack or physical media

COMMENTARY

In the first meeting of the work group (FRA, 06DEC04) it has been decided to reserve a block of ACARS message labels in 620 and use separate ACARS labels for separate functions. That means that at least for the time being and for ACARS a message can contain only one content element. However, it has also been decided to repeat the functional meaning in the element itself. In this strawman this is done by writing the element name in front of the list of element parameters.

In the second meeting of the work group (MEM, 16FEB05) it has been decided to use one label per application. An Application can not be spread over multiple end-systems, but an end-system can have multiple applications.

4.4.2 ACARS Message Envelope

4.4.2.1 General

- ACARS messages are defined by the use of two tables: Message Envelope Table and Supplementary Downlink Header Table
- The Message Envelope Table shows everything, except preceding MFI and/or supplementary addresses, which is inserted in the text field portion of an ARINC 620/618 message. The Supplementary Downlink Header and the Content Element are nested within the Message Envelope
- The Supplementary Header Table contains additional header parameters that may be used by an application for downlinks and uplinks
- The ARINC 618 label and optionally sublabel will be specific to each application, for example the label "RF" is used for the Refuel-CG Targeting application.

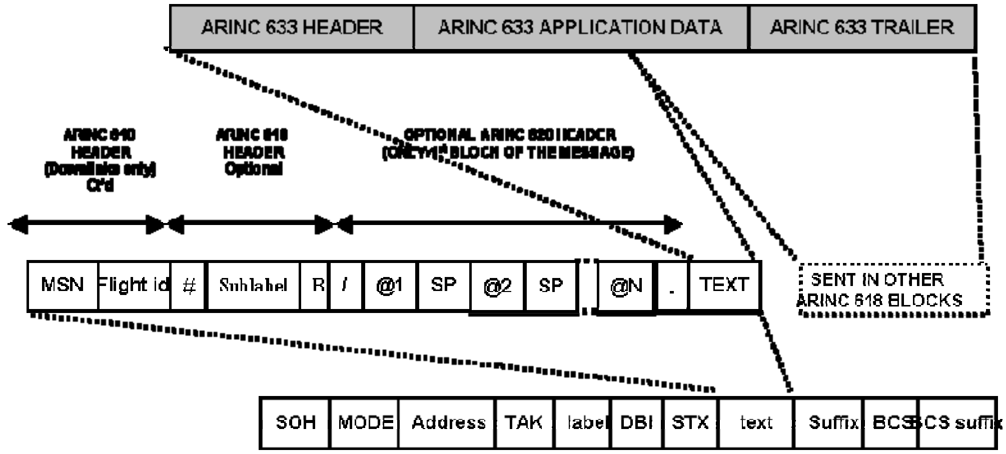
ACARS messages are sent and received by the aircraft using ARINC 618 protocol. ARINC 633 messages are encapsulated with ARINC 620 format, and then sent/received using ARINC 618 protocol. ARINC 618 protocol is outside the scope of the application (managed by the CMU). The present document only encompasses ARINC 633 header, application data and trailer. Specifications ARINC 618 and 620 cover other protocol fields.

The ARINC 633 message fields are inserted as text field of ARINC 620/618 messages – and may be preceded by MFI and/or supplementary addresses.

The ARINC 618 label and optionally sub label and MFI will be specific to each application.

Please refer to ARINC Specifications 618 and 620 for further details on ACARS message format. Below is an example of how an ARINC 633 ACARS Message, composed of a header, application data and trailer, is split and inserted by the CMU in ARINC 618 downlink blocks. Also refer to appendix B for illustration of ARINC Specifications 618, 620 and 633 protocols relationships.

4.0 MESSAGE STRUCTURE



Example of ARINC 618/620/633 protocol encapsulation for Downlinks

NB: @= supplementary address (1st one maybe an MFI)
"SP" stands for space character

4.4.2.2 ACARS Message Envelope Structure

Table 4.4.2.2.1 – ACARS Message Envelope

Size in Char	Parameter Name	Format or parameter type	Example	Option	Note
0-n1	Reserved			see note	1
1	delimiter	comma	,	No	
3	serviceID	Char3Type	FPA	No	5
3	messageID	Char3Type	SUB	No	5
2	versionNb	NN	01	No	5
12	headerTimestamp	datetime	05021711 5932	No	
0-n2	Supplementary Header Element	see table below	see example	Yes	3
1	ContentElementStartCharacter	slash	/	No	6
0-n3	Message Content Element	See messages definition		Yes	
1	Parameter Delimiter	comma	,	No	
1	FreeTextStartCharacter	slash	/ Free Text	see note	2
0-n4	FreeText	(0-n)I		Yes	
4	Application CRC (16 bit)	JJJJ	12AB	No	4

Notes:

1. This field is unassigned and available for future use.
2. The FreeTextStartCharacter is only needed if followed by FreeText.
3. Applications may or may not use this element. However, if an individual ACARS message definition mandates it, it is not optional.
4. CRC is mandatory for ACARS encoded ARINC 633 messages, the same calculation as used in ARINC 622, and covers all ARINC 633 fields of the message envelope table, except the CRC itself.

5. As defined in chapter 4.3.2, the element name, which split into application identifier (label), service identifier, message identifier and version number, is part of the Generic Message Structure in ACARS encoded messages and is part of the content element in IP encoded messages.
6. In order to stay compatible with possible future header changes, it is recommended that applications check for this character instead of counting the elements in the header in order to identify the start of the content element.

Table 4.4.2.2.2 – ACARS Supplementary Header Table

Size in Char	Parameter Name	Format or parameter type	Example	Option	Note
4	depICAOCd	ICAOType	LFPO	Yes	1, 2
1	delimiter	comma	,	No	
3	depIATACd	IATAType	CDG	Yes	2
1	delimiter	comma	,	No	
4	destICAOCd	ICAOType	LFBO	Yes	1, 2
1	delimiter	comma	,	No	
3	destIATACd	IATAType	TLS	Yes	2
1	delimiter	comma	,	No	
1-7	flightID	flightIdType	AFR5556	Yes	3, 4
1	delimiter	comma	,	No	
1-8	flightNum	flightNumType	AF5556	Yes	3, 4
1	delimiter	comma	,	No	
6	departureDate	Date	050217	No	
1	delimiter	comma	,	No	
4	departureTime	timeMin	1235	Yes	5
1	delimiter	comma	,	No	
3-4	acType	aircraftType	A380	No	
1	delimiter	comma	,	No	
3-4	acSubType	aircraftSubType	800	Yes	
1	delimiter	comma	,	No	
5-7	regist	RegistType	F-WWOW	No	
1	delimiter	comma	,	No	
0-n	Reserved			Yes	

The supplementary header fields should be used at least to associate the data to a given aircraft and flight. Some airborne applications may decide to check if the messages are intended to the right entity. In this case, it is advised to use primarily the aircraft registration (flightID and flightNum format can vary).

COMMENTARY

Supplementary header structure is designed to be consistent with XML supplementary header. Some differences exist but they only apply to parameter type definition. For example, IATA airport code is defined as ‘XXX’ in ACARS while it is a 3 characters string in XML.

Notes

1. Per ICAO Document 4444.
2. Destination and Departure airport codes are mandatory, but ICAO or IATA format can be used: ICAO code only, IATA code only, or both are allowed.
3. At least one identifier for the flight is required, but either Flight Id or Flight number can be used, or both.
4. The flight id or flight num used should designate the flight to which the message is applicable, that may be different from the current flight (for example next flight).
5. May be STD (Scheduled Time of Departure) or EOBT (Estimated OffBlock Time). This parameter is optional and should not be used for identification of flight (automatic matching of received messages with local flight information).

4.4.2.3 Examples

ACARS encoded uplink example with supplementary header (excluding ARINC 620 header and label)

```
,FRCSUB01050217065159
LFPG,,KJFK,,AFR5556,,050217,,A380,,F-WWOW,
/CAFS,BPC,H10556,1,8522A,58100,LT,802,45596,KG,,,xxxx
```

xxxx is the CRC

Notes:

1. The "633 part" of an ARINC 633 ACARS message is contained in the "free text" section of a 618/619/620 message. The samples show only this "633 part", the 618/619/620 header and trailer are not shown. Therefore, the ACARS address (e.g. the aircraft registration) and the label or SMI is not shown in the examples. For informational purposes, appendix B contains examples including the full 618/619/620 header and trailer.
2. The 633 part of an ARINC 633 ACARS messages does not contain control characters like carriage return, line feed or similar. If sample messages are split in separate lines, this is done for better readability only.
3. For the sample messages, the CRC has not been calculated. Therefore, it is added as "xxxx".

4.4.3 XML Message Definition

4.4.3.1 General

XML messages are defined using XML Schemas version 1.0 as defined by W3C.

4.4.3.2 XML Message Envelope Structure

Message Envelope XML Schema

XML message structure should be the same for uplinks and downlinks.

The root element of an XML encoded message should be named as follows:

```
<xs:element name="XXXYYY">
```

where XXX is the ServiceId (3 characters) followed by YYY, the MessageId ident (3 characters).

The message should also include the mandatory header

```
<xs:element ref="M633Header"/>
```

If needed, the message should include the supplementary header as follows:

```
<xs:element ref="M633SuppHeader" minOccurs="0"/>
```

or

```
<xs:element ref="M633SuppHeader"/>
```

See electronic document ‘ARINC633 AOC.zip’ for details and examples. Headers are defined in schema/m633headers/m633headers.xsd.

Depending on the application, and especially if there is a user involved in the message processing, it may be useful to add free text.

It is recommended that the free text be included as a specific element (string) under the service+message element definition. It is recommended to use the element name ‘FreeText’.

4.4.3.3 Examples

Hereafter is an example of how an XML definition of a set of messages looks like. This definition is embedded in a schema. There is one schema per application (here it is REFUELING). This schema also refers to the M633 headers definition (in a separate schema, called m633headers). The schema below defines several messages (please note that the refueling application (section 6.3) supports more than these 3 messages. The schema below is provided as EXAMPLE ONLY: it is not complete – please refer to the Electronic Document for complete definition of fuel XML schemas.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema targetNamespace="http://aeec.arinc.net/633"
   xmlns="http://aeec.arinc.net/633"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified">
<!--
   ***************************************************************
   ***************-->
  <xs:include schemaLocation="../m633headers/m633headers.xsd" />
<!-- ***************************************************************-->
<!--  FCA (Fuel CG Advisory) messages Definition-->
<!--
   ***************************************************************
   * -->
- <xs:element name="FCAIND">
   - <xs:complexType>
      - <xs:sequence>
           <xs:element ref="M633Header" />
           <xs:element ref="M633SuppHeader" />
           <xs:element name="ForwardTOCGLimit"
            type="xs:nonNegativeInteger" />
           <xs:element name="AftTOCGLimit"
            type="xs:nonNegativeInteger" />
           <xs:element name="CalculatedTOCG"
            type="xs:nonNegativeInteger" />
```

```
          <xs:element name="TOW"
           type="xs:nonNegativeInteger" />
          <xs:element name="TaxiFuel"
           type="xs:nonNegativeInteger" />
  </xs:sequence>
  </xs:complexType>
  </xs:element>
<!--
   *************************************************************
   ************************** -->
<!--   FRC (Fuel ReCeipt) messages Definition-->
<!--
   *************************************************************
   ************************-->
<xs:element name="FRCSUB">
   <xs:complexType>
     <xs:sequence>
          <xs:element ref="M633Header" />
          <xs:element ref="M633SuppHeader" />
          <xs:element name="IntoPlaneService"
           type="xs:string" />
          <xs:element name="Defuel" type="xs:boolean"
           minOccurs="0" />
          <xs:element name="FuelReceiptNumber"
           type="xs:string" />
          <xs:element name="SuppliedVolume"
           type="xs:nonNegativeInteger" />
          <xs:element name="SuppliedDensity"
           type="xs:nonNegativeInteger" minOccurs="0" />
          <xs:element name="SuppliedMass"
           type="xs:nonNegativeInteger" minOccurs="0" />
          <xs:element name="AircraftVolumeUnitDisplay"
           type="xs:string" minOccurs="0" />
       </xs:sequence>
     </xs:complexType>
  </xs:element>
<!--
   *************************************************************
   **************-->
 <xs:element name="FORACK">
   <xs:complexType>
     <xs:sequence>
             <xs:element ref="M633Header" />
             <xs:element ref="M633SuppHeader" />
             <xs:element name="OrderFlag" type="xs:boolean"
              />
             <xs:element name="AcknowledgeFlag"
              type="xs:boolean" />
             <xs:element name="FinalFlag" type="xs:boolean"
              />
             <xs:element name="ActualFlag"
              type="xs:boolean" />
             <xs:element name="ServiceAirport"
              type="xs:string" minOccurs="0" />
             <xs:element name="IntoPlaneService"
              type="xs:string" minOccurs="0" />
             <xs:element name="ServiceFlight"
              type="xs:string" minOccurs="0" />
             <xs:element name="BlockFuel"
              type="xs:nonNegativeInteger" />
             <xs:element name="TaxiFuel"
              type="xs:nonNegativeInteger" minOccurs="0" />
```

```
            <xs:element name="TripFuel"
             type="xs:nonNegativeInteger" minOccurs="0" />
            <xs:element name="Remaining"
             type="xs:nonNegativeInteger" minOccurs="0" />
            <xs:element name="AircraftMassUnitDisplay"
             type="xs:string" minOccurs="0" />
            <xs:element name="LimitMass"
             type="xs:nonNegativeInteger" minOccurs="0" />
            <xs:element name="LimitType" type="xs:string"
             minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
    </xs:element>
</xs:schema>
```

5.0 Common Elements

Application Name	Not Applicable
ACARS Label	corresponds to the application in which the element is used
Introduced	2005
Purpose	Common elements are elements that may be used in multiple applications, e.g. error or caution elements.
Stake Holder	all
Airborne System	all
Ground Systems	all
Network	ACARS and IP
Downlink Routing	

Element List

Element Name	Element ID	Service Version	DL/UL	Purpose
General Error Indication	GERIND	01	DL/UL	General error service

Chapter 5 defines how complex data sets used by various applications should be structured and encoded. It also defines common communication services, such as general error services.

The data addressed are generally complex and structured data, gathering strongly related parameters. For example, Flight plan is considered as a global data set. Weather, NOTAMs, are also separate data sets.

Definitions are essentially done using XML. The definitions are expected to be used/included in individual application schemas.

Thanks to the strength of XML, it is possible to dissociate data (XML instance) and the representation of the content of the data (XML Stylesheets).

The purpose here is not to standardize the way the data are to be displayed, but to standardize the naming and organization of some elements in order to easily find information in an XML instance of the data.

See electronic document 'ARINC633 AOC.zip' for data definition, details and examples. Schemas are defined in folder schema/CommonData.

5.1 Important parameters used in common Elements

TBD

5.2 Common communication services

5.2.1 General Error Service

5.2.1.1 General Error Indication Element Structure

Table 5.2.1.1 – GERIND01 ACARS Definition

Parameter Name	Meaning	Contents		Option	Range	Resolution	Unit	Example
		Fixed Text	Parameter Type					
ErroneousSMI	SMI of the application that caused the generation of this error element		Char3TypeType	YES				JRE
parameter separator		comma		NO				,
ErroneusService	Service that caused the generation of this error element		Char3Type	NO				FOR
ErroneusElement	Element which, upon reception, has triggered the generation of this error element		Char3Type	NO				SUB
ErroneusVersion	Version of the element which, upon reception, has triggered the generation of this error element		NN	NO				01
parameter separator		comma		NO				,
ErrorClass	see separate table		integer	YES	0...3			
parameter separator		comma		NO				,
ErrorType	see separate table		integer	YES	0...99			
parameter separator		comma		NO				,
ErrorData	to be interpreted by application		(0-n)I	YES				Message Undeliverable
parameter separator		comma		NO				,
TryAgain	Missing: not specified 0: Receiving application should not try again 1: Receiving application may retry to send the element that caused the error		boolean	YES				0

General Error XML Schema:

See electronic document ‘ARINC633 AOC.zip’ for details and examples. General errors formats are defined in folder schema/m633common/GeneralErrorServive/ERR.xsd.

5.2.1.2 Error Classes

Table 5.2.1.2 – Error Classes

Sender's Situation	Class Name	Class ID	Action to be taken by receiver	Suggested text to user (e.g. flightcrew)	Comment
not specified	undefined class	<nil> or 0	not specified	Undefined Error	
Upon reception of an element that caused an error, the transaction had to be aborted	fatal error	1	abort transaction and alert user	Transaction aborted	
The contents of a received element had been ignored, the transaction stays open	non-fatal error	2	inform user about the failure of last message	Message rejected	
The contents of a received element had been processed, but look strange	caution	3	inform user about the caution	Message accepted	

5.2.1.3 Error Types

Note: The following table defines standard error types, common to most applications (from 0 to 19) – application-specific error types are also possible, and they range from 20 to 99. Refer to individual application sections for more details.

Table 5.2.1.3 – Error Types

Sender's Situation	Type Name	Type ID	Suggested text to user (e.g. flightcrew)	Comment
not specified	unknown-error	0	Unknown Error	
Message was received too late	time-out	1	Time Out	
Message was received too early or out of order	flow	2	Too early or out of order	
Received message contains unknown syntax	syntax	3	Syntax Error	
Message triggered an internal error of receiving application	internal	4	Internal Error	
Message received uses a service version not supported	service	5	Service version not supported	
Message received contains a service or message ID not known	unknown-service	6	Unknown service	
Mandatory parameters not available	missing-data	7	Parameters mandatory to generate/process messages are not available or are invalid	
Invalid CRC	CRC	8	Checksum error	
Message related to a service which is temporarily not available	temporarily-unavailable	9	Service Temporarily Unavailable	
Message related to a service which is not available	unavailable	10	Service Unavailable	
Received message could not be delivered to the right end system	undeliverable	11	Message Undeliverable	

5.2.1.4 ACARS encoded example (excluding ARINC 620 header)

```
GERIND01 downlink:

,GERIND01050217065159KLAX,LAX,LFPG,CDG,AF1234,,051217,,A380,,F-
WWOW,
/JRE,FORSUB01,1,8,F3A2,1,xxxx
```

xxxx is the CRC

5.2.1.5 XML encoded example (GERIND Example 1.xml)


```
<?xml version="1.0" encoding="UTF-8" ?>
<GERIND xmlns="http://aeec.arinc.net/633"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://aeec.arinc.net/633
../schema/m633common/GeneralErrorService/ERR.xsd">
<M633Header versionNb="1" timestamp="2005-12-17T09:30:47-05:00"
/>
<M633SuppHeader>
<Flight departureTime="14:20:00+02:00" departureDate="2007-08-13">
         <FlightNumber>
            <FlightNum>AP0487</FlightNum>
         </FlightNumber>
         <Dep>
            <ICAOCd>CDGX</ICAOCd>
         </Dep>
         <Dest>
            <ICAOCd>JFKX</ICAOCd>
         </Dest>
      </Flight>
        <Aircraft regist="F-WWOW">
            <ACModel acType="A380"/>
        </Aircraft>
 </M633SuppHeader>
 <Error erroneousSMI="JRE" erroneousService="FOR"
erroneousElement="SUB" erroneousVersion="1" errorClass="1"
errorType="8" errorData="F3A2" tryAgain="1" />
</GERIND>
```


5.3 Flight Info

To be added in a future supplement.

5.4 Operational Flight Plan

5.4.1 Introduction

The XML Schema of the operational flight plan discussed in this section follows the requirements of JAR-Ops 1.1060 and the needs of major airlines. As usual with XML, the presentation depends on a stylesheet and can be freely chosen by the airlines.

In fact, the sending application does not even need to know the ofhow the datais presented. Also, by standardizing the naming and organization of the contained elements, the data becomes accessible by a number of applications without human intervention.

In cases where an airline needs to add elements not contained in FlightPlan.xsd, the schema contains attachpoints usually named "##other" where such elements can be added. However, it should be noted, that standard EFB applications will usually not be able to process elements added here, as long as the ARINC 633 schema has not been added.

See electronic document ‘ARINC633 AOC.zip’ for flight plan data definition, details and examples. Schemas are defined in folder schema/CommonData/FlightPlan/FlightPlan.xsd.

5.4.2 General Organization Of The Schema

An operational flight plan should be organized as follows:

- M633 Headers
- A flight information element (giving additional information that are not available in M633headers such as ATCCallSign for the flight)

- A general header (fpheader)
- A fuel header information
- A weight header information
- Waypoint information
- An alternate flight plan (if required)
- A flight plan summary

5.4.3 M633 Header

M633 headers are inserted in the flight plan mainly to identify the flight in terms of:

- ATC call sign (flightId))
- Commercial Flight Number (flightNum)
- Departure Date
- Departure Airport
- Arrival Airport

5.4.4 General header (FPHeader)

The general header is intended to store general information for the flight. Refer to the annex for a complete description of all identified elements.

It should mainly identify:

- The OutTtime (i.e. ‘commercial’ departure time)
- The dispatcher name (as per JAR regulations)
- The list of alternates airports
- Performance factor
- Average wind
- Average temperature
- Cost index or speed
- CostIndex : Cost index. When no value or element not available, it shall be assumed as ‘Variable’
- Speed : Speed constraint : When no value or element not available, it shall be assumed as ‘Variable’
- Mach : Mach constraint : When no value or element not available, it shall be assumed as ‘Variable’

5.4.5 Fuel header

The fuel header is intended to store general information regarding fuel. The XML element name should follow the here above rules.

It should mainly identify the following information:

- Trip: Trip fuel associated with the following information:
    - EFuel (Estimated Fuel for reaching destination airport)
    - AFuel (Actual fuel used for reaching destination airport)
    - ETime (Estimated flight time for EFuel)
    - Destination Airport code
    - NAM (Nautical Air Distance for reaching destination airport)
    - NM (Ground distance to destination airport)
    - FL (Flight Level planned for the cruise)

- Contingency: reserve fuel
  - EFuel (Estimated Fuel for reaching destination airport)
  - AFuel (Actual fuel used for reaching destination airport)
  - ETime (Estimated flight time for EFuel)
  - COEF (ratio of fuel used for reserve compared to Trip.EFuel)
- Alternates: List of fuel values for alternate airports
  (for each alternate airport)
  - EFuel (Estimated fuel for reaching alternate airport)
  - AFuel (Actual fuel used for reaching alternate airport)
  - ETime (Estimated flight time for EFuel)
  - Alternate Airport code
  - NAM (Nautical Air Distance for reaching alternate airport)
  - NM (Ground distance to alternate airport)
  - FL (Flight Level planned for the cruise)
- FinalReserve: Holding reserve
  - EFuel (fuel to fly for 30 minutes at holding speed at 1500 ft above aerodrome elevation) See Note 1.
  - AFuel (Actual fuel used for reaching destination airport)
  - ETime (Estimated flight time for EFuel)
- ETOPS: ETOPS reserve
  - EFuel (Estimated fuel for reaching to other airport)
  - AFuel (Actual fuel used for reaching to other airport)
  - ETime (Estimated flight time for EFuel)
  - COEF (ratio of fuel used for reserve compared to Trip.EFuel)
- Additional: Additional reserve
  - EFuel (Estimated fuel in reserve)
  - AFuel (Actual fuel in reserve)
  - ETime (Estimated flight time for EFuel)
  - CAPTCheck (Captain identification)
- EXTRA: Extra fuel confirmed by captain
  - EFuel (Estimated Extra Fuel)
  - AFuel (Actual Extra fuel)
  - ETime (Estimated flight time for EFuel)
- TOF: Take off fuel
  - EFuel (Estimated Fuel for take off)
  - AFuel (Actual fuel used for take off)
  - ETime (Estimated take off time for EFUel)
- Taxi: Taxi fuel
  - EFuel (Estimated Fuel for Taxi)
  - AFuel (Actual fuel used for Taxi)
  - ETime (Estimated taxiing time for EFuel)
- Block: Block fuel
  - EFuel (Estimated Fuel necessary for the complete flight)
  - AFuel (Actual fuel used for the complete flight)
  - ETime (Estimated flight time for EFuel)

  - EndOfFuelingFOB (Actual Fuel in tank after final refuelling)
- TransportCoef: Transport coefficient
  - This value represents the additional amount of fuel to accommodate weight above the zero fuel weight due to extra load. It has been used to compute the fuel estimation of the flight.

Note 1: Per JAR OPS 1.255 1.5 (for turbine powered airplanes).

5.4.6 Weight Header

The weight header should store weigh information. Each weight information is split in two values:

- The estimated weight
- The corrected weight (normally entered by the pilot)

The weight header should mainly identify the following information:

- BASICWT : Basic weight
- EPLD : Payload
- EBO : Estimated burn off

There is also some weight information that is associated with the following properties:

- An operational limit
- A structural limit
- A text describing the reason for the operational limit.

This concerns the following information:

- EZFW : Estimated zero fuel weight
- ETOW : Estimated take off weight
- ELAW : Estimated landing weight

5.4.7 Waypoints

Waypoints are used to identify the operational flight plan. Waypoints is a list of waypoint.

Each waypoint should identify the following information:

Element	Attribute	Definition
Waypoint	@id	Identifier of the waypoint (this shall be unique in the whole FlightPlan document)
	@passed	To mention that the waypoint has been actually flown
	@update_requested	Intends to store information to know if the pilot requested an update of this waypoint information from avionics data or not.
	@creation_method	Intends to say whether the waypoint is coming from a flight planning system (automatic) or if it comes from a data manually entered by the pilot (manual)
	@deleted	Used when the pilots wants to DIRTO. Intermediate waypoints are still present in the operational flight plan but marked as deleted
WPT		Name of the waypoint
COORD		Waypoints coordinates composed of: LAT : Waypoint Latitude LONG: Waypoint Longitude
AWY		Airway
FL		Flight Level
AFL		Actual Flight Level
MSA		Minimum safe altitude
Wind		Wind at that waypoint composed of

Element	Attribute	Definition
		- Direction - Speed
AWind		Actual Wind
OAT		Outside air temperature
Shear		Shear factor
DISA		Delta isa temperature
TAS		True air speed
MA		Speed in mach
GS		Ground speed
OTT		Outbound true track
ITT		Inbound true track
OMT		Outbound magnetic track
IMT		Inbound magnetic track
DST		Ground distance
RDST		Remaining ground distance
NAM		Air distance
RNAM		Remaining air distance
ET		Elapsed time
AT		Actual time
CT		Cumulated time
ETO		Estimated Time Over a Waypoint
ATO		Actual Time Over a waypoint
ERT		Estimated remaining time
RERT		Revised Estimated remaining time
EBO		Estimated burn off
ABO		Actual burn off
ECBO		Estimated cumulated burn off
ACBO		Actual cumulated burn off
EFOB		Estimated fuel on board
AFOB		Actual fuel on board
EACWT		Estimated aircraft weight
MFOB		Minimum fuel on board
ELEV		Elevation (for origin and destination)
TROP		Tropopause
DIRTO	@wptID	Identifier of the waypoint where to DIRTO from this waypoint.
Comment		Text comment associated to the pilot.

5.4.8 FP Summary

Gathers global parameters of the flight (calculated before flight and updated during flight).

Element	Attribute	Definition
EOffTime		Estimated Time Off (Take Off)
OffTime		Time Off (Take Off)
ETA		Estimated Time of arrival – generally calculated on ground before flight
RETA		Revised Estimated Time of arrival – ETA revised during the flight
OnTime		Time On (Land On)
EInTime		Estimated Time In (arrival at gate)
InTime		Time In (arrival at gate)
BlockTime		See appendix D
FlightTime		See appendix D
LW	@unit	Landing Weight
TOW	@unit	Take Off weight
EngineFuel	@unit	Fuel consumption for each engine

5.5 ATC Flight Plan

To be added in a future supplement.

5.6 Airport Weather

This section contains guidelines for interim implementations and will be revisited when Eurocae WG44 and WG53 will have defined OSED and standardized a format for these data.

See electronic document ‘ARINC633 AOC.zip’ for airport weather data definition, details and examples. Schemas are defined in folder schema/CommonData/AirportWeather/WeatherText.xsd

5.6.1 Airport Text Weather

Example of an Airport Weather xml instance

```
<WeatherText GenDate="20040630" GenTime="112359" FlightDate="20040630"
FlightID="AIB04">
<airport type="DEPT" IATACd="DUS" ICAOCd="EDDL" name="DUESSELDORF">
<weather type="SA " genDayTime="2005-12-17T09:30:47-05:00"
validDayTime="2005-12-17T09:30:47-05:00">
<wxdescr timeIntvlDescr="101206" wind="22005G12KT" vis="9999" rvr=""
rwycond="" presentWX="" oat="" dew="" qnh="" qfe="" supplInfo="">
         <clouds amount="FEW" type="CB" height="002"/>
         <clouds amount="SCT" height="080"/>
      </wxdescr>
<wxdescr timeIntvlDescr="101600"  wind="" vis="8000" rvr="" rwycond="">
         <clouds amount="SCT" height="012"/>
      </wxdescr>
   </weather>
   </airport>
</WeatherText>
```

5.7 SIGMET

To be added in a future supplement

5.8 Weather Chart

To be added in a future supplement

5.9 NOTAM

Editor’s Note: This section contains guidelines for interim implementations and will be revisited when Eurocae WG44 and WG53 will have defined OSED and standardized a format for these data.

See electronic document ‘ARINC633 AOC.zip’ for NOTAM data definition, details and examples. Schemas are defined in folder schema/CommonData/Notam/Notam.xsd

5.10 NOTOC

To be added in a future supplement

5.11 Load and Trim Sheet

To be added in a future supplement

5.12 Crew List

Editor’s Note: There is no schema definition for Crew list data. The example below is here to illustrate what could be crew list data. Further work is needed on this topic.

To be added in a future supplement

5.13 Pax list

To be added in a future supplement

5.14 Oceanic Track

To be added in a future supplement

5.15 Navigational Chart

To be added in a future supplement

5.16 Airport Information

To be added in a future supplement

5.17 Security Information

To be added in a future supplement

5.18 Airline Defined

To be added in a future supplement

6.0 Fuel

6.1 Introduction

Refuel applications are used to support the aircraft's refuel processes. The following applications have been defined:

Chapter	Application	SMI	Label	Description
6.2	CG Targeting	JRF	RF	On “CG targeting” capable aircraft such as the Airbus A380, the refueling of the aircraft does not support a predefined, static fuel distribution like “wings, center, trim”, but a dynamic fuel distribution that targets a certain take off center of gravity (TOCG). The entry parameters for this algorithm are ZFW, ZFCG and PFQ (Pre-selected fuel quantity).
6.3	Refueling	JRE	RE	Application supports the exchange of refueling related data between aircraft and providers like fueling into-plane-services or ground ops companies.

Common XML FuelTypes: See electronic document ‘ARINC633 AOC.zip’ for details and examples. Common fuel types are defined in the folder schema/FUEL/ fuelcommon.xsd.

6.2 Aircraft Refueling - CG Targeting

ZFW, ZFCG are usually calculated by a ground weight & balance system and PFQ is usually calculated by a ground flight planning system. All 3 parameters can be updated during and after the refuel process. To simplify the refueling task, the fuel quantity monitoring system may be controlled by ACARS messages.

This section specifies the messages to be supported for remote control of ‘CG targeting’ capable Fuel Quantity Monitoring airborne systems.

Fuel Distribution operation for CG targeting uses the following datalink services:

1. FUEL STATUS: the ground operations center can request a fuel status report indicating current refuel parameters . This service is named ‘FST’ (Fuel STatus).
2. FUEL DATA SUBMISSION: the ground operation center can initialize or update the refueling system with new values before refueling. This service is named ‘FDA’ (Fuel DAta submission).
3. AUTOMATIC REFUEL END: the aircraft fuel system informs the ground operation center that refueling have completed normally. This service is named FEN (reFuel ENd).
4. AUTO GROUND TRANSFER START: the aircraft fuel system informs the ground operation center that an auto ground transfer has started. This service is named FTB (Fuel auto ground Transfer Begins).
5. AUTO GROUND TRANSFER END: the aircraft fuel system informs the ground operation center that auto ground transfer has completed normally . This service is named FTE (Fuel auto ground Transfer End).
6. AUTO GROUND TRANSFER INTERRUPTION: the aircraft fuel system informs the ground operation center that auto ground transfer was prematurely interrupted. has failed to complete successfully for any reason (for example, due to crew deselection, due to a failure…) This service is named FTI (Fuel auto ground Transfer Interrupted).

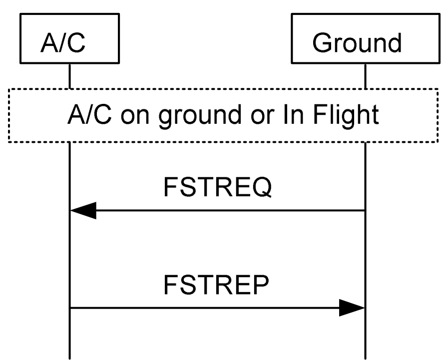
In addition to the 6 above operational services, an error service is defined to handle indication of errors to the ground.

These steps may require one or more of the following and any number of common elements to be sent. The following table shows the fuel elements defined in this chapter:

Table 6.2 – CG-Targeting Element Table

Element Name	ACARS Label	Service Version	DL/UL	Purpose
FSTREQ	RF	1	UL	**Fuel-STatus-REQuest** Requests a fuel status report downlink
FSTREP	RF	1	DL	**Fuel-STatus-REPort** Informs the ground about the aircraft fuel status
FDASUB	RF	1	UL	**Fuel-DAta submission-SUBmit :** Submits parameters to an aircraft's refueling system
FDAACK	RF	1	DL	**Fuel-DAta submission-ACK** Mirrors submitted parameters to the ground for confirmation
FDACOM	RF	1	UL	**Fuel-DAta submission-COMmand** Activates submitted parameters after fuel-data -ack has been received
FENIND	RF	1	DL	**reFuel ENd INDication** Informs the ground of normal completion of refueling process
FTIIND	RF	1	DL	**Fuel Transfer Interrupted INDication** is sent by the airborne system when automatic fuel ground transfer has not completed successfully for any reason (for example, due to crew deselection or abort due to a failure…)
FTEIND	RF	1	DL	**Fuel auto ground Transfer End INDication** Informs the ground of normal completion of fuel transfer process
FTBIND	RF	1	DL	**Fuel-Transfer-Begins-INDication** FTBIND is sent by the airborne system when an automatic ground transfer process starts.
FERIND	RF	1	DL	**Fuel-Error INDication** Indicates an error for fuel datalink services

6.2.1 Fuel-Status



Typical Fuel Status Exchange

The fuel-status element is used to inform ground systems about the fuel situation onboard, e.g. to transmit fuel figures to a ground based weight & balance system.

Following subsections define version 1 of FST service.

Operational parameters ranges and resolution in all messages will vary depending on aircraft type. Therefore, the airline and ground operator should ensure that only messages with ranges and resolutions fitting the aircraft type will be used.

6.2.1.1 Fuel-status-request Uplink (FSTREQ)

Element Name: **FSTREQ**

The Fuel-status-request uplink should contain no parameters – and thus only contain the header.

The fuel status request uplink may contain one or several optional supplementary address(es) (as per ARINC 620).

6.2.1.2 Fuel-status-report Downlink (FSTREP)

Element Name: **FSTREP**

The fuel status report downlink may contain one or several optional supplementary address(es) (as per ARINC 620), which will be the copy of the supplementary address(es) (if present) in the associated FSTREQ.

6.2.1.3 Dynamic Aspects

FSTREP should only be sent in response to FSTREQ.

The airborne application should also perform the actions specified in the following table. The standard application state is “listening for ACARS messages” (called “listening mode”).

Table 6.2.1.3.1 – Airborne Processing Of Non Nominal Conditions

Aircraft condition	Event	Aircraft action	Comment
Fuel system degraded mode or airborne application internal error	FSTREQ received	Send FERIND with appropriate error code and return to listening mode	
ANY	Erroneous FSTREQ received	Send FERIND with appropriate error code and return to listening mode	
ANY	FSTREQ received with a non supported version number	Send FERIND with appropriate error code 5 and supported version numbers and return to listening mode	

The ground application should also perform the actions specified in the following table.

Table 6.2.1.3.1.2 – Ground Processing Of Non Nominal Conditions

Ground Condition	Event	Ground action	Comment
FSTREQ was sent – wait for FSTREP	Receive FERIND with error code 5 (and list of supported versions)	Analyze list of supported versions – at user's discretion – it is recommended that if one matches the versions supported by the ground, then re-send the message with good version number	
FSTREQ was sent – wait for FSTREP	Receive FERIND with error code 2, 3 or 6 (message considered erroneous)	at ground user's discretion - it is recommended that the message is corrected and re-sent	
FSTREQ was sent – wait for FSTREP	FERIND received with error code 0, 1, 4, 7, 9, 10, 11 (A/C cannot reply)	at ground user's discretion – it is recommended that at least one retry is performed before giving up	
FSTREQ was sent – wait for FSTREP	More than 10 minutes (value at user's discretion) waiting	at ground user's discretion – it is recommended that at least one retry is performed before giving up refuel targets by data link and switching to manual back up – the ground application could check if FSTREQ was declared received by the aircraft (use of ARINC Message Assurance service)	
FSTREQ was sent – wait for FSTREP	FSTREP received containing invalid data	at ground user's discretion – it is recommended that at least one retry is performed before giving up refuel targets by data link and switching to manual back up	

6.2.1.4 ACARS format

The message should only contain element one element. No supplementary header should be used here.

Label used is “RF”

Element Name: **FSTREQ01**
No data field should be embedded in FSTREQ01 message (only header and CRC information and optional supplementary addresses).

Element Name: **FSTREP01**
The element parameters should be encoded in ACARS, as follows.

Table 6.2.1.4.1 – FSTREP01 ACARS Definition

Field name	Meaning	Field length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter type					
CurrentZFW	The current metric value of the aircraft's Zero Fuel Weight (Mass) parameter.	1-6		mass-kg	NO	0 to 999999	1	kg	320500
parameter separator		1	comma		NO				,
CurrentZFWSource	source that last changed ZFW (F=FMS, A=ACARS, N=EFB, Q=FQMS)	1		A	NO	'F', 'A', 'N', 'Q'	N/A		F
parameter separator		1	comma		NO				,
ZFWEntryDate	date when ZFW was last changed	6		date	NO				050125
parameter separator		1	comma		NO				,
ZFWEntryTime	time when ZFW was last changed	4		timeMin	NO				0055
parameter separator		1	comma		NO				,
CurrentZFCG	The current value of the aircraft's Zero Fuel Center-of-Gravity parameter.	3		center-of-gravity	NO	0 to 999	tenth of percent	tenth of percent	385
parameter separator		1	comma		NO				,
CurrentZFCGSource	source that last changed ZFW (F=FMS, A=ACARS, N=EFB, Q=FQMS)	1		A	NO	'F', 'A', 'N', 'Q'			A
parameter separator		1	comma		NO				,
ZFCGEntryDate	date when ZFCG was last changed	6		date	NO				050124
parameter separator		1	comma		NO				,
ZFCGEntryTime	time when ZFCG was last changed	4		timeMin	NO				2359
parameter separator		1	comma		NO				,
CurrentPFQ	The current metric value of the aircraft's Preselected Fuel Quantity parameter. This value specifies the end-quantity of the next aircraft refueling process	1-6		mass-kg	NO	0 to 999999	1	kg	230000
parameter separator		1	comma		NO				,
CurrentPFQSource	source that last changed ZFW (F=FMS, A=ACARS, N=EFB, Q=FQMS, I=Integrated Refuel panel)	1		A	NO	'F', 'A', 'N', 'Q', I			N
parameter separator		1	comma		NO				,
PFQEntryDate	date when PFQ was last changed	6		date	NO				050124

6.0 FUEL

Field name	Meaning	Field length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter type					
parameter separator		1	comma		NO				,
CurrentPFQTime	time when PFQ was last changed	4		timeMin	NO				2329
parameter separator		1	comma		NO				,
GWCG	The current value of the aircraft's Gross Weight Center-of-Gravity parameter	3		center-of-gravity	NO	000 to 999	tenth of percent	tenth of percent	397
parameter separator		1	comma		NO				,
GWCGAccuracyState	Value of parameter accuracy ['N' (normal = all sensors agree on a value) , 'D' (degraded = error on accuracy), 'F'(Failed = impossible to report a consolidated value)]	1		A	NO	'N', 'D', 'F'			N
parameter separator		1	comma		NO				,
FOB	The current metric value of the aircraft's Fuel On Board parameter	1-6		mass-kg	NO	0 to 999999	1	kg	230100
parameter separator		1	comma		NO				,
FOBAccuracyState	Value of parameter accuracy ['N' (normal = all sensors agree on a value) , 'D' (degraded = error on accuracy), 'F' (Failed = impossible to report a consolidated value)]	1		A	NO	'N', 'D', 'F'			N
parameter separator		1	comma		NO				,
TTK	The current value of the aircraft's Trim Tank Fuel quantity	1-6		mass-kg	YES	0 to 999999	1		4200
parameter separator		1	comma		NO				,
TTKAccuracyState	Value of parameter accuracy ['N' (normal = all sensors agree on a value), 'D' (degraded = error on accuracy), 'F' (Failed = impossible to report a consolidated value)]	1		A	YES	'N', 'D', 'F'			N

Field name	Meaning	Field length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter type					
parameter separator		1	comma		NO				,
Aircraft MassUnitDisplayed	Unit used for pilot display, e.g. FMS or ECAM/EICAS	2		AA	NO	KG' (for KG) or 'LB' (for Lbs)			KG

Example: (example below includes ARINC 633 header information)

```
FSTREQ01 uplink:

,FSTREQ01050125221536
/,xxxx


FSTREP01 downlink:

,FSTREP01050125221545
/320500,F,050125,0055,385,A,050124,2359,230000,N,050124,2329,397,N
,230100,N,4200,N,KG,xxxx
```

(note: xxxx is the CRC)

6.2.1.5 XML Format

Presently, this format should be used for ground-ground message exchanges only. Technically, it could also be used for air-ground message exchanges.

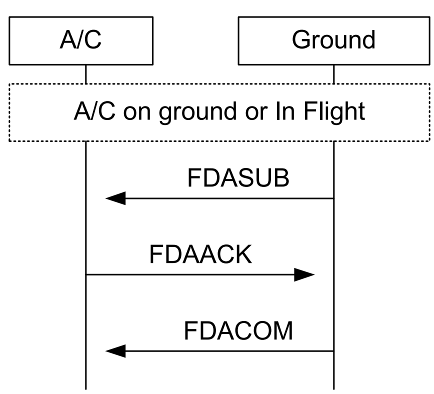
COMMENTARY

Presently, all refuel applications are connected to ACARS only and do not use any XML messages. The XML messages defined in section 6.2 may be used instead of the ACARS messages on IP-based ground-ground links only.

See electronic document 'ARINC633 AOC.zip' for details and examples. Schemas are defined in

schema/FUEL/CGTARGETING.xsd.

6.2.2 Fuel Data Submission Service



Typical Fuel Data Submission Exchange

The elements defined here are needed to specify the data required by the aircraft fuel system to perform refueling using CG targeting. Data uplinked to the aircraft "fuel system" is acknowledged by mirroring the uplinked values to the ground. A separate command uplink is needed for activation. This uplink also specifies if a message needs to be sent at completion of refuel operation (FENIND). Typically the three messages are sent in the order submit, ack and command. Typically the messages are used on the ground during preflight but may be exchanged at any time before, during or after the flight. Detailed use of these data by the fuel system is Aircraft implementation dependent and is outside the scope of this document.

This section defines version 1 of Fuel data service.

Operational parameters ranges and resolution in all messages will vary depending on aircraft type. Therefore, the airline and ground operator should ensure that only messages with ranges and resolutions fitting the aircraft type will be used.

6.2.2.1 Fuel-DAta-SUBmit Uplink (FDASUB)

Element Name: **FDASUB**

The fuel data submit uplink may contain one and a maximum of one optional supplementary address (as per ARINC 620).

All the data fields of this element are optional – at least one should be specified in FDASUB01 message.

6.2.2.2 Fuel-DAta-ACK Downlink (FDAACK)

Element Name: **FDAACK**

The fuel data ack downlink may contain one optional supplementary address (as per ARINC 620), which will be the copy of the supplementary address (if present) in the associated FDASUB.

This element should contain copies of the parameter values received in the FDASUB message. If a given value has been detected as invalid (out of range) by the airborne application, character ‘X’ should be sent instead of the concerned parameter.

6.2.2.3 Fuel-DAta-COMmand Uplink (FDACOM)

Element Name: **FDACOM**

The fuel data command uplink may contain one and a maximum of one optional supplementary address (as per ARINC 620).

6.2.2.4 Dynamic Aspects

- FDASUB, FDAACK and FDACOM should always be sent in the order specified above.
- When a valid FDASUB message has been received, the airborne refuel application should consider that a transaction has been ‘opened’. A transaction should be considered ‘closed’ when the airborne application has received the FDACOM from ground or when a time out after transmission of FDAACK has elapsed without receiving any FDACOM.
- The ground should generate FDACOM message within a delay following reception of FDAACK, compatible with the time-out value used by the airborne system.
- The aircraft should generate FDAACK message within a delay following reception of FDASUB, compatible with the time out value used by the ground system.

COMMENTARY

10mn can be considered as a reasonable value for airborne and ground time-out delays.

COMMENTARY

The airborne application may only support one transaction at a time. If no supplementary addresses are embedded in the messages, may consider that two ground applications are the same application.

- If multiple ground applications interact with the aircraft fuel system, supplementary addresses should be used to identify unambiguously the ground application.
- The airborne system is likely to clear received data at a point in the flight cycle as specified by the system supplier. Data should not be sent to the system during this flight phase(s).

COMMENTARY

The time when the airborne system can clear the received data is implementation dependent and may be obtained from supplier documentation.

The following tables define the required or recommended behavior (detailed in the tables) for the aircraft application and ground application for non-nominal conditions.

Table 6.2.2.4.1 – Airborne Processing Of Non Nominal Conditions

Aircraft Condition	Event	Aircraft Action	Comment
Any	Erroneous FDASUB (Includes condition where 'final refuel' flag is absent and current_PFQ is specified and the contrary) or FDACOM received	Send FERIND with appropriate error code No change on on-going transaction if started	
No transaction opened or FDAACK not yet sent	FDACOM received	- Send FERIND with error code 2 - current transaction continues	
Transaction opened	FDASUB received	- Send FERIND with error code 21 - current transaction continues	
FDAACK was sent following reception of FDASUB	No FDACOM received after FDAACK transmission before airborne time out delay	Send FERIND with error code 1 Abort transaction	
Any	FDASUB or FDACOM received with a non supported version number	-Send FERIND with error code 5 and list of supported versions of this service -- current transaction continues (if any)	
Fuel system degraded mode or airborne application internal error	Message received	-Send FERIND with appropriate error code - Abort transaction if any	
No transaction opened	FDASUB with "Out of range" parameters	-Send FDAACK with 'X' for the concerned parameter (s) -Transaction opened (wait for FDACOM with 'cancel')	

Table 6.2.2.4.2 – Ground processing of Non Nominal Conditions

Ground condition	Event	Ground action	Comment
Any	FERIND received with error codes 0, 1, 4, 7, 9, 10, 11 (Fuel system degraded mode or airborne application internal error)	At ground user's discretion – it is recommended that at least one retry appropriate to the error code is performed before giving up refuel targets by data link and switching to manual back up	
FDASUB transmitted to the aircraft	FDAACK not received by the ground before ground time out delay, following FDASUB transmission to the aircraft	At ground user's discretion – it is recommended that at least one retry is performed before giving up refuel targets by data link and switching to manual back up – the ground application should check if FDASUB was declared received by the aircraft (use of ARINC Message Assurance service)	
FDASUB transmitted to the aircraft	FDAACK received containing invalid data (Includes condition where 'final refuel' flag is absent and current_PFQ is specified and the contrary)	at ground user's discretion – it is recommended that at least one retry is performed before giving up refuel targets by data link and switching to manual back up and send FDACOM with cancel	
Any	Receive FERIND with error code 5 (and list of supported versions)	Analyze list of supported versions – at user's discretion – it is recommended that if one matches the versions supported by the ground, then re-send the message with good version number	
after any message sent	Receive FERIND with error code 2, 3 or 6 (message considered erroneous)	at ground user's discretion - it is recommended that the message is corrected and re-sent	
FDASUB transmitted to the aircraft	FERIND received with code 21	Wait for airborne time out delay before sending FSTREQ for checking new values then resubmitting the message if always needed, or check with other ground system that controls refueling.	
FDASUB transmitted to the aircraft	FDAACK received with 'X' value for some parameters	Ground system should cancel the transaction by sending FDACOM (cancel).	

6.2.2.5 ACARS format

The message will only contain one element. No supplementary header should be used here. Label used is "RF"

Table 6.2.2.5.1 – FDASUB01 ACARS Definition

Field name	Meaning	Field length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter type					
CurrentZFW	The current metric value of the aircraft's Zero Fuel Weight (Mass) parameter or, if out of value, the constant "X"	1-6		mass-kg	YES	0 to 999999	1	kg	320500
parameter separator		1	comma		NO				,
CurrentZFCG	The current value of the aircraft's Zero Fuel Center-of-Gravity parameter.	3		center-of-gravity	YES	000 to 999	tenth of percent	tenth of percent	385
parameter separator		1	comma		NO				,
CurrentPFQ	The current metric value of the aircraft's Preselected Fuel Quantity parameter. This value specifies the end-quantity of the next aircraft refueling process	1-6		mass-kg	YES	0 to 999999	1	kg	230000
parameter separator		1	comma		NO				,
FinalRefuelOperation	Indicates if this Fuel data submission will be the final one – (1=final refuel operation) - see section 6.2.3.1 FENIND dynamic description for details	1		boolean	YES, but always present, if Current PFQ para-meter is defined	'0' or '1'			1

FDASUB should contain at least one parameter (current ZFW, ZFCG or PFQ).

Note:

If FDASUB is sent without any value, it is regarded as erroneous and generates FERIND with appropriate error code.

Table 6.2.2.5.2 – FDAACK01 ACARS Definition

Field name	Meaning	Field length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter Type					
CurrentZFW	The current metric value of the aircraft's Zero Fuel Weight (Mass) parameter.	1-6		mass-kg or constant	YES (present if parameter was received in FDA-SUB message)	0 to 999999 or 'X'	1	kg	320500
parameter separator		1	comma		NO				,
CurrentZFCG	The current value of the aircraft's Zero Fuel Center-of-Gravity parameter.	3		center-of-gravity or constant	YES (present if parameter was received in FDA-SUB message)	000 to 999 or 'X'	tenth of percent	tenth of percent	385
parameter separator		1	comma		NO				,
CurrentPFQ	The current metric value of the aircraft's Preselected Fuel Quantity parameter. This value specifies the end-quantity of the next aircraft refueling process	1-6		mass-kg	YES	0 to 999999	1	kg	230000
parameter separator		1	comma		NO				,
FinalRefuelOperation	Indicates if this Fuel data submission will be the final one – (1=final refuel operation) - see section 6.2.3.1 FENIND dynamic description for details	1		boolean	YES, but always present, if CurrentPFQ parameter is defined	'0' or '1'			1

Table 6.2.2.5.3 – FDACOM01 ACARS Definition

Field name	Meaning	Field length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Variable text					
Confirm Status	'1'=Confirm (FDAACK received correctly and matches FDASUB), '0' or any other value means cancel	1		B	NO	0 to 1			1

Example

In the following scenario an A380 is fueled to 230000 kg. The estimated ZFW of 320500 kg and the estimated ZFCG of 38.5%, which are needed for the A380 specific CG Targeting process, become available later. Due to a ZFCG out of range (A380 Fuel system allows a max of ZFCG=44%), the ground canceled the transaction and a corrected message resent.

```
FDASUB01 uplink:

,FDASUB01050125221536
/,,230000,0,xxxx

FDAACK01 downlink:

,FDAACK01050125221539
/,,230000,0,xxxx

FDACOM01 uplink:

,FDACOM01050125221539
/1,xxxx


FDASUB01 uplink:

,FDASUB01050125221902
/320500,500,,,xxxx

FDAACK01 downlink:

,FDAACK01050125221909
/320500,X,,,xxxx

FDACOM01 uplink:

,FDACOM01050125221912
/0,xxxx


FDASUB01 uplink:

,FDASUB01050125223502/320500,385,,,xxxx

FDAACK01 downlink:

,FDAACK01050125223509
/320500,385,,,xxxx

FDACOM01 uplink:

,FDACOM01050125223512
/1,xxxx
```

6.2.2.6 XML Format

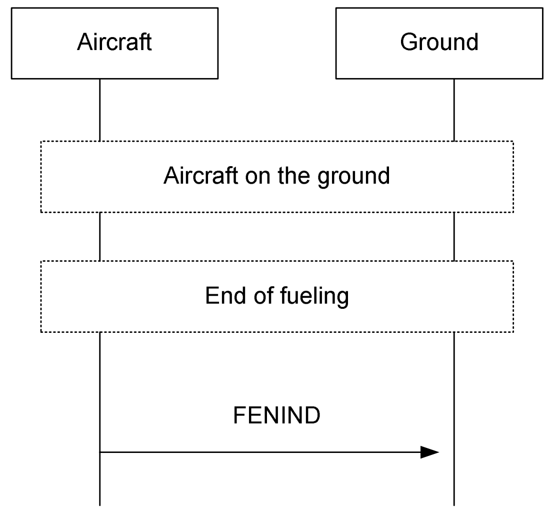
This format should be used for ground-ground message exchanges only.

COMMENTARY

Presently, all refuel applications are connected to ACARS only and do NOT use any XML messages. The XML messages defined in section 6.2 may be used instead of the ACARS messages on IP-based ground-ground links only.

See electronic document 'ARINC633 AOC.zip' for details and examples. Schemas are defined in schema/FUEL/ CGTARGETING.xsd.

6.2.3 reFuel ENd



Typical Fuel end exchange

End of refueling event is necessary to allow generation by ground system of a precise Load and Trim Sheet (fuel quantities in tanks may differ from the one known by ground system during refueling process).

This section defines version 1 of reFuel End service.

6.2.3.1 reFuel-ENd-INDication downlink (FENIND)

Element Name: **FENIND**

6.2.3.2 Dynamic Aspects

FENIND is sent by the airborne system at the end of a successful refueling process.

FENIND should not be generated if the PFQ used for refuel was received in a FDASUB message with ‘Final refuel operation’ flag set to ‘0’. FENIND should be generated in all other cases (remote, i.e. activation by FDASUB with flag set to ‘1’, or locally by airborne system or locally on the aircraft (from FMS, Integrated Refuel panel...)).

If fuel system data necessary for FENIND transmission the system are missing at the end of a Refuel operation, then FENIND message should not be sent and error message with error type ‘missing data, code 7 should be sent , with FENIND referenced as erroneous service and element

6.2.3.3 ACARS Format

The message will only contain one element. The refuel end indication downlink may contain one or several optional supplementary address(es) (as per ARINC 620), which will be the copy of the supplementary address(es) (if present) in the last RF uplink (if received).

Label used is “RF”

FENIND01

The element should be encoded in ACARS exactly as FSTREP01 element (refer to section 6.2.1.4).

Example: (example below includes header information)

```
FENIND01 downlink:

,FENIND01050125221545
/320500,F,050125,0055,385,A,050124,2359,230000,N,050124,2329,397,N
,230100,N,4200,N,KG,xxxx
```

(note: xxxx is the CRC)

6.2.3.3 XML Format

This format should be used for ground-ground message exchanges only.

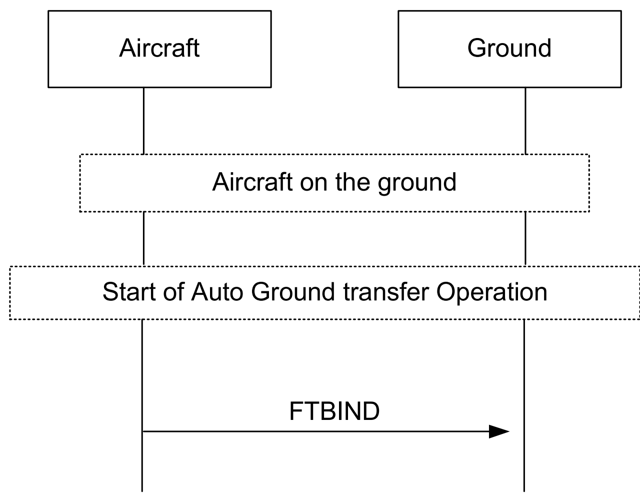
COMMENTARY

Presently, all refuel applications are connected to ACARS only and do NOT use any XML messages. The XML messages defined in section 6.2 may be used instead of the ACARS messages on IP-based ground-ground links only.

FEN XML Schema Definition (including the ARINC 633 header)

See electronic document 'ARINC633 AOC.zip' for details and examples. Schemas are defined in schema/FUEL/ CGTARGETING.xsd.

6.2.4 Automatic Ground Transfer Start



Typical Auto Ground Fuel Transfer Start Exchange

This section defines version 1 of Automatic ground transfer start indication.

6.2.4.1 Fuel-Transfer-Begins-INDication Downlink (FTBIND)

Element name: **FTBIND**

FTBIND message transfer should be optional.

6.2.4.2 Dynamic Aspects

FTBIND is sent by the airborne system when an automatic ground transfer process starts.

6.2.4.3 ACARS Format

The message will only contain one element. The fuel automated transfer end indication downlink may contain one optional supplementary address (as per ARINC 620), which will be the copy of the supplementary address(es) (if present) in the last RF uplink (if received)..

Label used is "RF"

FTBIND01

The Fuel-transfer-begin-indication downlink should contain no parameters – and thus only contain the basic header (not the supplementary header).

COMMENTARY

> Time and Date are needed for this message – it has been deemed acceptable to use the information already contained in the ARINC 633 header.

Example: (example below includes header information)

```
FTBIND01 downlink:

,FTBIND01050125221545
/,xxxx
```

(note: xxxx is the CRC)

6.2.4.4 XML Format

This format should be used for ground-ground message exchanges only.

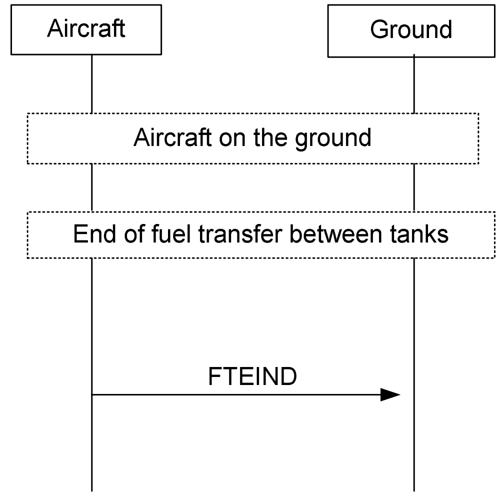
COMMENTARY

> Presently, all refuel applications are connected to ACARS only and do NOT use any XML messages. The XML messages defined in section 6.2 may be used instead of the ACARS messages on IP-based ground-ground links only.

FTB XML Schema Definition (including the ARINC 633 header)

See electronic document 'ARINC633 AOC.zip' for details and examples. Schemas are defined in schema/FUEL/ CGTARGETING.xsd.

6.2.5 Automatic Ground Transfer End



Typical Auto Ground Fuel Transfer End Exchange

End of auto ground transfer event is necessary to allow generation by ground system of a precise Load and Trim Sheet (fuel quantities in various tanks may differ from the one known by ground system during refueling process).

This section defines version 1 of automatic ground Transfer End INDication.

6.2.5.1 Fuel-automatic-Transfer-End-INDication Downlink (FTEIND)

Element Name: **FTEIND**

6.2.5.2 Dynamic Aspects

FTEIND is sent by the airborne system at the end of a successful automatic ground transfer process.

COMMENTARY

A380 examples of premature cessation include de-selection by the pilot of the ground transfer process and termination of the AGT due to engine start.

If fuel system data necessary for the system are missing at the end of an Auto ground fuel transfer operation, then FTEIND message should not be sent and the error message with the error type 'missing data', code 7, should be sent with FTEIND referenced as erroneous service and element.

6.2.5.3 ACARS Format

The message will only contain one element. The fuel automated transfer end indication downlink may contain one optional supplementary address (as per ARINC 620), which will be the copy of the supplementary address(es) (if present) in the last RF uplink (if received).

If fuel system data necessary for the system are missing at the end of a Refuel operation, then an FENIND message should not be sent and the error message with the error type 'missing data', code 7, should be sent.

Label used is "RF"

FTEIND01

The element should be encoded in ACARS exactly as FSTREP01 element (refer to Section 6.2.1.4).

Example: (example below includes header information)

```
FTEIND01 downlink:

,FTEIND01050125221545
/320500,F,050125,0055,385,A,050124,2359,230000,N,050124,2329,39
7,N,230100,N,4200,N,KG,xxxx
```

(note: xxxx is the CRC)

6.2.5.4 XML Format

This format should be used for ground-ground message exchanges only.

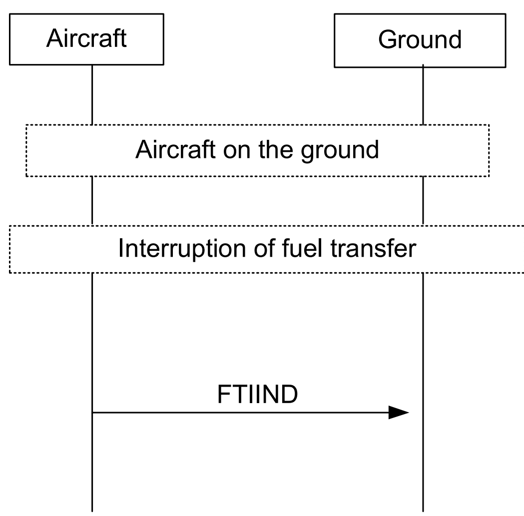
COMMENTARY

Presently, all refuel applications are connected to ACARS only and do NOT use any XML messages. The XML messages defined in section 6.2 may be used instead of the ACARS messages on IP-based ground-ground links only.

FTE XML Schema Definition (including the 633 header)

See electronic document 'ARINC633 AOC.zip' for details and examples. Schemas are defined in schema/FUEL/ CGTARGETING.xsd.

6.2.6 Automatic Ground Transfer Interrupted



Typical Auto Ground Fuel Transfer Interruption Exchange

This section defines version 1 of automatic ground Transfer Interruption Indication.

6.2.6.1 Fuel-automatic-Transfer-Interruption-Indication Downlink (FTIIND)

Element Name: **FTIIND**

6.2.6.2 Dynamic Aspects

FTIIND is sent by the airborne system when automatic fuel ground transfer has not completed successfully for any reason (for example, due to crew deselection or abort due to a failure). Automatic Fuel ground transfer operation is prematurely interrupted.

If fuel system data necessary for FTIIND transmission the system are missing at the end of an interrupted Auto ground fuel transfer operation, then FTIIND message should not be sent and error message with error type ‘missing data, code 7 should be sent with FTIIND referenced as erroneous service and element“.

6.2.6.3 ACARS Format

The message will only contain one element. The fuel automated transfer Interruption indication downlink may contain one optional supplementary address (as per ARINC 620), which will be the copy of the supplementary address (es) (if present) in the last RF uplink (if received).

Label used is “RF”

FTIIND01
The element should be encoded in ACARS exactly as FSTREP01 element (refer to section 6.2.1.4).

Example: (example below includes header information)

```
FTIIND01 downlink:

,FTIIND01050125221545
/320500,F,050125,0055,385,A,050124,2359,230000,N,050124,2329,397,N,230100
,N,4200,N,KG,xxxx
```

xxxx is the CRC

6.2.6.4 XML Format

This format should be used for ground-ground message exchanges only.

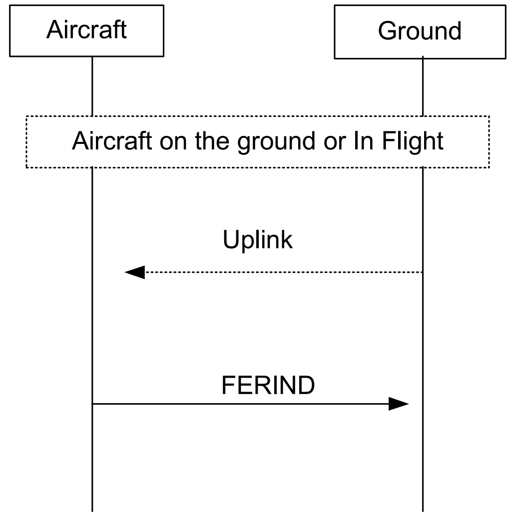
COMMENTARY

Presently, all refuel applications are connected to ACARS only and do **NOT** use any XML messages. The XML messages defined in section 6.2 may be used instead of the ACARS messages on IP-based ground-ground links only.

FTE XML Schema Definition (including the 633 header)

See electronic document ‘ARINC633 AOC.zip’ for details and examples. Schemas are defined in schema/FUEL/ CGTARGETING.xsd.

6.2.7 Error Service



Typical Error exchange

This service is used to report any error detected by the aircraft Fuel application to the Ground.

Depending on the type of received uplink and on its content, various types of errors will be reported by the aircraft (see in section 6.2.1 and 6.2.2 and in table below for list of possible error codes).

FERIND uplinks are not used by this Error service.

6.2.7.1 Fuel-Error-Indication uplink/downlink (FERIND)

Element Name: **FERIND**

6.2.7.2 Dynamic Aspects

Refer to other sections for details.

6.2.7.3 ACARS Format

The message will only contain one element. The Fuel error indication downlink may contain one or several optional supplementary address (as per ARINC 620), which will be the copy of the supplementary address(es) (if present) contained in last received uplink.

Label used is "RF"

Table 6.2.7.3.1 – FERIND01 ACARS Definition

Field name	Meaning	Field length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter type					
parameter separator		1	comma		NO				,
erroneous Service	Service that caused the error	3		Char3Type	NO	'XXX' is reserved to indicate that service is unknown			FST
erroneous Element	Element that caused the error	3		Char3Type	NO	'XXX' is reserved to indicate that service is unknown			REP

Field name	Meaning	Field length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter type					
service Version Nb	Version number of the service that caused the error	2		NN	NO				01
parameter separator		1	comma		NO				,
ErrorClass	category of error – indicates if the error is recoverable or not see section 5.1.2	1		Integer	NO	0-3			1
parameter separator		1	comma		NO				,
errorType	See table below	2		integer	NO	00-99			1
parameter separator		1	comma		NO				,
errorData	See table below	N		(0-n)I	YES				
Parameter separador		1	comma		NO				,

Table 6.2.7.3.2 – Application Specific Error Types

Sender's Situation	Type Name	Type ID	Suggested text to user (e.g. flightcrew)	Comment
as defined in Chapter 5		0		
as defined in Chapter 5		1		
as defined in Chapter 5		2		
as defined in Chapter 5		3		
as defined in Chapter 5		4		
as defined in Chapter 5		5		
as defined in Chapter 5		6		
as defined in Chapter 5		7		
Aircraft fuel system is not able to process the uplink (i.e. Submit Fuel data process on going with another Ground system)	Busy	21		
communication is broken with the avionics fuel system, datalink application is not able to send data to the fuel system – therefore, FDASUB is not available.	read-only	22		

Application Specific Use Of Error Data

Contains vendor specific detailed information for troubleshooting (software status, source of the error, parameters values…).

In case of error 5, should only list the supported versions of the service (NN separated by spaces)

6.2.7.4 XML format

This format should be used for ground-ground message exchanges only.

See electronic document ‘ARINC633 AOC.zip’ for details and examples. Schemas are defined in

schema/FUEL/ CGTARGETING.xsd.

6.3 Refueling

Application Name	Refueling
ACARS Label	RE
Introduced	2005
Purpose	The idea behind standardizing air-/ground messages related to the aircraft refueling process is the wish to refuel an aircraft - just in time and without transporting paper - with the help of airline independent into-plane-service or handling companies
Stake Holder	airlines, into-plane-services, ops handling agents
Airborne System	AOC
Ground Systems	Into plane service fuel handling system, airline or airport host, airline or airport departure control system, airline or handling agent weight & balance system
Network	ACARS
Downlink Routing	Label, Airport, Airline

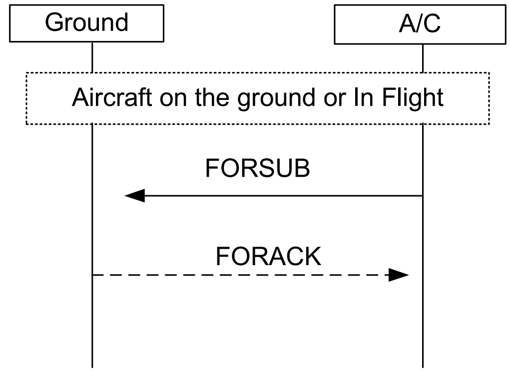
Table: 6.3.0.1 – Refueling Service Table

Service Name	Service ID	Description
Fuel Order	FOR	Fuel Order: The into plane service (fuel supplier) needs to know, that a customer wants him to serve a particular flight. The into plane service or the aircraft's refueling computer or the airline's weight and balance department needs to know with how much fuel the pilot wants to depart
Fuel Receipt	FRC	Fuel receipt: The into plane service should tell someone how much fuel has been refilled and who was the owner of the transferred fuel. "Someone" may be the pilot who may have to verify the commercial transaction after checking his fuel gauges with an (electronic) signature.
Fuel Process	FRP	Fuel Process: During aircraft turnaround, other processes may depend on the fueling process. Therefore handling partners may need to know when fueling starts and when it is finished.
Fuel CG Advisory	FCA	Fuel CG Advisory: The CG Advisory service should inform the flight crew that the aircraft is presently out of trim and that this situation can be corrected by a re-distribution of fuel.

Table 6.3.0.2 – Refueling Element Table

Element Name	ACARS Label	Service Version	DL/UL	Purpose
FORSUB	RE	1	DL	fuel-order-submit: Submits parameters to a ground handling system. Requests or cancels a fuel track
FORACK	RE	1	UL	fuel-order-acknowledge: Confirms the reception of a fuel-order-submit downlink
FRCSUB	RE	1	UL	fuel-receipt-submit: Tells the aircraft how much volume has been supplied. May replace paper fuel receipt
FRCACK	RE	1	DL	fuel-receipt-acknowledge: Confirms the reception of a fuel-receipt-submit uplink
FPRREP	RE	1	DL/UL	fuel-process-report: Informs about fueling progress, e.g. start or end of fueling
FCAIND	RE	1	UL	Fuel CG Advisory Indication: Informs about out of trim conditions

6.3.1 Fuel Order Service



Typical Fuel Order Exchange

The fuel-order element is needed to request fuel from an into-plane fuel service provider. It can be sent in all flight phases. It can be used to tell the fuel supplier that no fuel is needed today or to cancel a fuel truck previously ordered. It may contain the remaining fuel, so that the into-plane service can calculate the amount of fuel to be uploaded in the refueling process. It may either state the minimum fuel required (if block fuel has not yet been finalized, or the final block fuel.

Following subsections define version 1 of FOR service.

6.3.1.1 Fuel-order-submit Downlink

Element Name: **FORSUB**

Table 6.3.1.1 – FORSUB01 and FORACK01 ACARS Definition

Parameter Name	Meaning	Contents		Option	Range	Resolution	Unit	Example
		Fixed Text	Parameter Type					
OrderFlag	0 = No Fueling required, 1 = Fueling required		boolean	No				1
AcknowledgeFlag	0 = No acknowledge wanted 1 = Acknowledge requested		boolean	No				1
FinalFlag	0 = Preliminary fuel, e.g. minimum fuel before top of or not applicable 1 = Final fuel		boolean	No				0
ActualFlag	0 = Estimated remaining or not applicable 1 = Actual Remaining		boolean	No				1
parameter separator		comma		No				,
ServiceAirport	ICAO code of airport, where fueling shall take place		airport	Yes				KJFK
parameter separator		comma		No				,
IntoPlaneService	A three digit code unique to an into-plane-service company		provider-id	Yes				ABC

6.0 FUEL

Parameter Name	Meaning	Contents		Option	Range	Resolution	Unit	Example
		Fixed Text	Parameter Type					
parameter separator		comma		No				,
ServiceFlight	IATA flight number of flight to be fueled if different from current flight		flight-no	YES				AF123A
parameter separator		comma		No				,
BlockFuel	Amount of fuel that should be in the aircraft after refueling or also called total fuel		mass-tt	Yes	0 - 9999 (= 999.9 tons)	100 kg	tenth of metric tons	87
parameter separator		comma		No				,
TaxiFuel	Amount of fuel required from offblock until begin of takeoff run		mass-kg	Yes	0-99999 (=99.9 tons)	1 kg	Kilogram	200
parameter separator		comma		No	","			,
TripFuel	Amount of fuel required from begin of takeoff run until touchdown		mass-kg	Yes	0-999999 (= 999.9 tons)	1 kg	Kilogram	3450
parameter separator		comma		No	","			,
Remaining	Amount of fuel that is estimated to be in the aircraft or which is actual in the aircraft by the time the fueling starts		mass-tt	Yes	0 - 9999 (= 999.9 tons)	100 kg	tenth of metric tons	85
parameter separator		comma		No				,
AircraftMassUnitDisplay	Unit used for pilot display (Kilograms or pounds)		"KG" or "LB"	Yes				KG
parameter separator		comma		No				,
LimitMass	Most restrictive limitation on TOW, ZFW or LAW		mass-tt	YES	0-9999 (= 999.9 tons)	100 kg	tenth of metric tons	5232
parameter separator		comma		No				,
LimitType	Type of most restrictive limitation: "T" for TOW, "Z" for ZFW, "L" for LAW		U	Yes				T
parameter separator		comma		No				,
FuelTruckSTBY	Information whether the fuel truck can leave the airplane after refueling (NIL or 0) or not (1)		Boolean	Yes				1

6.3.1.2 Fuel-order-acknowledge Uplink

Element Name: **FORACK**

The FORACK element follows the same ACARS and XML definition as the FORSUB element.

6.3.1.3 Dynamic Aspects

FORSUB and FORACK should always be sent in the order specified above.

Upon reception of a correct FORSUB, the ground system should:

- Forward the data to the appropriate end-systems. If this is not possible, the ground system should send the appropriate GENERR message
- Check the AcknowledgeFlag parameter of the FORSUB downlink. If it is set to "1", the ground system should send the FORACK message

Upon reception of an erroneous FORSUB, the ground system should send the appropriate GENERR message.

Upon receiving a GENERR message, the airborne system should, if the FORSUB was manually triggered, display a failure message to the flightcrew. The text of the failure message depends on error class and error type.

Scenario 1

An A320 sends a general fuel truck cancellation before landing in EDDF. The cancellation is confirmed by the ground.

After touchdown in EDDF, this decision is revised. In the second downlink, it asks ABC to send a fuel truck and fill up the aircraft to the minimum fuel of 8.7 tons. Its actual remaining is 8.5 tons. Again, the message is confirmed.

The third downlink is sent, after the block fuel has been finalized to 9.0 tons. Taxi fuel is specified as 200 kg, TripFuel as 3450 kg. Again, the message is confirmed.

All message envelopes have been omitted.

ACARS Example

```
FORSUB01 downlink:

,FORSUB01050324115159LFPG,,KJFK,,AFR5556,,050324,,A320,,F-WWOW,
/0100,EDDF,,LH5678,,,,,,,,xxxx

FORACK01 uplink:

,FORACK01050324115202LFPG,,KJFK,,AFR5556,,050324,,A320,,F-WWOW,
/0100,EDDF,,LH5678,,,,,,,,xxxx


FORSUB01 downlink:

,FORSUB01050324122359LFPG,,KJFK,,AFR5556,,050324,,A320,,F-WWOW,
/1101,EDDF,ABC,,87,200,,85,KG,680,T,1,xxxx

FORACK01 uplink:

,FORACK01050324122407LFPG,,KJFK,,AFR5556,,050324,,A320,,F-WWOW,
/1101,EDDF,ABC,,87,200,,85,KG,680,T,1,xxxx


FORSUB01 downlink:

,FORSUB01050324123200LFPG,,KJFK,,AFR5556,,050324,,A320,,F-WWOW,
/1111,EDDF,ABC,,90,200,3450,85,KG,680,T,0,xxxx

FORACK01 uplink:

,FORACK01050324123205LFPG,,KJFK,,AFR5556,,050324,,A320,,F-WWOW,
/1111,EDDF,ABC,,90,200,3450,85,KG,680,T,0,xxxx
```

Scenario 2

An A380 in JFK asks to be filled up to a final block fuel of 125 tons. However, the fuel order service is temporarily not available.

ACARS Example

```
FORSUB01 downlink:

,FORSUB01050325050015LFPG,,KJFK,,AFR5556,,050325,,A380,,F-WWOW,
/1111,KJFK,,,1250,1000,95650,285,KG,5400,T,,xxxx
```

```
GERIND01 uplink:

,GERIND01050325050024LFPG,,KJFK,,AFR5556,,050325,,A380,,F-WWOW,
/JRE,FORSUB01,1,9,Fuel Order Error Transaction aborted Service
temporarily unavailable,0,/sorry crewxxxx
```

The flightcrew might receive the following error text:

```
Fuel Order Error

Transaction aborted

Service temporarily unavailable

sorry crew
```

6.3.1.4 ACARS Format

The ACARS format specified in table 6.3.1.1 may be used for air-ground and ground-ground messages

When using the ACARS format, the supplementary header should be used.

ACARS Label used should be "RE".

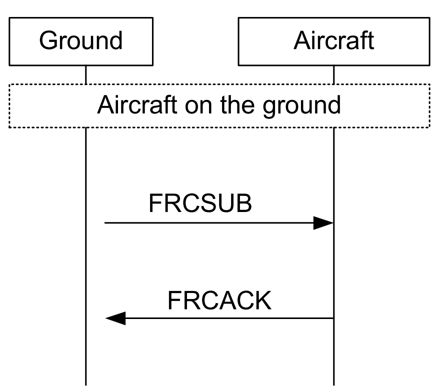
6.3.1.5 XML Format

The IP/XML format may be used for air-ground and ground-ground messages.

When using the IP/XML format, the supplementary header should be used.

See electronic document 'ARINC633 AOC.zip' for details and examples. Schemas are defined in schema/FUEL/ REFUELING.xsd.

6.3.2 Fuel Receipt Service



Typical Fuel Receipt Exchange

Purpose of the Fuel Receipt Service is to eliminate the need for a paper fuel receipt after the end of aircraft refueling. Such a receipt fulfills at least two functions.

The fuel receipt is used to check that the aircraft has enough fuel for the next flight. This check is usually performed by the flight crew

The fuel receipt is used to check the commercial bill from the fuel supplier. This check is usually performed by ground personnel

The fuel-receipt element is used to uplink the fuel receipt to the aircraft after the end of refueling. It contains data needed to check the international fuel supplier code, a refuel-defuel flag, a fuel receipt number, the volume or the mass of fuel uplift and the corresponding units, as well as the fuel density,. Note that the fuel volume or mass unit used is, if different from the metric units, liters and kg, not the unit used for the transmission of the supplied fuel uplift value. All transmitted values are always metric and should be converted by the application if required into the airline's or pilot's chosen unit.

Following subsections define version 1 of FRC service.

6.3.2.1 Fuel Receipt Submit Uplink

Table 6.3.2.1 – FRCSUB01 and FRCACK01 ACARS Definition

Parameter Name	Meaning	Contents		Option	Range	Resolution	Unit	Example
		Fixed Text	Parameter Type					
IntoPlaneService	A three digit code unique to an into-plane-service company		provider-id	No				AFS
parameter separator		comma		No	","			,
FuelSupplier	A three digit international supplier code unique to a fuel supplier		provider-id	Yes				BPC
parameter separator		comma		No	","			,
FuelTruckID	Fuel truck identification number		(0-10)Y	Yes	up to 10 chars			H10556
parameter separator		comma		No	","			,
Refuel/Defuel mode	<nil> or 1 = Refueling with one fuel truck, 2=Refueling with 2 trucks, 5 = Defueling with one trucks, 6= Defueling with 2 trucks		integer	Yes				1
parameter separator		comma		No	","			,
FuelReceiptNumber	Fuel receipt number		(0-20)Y	No	up to 20 chars			8522A
parameter separator		comma		No	","			,
SuppliedVolume	supplied (uplifted) fuel volume (should be given, if measured directly)		volume-lt	Yes	0...999 999	1 liter	liter	58100 see note 2
parameter separator		comma		No	","			,
FuelVolumeMeasureUnit	Unit used by the IntoPlaneService if the fuel volume has been measured (Liters, Imperial Gallons or US Gallons)		"LT" or "IG" or "UG"	Yes,				LT See note 3
parameter separator		comma		No	","			,
SuppliedDensity	mean density of		density	No	700...9	kg/liter/1000	kg/liter	802

Parameter Name	Meaning	Contents		Option	Range	Resolution	Unit	Example
		Fixed Text	Parameter Type					
	supplied fuel				00		*1000	
parameter separator		comma		No	","			,
SuppliedMass	supplied (uplifted) fuel mass (should be only given, if measured directly)		mass-kg	Yes	0...999 999	1 kg	kg	46596 See note 2
parameter separator		comma		No	","			,
FuelMassMeasureUnit	Unit used by the IntoPlaneService if the fuel mass has been determined (Kilograms or pounds))		“KG” or “LB”	Yes,				KG See note 3
parameter separator		comma		No	","			,
FuelType	Supplied fuel type		(0-10)Y	Yes	Up to 10 chars			JET A1 See note 1

Notes:

1. There are several fuel types used over the world, most common are JET A1 and JET A. All of the fuel types have different freezing points requiring the pilots to work special procedures when a “LOW FUEL TEMPERATURE” WARNING is announced in-flight.
2. SuppliedVolume has to be specified if during refueling the fuel volume has been measured by the refueler. SuppliedMass should not be filled in this case. If elsewise the mass of the supplied fuel has been determined during refueling, SuppliedMass should be specified instead of SuppliedVolume.
3. This/these parameter(s) should show the unit in which the supplied volume or the supplied mass was originally measured. This parameter should be present if associated parameter (SuppliedVolume or SuppliedMass) has been specified.

6.3.2.2 Fuel Receipt Acknowledge Downlink

Same data as fuel-receipt-submit uplink

6.3.2.3 ACARS Format

The ACARS format specified in table 6.3.2.1 may be used for air-ground and ground-ground messages

When using the ACARS format, the supplementary header should be used.

ACARS Label used should be “RE”.

6.3.2.4 XML Format

The IP/XML format may be used for air-ground and ground-ground messages.

When using the IP/XML format, the supplementary header should be used.

See electronic document ‘ARINC633 AOC.zip’ for details and examples. Schemas are defined in schema/FUEL/ REFUELING.xsd.

6.3.2.5 Dynamic Aspects

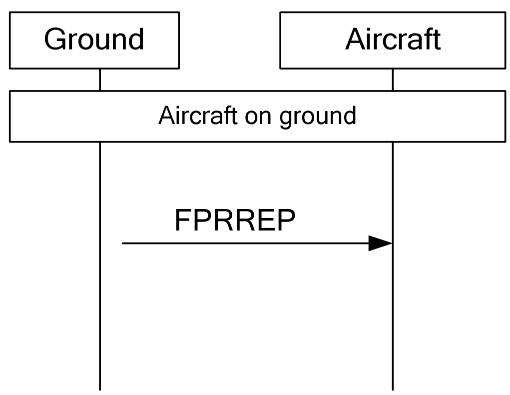
As multiple fuel receipt submit messages could be uplinked, the aircraft application should use a combination of IntoPlaneService and FuelReceiptNumber to identify a consecutive fuel receipt submit message.

In the following scenario a B747 is refueled by 2 BP fuel trucks (with fuel truck numbers H10556 and H10889) simultaneously. One fuel truck measures volume, the other measures mass and density and calculates the volume. The fuel receipts are uplinked independently. Together, both trucks have supplied 123530 liter.

ACARS Example:

```
FRCSUB01 uplink:

,FRCSUB01050324123200LFPG,,KJFK,,AFR5556,,050324,,B747,,F-WWOW,
/AFS,BPC,H10556,,8522,65430,,,JETA1,LT,xxxx

FRCACK01 downlink:

,FRCACK01050324123205LFPG,,KJFK,,AFR5556,,050324,,B747,,F-WWOW,
/AFS,BPC,H10556,,8522,65430,,,LT,xxxx


FRCSUB01 uplink:

,FRCSUB01050324123404LFPG,,KJFK,,AFR5556,,050324,,B747,,F-WWOW,
/AFS, ,H10889,,8522A,58100,802,46596,JETA1,LT,xxxx

FRCACK01 downlink:

,FRCACK01050324123409LFPG,,KJFK,,AFR5556,,050324,,B747,,F-WWOW,
/AFS,,H10889,,8522A,58100,802,46596,JETA1,LT,xxxx
```

6.3.3 Fuel Process Service



Typical Fuel Process Report Exchange

The fuel-process-report element is used to inform ground or airborne systems about the progress of the current refueling, e.g. fueling has started or fueling is finished. It should be used only, while the aircraft is on ground. There is no message confirmation on application level.

Following subsections define version 1 of FPR service.

6.3.3.1 Fuel Process Report Element

Element Name: **FPRREP**

6.3.3.2 ACARS Format

ACARS format should be used for air-ground communications. It may also be used end to end, i.e. on ground.

The ACARS format specified in table 6.3.3.2 may be used for air-ground and ground-ground messages

When using the ACARS format, the supplementary header should be used.

ACARS Label used should be “RE”.

Table 6.3.3.2-1 – FPRREP01 ACARS Definition

Parameter Name	Meaning	Contents		Option	Range	Resolution	Unit	Example
		Fixed Text	Parameter Type					
IntoPlaneService	A three digit code unique to an into-plane-service company		provider-id	No				AFS
parameter separator		Comma		No	","			,
FuelSupplier	A three digit international supplier code unique to a fuel supplier		provider-id	Yes				BPC
parameter separator		Comma		No	","			,
FuelTruckID	Fuel truck identification number		(0-10)Y	Yes	up to 10 chars			H10556
parameter separator		Comma		No	","			,
RefuelDefuel Mode	<nil> or 1=Refueling with one fuel truck, 2=Refueling with 2 trucks, 5=Defueling with one fuel truck, 6=Defueling with 2 trucks		integer	Yes				1
parameter separator		Comma		No	","			,
FuelReportStatus	see table below		integer	No	0-99			1
parameter separator		Comma		No	","			,
ProcessTimestamp	date and time when the process has been archived		datetime	Yes		seconds	YYMMDDHHMMSS	050324 235959

Table 6.3.3.2-2 – Fuel Report Status Report Table

Trigger Condition	Type Name	Type ID	Meaning	Comment
Into plane service company receives notice that no fuel is needed	fueling-cancelled	0	Fuel truck has been cancelled	
Into plane service company receives notice that fuel is needed	ordered-for-fueling	1	Fuel truck has been ordered	
Into plane service company receives preliminary fuel notice (MINTOF, BLOCK w/o Final flag)	ordered-for-fueling-preliminary	11	Non-Final fuel data received	
Into plane service company receives final fuel notice (BLOCK w/ Final flag)	ordered-for-fueling-Final	12	Final fuel data received	
Into plane service confirms fuel truck assignment	fueling-assignment	21	Work order to refuel/defuel an aircraft has been generated	
Fuel truck driver indicates that	available-for-fueling	22	Fuel truck has arrived at the aircraft	

6.0 FUEL

Trigger Condition	Type Name	Type ID	Meaning	Comment
fuel truck is available for fueling			or is waiting for the aircraft	
Fuel hose connected (either sensed by the aircraft or signaled by the fuel truck driver)	ready-for-refueling	23	Fuel hose is connected	
Refueling starts (fuel is flowing through fuel hose to the aircraft)	start-of-fueling	34	Start of Refueling	
Refueling is finished (either no more fuel flowing through fuel hose, pump switched off or amount ordered equals indicated fuel on board)	end-of-fueling	35	End of Refueling	
Fuel truck driver has released fuel receipt		36	Fuel Receipt transferred	
Defueling starts (fuel is flowing through fuel hose from the aircraft)		44	Start of Defueling	
Defueling is finished (either no more fuel flowing through fuel hose, pump switched off		45	End of Defueling	
Fuel transfer starts		54	Start of Fuel Transfer	
Fuel transfer ends		55	End of Fuel Transfer	

In the following scenario a fuel truck uplinks that he is now ready for fueling.

Later the aircraft reports the time, when it has sensed that fueling is finished.

ACARS Example

```
FPRREP01 uplink:

,FPRREP01050324123205LFPG,,KJFK,,AFR5556,,050324,,A380,,F-WWOW,
/TFG,,H10483,,23,,xxxx


FPRREP01 downlink:

,FPRREP01050324125200LFPG,,KJFK,,AFR5556,,050324,,A380,,F-WWOW,
/TFG,,H10483,5,55,050324125159,xxxx
```

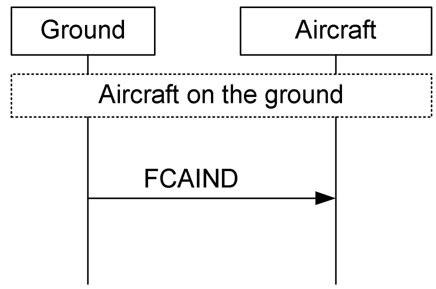
6.3.3.3 XML Format

The IP/XML format may be used for ground-ground messages and for air-ground messages as an alternative to ACARS.

When using the IP/XML format, the supplementary header should be used.

See electronic document ‘ARINC633 AOC.zip’ for details and examples. Schemas are defined in schema/FUEL/ REFUELING.xsd.

6.3.4 Fuel CG Advisory Service



Typical Fuel CG Advisory INDication Message Exchange

Purpose of the Fuel CG Advisory service is to inform the flight crew that a fuel distribution change is needed to reach a takeoff center of gravity value within operational limits. Besides, with this service the flight crew should be enabled to decide to end a fuel distribution change (fuel ground transfer) prematurely, e.g. as soon as the takeoff center of gravity has reached the operational limit. Decision against a complete fuel transfer could be useful to get the aircraft ready for takeoff with minimum departure delay.

The service is mainly applicable for operation of ground based load and trim control.

The Fuel CG Advisory INDication element is used to uplink the actual limit of the operational CG envelope.

Following subsections define version 1 of FCA service.

6.3.4.1 Fuel CG Advisory INDication Uplink

Table 6.3.4.1 – FCAIND01 ACARS Definition

Parameter Name	Meaning	Contents		Option	Range	Resolution	Unit	Example
		Fixed text	Parameter Type					
ForwardTOCGLimit	The forward limit value of the aircraft's Takeoff Weight Center-of-Gravity parameter		center-of-gravity	Yes	NO	000 to 999	tenth of percent	365
parameter separator		comma		No	","			,
AftTOCGLimit	The aft limit value of the aircraft's Takeoff Weight Center-of-Gravity parameter		center-of-gravity	Yes	NO	000 to 999	tenth of percent	425
parameter separator		comma		No	","			,
CalculatedTOCG	The limit value of the aircraft's Gross Weight Center-of-Gravity parameter		center-of-gravity	Yes	NO	000 to 999	tenth of percent	351
parameter separator		comma		No	","			,
TOW	Take Off Weight value used for TOCG computation		mass-kg	Yes	0 to 999999	1	kg	500300
parameter separator		comma		No	","			,
TaxiFuel	Planned Taxi Fuel		mass-kg	Yes	0 to 999999	1	kg	1125

Example:

In the following scenario the ground based load and trim control system detected the aircraft being currently outside of the operation CG envelope. The TOCG as presently calculated on the ground is 35.1%, the operational limits are 36.5% and 42.5%. The CG needs to be shifted at least 1.4% aft. Aircraft GW is 500.3 tons. Estimated taxi fuel is 1125kg. The onboard AOC application should create a text advisory to the flight crew.

6.3.4.2 ACARS format

The ACARS format specified in table 6.3.4.1 may be used for air-ground and ground-ground messages.

When using the ACARS format, the supplementary header should not be used.

ACARS Label used should be “RE”.

ACARS Example

```
FCAIND01 uplink:

,FCAIND01050125221536LFPG,,KJFK,,AFR5556,,050125,,A380,,F-WWOW,
/365,425,351,500300,1125,xxxx
```

6.3.4.3 XML format

The IP/XML format may be used for air-ground and ground-ground messages.

When using the IP/XML format, the supplementary header should be used.

See electronic document ‘ARINC633 AOC.zip’ for details and examples. Schemas are defined in schema/FUEL/ REFUELING.xsd.

7.0 Electronic Logbook

Application	SMI	Downlink		Uplink		Description
		Label		Sub Label	Label	
Tech logbook	JLT JLS	LT LS		N/A	LT LS	Technical (Cockpit) E-Logbook application messages
Cabin Logbook	JLC JLB	LC LB		N/A	LC LB	Cabin E logbook application messages

This section addresses the Electronic logbook data. These data are exchanged between airborne logbook application and ground logbook application, and then between ground logbook application and ground maintenance system.

Logbook Data structure exchanged between GROUND logbook application and GROUND maintenance system isl being defined by ATA. The XML definition below is also provided for illustration only (ATA document supersedes the XML definition in ARINC 633).

This section will be completed in a future supplement.

See electronic document ‘ARINC633 AOC.zip’ for details and examples. Schemas are defined in schema/LogBook/ ELogbook.xsd.

8.0 Configuration Control

Application	SMI	Downlink		Uplink		Description
		Label		Sub Label	Label	
DLCS	JDL	DL		N/A	DL	DataLoading and Configuration System

To be added in a future supplement.

9.0 Weight and Balance (WBA)

On some aircraft, the pilot will have the possibility to use an airborne electronic weight and balance application to perform weight and CG calculation and issue on board the aircraft a Load and Trim Sheet (LTS).

To do so, an air ground communication means will be available to allow the initialization of the on board application and the sending of the LTS to the ground.

Weight and balance operations are usually composed of the following operations:

- The ground airline application computes the initialization parameters needed to initialize the On board weight and balance application
- The ground sends those data to the onboard weight and balance application (WIM, WII or WIF services)
- The WBA initializes its inputs with the data sent from ground. If an error is raised a downlink message (WIRREP) is sent to the ground base
- The WBA computes and issue a LTS.
  Today, the WBA generates
  AHM517 (full format) and AHM518 (ACARS-compatible format), that are Load and Trim Sheet formats specified by IATA.
- The pilot performs necessary checks the LTS and sends it back to the ground before dispatch using same service as eFF (EFF download service: EFDREP01). The LTS is included as a file in an eFFpackage

Note: The Weight and Balance application deviates from the general "unit use rule" (4.1) and allows the use of non-metric units in its air-ground messages.

Element Name	Gatelink Label	Service Version	DL/UL	Purpose
WIMSUB	N/A	1	UL	**Weight-balance-Init Minimum - SUBmit :** Submits initialization parameters to an aircraft's weight and balance system with minimum profile (Only Mandatory data sent)
WIISUB	N/A	1	UL	**Weight-balance-Init Intermediate - SUBmit :** Submits initialization parameters to an aircraft's weight and balance system with intermediate profile (Mandatory + data contained on AHM517 for ZFW and CG computation sent)
WIFSUB	N/A	1	UL	**Weight-balance-Init Full - SUBmit :** Submits initialization parameters to an aircraft's weight and balance system (full data are sent)
EFDREP	N/A	1	DL	**EFf Download REPort:** Send the LTS to ground – as a specific file embedded in an eFF package.
WIRREP	N/A	1	DL	**Weight-balance-Init- eRror- REPort** Indicates an error for weight and balance initialization

COMMENTARY

AHM517 (full format) and AHM518 (ACARS-compatible format) are Load and Trim Sheet formats specified by IATA.

9.0 WEIGHT AND BALANCE (WBA)

9.1 Weight-and Balance – Common Elements And Data

This section defines the parameters and data structures that are shared by various types of Weight and Balance services.

9.1.1 ACARS Format

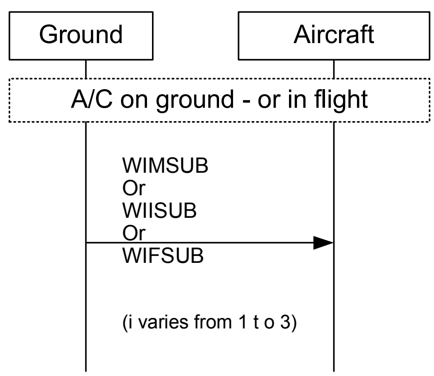
N/A

9.1.2 XML Format

A number of common schemas are defined for Weight and Balance services – these schemas are called by the appropriate services (see “include” command in the service schemas).

See electronic document ‘ARINC633 AOC.zip’ for details and examples. Schema is defined in schema/WBA/WBAcommon.xsd.

9.2 Weight and Balance Initialization



Typical Weight and Balance Init Transmission

The Weight and Balance Init data is used to initialize the onboard W&B application. The aim of this transmission is to send preliminary W&B data from a W&B ground system to the airborne Weight and balance application (WBA).

3 levels of service are possible for Weight & Balance application initialization: minimum, intermediate and full – see appropriate sections for detailed description of these services.

9.2.1 Weight-and Balance- Minimum Init Service

This section defines version 1 of the Minimum initialization service (WIM service).

9.2.1.1 Weight and Balance Initialization Minimum – SUBmit Uplink

Element Name: **WIMSUB**

While operating using this mode the ground system sends only mandatory data to the aircraft and the aircraft can only generate and send to the ground AHM518. The crew may also have to enter some parameters to complete WBA application initialization and/or to complete the LTS computation (see list of optional parameters in the element definition).

9.2.1.1.1 ACARS Format

N/A

9.2.1.1.2 IP Format

See electronic document ‘ARINC633 AOC.zip’ for details and examples of Element WIMSUB.

Schemas are defined in schema/WBA/WBA.xsd.

9.2.2 Weight-and Balance- Intermediate Init Service

9.2.2.1 Weight and Balance Initialization Intermediate – SUBmit Uplink

Element Name: **WIISUB**

While operating using this mode the ground system sends Minimum init service data plus the additional data to the aircraft corresponding to the AHM517 input and the aircraft can generate and send to the ground AHM518 and AHM517.

9.2.2.1.1 ACARS Format

N/A

9.2.2.1.2 IP Format

See electronic document ‘ARINC633 AOC.zip’ for details and examples of Element WIMSUB.

Schemas are defined in schema/WBA/WBA.xsd.

9.2.3 Weight-and Balance- Full Init – Service

9.2.3.1 Weight and Balance Initialization Full – SUBmit Uplink

Element Name: **WIFSUB**

While operating using this mode the ground system sends a complete set of data to the aircraft in order to fully initialise the W&B airborne application and the aircraft can generate and send to the ground AHM518 and AHM517.

9.2.3.1.1 ACARS Format

N/A

9.2.3.1.2 IP Format

See electronic document ‘ARINC633 AOC.zip’ for details and examples of Element WIMSUB.

Schemas are defined in schema/WBA/WBA.xsd.

9.2.4 Dynamic Aspects

The initialization should only consist in simple transmission of the SUBmit element from the ground to the aircraft WBA.

COMMENTARY

> Transport services will ensure reliable and error free transmission of the data to the airborne application.

The initialization service should be usable at any time (in ground, during flight preparation, or in flight, if the crew decides to anticipate next flight preparation).

WIxSUB element should apply to flight under preparation. The supplementary header should specify the aircraft and the flight to which the data are associated (flight under preparation).

COMMENTARY

The crew may not have yet entered a valid flight number when the init data are received (the communication systems may use last flight number).

Upon reception of an erroneous initialization message (WIMSUB, WIISUB or WIFSUB), the airborne system should log an error message on board the aircraft, inform the crew that the received message is erroneous, and generate a corresponding error message (WIRREP) (see section 9.4).

9.3 Weight and Balance Report (LTS) Transmission Services

This service consists in the airborne application transmitting to the ground airline application the Load and Trim Sheet (LTS) calculated on board.

There is no specific service for the Weight and balance Application (WBA) – WBA should use the eFF download service (EFDREP) specified in section 10.3.

The only exception to the use of EFDREP service is the file extension to be used: WBA should generate an '.lds' file (and not '.eff'), containing eFF structure and the LTS as a specific document. Other documents may be included in the eFF.

The LTS should be embedded in the EFF folder associated with the topic 'LoadTrimSheet '(see section 10 for more details).

9.4 Weight and Balance Init Error Service.

Weight-balance-Init-eRror- REPort Service

This service is composed of a single message: WIRREP downlink.

Element Name: **WIRREP**

This service allows the airborne WBA application to indicate to the ground that an error was detected on a received initialization message (WIMSUB, WIISUB, WIFSUB).

9.4.1 Weight and Balance Init eRror REPort (WIRREP) Downlink

9.4.1.1 ACARS Format

N/A

9.4.1.2 IP Format

See electronic document 'ARINC633 AOC.zip' for details and examples of Element WIRREP.

Schemas are defined in schema/WBA/WBA.xsd.

9.4.2 Dynamic Aspects

The weight and balance init report service should only consist in simple transmission of the REPort element (WIRREP) from the aircraft WBA to the ground.

COMMENTARY

Transport services will ensure reliable and error free transmission of the data to the airborne application.

9.5 WBA Files Packaging

WBA messages should be transported as files with the extension '.wba' and filenames with the following structure:

See section 2.3 for general live IP transportation rules.

Each WBA file message (except LTS download) should be transported through a unique file with extension ".wba".

It should be compliant with the corresponding element of the WBA.xsd schema:

- WIMSUB
- WIISUB
- WIFSUB
- WIRREP

It should be named as depicted in section below:

The name of the files for the above mentioned messages should be structured as follows:

Subfield Name	Description	Format (see Table 4.1.2)	Example
flightNum or flightID	As defined in Table 4.4.2.2.2.	(1-10)C	“DLH1234ABC
Delimiter	One Underscore		“_”
departureDate	Identifying flight date	YYMMDD	“050822”
Delimiter	One Underscore		“_”
depICAOCd or depIATACd	3-letter IATA code or 4-letter ICAO code	XXX or (exclusive) XXXX	“FRA”
Delimiter	One Underscore		“_”
destICAOCd or destIATACd	3-letter IATA code or 4-letter ICAO code	XXX or (exclusive) XXXX	“TLS”
Delimiter	One Underscore		“_”
DateTimestamp	YYMMDDHHMMSS		050822173214
File Type	One period and the letters “wba”		“.wba”

Examples:

```
LH436_050822_DUS_ORD_050822122634.wba
UA575_050822_ORD_SEA_050821122634.wba
CFG3214D_050401_XRY_PMI_050402052658.wba
```

The file extension “.wba” indicates that this is an xml File.

Concerning Weight and balance Report (LTS) transmission services (download of LTS using EFDREP service), refer to section 10.3 (eFF file packaging), except that the extension shall be ‘.lds’ and not ‘.eff'. The LTS file enclosed in the eFF can be of any format.

10 Electronic Flight Folder

10.1 General

The purpose of this section is to standardize how operational information regarding a flight segment is technically organized in a flight folder that is accessed by the pilot. This does not intend to provide a standard organization of the documents in the flight folder, but rather a technical mean that would allow carrying a structured organization of flight documents.

As the electronic flight folder is also the support of the operational flight plan, and as operational flight plan contain many important data for the flight that can be passed to other applications, it was deemed necessary to standardize data in an operational flight plan. As there are many different ways to display an operational flight plan, the aim here is to try to have a generic model to handle the data that are available in the operational flight plan, but not on this format. This could be easily handled by XML, as XML is able to separate the data and the way it is displayed.

Other data than operational flight plan may be defined in future versions of this document.

This section defines:

- An overview of EFF operational concept (how the data package is elaborated, updated, and finalized).
- The structure of eFF data package, which parameters are defined and how they are used.
- Then the packaging into an '.eff' file of the EFF data package is covered.
- The rules of transportation via an IP network or with physical media (e.g. USB media) are described. The transportation subsection also covers the option to request and transport updates to an existing EFF. Updates can apply either to a part of the EFF which the user wants to update or the whole EFF.

The definition of some document contents can be found in section 5.2 'Common Data'.

Note: the way eFF is transported is outside the scope of this section and is addressed in section 2.3.

10.1.1 Review of the eFF workflow

Once the documents for the EFF are gathered or created and the EFF is packed properly the question of the applicable usage arises. The following actions are possible with an EFF:

- Transport an EFF from one System to another (e.g. from ground to an airborne system and vice versa).
- Update the whole EFF which was fetched before.
- Update documents marked with specific topic identifiers in an EFF which was fetched before.
- Update single documents in an EFF which was fetched before.
- Use content of the EFF.

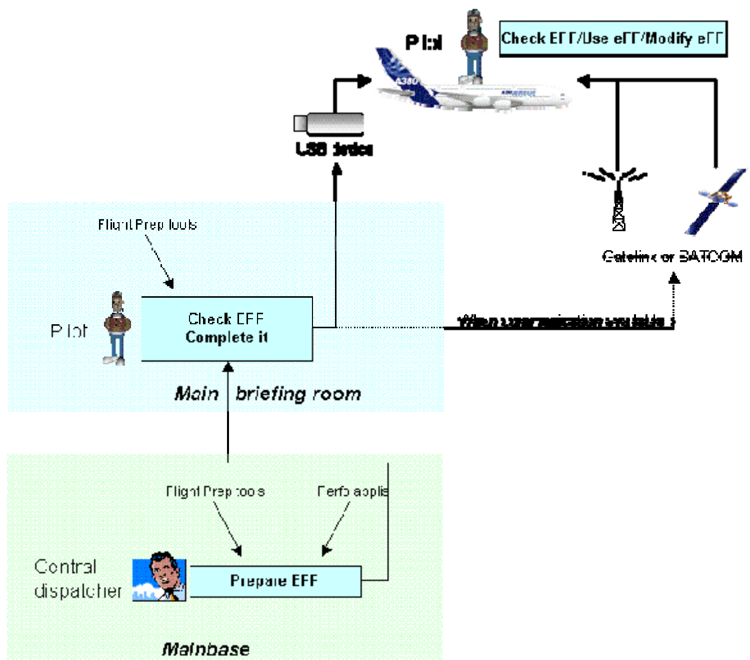
The last issue in the list above is out of scope of this document. It is the responsibility of the Applications working with the EFF.

The EFF data package is exchanged between ground and airborne applications. These applications are known as:

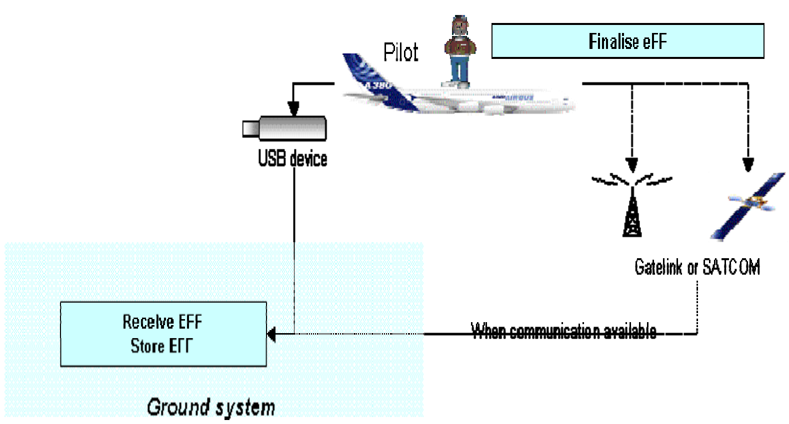
- Ground tool at mainbase: (ground application used by dispatcher to generate the eFF).
- EFF ground application: (used by the flight crew on ground at main briefing room).

- EFF airborne application (used by the crew onboard the aircraft).

During its lifecycle, the EFF data package (known as ‘EFF’) is produced, modified and processed by these 3 above mentioned applications.



Example of process for uplinking of the eFF data package (PreFlight) (for a long haul flights from Home base)



Typical Process For Downlinking Of The Eff Data Package (Postflight)

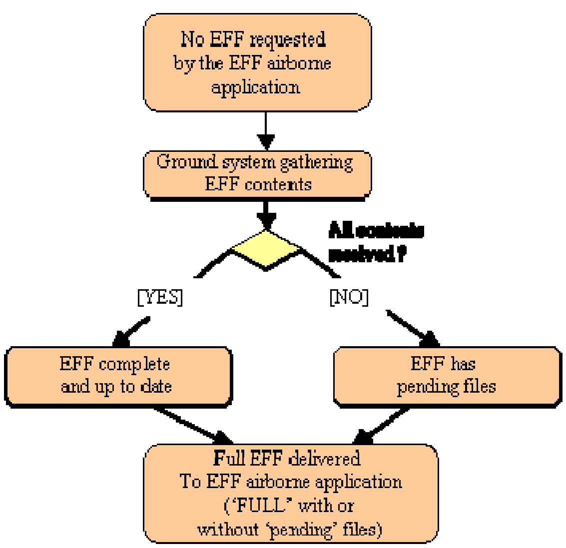
10.1.2 The EFF Lifecycle and associated communication services

The EFF has the following life cycle:

- A ground based application, e.g. an Airline Flight Data Processing System, generates EFF data.
- The eFF data package is here complete (marked as ‘FULL’), with all needed documents, or with documents or folders to be later sent (in this case these documents are identified in the EFF data package but marked as ‘pending’ in the

EFF data package). The system which composes the EFF decides when not to include a document into the EFF. This option can be used in case of a timed out request to a subsystem or for performance reasons also.

- During flight preparation, the pilot may complete the eFF data package using his ground EFF application. The result of this is generally a FULL EFF data package.
- Preflight, the whole EFF is transported from the ground application to the EFB by the use of the **EFF Uplink service** described later. Transport can be physical (USB memory stick, CD/DVD..) or via IP networks.
- Preflight or in-flight updates may be sent from the ground application to the EFF airborne application, again by the use of the EFF Uplink service.
    - In this case, the full EFF with new documents added may be sent to the aircraft (EFF data package marked as ‘FULL’ with all documents or pending documents). In this case, previous EFF data package for the same flight will be replaced by the newly received one (it may be useful to keep track of the eventual modification performed on the previous EFF data package by the crew on board the aircraft and copy them on the new EFF data package).
        - Or only an EFF update is sent to the aircraft (EFF data package marked as ‘UPDATE’): this avoids sending again all documents that were previously sent to the aircraft. In this case, the EFF airborne application will add the update to the previous updates and the last FULL EFF data package received.
- In-flight and post flight the EFF is enriched by flight crew or airborne systems.
- Parts or the whole of the EFF is sent from airborne EFF application to the ground application by the use of the EFF Downlink service.



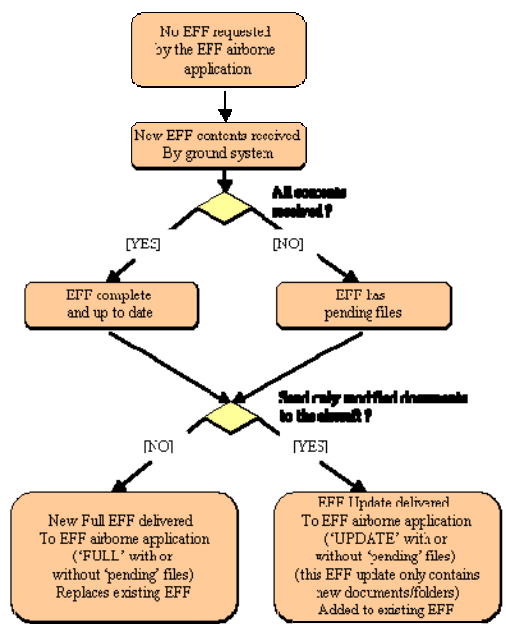
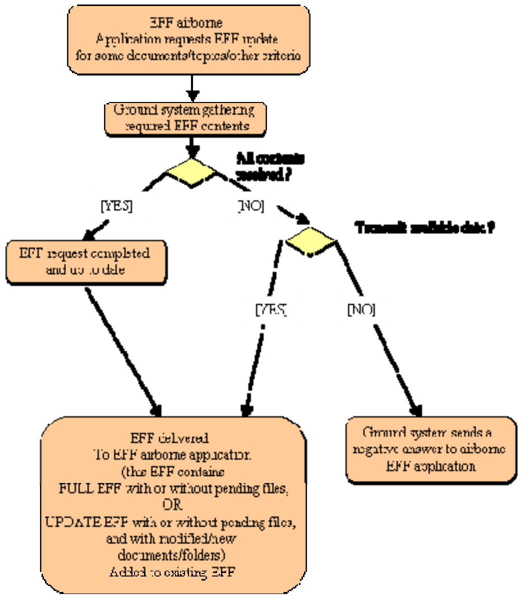


Illustration of Initial EFF generation and uplink (sent in EFUSUB message)

Illustration of Unsolicited EFF modification (sent in EFUSUB message)

10.0 ELECTRONIC FIGHT FOLDER



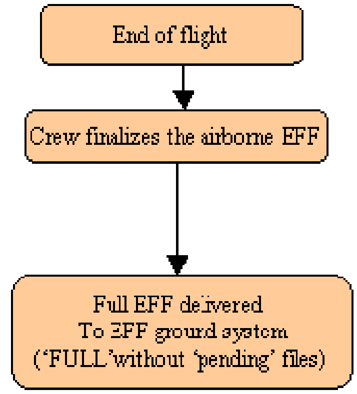


Illustration of EFF requested by Aircraft (EFUSUB message sent following EFFREQ message)

Illustration of EFF sent to ground system at flight closure (sent in EFDREP message)

Note: 'Unsollicited' means here 'unsollicited by the application' – see section 2 for details on how the transport level exchanges are managed (generally all communications are initiated by the aircraft).

10.2 EFF Structure

10.2.1 Introduction

As mentioned above the pure documents of a briefing are not sufficient to generate a usable system for a crew member who has to operate a flight leg. The documents have to be assigned to some structure and various parameters have to be provided to turn a heap of documents into a usable data package.

The main idea is to add an XML-File with the essential structural information and metadata to the documents.

The interrelation between the documents and this XML-file (which is called "eff.xml", in the following text also referenced as "structure file") is described in the following subsections.

See electronic document 'ARINC633 AOC.zip' for EFF structure specification in XML and examples. EFF structure XML schema is defined in schema/EFF/EFF.xsd.

This section is intended to provide global overview of EFF structure as well as information complementary to the EFF.xsd schema. If information contained in this section is contradictory to the EFF.xsd schema definition, the EFF.xsd schema should have precedence.

10.2.2 Documents Storable In An EFF

The standard presently does not deal with the structure of all documents that can be stored in the EFF. The standard deals only about how to attach a document in the structure. The format of the document is dependent of the tool used to display it. Considering so, one EFF tool can allow storing PDF documents for instance, whereas another one may not allow it as it may not be able to display them.

The enclosed definition of EFF XML schemas provide a predefined list of common MIME types that should be supported. (See definition of DocumentMimeType is XML schemas)

10.2.3 Documents / Operational Flight Plan Documents

Documents as they are described in the XML structure file should be of one of the following types:

- Basic documents
  - A basic document is made of:
    - The document file itself (ex : A text file, a graphic file, a PDF file).
    - Optional comment file(s) which contains annotations on the document.
  - One optional parameter file These parameters can be used by other applications, e.g.: result of a take-off performance computation.
- Operational Flight plan documents
  - Operational Flight plan documents are data that use a style sheet to be displayed. They are not associated to a static representation to be displayed.
  - Operational Flight plan documents are associated with a set of style sheets (called a display model) Display models are part of the configuration of the application. Display models shall be managed in configuration by the operator to ensure the display of information after a long period of time.
  - Further information for the Operational Flight plan can be found in paragraph 5.4, Operational Flight Plan.

10.2.4 Contents of the Electronic Flight Folder

The following files can be found in an EFF. There are three files that are mandatory: The XML structure file, the operational flight plan and the integrity check file. The rest of the contents of the EFF are highly depended on the operational procedures of the airline that operates the flight.

An Electronic Flight Folder should be made of the following files:

- One XML structure file (‘eff.xml’ file)
  - XML elements called subfolders are used to organize the documents in the structure file.
- All the document files
  - Type is arbitrary.
  - Each of them is referenced in the structure file.
  - There can be several operational flight plans (one can be declared as ‘active’).
- Comment files
  - Comment files are Optional.
  - Comments are associated to documents by the structure file.
- Parameters files
  - For documents that have associated parameters.
  - Parameter files are associated to documents by the structure file.

- Integrity check file
    - This file references all the files in the packages and provides a hashcode for each of them to ensure their integrity.
    - The modification of any of these file will modify their hashcode so that it is no more compliant with the stored hashcode in the integrity check file.

COMMENTARY

These files will be zipped together in a single '.eff' file – see Section 10.3 for details.

10.2.5 Topic Concept

The EFF brings the notion of topic concept.

The topic represents a functional identifier for one or more document(s) of the same kind. It is a technical identification of the subject of the document. This identification can be attached to each document mentioned in the XML structure file (eff.xml). This may help other applications that need to get data from the EFF to search for a technical identifier instead of a human readable identifier.

For example, an ATC flight plan can be named "Flight Plan" or "Plan de vol", or something else. If we identify the ATC Flight plan with a Topic "FlightPlan/ATC", then any application that requires getting the ATC Flight Plan can look in the EFF for the relevant topic "FlightPlan/ATC". Flight briefing tool providers are free to define additional topics. The following list shows the predefined topics which should be used when applicable.

The topic identifiers consists of one or more parts divided by a slash. The first part of the topic identifier describes the general scope (e.g. "FlightPlan" or "Weather"). Additional parts of the topic identifiers are applied cumulative. For example the topic "Weather" could be applied to all documents that contain weather information, "Weather/Text" is applied to all documents that contain textual information about weather conditions and "Weather/Text/METAR" is applied to textual METAR information.

Please note that a topic identifier can be associated to several documents but a document can refer only to one topic identifier.

See electronic document 'ARINC633 AOC.zip' for Topic specification in XML and examples. Topic XML schema is defined in schema/EFF/Topic.xsd.

This section is intended to provide global overview on eFF Topics structure as well as information complementary to the Topic.xsd schema. If information contained in this section is contradictory to the Topic.xsd schema definition, the Topic.xsd schema should have precedence.

Topic	Definition
FlightInfo	General flight information summary on the flight
FlightPlan	Gives general information on a flight plan
FlightPlan/ATCNon-ICAO	This topic should be used for the ATC-Flightplan only (for CAAs not compliant with ICAO).
FlightPlan/ATCICAO	This topic should be used for the ATC-Flightplan only (for CAAs compliant with ICAO).
FlightPlan/Operational	This topic should be used for the OFP only.
Weather	General weather information
Weather/Text	General textual weather information
Weather/Text/METAR	The airport weather topic contains the latest METAR information for all airports relevant for the planned flight
Weather/Text/TAF	The airport weather topic contains the latest TAF information for all airports relevant for the planned flight
Weather/Text/SIGMET	The SIGnificant METerological topic contains information about all significant weather phenomena in airspaces (e.g. FIRs or UIRs) relevant for the planned flight
Weather/Chart	The weather chart topic contains wind charts and significant weather charts relevant for the planned flight
Weather/Chart/Profile	Weather profile chart
Weather/Chart/Sat	Satellite image
Weather/Chart/SigWX	Signification weather chart
Weather/Chart/WindTemp	Wind & Temperature chart
NOTAM/ATC	The NOtice To AirMan topic contains other (non-weather) geo-referenced information relevant for the planned flight issued by ATC.
NOTAM/Company	The NOtice To AirMan topic contains other (non-weather) geo-referenced information relevant for the planned flight issued by the company.
NOTOC	The NOtification TO Captain topic contains dangerous goods and other information relevant of the loading of the aircraft. (See also IATA Airport Handling Manual)
NOTOC/DG	Dangerous goods
NOTOC/Others	Other notification to captain (not a dangerous good)
LoadTrimSheet	The Load and Trim Sheet topic contains information about the distribution of masses along the aircraft's longitudinal axes. See also IATA Airport Handling Manual
CrewList	The Crew list topic contains names and other information about the crew member aboard the aircraft
PaxList	The Pax list topic contains information on passengers
OceanicTracks	The Oceanic Tracksystem Track topic contains general information about Oceanic Tracks
OceanicTracks/Chart	The oceanic Tracksystem Track Chart topic contains a chart showing the current organization of the oceanic track system in graphical form
OceanicTracks/Text	The oceanic Tracksystem Track Text topic contains the coordinates of the current oceanic track system in text form
NavChart	The Navigation Chart topic contains navigational charts that are relevant to the planed flight, e.g. updates of charts
AirportInformation	General information regarding airports
SecurityInformation	General information regarding security
AirlineDefined	User defined information – use of additional parts of the topic identifier is recommended

10.2.6 Subfolders Properties

Subfolders are structural information (elements) in the XML structure file. The EFF itself contains no folders in terms of a traditional file system. Subfolder elements in the XML structure file can contain:

- Other subfolders
- Documents

A subfolder has several properties that are represented in the following table.

Attribute	Definition
title	Title of the subfolder
changed	Change information. This information describes the differences between updates of an EFF. When an EFF is fetched the first time all subfolders will have the changed status New. In the frame of incremental deliveries, it helps to know the difference between data that are New, data that existed but Revised. Unchanged may be used in the case of a complete delivery of an EFF, but partially updated. note Value — Semantics 'New' — New Subfolder; in the last received eFF package weren't such a subfolder. 'Unchanged' — Unchanged. 'Revised' — The contents of the subfolder have been updated. 'Deactivated' — The subfolder is outdated. Its contents should not be used anymore.
documentActivatable	Boolean state. Specifies that we can have an active document in this subfolder. Typically, active document are mostly used for flight plan. See Note [1] below.
activeDocument	Specifies the identifier of one document which is active in this subfolder (optional). Active notion is similar to the notion of active flight plan in the FMS. Only one document can be active any time. Active documents relates mainly to flight plan documents. An active flight plan is the flight plan known as "active" (meaning currently in use) by the FMS. It is there to be consistent with the FMS philosophy of activation. See Note [1] below.
mandatory	Specifies that documents in this subfolders or recursive subfolders must be exchanged. This attribute should be set by the ground system to specify to the airborne system which folders are mandatory in eFF downlinks.
constraint	Defines the constraints on these document subfolders. Constraints can be one of the following: • "None" : No specific constraint • "CheckOne" : The pilot is required to check at least one document in this subfolder • "CheckAll" : The pilot is required to check all the documents in this subfolder • "Sign" : The pilot is required to sign documents in this subfolder
resetConstraint	Tells when to reset the constraint: • "Never" • "OnLoadOnBoard" : When loaded on board • "OnLoadOnGround" : When loaded on ground • "OnNewDocument": When a new document is inserted in this folder • "OnLoad" : On any load
resetReadStatus	Tells when the read status (by crew operator) is to be reset: • "Never" • "OnLoadOnBoard" : When loaded on board • "OnLoadOnGround" : When loaded on ground • "OnNewDocument": When a new document is inserted in this folder • "OnLoad" : On any load
transferPending	Declares that the documents in this folder are missing for some reason. The same as if all documents in this folder where marked "transferPending". This attribute may be used for partly delivery of an EFF, e.g. when gathering of some Files would take too much time. The pending files should be delivered with the next update of the EFF. The Pilot should be informed if there is pending information and the EFF therefore is not complete.

Note: The notion of 'active' document was defined mainly for flight plan, Load and Trim sheet, Take Off performance documents – it allows the pilots to identify without ambiguity which document is to be applied for the flight. The attribute 'documentactivatable' is used to specify that this folder can contain a document tagged as 'active'. If documentactivatable' attribute is absent or not set, it means that no document should be declared active in this folder.

10.2.7 Subfolder Associations

A subfolder has several associations:

Element	Attribute	Definition
StageOfFlight	@code	@code represents the stage of flight e.g.: • Pre-flight • In-flight • Post-flight
Application	@name	Association with an application. Represents the identifier of the application on the target platform
	@callable	Specifies is the application can be called when on the target platform.
Topic	@name	Represents a topic associated to this subfolder. (See topic concept for definition of list of topics)

10.2.8 Documents Properties

The following definitions apply to all documents

Attribute	Definition
id	Technical identifier of the document. It shall be unique in the whole EFF. Even when considering an EFF with multiple updates an id must be unique in the set of files. E.g. when a document is replaced with a newer version the id must not be identical to any id of a file which was ever part of the EFF.
type	Defines the mime type associated with the document. Mime type is used as a recognized international standard for file typing. XML schemas define a standard set of MIME types that shall be supported
file	If in the eff.xml structure, this references the physical file as embedded in the .EFF package
title	Title of the Document – can be different from the filename
priority	Number between 0 and 15 indicating the priority for data transfers or other treatments. The higher the number the more important the Document.
application	This attribute intends to store the application identifier to which this data relates.
mandatory	States that this document is mandatory. Mandatory attributes shall be processed in exchanging EFF information as forcing documents to be exchanged.
Constraint	Defines the constraints on this document. Constraints can be one of the following: • “None” : No specific constraint • “Check” : The pilot is required to check one document in this subfolder • “Sign” : The pilot is required to sign documents in this subfolder
captainOnly	Information stating that only the captain shall be able to display that information
originalDateTime	Date / Time of the first release of the document in ARINC 633 format
updateDateTime	Data / Time of the last update of the document in ARINC 633 format
expectedUpdateAt	TimeDateStamp in ARINC 633 format at which it is expected that an update to the present document may be available
timeToLive	TimeDateStamp in ARINC 633 format from which on the document should no longer be presented as a valid document
status	Status of the document - Possible values are: • ‘None’ • ‘Read’ • ‘Checked’ • ‘Signed’
preliminary	Indicates Data that is provided as information only.
changed	Change information. Allowed values are: (‘New’, ‘Unchanged’, ‘Revised’; ‘Deactivated’)
authorProfile	Profile of the author of the document – Predefined values are: • Administrator • Dispatcher • CrewMember Other values can be used
AuthorName	Name of the author of the document (string)
displayModel	Associated reference to a display model. This is valid only for documents of type (Text/XML)
transferPending	Declares that this document is missing for some reason.

10.2.9 Document associations

A document has several associations (XML elements):

Element	Attribute	Definition
Parameter	name value	Used by the application to trace the context in which a document was generated (eg Performance calculation input parameters)
ParameterFile	file	Function similar to ‘parameter’ element except that all parameters are stored in a file
Signature	@date	
	@type	
	    LegalSign, or SimpleSign	These date are in fact elements and not attributes Specify the type of signature used Contains: Digest, Cert Contain: Digest, userId
Topic	@name	Represents a topic associated to this subfolder. (See topic concept for definition of list of topics)
CommentDocument		Optional commentary file associated to the document – several commentary files can be specified

10.2.10 Flight Folder Properties

Several information entries should be attached at the level of the EFF itself. The following table summarizes this information:

Element	Attribute	Definition
EFUSUB Or EFDREP	@fullpackage	Indicates that this EFF is a fullpackage and not an update. If any EFF is found on the target system with the same flight identification, this new EFF should replace the EFF that was found.
	@ initialEFFGenerationTimestamp	Will be added in a future supplement
	@ prevTimestamp	Gives the timestamp (timestamp element of m633header) of the previous eFF. This attribute is mandatory when eFF is an update (i.e. there is a previous eFF) – it should be omitted if eFF is not an update but a full version of an eFF. When an EFF is an update, this data can be used to ensure that previous update is not missing in the update process. The application may choose to discard this update if the previous one does not have this value for timestamp.
	@templateRef	This identifies the reference of the EFF template model. This is to be used when importing an EFF from the ground using the template possibilities
	@state	Optional attribute - Denotes the current state in the Lifecycle of an eff. The following values are defined: • “New” - denotes that the EFF has not been sent to an AC. • “Uplink” – the EFF is on transportation to the AC. This value is especially applicable by transport on physical media. • “OnAC” – the EFF is stored on the EFB and ready to be accessed by the applications. • “Active” – the EFF is on the EFB and currently used for operating a flight. • “DownlinkPartial” – the EFF is on transportation back to the ground. This value is especially

Element	Attribute	Definition
		applicable for transport on physical media and the flight is not finished (eFF will still evolve until flight closure (see below)). • "DownlinkClosed" – the EFF was used for operating a flight in the past and is is on transportation back to the ground – eventually stored on the ground system.
Subfolder		The EFF itself contains no folders in terms of a traditional file system. Subfolder elements in XML structure file can contain: - Other subfolders - Documents
M633Header		General information of the EFF message
M633SuppHeader		General information allowing to identify a flight
Signature	@timestamp	Signature generation date/time
	@type	
	LegalSign, or SimpleSign	These date are in fact elements and not attributes Specify the type of signature used Contains: Digest, Cert Contains: Digest, userId

The EFF flight identification is defined in ARINC 633 supplementary header (see section 4.4.3) and the EFF.xsd schema. By the nature of the typically used mobile devices there will be multiple instances of an EFF (different versions) at the same time at different locations. It is the responsibility of the involved systems to handle this circumstance correctly. For example, at a given time, the version of eFF onboard is older than the new eFF computed onground and pending for transmission. The version of eFF is identified by the timestamp of its generation. Additionnally, a given eFF contains the timestamp of the previous version. By this way, the system is able to track all versions of eFF document.

If parts of an EFF are to be updated these updates are generated as another EFF with the same flight identification as the target EFF. The "Changed" attribute is used to control how the documents are to be merged with existing EFF.

10.2.11 eFF Schema

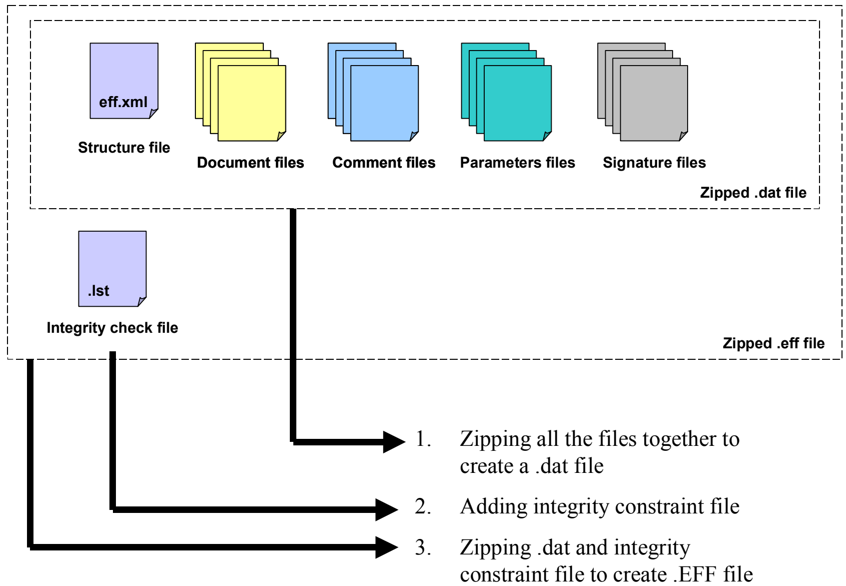
See electronic document 'ARINC633 AOC.zip' for details and examples of eff. Schemas are defined in schema/EFF/EFF.xsd.

The definition of the operational flight plan and the M633Header and M633SuppHeader are explained in section 5.2.2.

10.3 EFF File Creation

10.3.1 EFF Packaging

The above picture shows the principle of .EFF packaging:



COMMENTARY

For transportation convenience, all of these files are embedded in a zipped file (file extension '.eff'). These will easy carrying of flight information as all the files are stored in a single file. This will improve the consistency by making sure that we are not missing any file during the transportation.

EFF should be transported in the form of a unique file. Same conventions apply whether the file is transported over IP connections or via physical media.

EFF should be transported through a unique file with extension ".eff".

This file should be generated as follows:

- Creation of structure file: eff.xml file (compliant with eff.xsd' schema)
- All files referenced in eff.xml are gathered
- All these files are Zipped together in a ".dat" file
- An ".lst" integrity xml file is created (encompasses all files)
- A final '.eff' file is created, by zipping together ".dat" and ".lst" files
- The file is named as depicted in section below.

Zip algorithm

The compression format and algorithm that should be used for eFF compression are specified in RFC 1950 (ZLIB Compressed Data Format Specification version 3.3), and RFC 1951 (DEFLATE Compressed Data Format Specification version 1.3).

COMMENTARY

At the time of writing ZIP algorithm can be found through the following hyperlink:

http://java.sun.com/j2se/1.4.2/docs/api/java/util/zip/package-summary.html

10.3.2 Structure of EFF File Name

When the file is created as described above it has to get a name that makes the most important properties of the EFF visible without unpacking.

The EFF filename is defined in the following table:

Sub field name	Description	Format (see table 4.1.2)	Example
flightNum or flightID	As defined in Table 4.4.2.2.2.	(1-10)C	“DLH1234ABC
Delimiter	One Underscore		“_”
departureDate	Identifying flight date	YYMMDD	“050822”
Delimiter	One Underscore		“_”
depICAOCd or depIATACd	3-letter IATA code or 4-letter ICAO code	XXX or (exclusive) XXXX	“FRA”
Delimiter	One Underscore		“_”
destICAOCd or destIATACd	3-letter IATA code or 4-letter ICAO code	XXX or (exclusive) XXXX	“TLS”
Delimiter	One Underscore		“_”
EFF update or Full EFF	‘F’ (for Full eFF) or ‘U’ for Update	‘F’ or ‘U’	‘F’
DateTimestamp	YYMMDDHHMMSS		050822173214
File Type	One period and the letters “eff”		“.eff”

Examples:

```
LH436_050822_DUS_ORD_U050822122634.eff
UA575_050822_ORD_SEA_F050821122634.eff
CFG3214D_050401_XRY_PMI_U050402052658.eff
```

COMMENTARY

There is a possibility of inconsistency in the assignment of the Flight Date. When the current flight is the second leg of a multi-leg flight, some airlines use the scheduled date of the initial leg. While the use of the date of the current flight leg would be equally valid, it is essential that a consistent method be used within an airline.

10.3.3 Structure of the Integrity.xml File

This file references all the files in the package and provides a non cryptographic MD5 hash code for each of them to ensure their integrity, providing protection against unintentional modification.

In the integrity.xml file all contents of the eff.dat file are listed and the MD5 hash generated with the according file is given. So every application can easily check if the data in the EFF is corrupted.

The integrity.lst file should follow the structure of the sample given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<hashfilelist version="1.0" toolVersion="1.0">
   <checkcode>bzGFcO/mO5v8KFfZgEQ+lg==</checkcode>
   <hashedfile href="file1.xml">bro+2/iN72G15gj/1ZdvgA==</hashedfile>
   <hashedfile href="file2.doc">CbMgaSKuEKYwg5k74cGlIQ==</hashedfile>
   <hashedfile href="file3.txt">Ruxb8AIwEPc+XdtFCiwV4w==</hashedfile>
   <hashedfile href="file4.txt">VqzvypjT7DA2RBBRj2brHA==</hashedfile>
   <hashedfile href="file5.txt">ua0XS0ZpgE8wJcL+GbhXQw==</hashedfile>
   <hashedfile href="file6.txt">VqzvypjT7DA2RBBRj2brHA==</hashedfile>
   <hashedfile href="file7.doc">eFvP4D0RDo0XMxo4JnQIYQ==</hashedfile>
</hashfilelist>
```

where:

'version' attribute of hashedfilelist represents the version of the integrity computation function, i.e. version 1 supports only MD5 hash.

'toolversion' attribute of hashedfilelist represents the version of the tool that generates the integrity computation

<checkcode> represents the MD5 digest of the .DAT file, represented in base64 (see section 10.3.1).

<hashedfile href="file.txt"> represents the MD5 digest of the file mentioned in href.

10.4 EFF Communication Services

This EFF data package can be transmitted between airborne EFF application and ground application using the communication services defined in this section.

Three communication services are defined:

- **EFU**
  - EFF Uplink service allows an EFF data package to be sent by a ground tool to an EFB
  - EFF can be sent asynchronously or upon request (answer to EFR service)
- **EFD**
  - EFF Downlink service allows the EFF data package to be downlinked to the ground or to another EFB
  - EFF is sent asynchronously
- **EFF**
  - EFF request service allows the EFF airborne updates of an EFF data package
  - This service is synchronous (requires real time interaction between airborne and ground EFF applications)

COMMENTARY

Asynchronous means here that there is no guarantee that the receiving application is active and has received the data. However, the sending application may receive indication that the message was properly received by the ground communication system.

COMMENTARY

The use of 'EFUSUB' as root element denotes that this eFF was generated by a ground tool for use by an EFB, while the use of 'EFDREP' root element denotes that the eFF is generated by an EFB.

These services are fulfilled by the exchange of messages (elements identified below):

Element Name	Gatelink Label	Service Version	DL/UL	Purpose
EFUSUB	N/A	1	UL	Eff Uplink- SUBmit : Ground tool submits a full or update eFF to an EFB
EFDREP	N/A	1	DL	EFf Downlink REPort: An EFB exports/sends EFF to the ground or other EFB
EFFREQ	N/A	1	DL	Eff uplink- REQuest Requests a full or update (with specific criteria (e.g. some topics only…)) eFF to be uplinked.

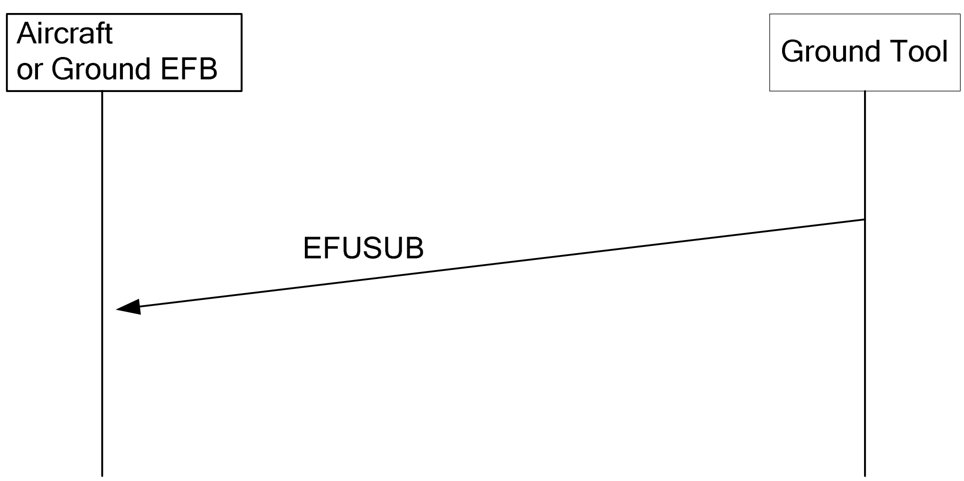
10.4.1 EFF Uplink Service (EFU)

This service is used to uplink an EFF data package to an EFB application, i.e. from a ground EFF data package generation to to the airborne EFF application or to a class 1 or 2 EFB on ground.The EFF can either be a FULL EFF, or an UPDATE. EFU service only consists in transmission from the ground to the aircraft of EFUSUB message.

EFUSUB message should contain the EFF data package to be sent to the aircraft.

EFUSUB can either be sent asynchronously, or following a specific request (EFFREQ).

Asynchronous transmission of EFUSUB message should be supported, while the support of synchronous EFUSUB message transmission following EFRREQ reception is optional.



Typical EFUSUB Exchange

COMMENTARY

While EFF application level communications only consist in simple transmission of EFUSUB message, lower communication layers are in charge of ensuring reliable transmission of this message – these lower communication layers will thus involve bidirectional communications (initiated by the aircraft for security reasons).

10.4.1.1 EFUSUB Message

Message: EFUSUB (version 1).

This message should be only sent as a .eff file in the appropriate format for the transport over physical media or IP communication links.

10.4.1.2 Dynamic Aspects

For asynchronous transmission, EFUSUB message should be transmitted either through physical media (see section 2.3) or using IP air-ground communication links (see section 2.2).

Synchronous transmission of EFUSUB message (following reception of EFRREQ message) should be performed only through IP air-ground communication links (see section 2.2).

Several EFF files can be transported simultaneously over the same media (especially when a physical media is used). When the transport system will have sent the'.eff' files to the application, the EFF application should sort out the various files according to the flight segment to which they apply, the generation time stamp, and the 'fullPackage' attribute of the root element.

Upon reception of an EFUSUB message, the EFF airborne application should determine to which flight segment the EFF data package is associated, by using the M633SuppHeader element received in eff.xml file.

Reception of a single EFF data package (one EFUSUB reception):

If a single EFF data package is received, the EFF airborne application should compose the EFF data package as follows:

- If the received EFF is FULL ('fullPackage' attribute of the root element is set), then this new EFF data package should replace the preceding one for same flight segment, if timestamp is more recent than last received EFF data package. The EFF application should keep track of the modifications done on the preceding EFF data package (the way to retain the history of modifications is implementation dependent).
- If the received EFF is an UPDATE ('fullPackage' attribute of the root element is NOT set), then this new EFF data package should be added to the current one without impacting the documents and folders not updated in the received data package. The EFF application should keep track of the modifications done on the preceding EFF data package (the way to retain the history of modifications is implementation dependent).

Reception of multiple EFF data packages (several EFUSUB receptions):

For same flight segment ('flight' attribute of M633 header with same values), if several EFF files are received simultaneously, the EFF airborne application should compose the EFF data package as follows:

- Find the most recent FULL EFF data package received ('fullPackage' attribute of the root element is set).
- Ignore all data packages generated before the last FULL EFF data package.
- Then add the more recent EFF data packages if they are 'UPDATE' data packages as defined in the single data package reception processing.

Example of multiple eff files received (multiple EFUSUB messages received):

```
UA575_050822_ORD_SEA_F050821222634.eff
UA575_050822_ORD_SEA_U050821235712.eff
UA575_050822_ORD_SEA_U050822021246.eff
UA575_050822_ORD_SEA_U050822022355.eff
```

The EFF with the earliest time stamp must be the complete EFF which was fetched first (‘F’ character in the filename and ‘fullPackage’ attribute set. The following EFFs are updates. By reading the EFFs by the order of their timestamps a EFF can be compose that has exactly the same status and contents as if all the updates happened via an IP network.

Non nominal cases

Reception of reception of invalid or out of sequence message:

- If the airborne application receives an erroneous message, for example EFDREP instead of EFUSUB, the application should at least log locally the error, and, if supported by the application, send a GERIND (general error message) to the ground system (e.g. ground administrator log).

10.4.1.4 XML Format

The .eff file for EFUSUB message should contain a eff.xml file using the EFUSUB element as root element. This allows unambiguous determination that it is an uplink EFF. ‚versionNb’ attribute of M633Header element should be set to 1.

See electronic document ‘ARINC633 AOC.zip’ for details and examples. Schemas are defined in schema/EFF/EFF.xsd.

10.4.1.4.1 Transportation Over IP

The service should here use the IP communication services defined in section 2.2.

10.4.1.4.2 Transportation Via Physical Media

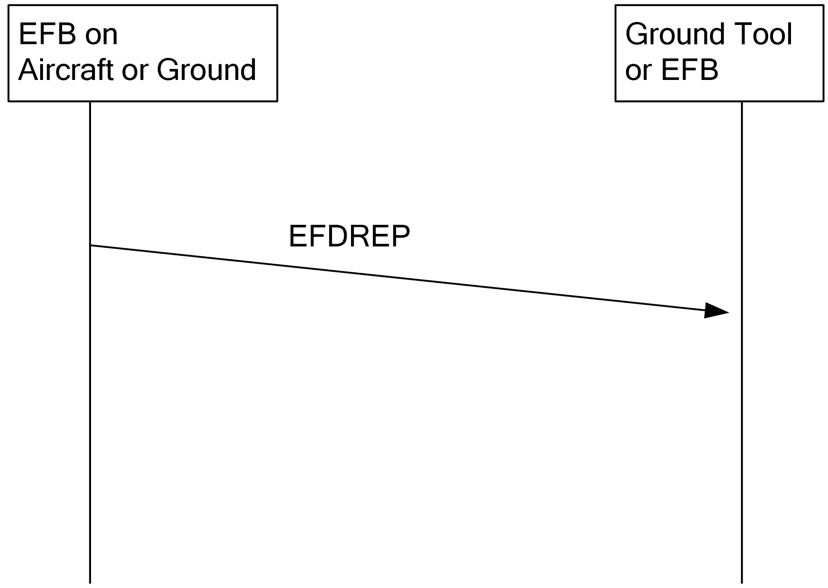
Uplink via physical media is nothing more than copying the EFF on a physical media, carrying the media to the EFB and then copying the EFF from the physical media to the EFB. The service should here use the physical media transportation rules defined in section 2.3.

10.4.2 EFF Downlink Service (EFD)

This service is used to downlink an EFF data package to the ground system or to another EFB (export-import function). The EFF can either be a FULL EFF, or an UPDATE. EFD service only consists in transmission from the aircraft to the ground of EFDREP message.

EFDREP message should contain the EFF data package to be sent to the ground.

EFDREP should always be sent asynchronously.



Typical EFDREP Exchange

COMMENTARY

While EFF application level communications only consist in simple transmission of EFDREP message, lower communication layers are in charge of ensuring reliable transmission of this message – these lower communication layers will thus involve directional communications (initiated by the aircraft for security reasons).

10.4.2.1 EFDREP Message

Message: EFDREP (version 1).

This message should be only sent as a .eff file (packaged as per Section 10.3) in the appropriate format for the transport over physical media or IP communication links.

10.4.2.2 Dynamic Aspects

EFDREP message should be transmitted asynchronously (i.e. without setting up a transaction with the ground application); either through physical media (see section 2.3) or using IP air-ground communication links (see section 2.2).

10.4.2.3 XML Format

The .eff file for EFDREP message should contain a eff.xml file using the EFDREP element as root element. This allows unambiguous determination that it is a downlink EFF. ‚versionNb’ attribute of M633Header element should be set to 1.

10.4.2.3.1 Transportation Over IP

The service should here use the IP communication services defined in section 2.2.

10.4.2.3.2 Transportation Via Physical Media

Uplink via physical media is nothing more than copying the EFF on a physical media, carrying the media to the EFB and then copying the EFF from the physical media to the EFB. The service should here use the physical media transportation rules defined in Section 2.3.

10.4.3 EFF Request Service

Up-to-dateness is essential for the usage of EFF data or to in other words: The documents in an EFF are aging rapidly. The document properties “ExpectedUpdateAt” and “TimeToLive” should be used to display the pilot when a document is outdated. In various cases the pilot might be willing to update the contents of the EFF without such a notification (e.g. WX).

Thus there have to be a possibility to get updates for an EFF that was fetched earlier. This update has to be transferred over potentially slow or expensive connections which demands the possibility to update only parts of the EFF or even single documents.

This service (eFF request service) allows the EFF airborne application to request an EFF data package update. The answer to this request should be an EFF data package tagged ‘UPDATE’ (i.e. not a FULL EFF), or a FULL eFF package (i.e. tagged ‘FULL’).

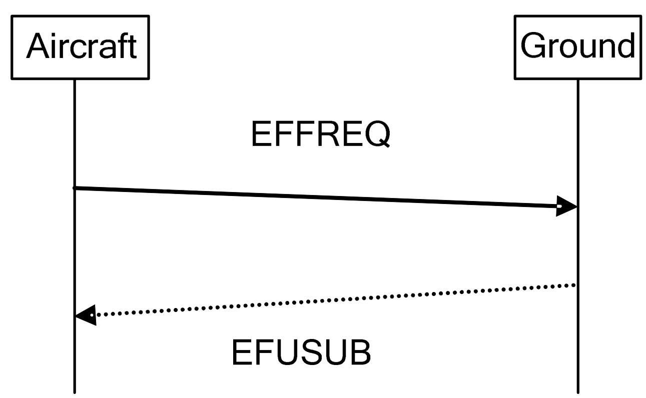
The EFF request service should thus allow to request:

- Full EFF data package containing all documents (even those that did not change since generation of last EFF data package on the aircraft).
- Update of ALL modified/new documents in a given EFF data package (identified by its flight segment and timestamp) – the answer should be a ‘UPDATE’ EFF data package containing all new or modified documents since the generation of the EFF data package referenced in the request. The answer should be a ‘UPDATE’ EFF data package containing only the documents required in the request.
- Selective updates: updates of documents of a given list of topic, or update of an explicit list of documents of a given EFF data package (identified by its flight segment and timestamp). The answer should be a ‘UPDATE’ EFF data package containing only the documents required in the request.

The key to partial updates of an EFF is the XML-Structure file – this information is common to the ground system and the EFB – as the ground system deliver this information to the EFB.

Parts of the EFF are referenced using the topic identifier. If a topic identifier is given in a update request all documents marked with this topic identifier are checked if they are outdated.

If a document ID is given only the referenced document is delivered in the update.Other criteria, such as ‘mandatory’ attribute of a subfolder, can be used.



Typical EFFREQ Exchange

10.4.3.1 EFFREQ Message

Message: EFFREQ (version 1).

This message should be only sent as a .xml file (i.e. NOT packaged as per Section 10.3) for the transport over IP communication links.

10.4.3.2 Dynamic Aspects

The answer to an update request should always be an EFF with complete information in the eff.xml structure file (EFUSUB message of EFU service). The eff.xml in this package has the 'changed' flag set properly for all files. If no full EFF is requested, documents are included only when their 'change' status is "new" or "revised".

The aircraft can specify in the request a priority level associated to the requested documents/topics for requested eFF. 3 levels are defined and, when specified, the ground should use the priority attribute as a hint to package eFF data and answer the request (trying to send first higher priority data) .

A 'document' (file tagged as 'document' in the EFF structure by using the 'document' element in the eff.xml file) can be associated with various files (if necessary), in which case these are mentionned in the EFF ('struct.eff' file):

- The document content itself.
- The associated parameters (if any).
- The associated signature (if any).
- The associated comments (if any).

With this package the EFB does not only receive a new document or new documents of the requested topic, it receives complete information about the status of all documents in the EFF.

Please note that the server which delivers the updates has to track which file was sent to whom to be able to compose the appropriate updates.

10.4.3.3 XML Format

The EFFREQ message should be embedded in an XML file, with EFFREQ as root element only.

See electronic document 'ARINC633 AOC.zip' for details and examples of EFF request. EFF request is defined in schema/EFF/EFF.xsd.

10.4.3.3.1 Request Associations

A request has several associations (see XML schema for more details).

Element	subelement	attribute	Definition
		@previousTimestamp	Time of last received eFF
FullPackage			
OR			
UpdateRequest		@pendingDocument	Request only pending documents
		@mandatoryDocument	Requests only mandatory document
	Document	@id	Document = optional element
		@transportPriority	
	Topic	@transportPriority	Topic=Optional element

All criteria are 'AND associated', i.e. if 'mandatory' is set, and topic(s) are requested, then the answer is: all mandatory documents PLUS all documents (mandatory or not) of the requested topics.

10.4.3.3.1 Transportation Over IP

The service should here use the IP communication services defined in section 2.2.

10.4.3.3.2 Transportation Via Physical Media

Not recommended – can be used if needed – however, the EFUSUB message (eFF sent from ground) could be conveyed by physical media. The airborne system should handle these situations.

11.0 De-Icing

Table 11.0-1 – De-Icing Information Table

Application Name	**De-Icing**
ACARS Label	DI (SMI=JDI)
Introduced	2005
Purpose	De-icing messages should be standardized to enable de-icing providers in directly providing the flight crew with digital information on the de-icing process. Voice frequency congestion in busy de-icing periods and therefore departure delays can be reduced.
Stake Holder	airlines, de-icing providers, airport operators
Airborne System	AOC
Ground Systems	De-icing provider dispatch system, airline or airport host, airline, airport and ATC departure management system
Network	ACARS
Downlink Routing	Label, Airport, Airline

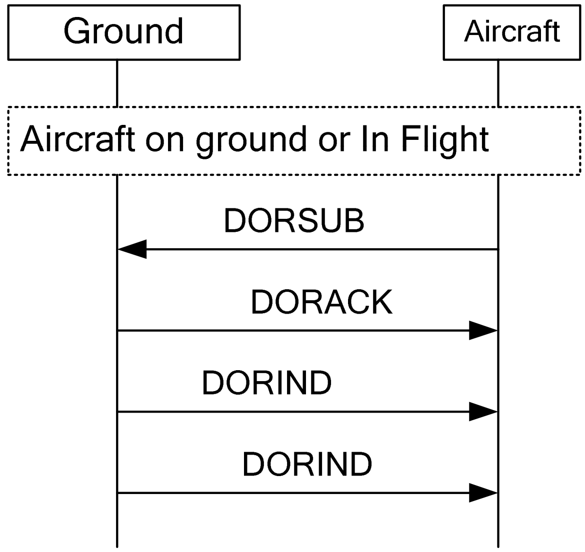
Table 11.0-2 – De-Icing Service Table

Service Name	Service ID	Description
De-Icing Order	DOR	**De-Icing Order:** The de-icing provider needs to know, that a customer wants him to serve a particular flight. The flight crew needs information on the estimated times and/or the sequence of the de-icing process.
De-Icing Receipt	DRC	**De-Icing Receipt:** The de-icing provider should inform the flight crew (and other entities), which fluids have been used for de-icing and anti-icing the aircraft and when the de-icing/anti-icing processes were conducted. With this information the pilot can calculate the hold-over time.
De-Icing Report	DPR	**De-Icing Process Report:** A de-icing application service enabling the flight crew in sending a de-icing process monitoring report and documenting the anti-icing code.

Table 11.0-3 – De-Icing Element Table

Element Name	ACARS Label	Service Version	DL/UL	Purpose
DORSUB	DI	1	DL	**De-icing order-submit:** Submits information to a de-icing providers dispatch system. Requests or cancels the de-icing of an aircraft.
DORACK	DI	1	UL	**De-icing order-acknowledge:** Confirms the reception of a de-icing request-submit downlink
DORIND	DI	1	UL	**De-Icing-order-indication:** Gives details on the de-icing order, e.g. estimated begin times
DRCSUB	DI	1	UL	**De-icing-receipt-submit:** Informs on the de-icing/anti-icing fluids and times.
DRCACK	DI	1	DL	**De-Icing-receipt-acknowledge:** Confirms the reception of a de-icing-receipt-submit uplink
DPRREP	DI	1	DL	**De-Icing report:** Flight crew confirmation of anti-icing code and de-icing/anti-icing times.

11.1 De-Icing Order Service



Typical De-Icing Order exchange

With the use of the de-icing order submit element, the de-icing provider will be informed on the need for de-icing/anti-icing an aircraft. It can be sent in all flight phases.

It can also be used to cancel the de-icing/anti-icing previously ordered.

The de-icing-order-indication element is used by the de-icing provider to inform the flight crew about time and place of the de-icing/anti-icing event.

Following subsections define version 1 of DOR service.

11.1.1 De-icing-ORder-SUBmit Downlink

Element Name: **DORSUB**

DORSUB message should convey a request for de or anti icing, together with the appropriate data.

11.1.2 De-icing-ORder-ACKnowledge Uplink

Element Name: **DORACK**

The de-icing order acknowledgement downlink should only contain the ARINC 633 application header and trailer, no application data itself.

DORACK message is optional (see Section 11.1.3 for details).

11.1.3 De-icing-ORder-INDication Uplink

Element Name: **DORIND**

The de-icing-order-indication element should be used by the de-icing provider to inform the flight crew about time and place of the de-icing/anti-icing event.

11.1.4 DOR ACARS Format

The ACARS format may be used for air-ground and ground-ground messages.

When using the ACARS format, the supplementary header should be used.

ACARS Label used is “DI”

Free Text is allowed.

Table 11.1.4-1 – DORSUB01 ACARS Definition

Parameter Name	Meaning	Field Length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter Type					
OrderFlag	0 = No De-Icing/Anti-icing required, 1 = De-Icing/Anti-Icing required	1		Boolean	No				1
AcknowledgeFlag	0 = No acknowledge wanted 1 = Acknowledge requested	1		Boolean	No				1
Parameter separator		1	comma		No				,
ServiceAirport	ICAO code of airport, where de-icing/anti-Icing shall take place or has to be cancelled	4		Airport	Yes				KJFK
Parameter separator		1	comma		No				,
De-IcingProviderID	A three digit code unique to an de-icing provider	3		provider-id	Yes				ABC
Parameter separator		1	comma		No				,
ServiceFlight	IATA flight number of flight to be de-iced/anti-iced if different from current flight	5-8		flight-no	YES				AF123A

Table 11.1.4-2 – DORIND01 ACARS Definition

Parameter Name	Meaning	Field Length (in Char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed Text	Parameter Type					
De-IcingPlaceType	P=Position, R=Remote Position (Pad)	1		A	YES	'P', 'R'	N/A		P
Parameter separator		1	comma		No				,
De-IcingPlace	Apron position identifier or Pad name	1-4		String	YES				V153, PAD3, D3
Parameter separator		1	comma		No				,
EstDe-IcingBeginTime	De-Icing/Anti-Icing begin time as calculated by the de-icing provider	4		timeMin	YES				0058
Parameter separator		1	comma		No				,
EstDe-IcingEndTime	De-Icing/Anti-Icing end time as calculated by the de-icing provider	4		timeMin	YES				0113
Parameter separator		1	comma		No				,
De-IcingSequenceNbr		2		Integer	YES	1-99			12
Parameter separator		1	comma		No				,
De-IcingOpsIndicator	N = normal operation, A = adverse weather operation	1		A	NO	'N', 'A'	N/A		N

COMMENTARY

At some airports, de-icing providers distinguish between "normal" de-icing operation and "adverse weather" de-icing operation. In the second case, de-icing/anti-icing must be conducted near the takeoff runway threshold because of very short hold-over time. For instance, after announcing "adverse weather" de-icing operation, all previously sent information on times and positions becomes invalid and only remote position de-icing/anti-icing will be performed. Instead of providing estimated begin and end times, the de-icing provider publishes the de-icing sequence number with the de-icing position, only.

11.1.5 DOR XML Schema

The IP/XML format may be used for air-ground and ground-ground messages.

When using the IP/XML format, the supplementary header should be used.

See electronic document 'ARINC633 AOC.zip' for details and examples of DOR service messages. Schemas are defined in schema/DeIcing/ DEICING.xsd.

11.1.6 Dynamic Aspects

In contrast to many local procedures, all times included in the DOR service should be UTC times.

DORSUB and DORACK should always be sent in the order specified above.

Upon reception of a correct DORSUB, the ground system (see appendix C – routing options) should forward the data to the appropriate end-system. If this is not possible, the ground system should send the appropriate GERIND message.

The de-icing provider's ground system should check the AcknowledgeFlag parameter of the DORSUB downlink. If it is set to "1", the ground system should send the DORACK message.

Upon reception of an erroneous DORSUB, the ground system should send the appropriate GERIND message.

Upon receiving a GERIND message, the airborne system should, if the DORSUB was manually triggered, display a failure message to the flightcrew. The text of the failure message depends on error class and error type.

The DORACK message should be seen as technical acknowledgement to the corresponding DORSUB element, whereas the DORIND service, by providing planning data for the de-icing/anti-icing event, can confirm the reception of an order from a commercial and dispatch point of view.

After receiving the de-icing order submit downlink, the de-icing sequence within the provider's ground system is normally being updated according to request time, scheduled departure time etc. If de-icing place and time are defined/calculated, this information will be forwarded to the requesting aircraft. More than 1 DORIND messages can be triggered within the De-icing Order service.

EXAMPLES:

Scenario 1

Based on latest ATIS information, the flight crew of an MD83 aircraft decides to request de-icing/anti-icing before the next takeoff from EKCH by sending a DORSUB downlink. A technical acknowledgement is generated from the provider's EDP-system.

Due to aircraft rotation changes, this decision is revised after onblock in EKCH. An appropriate ground cancellation confirmation is not generated because it was not required. All message envelopes have been omitted.

ACARS Example

```
DORSUB01 downlink:

,DORSUB01050324052300LFPG,,KJFK,,AFR5556,,050324,,A380,,F-WWOW,
/11,EKCH,LRV,SA345,xxxx


DORACK01 uplink:

,DORACK01050324052613LFPG,,KJFK,,AFR5556,,050324,,A380,,F-WWOW,
/,xxxx


DORSUB01 downlink:

,DORSUB01050324055944LFPG,,KJFK,,AFR5556,,050324,,A380,,F-WWOW,
/00,EKCH,LRV,SA345,xxxx
```

Scenario 2

Pre-flight, the aircrew of flight SQ103H requests de-icing, which is acknowledged by the ground system.

Later on, an information uplink on “normal” position de-icing with estimated de-icing begin time 1429 and position B12 are sent by the de-icing provider. The second DORIND message is a revision to the first one and informs about position de-icing changed to remote position de-icing at PAD2 with estimated de-icing begin time 1438 and estimated de-icing end time 1452.

Due to severe snow precipitation, at 1424 the de-icing provider announces “adverse weather” de-icing operation switching to remote position de-icing only. For detailed Information on the de-icing process the free_text field is used. After recalculating the de-icing sequence, a new uplink informing about remote position de-icing at PAD2 and sequence number 7 is triggered.

ACARS Example

```
DORSUB01 downlink:

,DORSUB01051204135850LFPG,,KJFK,,AFR5556,,051204,,A380,,F-WWOW,
/11,EDDF,,SQ103H,xxxx


DORACK01 uplink:

,DORACK01051204135859LFPG,,KJFK,,AFR5556,,051204,,A380,,F-WWOW,
/,xxxx


DORIND01 uplink:

,DORIND01051204140327LFPG,,KJFK,,AFR5556,,051204,,A380,,F-WWOW,
/1425,1450,P,B12,1429,,,N,xxxx


DORIND01 uplink:

,DORIND01051204141303LFPG,,KJFK,,AFR5556,,051204,,A380,,F-WWOW,
/1425,1427,R,PAD2,1438,1452,,N,xxxx
```

DORIND01 uplink:

```
,DORIND01051204142411LFPG,,KJFK,,AFR5556,,051204,,A380,,F-WWOW,
/1425,,,,,,,A,/DISREGARD ALL RECEIVED DE-ICING MESSAGES,xxxx
```

DORIND01 uplink:

```
,DORIND01051204142520LFPG,,KJFK,,AFR5556,,051204,,A380,,F-WWOW,
/1425,,R,,,,,A,/EXPECT DE-ICING DELAY OF 1 HOUR – STANDBY ON
135.225 FOR NEXT INFO,xxxx
```

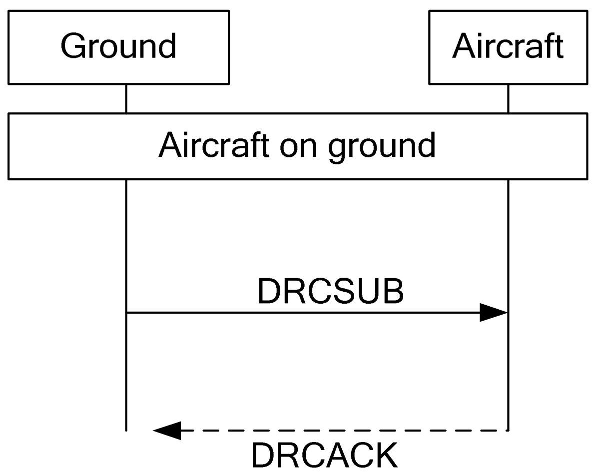
DORIND01 uplink:

```
,DORIND01051204143736LFPG,,KJFK,,AFR5556,,051204,,A380,,F-WWOW,
/1440,1443,R,PAD2,,,7,A,xxxx
```

XML Examples

See electronic document 'ARINC633 AOC.zip' for details and examples of DOR service messages. Examples are available in examples folder (DORACK Example 1.xml…..).

11.2 De-icing Receipt service



Typical De-Icing Receipt Exchange

Purpose of the De-Icing Receipt Service is to inform flight crews and airlines on the de-icing details, which are required to a) calculate the hold-over-time and b) establish a cost-effective de-icing provider monitoring process.

The De-Icing receipt element is used to provide the anti-icing code to the flight crew as well as informing on additional data regarding the De-Icing/Anti-icing process. The uplink message will be generated after the de-icing/anti-icing has been performed. Note that the volume unit used for display is optional and, if different from the metric unit liters, not the unit used for the transmission of the amount of de-icing fluid. All transmitted values are always metric and should be converted by the application into the airline's chosen unit.

Following subsections define version 1 of DRC service.

COMMENTARY

One key data of DRC service is the anti icing code received in the message that describes the fluid type and timestamp. These data are important because, when the take off clearance is received by the crew, the crew will check that the anti icing operation is still valid (timestamp acceptable). Therefore the integrity of the timestamp is important. Before going, as part of the procedure, the crew will make a visual observation of whether or not light freezing rain condition is existing. Errors in the messages, or during visual observation by the crew, can have negative effects on the flight. Airlines should perform hazard analysis to confirm with their authorities that the definition specified in this document is acceptable.

11.2.1 De-Icing Receipt Submit Uplink

Element Name: **DRCSUB**

Message DRCSUB should convey anti icing code to the crew as well as additional data on the de/anti icing process.

11.2.2 De-Icing Receipt Acknowledge Downlink

Element Name: **DRCACK**

Message DRCSUB should acknowledge correct reception and validation of DRCSUB message.

This message is optional (see details on section 11.2.5).

11.2.3 DRC ACARS Format

The ACARS format may be used for air-ground and ground-ground messages.

When using the ACARS format, the supplementary header should be used.

ACARS Label used is “DI”

Table 11.2.3.1 – DRCSUB01 ACARS Definition

Parameter Name	Meaning	Field length (in char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter Type					
De-IcingProviderID	A three digit code unique to a de-icing provider	3		provider-id	Yes				NCE
parameter separator		1	comma		No	","			,
De-IcingReceiptNbr	Commercial De-Icing receipt number	0-20		(0-20)Y	Yes	up to 20 chars			8522A
parameter separator		1	comma		No	","			,
De-Icing performed	0=No 1=Yes	1		boolean	No				0
parameter separator		1	comma		No	","			,
De-IcingFluidType	Fluid type used for aircraft de-icing	1		Y	Yes	[1-4]			2
parameter separator		1	comma		No	","			,
De-IcingFluidMix	De-Icing fluid percentage of the fluid-water mixture used for aircraft de-icing	2-3		(2-3)Y	Yes	[25-100]	1 percent		50
parameter separator		1	comma		No	","			,
De-IcingFluidVolume	De-Icing fluid volume	0-6		volume-lt	Yes	0... 999999	1 liter	liter	600
parameter separator		1	comma		No	","			,
ActualDe-icingBeginTime	Time when the process of de-icing was started	4		timeMin	Yes				1753

Parameter Name	Meaning	Field length (in char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter Type					
parameter separator		1	comma		No	","			,
ActualDe-icingEndTime	Time when the process of de-icing was finished	4		timeMin	Yes				1753
parameter separator		1	comma		No	","			,
Anti-Icing performed	0=No 1=Yes	1		boolean	No				0
parameter separator		1	comma		No	","			,
Anti-IcingFluidTyp	Fluid type used for aircraft anti-icing	1		Y	Yes, but mandatory if anti-icing was performed	[1-4]			4
parameter separator		1	comma		No	","			,
Anti-icingFluidMix	Anti-Icing fluid percentage of the fluid-water mixture used for aircraft anti-icing	2-3		(2-3)Y	Yes, but mandatory if anti-icing was performed	[25-100]	1 percent		100
parameter separator		1	comma		No	","			,
Anti-IcingFluidVolume	Anti-Icing fluid volume	0-6		volume-lt	Yes	0...999999	1 liter	liter	600
parameter separator		1	comma		No	","			,
ActualAnti-icingBeginTime	Time when the process of anti-icing was started	4		timeMin	Yes, but mandatory if anti-icing was performed				1753
parameter separator		1	comma		No	","			,
ActualAnti-IcingEndTime	Time when the process of anti-icing was finished	4		timeMin	Yes				1753
parameter separator		1	comma		No	","			,
AircraftVolumeUnitDisplay	Unit used by the IntoPlaneService (Liters, Imperial Gallons or US Gallons)	2		"LT" or "IG" or "UG"	Yes				LT

Table 11.2.3.2 – DRCACK01 ACARS Definition

Parameter Name	Meaning	Field length (in char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter Type					
Anti-IcingFluidTyp	Fluid type used for aircraft anti-icing	1		Y	Yes, but mandatory if anti-icing was performed	[1-4]			4
parameter separator		1	comma		No	","			,
Anti-icingFluidMix	Anti-Icing fluid percentage of the fluid-water mixture used for aircraft anti-icing	2-3		(2-3)Y	Yes, but mandatory if anti-icing was performed	[25-100]	1 percent		100
parameter separator		1	comma		No	","			,
ActualAnti-icingBeginTime	Time when the process of anti-icing was started	4		timeMin	Yes, but mandatory if anti-icing was performed				1753

11.2.4 DRC XML Schema

The IP/XML format may be used for air-ground and ground-ground messages.

When using the IP/XML format, the supplementary header should be used.

See electronic document 'ARINC633 AOC.zip' for details and examples of DRC service messages. Schemas are defined in schema/DeIcing/ DEICING.xsd.

11.2.5 Dynamic Aspects

In contrast to many local procedures, all times included in the DRC service should be UTC times.

EXAMPLES

Scenario 1

In the following scenario a B735 has been de-iced and anti-iced. For de-icing purposes, 600 liters of fluid type 1 with a mixture of 50% fluid and 50% hot water were used. Process started at 0624 and took until 0631. Beginning at 0631, the aircraft has then been protected against new icing with 700 liters of 100% concentrated fluid type 4. Anti-icing was finished with the generation of the DRCSUB message at 0639.

ACARS Example:

```
DRCSUB01 uplink:

,DRCSUB01050107063912LFPG,,KJFK,,AFR5556,,050107,,A380,,F-WWOW,
/NCE,,1,1,50,600,0624,0631,1,4,100,700,0631,0639,LT,xxxx

DRCACK01 downlink:

,DRCACK01050107063914LFPG,,KJFK,,AFR5556,,050107,,A380,,F-WWOW,
/4,100,0631,xxxx
```

Scenario 2

De-icing provider sends information on the anti-icing process. De-icing was not performed, but anti-icing the aircraft was conducted with 400 liters of fluid type 2 and a 75:25 fluid: hot water mixture.

However, this uplink is rejected due to missing mandatory data (de-icing/anti-icing performed indicators). Error is classified as non-fatal error, error type is syntax error.

After processing the error downlink, the de-icing service provider sends a revised DRCSUB message.

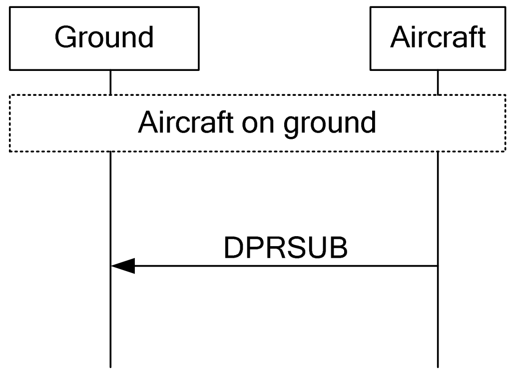
ACARS Example:

```
DRCSUB01 uplink:

,DRCSUB01050311081212LFPG,,KJFK,,AFR5556,,050311,,A380,,F-WWOW,
/ABC,E4567,,,,,,2,75,400,0801,0809,LT,xxxx

GERIND01 downlink:

,GERIND01050311081214LFPG,,KJFK,,AFR5556,,050311,,A380,,F-WWOW,
/,DRCSUB01,2,3,,1,xxxx

DRCSUB01 uplink:

,DRCSUB01050311081935LFPG,,KJFK,,AFR5556,,050311,,A380,,F-WWOW,
/ABC,E4567R,0,,,,,1,2,75,400,0801,0809,LT,xxxx
```

XML Example

See electronic document 'ARINC633 AOC.zip' for details and examples of DRC service messages. Examples are available in examples folder (DRCSUB Example 1.xml…..).

11.3 De-Icing Process Report Service



Purpose of the De-Icing Process Report Service is to document anti-icing code as well as to monitor the de-icing provider performance.

Following subsections define version 1 of DPRREP service.

11.3.1 De-Icing Report Downlink

Element Name: **DPRREP**

The De-Icing process report element is used to downlink the anti-icing code and other data related to the de-icing process to the airline's ground host. Even if not a "third-party" message, it is included to enable application designers in providing a de-icing application covering all the airline's needs.

11.3.2 DPR ACARS Format

The ACARS format may be used for air-ground and ground-ground messages.

When using the ACARS format, the supplementary header should be used.

ACARS Label used is "DI"

Table 11.3.1.1 – DPRREP01 ACARS Definition

Parameter Name	Meaning	Field length (in char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter Type					
De-IcingFluidTyp	Fluid type used for aircraft de-icing	1		Y	Yes	[1-4]			2
parameter separator		1	comma		No	","			,
De-IcingFluidMix	De-Icing fluid percentage of the fluid-water mixture used for aircraft de-icing	2-3		(2-3)Y	Yes	[25-100]	1 percent		50
parameter separator		1	comma		No	","			,
ActualDe-icingBeginTime	Time when the process of de-icing was started	4		timeMin	Yes				1753
parameter separator		1	comma		No	","			,
ActualDe-icingEndTime	Time when the process of de-icing was finished	4		timeMin	Yes				1753
parameter separator		1	comma		No	","			,

11.0 DE-ICING

Parameter Name	Meaning	Field length (in char)	Contents		Option	Range	Resolution	Unit	Example
			Fixed text	Parameter Type					
Anti-IcingFluidTyp	Fluid type used for aircraft anti-icing	1		Y	Yes	[1-4]			4
parameter separator		1	comma		No	","			,
Anti-icingFluidMix	Anti-Icing fluid percentage of the fluid-water mixture used for aircraft anti-icing	2-3		(2-3)Y	Yes	[25-100]	1 percent		100
parameter separator		1	comma		No	","			,
ActualAnti-icingBeginTime	Time when the process of anti-icing was started	4		timeMin	Yes				1753
parameter separator		1	comma		No	","			,
ActualAnti-IcingEndTime	Time when the process of anti-icing was finished	4		timeMin	Yes				1753

11.3.3 DPR XML Schema

The IP/XML format may be used for air-ground and ground-ground messages.

When using the IP/XML format, the supplementary header should be used.

See electronic document ‘ARINC633 AOC.zip’ for details and examples of DPR service messages.

Schemas are defined in schema/DeIcing/ DEICING.xsd.

11.3.4 Dynamic Aspects

In contrast to many local procedures, all times included in the DPR service should be UTC times.

EXAMPLES

In the following scenario a CRJ7 has been de-iced and anti-iced. For de-icing purposes, fluid type 1 with a mixture of 50% fluid and 50% hot water were used. Anti-icing was started at 06:31 with fluid type 3 and mixture of 75:25. The de-icing service provider did not forward any other than these data. For de-icing provider monitoring purposes, flight crew therefore entered de-icing begin (06:20) and anti-icing end-time (06:39). Free text was added by the flight crew to inform the airline operation people about new fluid types available at the departure airport. Ground host appropriately handles the free text.

ACARS Example:

```
DPRREP01 downlink:

,DPRREP01050107064155LFPG,,KJFK,,AFR5556,,050107,,A380,,F-WWOW,
/1,50,0620,0631,3,75,0631,0639,/ACC.TO PROVIDER, TYPE 4 FLUID
IS NOW AVBL IN EPWAxxxx
```

XML Example

See electronic document ‘ARINC633 AOC.zip’ for details and examples of DPR service messages.

Examples are available in examples folder (DPRREP Example 1.xml…..).

Attachment 1 AOC Application Schema

A1.0 Schema

This attachment consists of multiple XML schemas. The schemas are contained in a separate .zip file with a schema folder containing files with the XML extension of .xsd. The .zip file is available at the namespace *www.aviation-ia.com/aeec/xmlschema/633/.*

COMMENTARY

These schemas are provided to ensure application interoperability and to assist the software designer in proper coding of applications defined in this specification. While the best intent and care has been taken in their construction, no assurances can be made that they are flawless. The reader should report any omissions or inconsistencies to the ARINC IA staff using the attached errata sheet.

COMMENTARY

The schemas are located within the ARINC Store and therefore access is limited. AEEC Member Organizations (AMO) and ARINC IA Corporate Sponsors are provided a password with their membership and may download this file at no additional charge. All others will be required to place an order through the ARINC store.

Those who download the schema file will also receive the associated XML schema example files. See Appendix E.

Appendix A: Glossary & Acronyms

The following table contains acronyms and definitions used in the document text.

For the definition of parameters used in actual messages, see Appendix D.

Abbreviation	Description	Note/Clarification
ACARS	Aircraft Communication Addressing and Reporting System	In the context of this document, the acronym ACARS is usually used for the ACARS air/ground link consisting of a protocol stack defined in the 1970s
AOA	ACARS Over AVLC	A 1990s extension of the original ACARS protocol stack, allowing the use of classical ACARS messages over a modernized subnetwork called VDL Mode 2
AOC	Aeronautical Operational Control	A group or the entirety of applications used for communication of an aircraft with its airline or service partner pendants on the ground An "AOC" super-application could be hosted on a CMU or an EFB platform
ARINC	Aeronautical Radio Incorporated	
AMS	ACARS Message Security	
ATA	Air Transport Association	
ATS	Air Traffic Service(s)	Group of applications used for communication of an aircraft with Air Traffic Control entities on the ground
CA	Certificate Authority	
CG	Center of Gravity	usually expressed in percent of MAC (Mean aerodynamic coord)
CMU	Communication Management Unit	Avionics system used as an intermediate system (communication router or gateway) or endsystem.
CRC	Cyclic Redundancy Code	A method to assure integrity of messages against unattended corruption
DL	Downlink	message from aircraft to ground
DSP	Datalink communication Service Provider	e.g. SITA and ARINC
ECAM	Electronic Centralized Monitoring System	
EFB	Electronic Flight Bag	An EFB is an airborne system capable of communicating with ground systems and providing an interface to the flight crew. The EFB is not part of the classical avionics systems. Note that other ARINC documents use the acronym NSS instead of EFB
EICAS	Engine Indication and Crew Alerting System	
EFF or EFF system	Electronic Flight Folder	An application, typically hosted on an EFB, which allows pilots to access and manipulate briefing information during all flight phases.

APPENDIX A
GLOSSARY & ACRONYMS

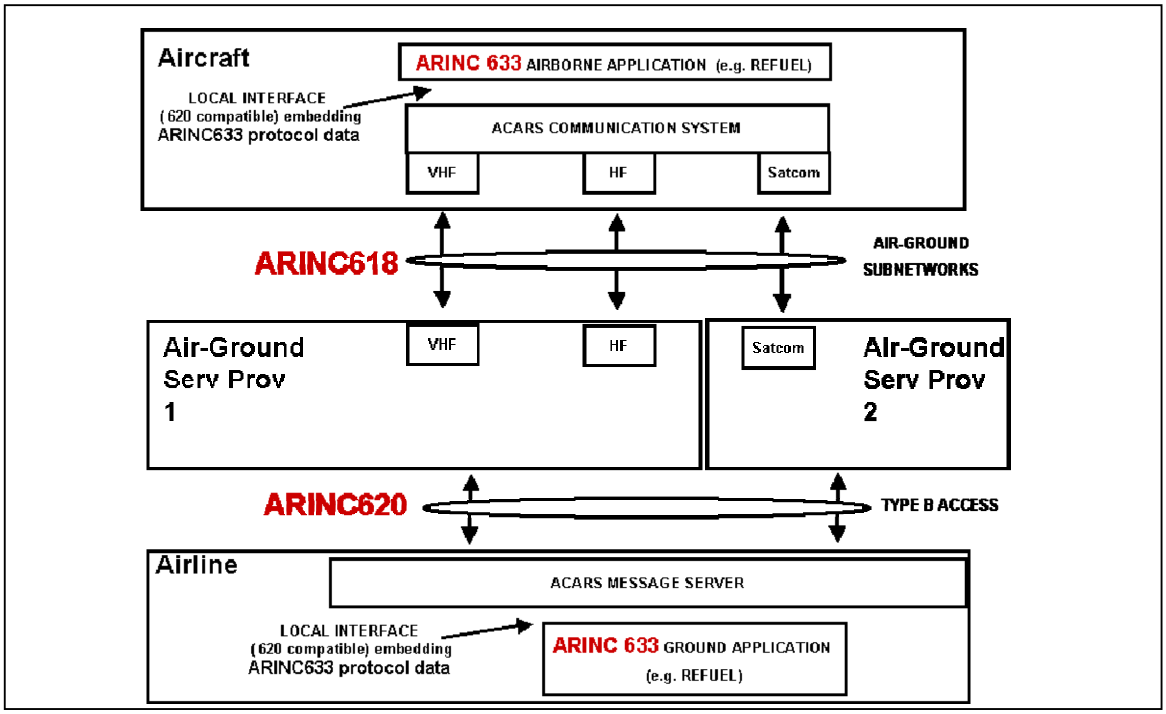
Abbreviation	Description	Note/Clarification
eff data package		The data processed by the EFF application. Always related to one specific flight sector. A full eff data package usually contains more than 100 kByte of flight related data.
FMS	Flight Management System	Avionics system used by the flight crew for navigation and other flight related tasks. A FMS can be the airborne endsystem of air-/ground messages
FOB	Fuel On Board	Amount of fuel mass contained in aircraft fuel tanks
FQMS	Fuel Quantity Management System	Avionics system managing fuel related tasks, e.g. distribution of fuel in multiple aircraft tanks
FCA	Fuel CG Advisory	
FCAIND	Fuel CG Advisory Indication	
FDA	Fuel Data	
FDAACK	Fuel Data Acknowledge	
FDACOM	Fuel Data Submit Command	
FDASUB	Fuel Data Submit	
FEN	Automatic Refuel End	
FENIND	Automatic Refuel End Indication	
FERIND	Fuel Error Indication	
FO	Fuel Order	
FORACK	Fuel Order Acknowledge	
FORSUB	Fuel Order Submit	
FPR	Fuel Process	
FPRACK	Fuel Process Acknowledge	
FPRREP	Fuel Process Report	
FQMS	Fuel Quantity Management System	
FR	Fuel Receipt	
FRCACK	Fuel Receipt Acknowledge	
FRCSUB	Fuel Receipt Submit	
FST	Fuel Status	
FSTREP	Fuel Status Report	
FSTREQ	Fuel Status Request	
FTB	Fuel auto ground Transfer Begins	
FTBIND	Fuel Transfer Begin Indication	
FTE	Fuel auto ground Transfer End	
FTEIND	Fuel auto ground Transfer End Indication	
FTI	Fuel auto ground Transfer Interrupted	
FTIIND	Fuel auto ground Transfer Interrupted Indication	
HF	High Frequency	
ICAO	International Civil Aviation Organization	
IEC	International Electrotechnical Committee	
IEEE	Institute of Electrical and Electronic Engineers	
IETF		
IP	Internet Protocol	
ISO	International Standards Organization	
LLC	Limited Liability Corporation	
LTS	Load Trim Sheet	A document or information package containing data about the aircraft's weights and weights

APPENDIX A
GLOSSARY & ACRONYMS

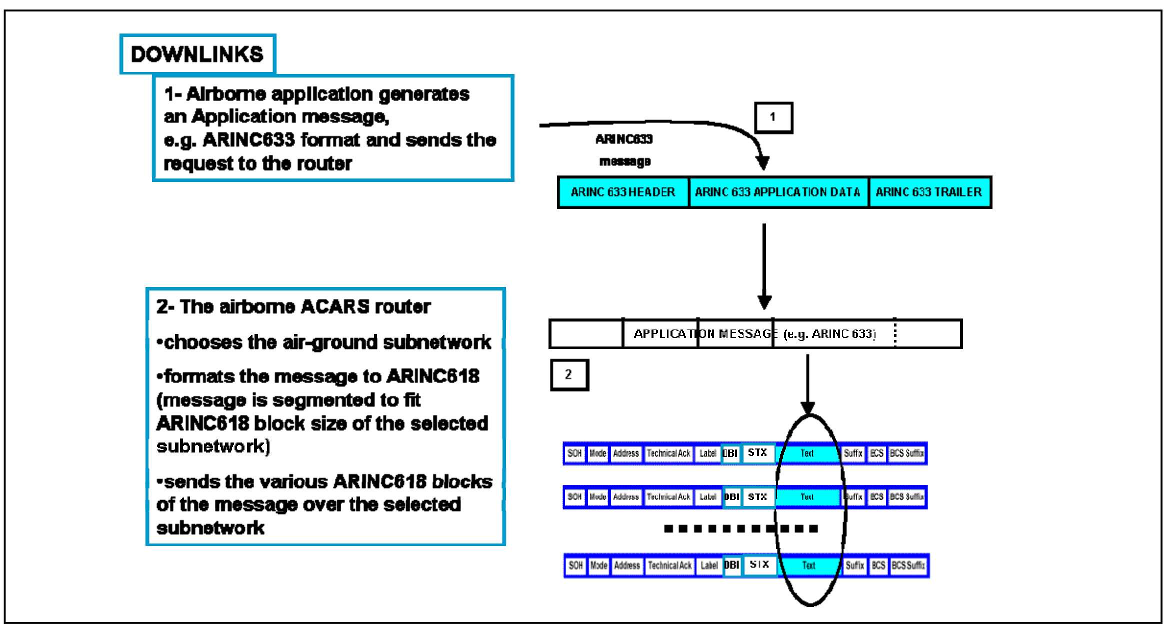
Abbreviation	Description	Note/Clarification
		distribution
MFI	Message Function Identifier	
N/A	Not Applicable	
PKI	Public Key Infrastructure	
RFC	Request for Comment	
SATCOM	Satellite Communication	
SMI	Standard Message Identifier	
TCP	Transmission control Protocol	
UL	Uplink	message from ground to aircraft
VHF	Very High Frequency	
WBA	Weight and Balance Application	
WERIND	Weight and Balance Error Indication	
WIFSUB	Weight and Balance Initialize (Full) Submit	
WIISUB	Weight and Balance Initialize (Intermediate) Submit	
WIMSUB	Weight and Balance Initialize (Minimum) Submit	
WISUB	Weight and Balance Initialization Submit	
WR1REP	Weight and Balance Report	
WR2REP	Weight and Balance Report	
XML	Extensible Markup Language	

Appendix B Tutorial on Transportation of ARINC 633 Messages Over ACARS

This appendix explains how ARINC 633, 618 and 620 interact, gives examples and references for message structure for each of these protocols.

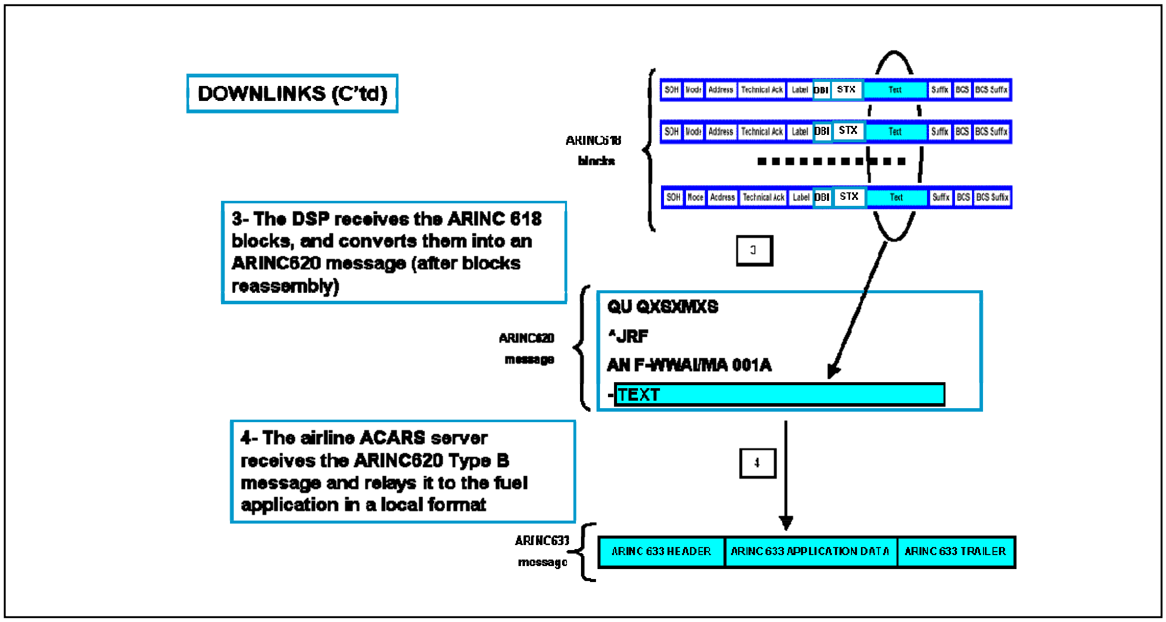


Location and Interaction Between ARINC ACARS Protocols

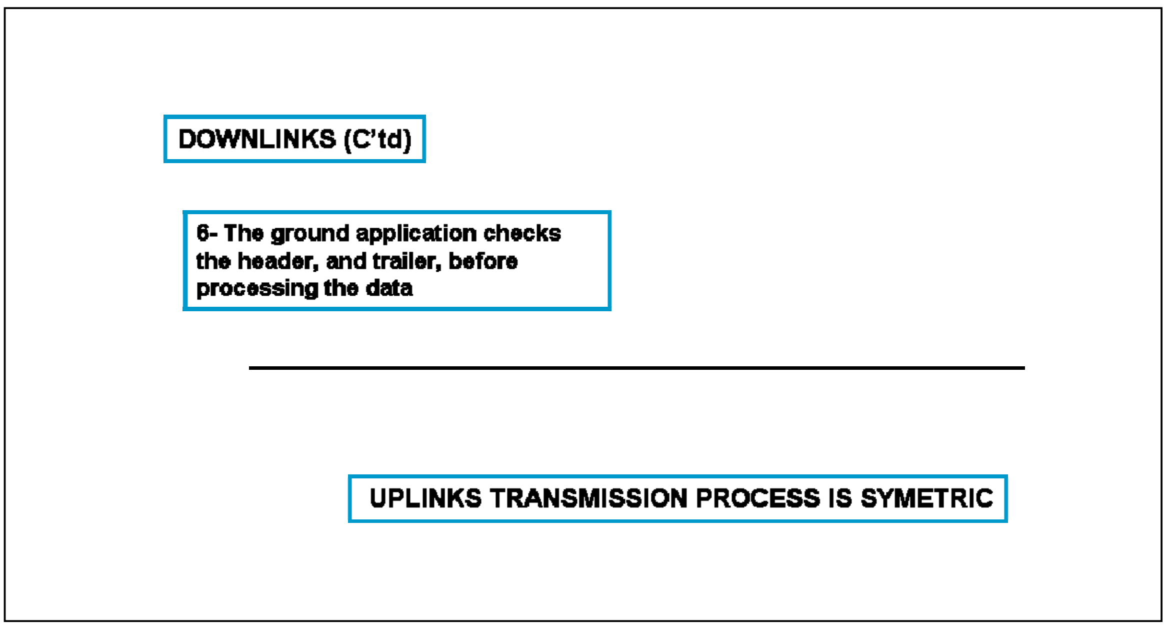


Protocol Interactions for ARINC 633 Downlinks

APPENDIX B
TUTORIAL ON TRANSPORTATION OF 633 MESSAGES OVER ACARS



Protocol Interactions for ARINC 633 Downlinks (Continued)



Protocol Interactions for ARINC 633 Downlinks (Continued)

APPENDIX B
TUTORIAL ON TRANSPORTATION OF 633 MESSAGES OVER ACARS

4.4.2.2 ACARS Message Envelope Structure

Table 4.4.2.2.1: ACARS Message Envelope

ARINC633 HEADER

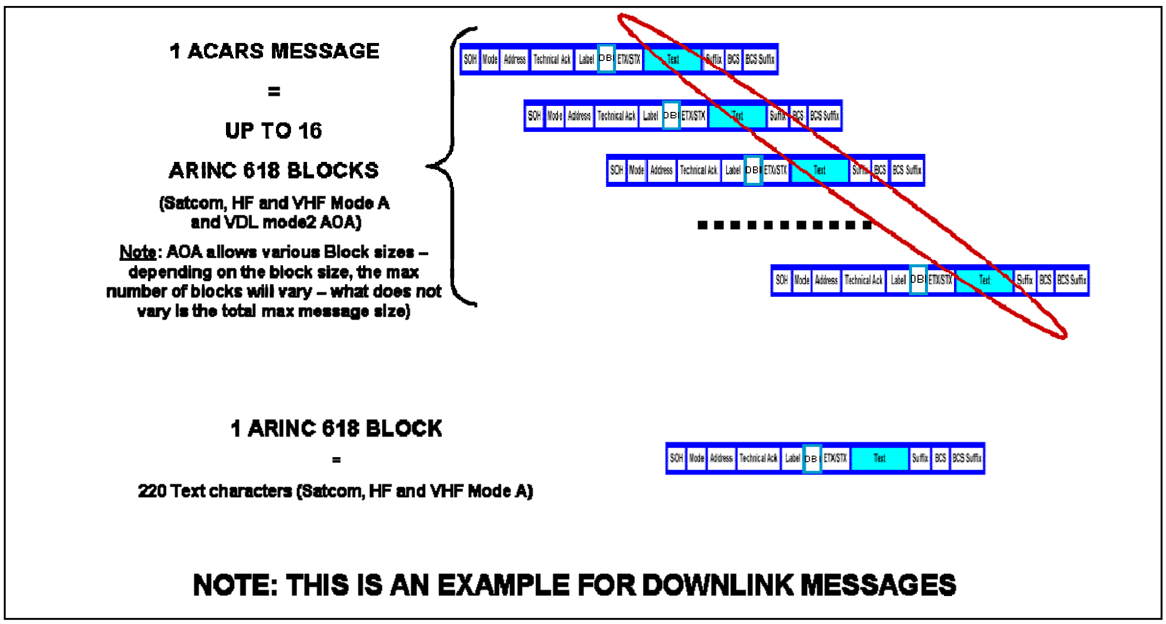
ARINC633 APP DATA

ARINC633 TRAILER

Char	Parameter Name	Format	Example	Option
0-n1	ProtectionIdentifier	(0-n)I or (0-n)J	(empty)	see note
1	delimiter	","	,	No
3	ServiceIdentifier	3char	FDA	No
3	MessageIdentifier	3char	SUB	No
2	VersionNumber	NN	01	No
12	HeaderTimestamp	datetime	050125131605	No
0-n2	Supplementary Downlink Header Element	see table below	(empty)	Yes
1	ContentElementStartCharacter	"/"	/	No
0-n3	Message Content Element		,,230000,0	Yes
1	Parameter Delimiter	"," (comma)	,	No
1	FreeTextStartCharacter	"/"	(empty)	see note
0-n4	FreeText	(0-n)I	(empty)	Yes
4	Application CRC (16 bit)	JJJJ	xxxx	No

EXAMPLE ARINC 633 FDASUB REFUEL MESSAGE IS THUS:
« ,FDASUB01050125131605/,,230000,0,xxxx »

ARINC 633 MESSAGE STRUCTURE



ARINC 618 Message Structure In Blocks

APPENDIX B
TUTORIAL ON TRANSPORTATION OF 633 MESSAGES OVER ACARS

HEREAFTER IS THE FORMAT OF AN ARINC 618 BLOCK

SAME FORMAT IS APPLICABLE TO ALL SUBNETWORKS, EXCEPT FOR VDL MODE 2 – AOA

<u>for some DSPs only</u> WHERE THE ONLY DIFFERENCE IS THE BLOCK MAX TEXT SIZE, THAT CAN BE BIGGER (1008 instead of 220 for some implementations, can go up to 2032 bytes)

<u>NAME OF FIELD</u>	**<u>FIELD LENGTH (CHARACTERS)</u>**
Start of Header	1
Mode Address	1 7
Technical Acknowledgment	1
Label	2
Uplink/Downlink Block Identifier	1
Start of Text Text Suffix	1 0 to 220 max (except for VDL2 AOA: 220 or 1008 depending on the DSP) <u>Note</u>: AOA text size= (N1 value (in bits)/8) – 31 Default N1 value=8312b → 1008 characters for SITA Min value for AOA= 2008b → 220 characters for ARINC Max value of N1=16504b → 2032 characters 1
Block Check Sequence	2
BCS Suffix	1

ARINC 618 Block Structure

UPLINK TO A LOCAL APPLICATION

NAME	SOH	MODE	address	TAK	LABEL	UBI	STX (<u>if text</u>)	APPL TEXT (<u>option</u>)	SUFFIX	BCS	BCS Suffix (<DEL>)
SIZE	1	1	7	1	2	1	1	0 - 220	1	2	1
example	<SOH>	2	.N123XX	3	10	A	<STX>		<ETX> or <ETB> for last block of msg		<DEL>

UPLINK TO A REMOTE APPLICATION (ARINC 619 USER)

NAME	<SOH>	MODE	address	TAK	LABEL	UBI	<STX>	"-"	<SP> (space char)	"#"	SUB LABEL	APPL TEXT (option)	SUFFIX	BCS	BCS Suffix (<DEL>)
SIZE	1	1	7	1	2	1	1	1	1	1	2	0-215	1	2	1
EXAMPLE	<SOH>	2	.N123XX	3	H1	A	<STX>	'-'	<SP>	#	MD		<ETB> or <ETX>		<DEL>

<u>Uplink supplementary address(es):</u>
Some uplink messages permit the inclusion of a supplementary address field. The supplementary address field of the uplink message, if present, will contain the address of the source (End system) that is providing the uplink message. In the case of ATS messages, the Supplementary address will typically be that of the ATC facility that is responsible for the airspace in which the aircraft is located, or into which the aircraft is expected to travel.
The information in the supplementary address field is used to address the response, to the message, if any. The supplementary address field may contain additional addresses, designating secondary recipients of any subsequent downlink generated by the aircraft avionics in response to the uplink message. First supplementary address can be the message MFI (2 characters only).

ARINC 618 Block Structure for Uplink

APPENDIX B
TUTORIAL ON TRANSPORTATION OF 633 MESSAGES OVER ACARS

DOWNLINK FROM A LOCAL APPLICATION

NAME	SOH	MODE	A/C Address	TAK	LABEL	DBI	STX (if text)	MSN	Flight ID	APPL TEXT (option)	SUFFIX	BCS	BCS Suffix (<DEL>)
SIZE	1	1	7	1	2	1	1	4	6	0 - 210	1	2	1
example	<SOH>	2	.N123XX	3	10	B	<STX>				<ETX> or <ETB>	|	<DEL>

DOWNLINK FROM A REMOTE APPLICATION (ARINC 619 USER)

NAME	<SOH>	mode	A/C address	TAK	label	DBI	<STX>	MSN	Flight ID	#	SUB LABEL +'B'	APPL TEXT	suffix	BCS	BCS Suffix
SIZE	1	1	7	1	2	1	1	4	6	1	3	0-206	1	2	1
EXAMPLE	<SOH>	2	.N123XX	3	H1	A	<STX>	S02A	AP1234	#	MDB		ETX		

Downlink supplementary address(es):

For some downlink message types it is possible to include Supplementary Addresses in the message text. The DSP will determine the ground address(es) of the downlink message and will add the addresses included in the Supplementary Address field to line 1 of the ground/ground message.

When Supplementary Addresses are included in the downlink, the Supplementary Address field begins with '/' and is immediately followed by the first character of the first address. Subsequent addresses in the Supplementary Address field are separated by a space character. The last address in the list is followed by a period character.

The maximum number of Supplementary Addresses is 16.

A Supplementary Address is normally either 7, 4 or 3 letters in length. All other Supplemental addresses in a single message must be of the same length. First supplementary address can be the message MFI (2 characters only).

ARINC 618 Block Structure for Downlink

REFER TO ARINC 620 SECTION 3.2.2

Title	Format	example
Destination address line	QU (priority) Destination address (airline host)	"QU XYTGK7X" (normal priority + to airline host)
Signature line	'.' Source address Date and time	".QXSXMXS 251318" (from DSP on the 25th of current month at 13h18mn
SMI line	<01h> SMI value	"^CFD" (CMS message)
Text Element line	TEI + values list separated by '/' In general: Flight identifier value and Aircraft number value	"FI AP0001/AN F-WWAI"
Communication service line	'DT' DSP RGS or VGS or GES or HFDL ground station id Date and time MSN	"DT QXS FRA2 251318 C01A"
Free text line	Dash Space Application Data	"- data" where 'data' is application data
Free text line with supplementary addresses	Dash Space *Slash* *Supplementary address #1* *Space* *Supplementary address #2* *…list of addresses separated by spaces* *Supplementary address #16* *Period* Application Data	"- /XYTGL7X XYTGK7X.data" where 'data' is application data

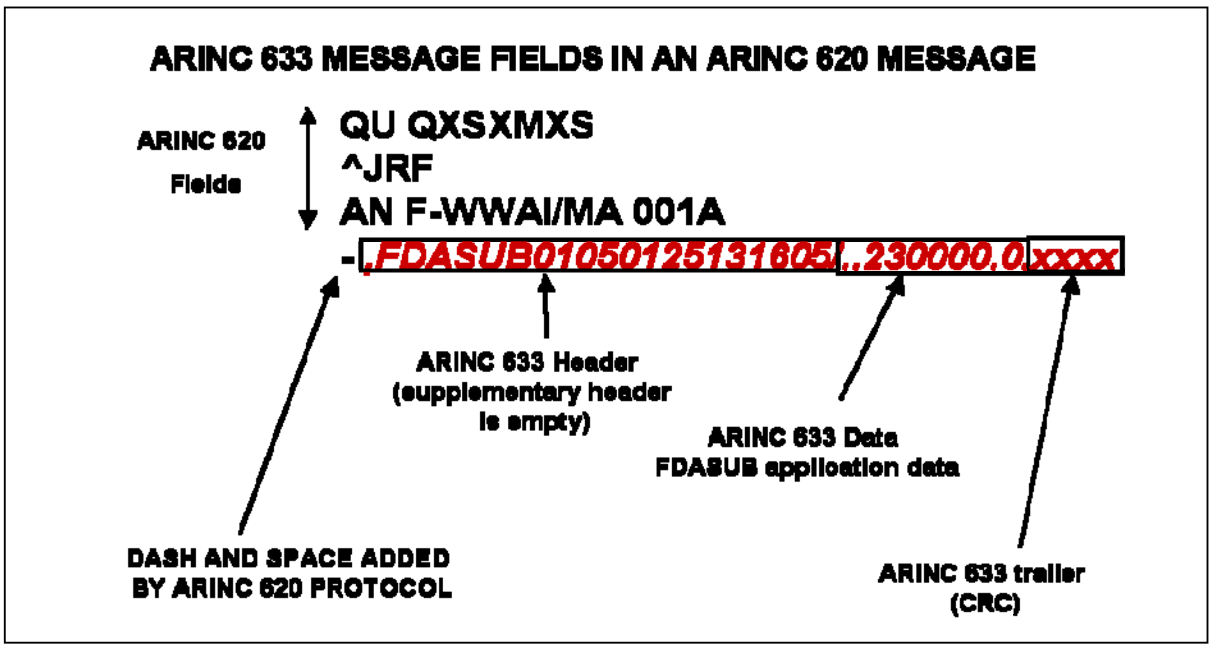
ARINC 620 Block Structure for Downlink

APPENDIX B
TUTORIAL ON TRANSPORTATION OF 633 MESSAGES OVER ACARS

REFER TO ARINC 620 SECTION 3.2.3

Title	Format	Example
Destination address line	QU (priority) Destination address (airline host)	"QU QXSXMXS" (normal priority + to airline host)
Signature line *(in general, not used for uplinks)* SMI line	'.' Source address Date and time <01h> SMI value	"XYTGK7X 251318" (from airline on the 25th of current month at 13h18mn "^CFD" (CMS message)
Text Element line *(in general, AN or FI exclusives)*	TEI + values list separated by '/' In general: Flight identifier value and/or Aircraft number value and possibly MAS element The airline host may also specify a transmission path (TP) to force transmission on a given subnetwork	"AN F-WWAI/MA 007A/TP SAT » (to aircraft number F-WWAI, message number 7, MAS requested, routed only by Satcom allowed)
Free text line	Dash Space Depends on message	"- data" where 'data' is application data
Free text line with supplementary addresses	Dash Space Depends on message	"- xxxxxxxxx" where xxxxxxxxx is application data

ARINC 620 Block Structure for Uplink



Example of an ARINC 633 Message Decoded at ARINC 620 Level

APPENDIX B
TUTORIAL ON TRANSPORTATION OF 633 MESSAGES OVER ACARS

1. GND sends fuel data submit uplink (FDASUB) (with Message Assurance required, code A aircraft ack requested)

```
QU QXSXMXS
^JRF
AN F-WWAI/MA 001A
- ,FDASUB010501251316O5/,,230000,0,xxxx
```

(where xxxx is the CRC computed on the free text)
(ARINC633 fuel message is in red and italic)

2. Reception of MAS (acknowledgement) of uplink (MA 001S: S=Success, F=fail)(optional)

```
QU QXSXMXS
^MAS
AN F-WWAI/FI AP0001/MA 001S
DT QXS FRA2 251317 S10A
```

3. Fuel Data submit ACK is sent by the aircraft

```
QU XYTGK7X
.QXSXMXS 251318
^JRF
FI AP0001/AN F-WWAI
DT QXS FRA2 251318 U01A
- ,FDAACK0105012513170 1/,,230000,0,xxxx
```

(where xxxx is the CRC computed on the free text)
(ARINC633 fuel message is in red and italic)

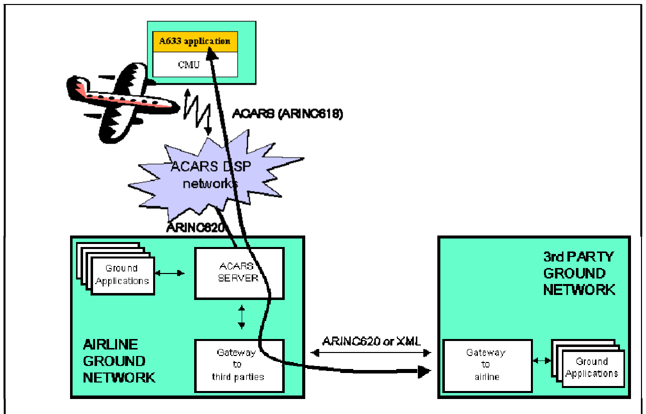
Example of Fuel Exchanges at ARINC 620 Level

Appendix C Possible Scenarios for Routing ACARS Messages to Multiple Parties

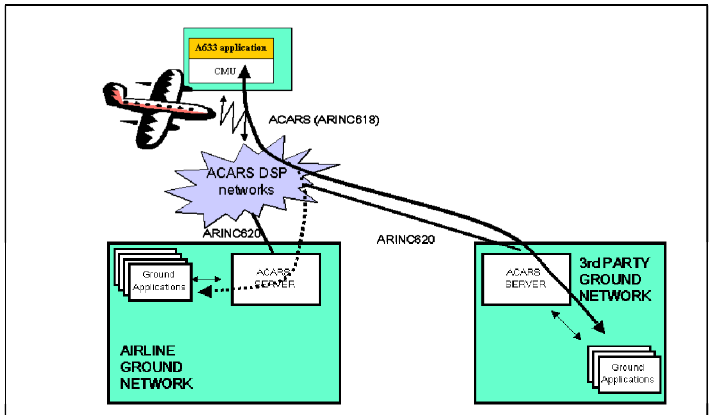
This appendix gives some examples of implementation of routing of downlink messages to multiple ground parties.

3 main scenarios are identified:

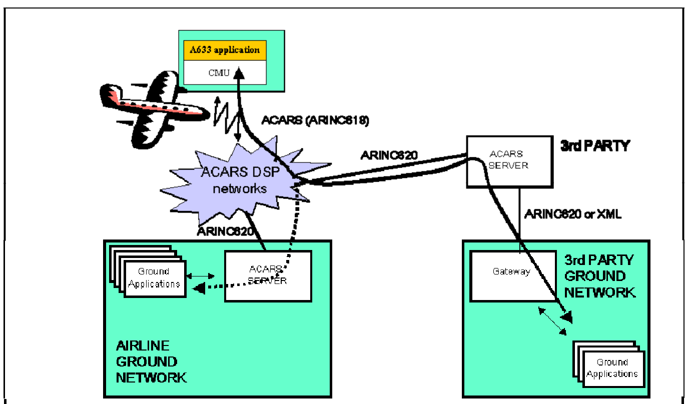
1. Centralized at airline
2. Decentralized: all parties have access to the ACARS network and DSP routes downlink messages to all interested parties (identified either statically or via supplementary addresses)
3. Centralized at a third party: maybe the DSP or any other party.



Scenario #1 for routing downlink 633 messages to ground third parties: gateway in the airline



Scenario #2 for routing downlink 633 messages to ground third parties: 3rd party has access to ACARS network



Scenario #3 for routing downlink 633 messages to ground third parties: a specific 3rd party relays the messages

Appendix D Parameter Table

The following table lists parameters that may be available to typical Datalink installations and could be used in ARINC 633 message definition.

In a given aircraft installation, only a subset of these parameters may be available. Parameters that appear in standardized messages have been added to this list.

Parameter Name	Def	Meaning	Example
A/THREngaged			
A623Connected			
AcrStatus			
Airborne			
AircraftConfiguration			
AircraftFleetIdentification			
AircraftType			
AircraftMassUnitDisplay			
AirlineIdentification			
AirportDestinationAlternate			
Altitude			
AltitudeCabinCurrent			
AltitudeCabinTarget			
APEngaged			
APEngagedMessageOnFMA			
APUAirIntake			
APUAvailable			
APUFaultPresent			
APUFuelLowPressure			
APURequired			
APUShutDown			
APUBleedValveClosed			
APUEGT			
APUFuelFlow			
APULowOilPressure			
APUOilPressure			
APUOilQuantity			
APUOilTemperature			
APUOperatingHours			
APURPMOutput			
APURunning			
APUStart			
APUStartCycles			
AverageWheelSpeed			
BaroAltitude			
Citypair			
CommandType			
CompanyRoute			
ComputedAirspeed			
CurrentDate			
CurrentFOB			
CurrentPFQ			

APPENDIX D
PARAMETER TABLE

Parameter Name	Def	Meaning	Example
CurrentPFQDate			
CurrentPFQSource			
CurrentPFQTime			
CurrentTime			
CurrentTTF			
CurrentZFCG			
CurrentZFCGDate			
CurrentZFCGSource			
CurrentZFCGTime			
CurrentZFW			
CurrentZFWDate			
CurrentZFWSource			
CurrentZFWTime			
DesiredTrack			
DestinationLatitude			
DestinationLogitude			
DistanceToDestination			
DistanceToWaypoint			
DoorCargo1Closed			
DoorCargo2Closed			
DoorCargo3Closed			
DoorCargo4Closed			
DoorEbayClosed			
DoorEntry1Closed			
DoorEntry2Closed			
DoorEntry3Closed			
DoorEntry4Closed			
DoorEntry5Closed			
DoorEntry6Closed			
DoorEntry7Closed			
DoorEntry8Closed			
DoorFlightDeckClosed			
DoorMainFwdClosed			
DoorsAllCabinClosed			
DoorsAllClosed			
DoorServiceClosed			
DriftAngle			
ECSPack1On			
ECSPack2On			
ECSPack3On			
Eng1BleedValveClosed			
Eng1EGT			
Eng1EPR			
Eng1FuelCutoff			
Eng1FuelFlow			
Eng1InletTurbineTemp			
Eng1LowOilPressure			
Eng1Mode			
Eng1N1			
Eng1N2			
Eng1OilPressure			

APPENDIX D
PARAMETER TABLE

Parameter Name	Def	Meaning	Example
Eng1OilQuantity			
Eng1OilTemperature			
Eng1Running			
Eng1TargetN1			
Eng1VibHP			
Eng1VipLP			
Engine identification			
Engine type			
ETAToDestination			
ETAToWaypoint			
FD engagement status			
FlapPosition			
Flight Phase FWC			
FlightIdentifier			
FlightNumber			
FMA modes engaged			
FMSActiveRoute			
FMSDate			
FMSDestinationAirport			
FMSDestinationRunway			
FMSFlightID			
FMSFlightPhase			
FMSOriginAirport			
FMSTime			
FOB			
FOBAccuracyState			
FQMS Fault			
FuelCenterTank			
FuelFlow			
FuelOnBoard-Unit			
FuelQuantity-Unit			
FuelDensity			
FuelOnBoard			
FuelQtyTank1			
FuelQtyTank2			
FuelQtyTank3			
FuelTemperature			
GearLeverDown			
Grossweight			
GroundSpeed			
GWCG			
GWCGAccuracyState			
ICAOAircraftAddrXPDR			
IndicatedAirspeed			
Jettison fault			
LandingCenterGear			
LandingGearOnGround			
LandingGearRetactionFault			
LandingGearStatusFailures			
LandingGearSystemFault			
LandingGearSystemInControl			

APPENDIX D
PARAMETER TABLE

Parameter Name	Def	Meaning	Example
LandingGearDown			
Mach			
MagneticHeading			
MagneticVariation			
MinimumAirspeed			
NavDatabase-Active			
OOOI1InDiscrete			
OOOI2InDiscrete			
OOOI3InDiscrete			
OOOI4InDiscrete			
OOOI5InDiscrete			
OOOI6InDiscrete			
OOOI7InDiscrete			
OOOI8InDiscrete			
OOOIOnOffInDiscrate			
OutsideAirTemperature			
PackValve1CmdOpen			
PackValve2CmdOpen			
ParkingBrakeSet			
PitchAngle			
PresentPositionLatitude			
PresentPositionLatLon			
PresentPositionLongitude			
PressureAltitude			
RadioHeightC			
RadioHeightL			
RadioHeightR			
ReceivedDatalinkPFQ			
ReceivedDatalinkPFQOutOfRange			
ReceivedDatalinkZFCG			
ReceivedDatalinkZFCGOutOfRange			
ReceivedDatalinkZFW			
ReceivedDatalinkZFWOutOfRange			
Refuel Panel Status			
Refuel Progress Parameter			
RegistrationNumber			
RollAngle			
RotatingBeaconOn			
SCI-ApplicationStatus			
SCI-Status			
SDUInstalled			
SelectedAirspeed			
SelectedAltitude			
SlideStatus			
StaticAirTemperature			
SubmittedPFQ			
SubmittedZFCG			
SubmittedZFW			
ThrottleLeverAngle1			
ThrottleLeverAngle2			
ThrottleLeverAngle3			

APPENDIX D
PARAMETER TABLE

Parameter Name	Def	Meaning	Example
ThrottleLeverAngle4			
TimeToDestination			
TimeToTopOfClimb			
TimeToTopOfDescent			
TimeToWaypoint			
TotalAirTemperature			
TrackAngleTrueValue			
TrackAngle			
TrimTankFuelQuantity			
TrueAirspeed			
TrueHeading			
TTF			
TTFAccuracyState			
Waypoint sequenced			
Weight-GWReinitRequest			
Weight-GW-ZFWUnit			
Weight-ZFW(used for performance computation)			
Weight-ZFWReinitRequest			
WeightOnWheels			
WindDirection			
WindSpeed			
WingAntiIceSwOn			
ZeroFuelWeight			

Parameter Name	Def.	Meaning	Example
acMod	acType	Aircraft Equipment subtype according to ICAO 4444	A346
aircraftMassUnitDisplay	AA	Unit used for pilot display, e.g. FMS or ECAM/EICAS	KG
applicationID	Char3Type	Identification of the airborne application	JRE
ATO		Actual Time Over a Waypoint	
commandType	boolean	1=Confirm, 0 or any other value means cancel	1
departureTime	time	Estimate of the time the aircraft leaves its parking stand	
departureDate	date	Estimate of the date the aircraft leaves its parking stand	
ETA	time	Estimate of the time the aircraft touches down	
ETimeOff	time	Estimate of the time the aircraft takes off	
ETimeIn	time	Estimate of the time the aircraft reaches parking stand	
FOB	mass-kg	The current metric value of the aircraft's Fuel On Board parameter	98100
TTK	mass-kg	The current metric value of the aircraft's Trim Tank Fuel parameter	7500

APPENDIX D
PARAMETER TABLE

Parameter Name	Def.	Meaning	Example
currentPFQ	mass-kg	The current metric value of the aircraft's Preselected Fuel Quantity parameter. This value specifies the end-quantity of the next aircraft refueling process	230000
PFQ	mass-kg	Amount of fuel that is wished to be in the aircraft after refueling. After fueling is finished PFQ should equal FOB	230000
PFQEntryDate	date	date when PFQ was last changed	050124, 061224
currentPFQSource	A	source that changed PFQ	F (for FMS)
currentPFQTime	time	time when PFQ was last changed	2329
currentZFCG	center-of-gravity	The current value of the aircraft's Zero Fuel Center-of-Gravity parameter.	385
ZFCGEntryDate	date	date when ZFCG was last changed	050124
currentZFCGSource	A	source that changed ZFCG	'A' (for Acars)
ZFCGEntryTime	time	time when ZFCG was last changed	2359
currentZFW	mass-kg	The current metric value of the aircraft's Zero Fuel Weight (Mass) parameter.	320500
ZFWEntryDate	date	date when ZFW was last changed	050125
currentZFWSource	A	source that changed ZFW	'F' (for Fms)
ZFWEntryTime	time	time when ZFW was last changed	000055
dep	airport		LFPG
dest	airport		KJFK
flightDate	String	Flight date (YYMMDD)	051231
FOB	mass-kg	The current metric value of the aircraft's Fuel On Board parameter. Amount of fuel mass contained in aircraft fuel tanks	230100
FOBAccuracyState	boolean	1=Normal, 0=Degraded	0
GW	mass-kg	Total aircraft mass. GW includes taxi fuel and therefore GW is higher than TOW	56700
GWCG	center-of-gravity	The current value of the aircraft's Gross Weight Center-of-Gravity parameter	397
GWCGAccuracyState	boolean	1=Normal, 0=Degraded	1
messageID	Char3Type		SUB
receivedDatalinkPFQ	mass-kg	Contains the metric Preselected Fuel Quantity value to which the aircraft's refuel panel is to be set. This value specifies the end-quantity of the next aircraft refueling process	230000

APPENDIX D
PARAMETER TABLE

Parameter Name	Def.	Meaning	Example
receivedDatalinkPFQOutOfRange	mass-kg	same parameter, but received value out of range	
receivedDatalinkZFCG	center-of-gravity	Contains the Zero Fuel Weight Center-of-Gravity value. An A380 needs this value to distribute the fuel correctly	
receivedDatalinkZFCGOutOfRange	center-of-gravity	same parameter, but received value out of range	585
receivedDatalinkZFW	mass-kg	Contains the Zero Fuel Weight value. An A380 needs this value to distribute the fuel correctly.	320500
receivedDatalinkZFWOutOfRange	mass-kg	same parameter, but received value out of range	
serviceID	Char3Type		FPR
submittedPFQ	mass-kg	Contains the metric Preselected Fuel Quantity value to which the aircraft's refuel panel is to be set. This value specifies the end-quantity of the next aircraft refueling process	230000
submittedZFCG	center-of-gravity	Contains the Zero Fuel Weight Center-of-Gravity value. An A380 needs this value to distribute the fuel correctly.	385
submittedZFW	mass-kg	Contains the Zero Fuel Weight value. An A380 needs this value to distribute the fuel correctly.	320500
TimeOut	time	Actual time the aircraft leaves its parking stand	
TimeIn	time	Actual time the aircraft reaches parking stand	
TimeOff	time	Actual time the aircraft takes off	
TOCG	center-of-gravity	Take-off Center of Gravity	350
TTF	mass-kg	The current value of the aircraft's Trim Tank Fuel quantity	4200
TTFAccuracyState	boolean	1=Normal, 0=Degraded	1
TTK	mass-kg	Amount of Fuel in Trim Tank. While aircraft fuel tanks are usually in the wings and the airframe between the wings, the trim tank is usually in the aft of the aircraft, e.g. the stabilizer. Therefore filling or emptying the trim tank strongly effects the aircraft's CG	4500
versionNb	Int	Version Number	1
ZFW	mass-kg	Zero Fuel Weight. Mass of a fully loaded aircraft, except for the fuel	

APPENDIX D
PARAMETER TABLE

Parameter Name	Def.	Meaning	Example
ZFWCG	center-of-gravity	Zero Fuel Weight Center of Gravity	390

Avionics Parameter Table

Editor's Note: *Parameter definitions taken from Strawman of ARINC document 821.*

Parameter Name	Def.	Meaning	Example
WhoAmI			
AirlineID	IATA 767, 1-999		
AircraftID	ICAO, 1-7 char		
AircraftType	ICAO, 4 char		
AircraftTailNumber	alpha, 1-7 char		
FlightNumber	alpha, 1-6 char		
WhereAmI			
Latitude	degrees +/-180		
Longitude	degrees +/-180		
BaroCorrectedAltitude	feet (0-131072)		
ComputedAirspeed	knots +/-1024		
TrueAirspeed	knots +/- 2048		
Mach	0 - 4.096		
TrueHeading	degrees +/-180		
StaticAirTemp	degreesC, +/-180		
VerticalSpeed	FPM, +/-32768		
Origin	alpha, 4 char		
Destination	alpha, 4 char		
DistanceToDestination	NM, 0-32768		
TimeToDestination	Minutes, 0-8192		
WindSpeed	knots, 0-256		
WindDirection	degree, +/-180		
TrueTrack	degree, +/-180		
OOOI	packed discrete	as per ARINC 758	
WhenAmI			
TimeUTC	alpha, HH:MM:SS.SS		
Date	DD:MM:YY		
WeightOnWheels	0 = ground		
FlightPhase	{Gate, Taxi, TakeOff, Climb, Cruise, Descent}		

COMMENTARY

All mass parameters in this specification have been named as weights, because the term weight is more commonly used than the term mass

Appendix E AOC Application Schema

E1.0 Examples

This appendix consists of multiple example XML schemas. The examples are contained in a separate .zip file with a schema examples folder containing files with the XML extension of .xsd. The file is available at the namespace *www.aviation-ia.com/aeec/xmlschema/633/.*

COMMENTARY

These examples are provided as guidance to assist the software designer in proper coding of applications defined in this specification. While the best intent and care has been taken in their construction, no assurances can be made that they are flawless. The reader should report any omissions or inconsistencies to the ARINC IA staff using the attached errata sheet.

COMMENTARY

The schema examples are located within the ARINC Store and therefore access is limited. AEEC Member Organizations (AMO) and ARINC IA Corporate Sponsors are provided a password with their membership and may download this file at no additional charge. All others will be required to place an order through the ARINC store.

The schema examples are integrally linked to their associated schema. Both the schema and the examples are packaged within a single folder. This folder and its contents must be downloaded for the examples to execute properly.

ARINC Standard – Errata Report

1. Document Title

ARINC Specification 633: *AOC Air-Ground Data and Message Exchange Format*
Published: June 30, 2007

2. Reference

Page Number: ______ Section Number: ______________ Date of Submission: ______________

3. Error

(Reproduce the material in error, as it appears in the standard.)

4. Recommended Correction

(Reproduce the correction as it would appear in the corrected version of the material.)

5. Reason for Correction *(Optional)*

(State why the correction is necessary.)

6. Submitter (Optional)

(Name, organization, contact information, e.g., phone, email address.)

Please return comments to fax +1 410-266-2047 or standards@arinc.com

Note: Items 2-5 may be repeated for additional errata. All recommendations will be evaluated by the staff. Any substantive changes will require submission to the relevant subcommittee for incorporation into a subsequent Supplement.

[To be completed by IA Staff]

Errata Report Identifier: ______________ **Engineer Assigned:** ________________________

Review Status: ____

ARINC IA Project Initiation/Modification (APIM) Guidelines for Submittal

1. ARINC Industry Activities Projects and Work Program

A project is established in order to accomplish a technical task approved by one or more of the committees (AEEC, AMC, FSEMC) Projects generally but not exclusively result in a new ARINC standard or modify an existing ARINC standard. All projects are typically approved on a calendar year basis. Any project extending beyond a single year will be reviewed annually before being re-authorized. The work program of Industry Activities (IA) consists of all projects authorized by AEEC, AMC, or FSEMC (The Committees) for the current calendar year.

The Committees establish a project after consideration of an ARINC Project Initiation/Modification (APIM) request. This document includes a template which has provisions for all of the information required by The Committees to determine the relative priority of the project in relation to the entire work program.

All recommendations to the committees to establish or reauthorize a project, whether originated by an airline or from the industry, should be prepared using the APIM template. Any field that cannot be filled in by the originator may be left blank for subsequent action.

2. Normal APIM Evaluation Process

Initiation of an APIM

All proposed projects must be formally initiated by filling in the APIM template. An APIM may be initiated by anyone in the airline community, e.g., airline, vendor, committee staff.

Staff Support

All proposed APIMs will be processed by committee staff. Each proposal will be numbered, logged, and evaluated for completeness. Proposals may be edited to present a style consistent with the committee evaluation process. For example, narrative sentences may be changed to bullet items, etc. When an APIM is complete, it will be forwarded to the appropriate Committee for evaluation.

The committee staff will track all ongoing projects and prepare annual reports on progress.

Committee Evaluation and Acceptance or Rejection

The annual work program for each Committee is normally established at its annual meeting. Additional work tasks may be evaluated at other meetings held during the year. Each committee (i.e., AMC, AEEC, FSEMC) has its own schedule of annual and interim meetings.

The committee staff will endeavor to process APIMs and present them to the appropriate Committee at its next available meeting. The Committee will then evaluate the proposal. Evaluation criteria will include:

- Airline support – number and strength of airline support for the project, including whether or not an airline chairman has been identified
- Issues – what technical, programmatic, or competitive issues are addressed by the project, what problem will be solved
- Schedule – what regulatory, aircraft development or modification, airline equipment upgrade, or other projected events drive the urgency for this project
- Accepted proposals will be assigned to a subcommittee for action with one of two priorities:
- High Priority – technical solution needed as rapidly as possible
- Routine Priority – technical solution to proceed at a normal pace
- Proposals may have designated coordination with other groups. This means that the final work must be coordinated with the designated group(s) prior to submittal for adoption consideration.
- Proposals that are not accepted may be classified as follows:
- Deferred for later consideration - the project is not deemed of sufficient urgency to be placed on the current calendar of activities but will be reconsidered at a later date
- Deferred to a subcommittee for refinement – the subcommittee will be requested to, for example, gain stronger airline support or resolve architectural issues
- Rejected – the proposal is not seen as being appropriate, e.g., out of scope of the committee

3. APIM Template

The following is an annotated outline for the APIM. Proposal initiators are requested to fill in all fields as completely as possible, replacing the italicized explanations in each section with information as available. Fields that cannot be completed may be left blank. When using the Word file version of the following template, update the header and footer to identify the project.

ARINC IA Project Initiation/Modification (APIM)

Name of proposed project APIM #: _____

Name for proposed project.

Suggested Subcommittee assignment

Identify an existing group that has the expertise to successfully complete the project. If no such group is known to exist, a recommendation to form a new group may be made.

Project Scope

Describe the scope of the project clearly and concisely. The scope should describe "what" will be done, i.e., the technical boundaries of the project. Example: "This project will standardize a protocol for the control of printers. The protocol will be independent of the underlying data stream or page description language but will be usable by all classes of printers."

Project Benefit

Describe the purpose and benefit of the project. This section should describe "why" the project should be done. Describe how the new standard will improve competition among vendors, giving airlines freedom of choice. This section provides justification for the allocation of both IA and airline resources. Example: "Currently each class of printers implements its own proprietary protocol for the transfer of a print job. In order to provide access to the cockpit printer from several different avionics sources, a single protocol is needed. The protocol will permit automatic determination of printer type and configuration to provide for growth and product differentiation."

Airlines supporting effort

Name, airline, and contact information for proposed chairman, lead airline, list of airlines expressing interest in working on the project (supporting airlines), and list of airlines expressing interest but unable to support (sponsoring airlines). It is important for airline support to be gained prior to submittal. Other organizations, such as airframe manufacturers, avionics vendors, etc. supporting the effort should also be listed.

Issues to be worked

Describe the major issues to be addressed by the proposed ARINC standard.

Recommended Coordination with other groups

Draft documents may have impact on the work of groups other than the originating group. The APIM writer or, subsequently, The Committee may identify other groups which must be given the opportunity to review and comment upon mature draft documents.

Projects/programs supported by work

If the timetable for this work is driven by a new airplane type, major avionics overhaul, regulatory mandate, etc., that information should be placed in this section. This information is a key factor in assessing the priority of this proposed task against all other tasks competing for subcommittee meeting time and other resources.

Timetable for projects/programs

Identify when the new ARINC standard is needed (month/year).

Documents to be produced and date of expected result

The name and number (if already assigned) of the proposed ARINC standard to be either newly produced or modified.

Comments

Anything else deemed useful to the committees for prioritization of this work.

Meetings

The following table identifies the number of meetings and proposed meeting days needed to produce the documents described above.

Activity	Mtgs	Mtg-Days
Document a	*# of mtgs*	*# of mtg days*
Document b	*# of mtgs*	*# of mtg days*

For IA Staff use

Date Received:_________________ **IA Staff Assigned:**___________________________________

Potential impact:__________

*(**A**. Safety **B**. Regulatory **C**. New aircraft/system **D**. Other)*

Forward to committee(s) (AEEC, AMC, FSEMC):___________ **Date Forward:** ____________

Committee resolution: _________

*(**0** Withdrawn **1** Authorized **2** Deferred **3** More detail needed 4 Rejected)*

Assigned Priority:_________ **Date of Resolution:** ___________________

A. – High (execute first) **B.** – Normal (may be deferred for A.)

Assigned to SC/WG:____