1. INTRODUCTION

1.1 Purpose

1.1.1 In line with normal ICAO practice, this document was developed as a companion document to the ATN Flight Information Services (Automatic Terminal Information Service) - FIS(ATIS) - SARPs. It may be read alongside the FIS(ATIS) SARPs, in order to provide a greater understanding of the specification itself, or it may be read instead of the ATN FIS(ATIS) SARPs by readers that simply want to understand the purpose of the FIS(ATIS) Application rather than the detail of the specification.

1.2 Scope

1.2.1 This document provides guidance material for those implementing of the Automatic Terminal Information Service as part of the ATN Flight Information Service Application.

1.2.2 This document does not define any mandatory or optional requirements for ADS, neither does it define any recommended practices.

1.3 History

1.3.1 The FIS application allows a pilot to request and receive FIS services from ground FIS systems. In a fully operational ATS data link environment, FIs is expected to be used as the main means of passing flight information (e.g. automatic terminal information (ATOS), notices to airmen (NOTAMs), meteorological aerodrome reports (METARs) and extracts from Aeronautical information circulars (AICs)) to aircraft, whether in flight or on the ground.

1.3.2 In the initial implementation of FIS, only ATIS information will be passed. ATIS messages, their format and intent, are based on the relevant ICAO documentation, in particular Annexes 3 and 11 and Doc 4444, Procedures for Air Navigation and Rules of the Air (PANS/RAC). The format and content of the messages will be identical to the current voice based systems.

1.3.3 The use of data link is not as flexible as voice, and a set of rules has had to be developed indicating, for example, how a dialogue is opened and closed, and how a particular sequence of messages within a dialogue is ended. However, the intention is that this should be as automatic as possible, with an apparently seamless line of communication between end users. The extent of automation will ultimately be the responsibility of the system designers, both from the engineering and operational aspects.

1.3.4 In addition to the ICAO documentation noted in paragraph 1.3.2 above, the main document from which the FIS SARPs have been developed are the Draft ICAO Manual of ATS Data Link Applications, submitted to the 2nd Meeting of the ADS Panel in September 1996. This specifies operating concepts in some detail. ICAO has specified that the FIS application should conform to the ATN protocols for its data link operations.

1.3.5 The initial development of the SARPs centred around the requirement to replace a broadcast service with a service based on individual contact between the user (the aircraft) and the provider (the ground system). An ATIS broadcast system allows the pilot to obtain current information almost exactly when he/she wants it, and if the information becomes obtrusive, it can be switched off. If at any stage in the flight a controller detects that the aircraft is not in possession of the current information, it can easily be updated by voice.

1.3.6 This functionality is replicated to the extent possible by having two basic modes of operation in a data link FIS, namely a single request capability, and a 'contract' with the ground system, which provides updates to the aircraft as and when the information is updated by the ground. In stable conditions update rates may be virtually nil, whereas during the passage of an active front the ATIS may be updated several times per hour.

1.3.7 Although states would be obliged to make data link FIS information available on request, they may not wish to implement the provision of the update functionality. States might not be willing to incur the costs of implementing a complete system if they only ever intended to use certain elements of the application. On the basis of this argument, therefore, the functionality have been separated out to enable part implementation, whilst still retaining the interoperability required by the ICAO standards. This has led to the development of subsetting rules, and the identification of conformant configurations.

1.3.8 The subgroup of ATNP/Working Group 3 which has been responsible for the production of the FIS SARPs material has worked very closely with the relevant Working Groups of the ADSP, to ensure that the development of both the operational concepts, and the technical means of achieving them keep in step with each other. However, the ADSP is generally looking at a longer timescale than the current ATNP initial implementation programme, and this will inevitably mean that some elements of their work has not been incorporated into the present SARPs.

1.4 Structure

1.4.1 Chapter 1 **-** INTRODUCTION **-** contains the reason for providing guidance material as well as the scope. In addition, it provides a brief overview of the FIS(ATIS) functionality, FIS' relationship with other SARPs, and identifies applicable reference document.

1.4.2 Chapter 2 **-** OVERALL GENERAL FUNCTIONALITY **-** describes generic concepts that are used throughout the FIS(ATIS) SARPs and guidance material. This chapter also covers some implementation issues that are not addressed in the SARPs.

1.4.3 Chapter 3 **-** FIS SERVICE DESCRIPTION **-** gives a functional breakdown of the various services that CM provides. It describes a peer to peer interaction, including reasons for why particular information is used or not used, and what operations on the information are expected.

1.4.4 Chapter 4 **-** FIS(ATIS) CHAPTER **-** clarifies any functionality that was not addressed in Chapter 3 on a chapter by chapter basis.

1.4.5 Chapter 5 **-** DIMENSIONS **-** gives some sample encoding sizes for guidance on what capacities need to be allowed for in order to meet normal operational expectations.

1.4.6 Chapter 6 **-** INDEXES / TABLES **-** is tbd.

1.4.7 Chapter 7 **-** EXAMPLE SCENARIOS **-** gives some examples as to what typical scenarios one can expect in course of normal CM operation.

1.4.8 Chapter 8 **-** EXAMPLE ENCODING **-** outlines some actual sample PER encoding of typical FIS(ATIS) messages.

1.5 FIS(ATIS) Data Link Application Overview

1.5.1 In a data link ATS system, flight-related information (e.g. meteorological information and situational awareness) can be made available to aircraft in digital form. This information will assist the pilot by increasing flight safety and improving situational awareness. Currently Flight Information Services (FIS) information is provided primarily through voice channels.

1.5.2 There are multiple DFIS services that may be supported by the FIS Data Link Application, including:

a) Automatic Terminal Information Services (ATIS),

b) Aviation Routine Weather Report (METAR) Service,

c) Aerodrome Forecast (TAF) Service,

d) Precipitation Map Service,

e) Terminal Weather Service (TWS),

f) Windshear Advisory Service,

g) Pilot Report (PIREP) Service,

h) Notice to Airmen (NOTAM) Service, and

i) Runway Visual Range (RVR) Service.

1.5.3 Each of these services is accessed and used independently of the others and is initiated by the aircraft avionics (and/or pilot). It is not required that aircraft avionics include the capability for all of the FIS services.

1.5.4 The ICAO Manual of ATS Data Link Applications [ref. x.] describes two initial services provided through the FIS Application: ATIS Service and METAR Service.

1.5.5 The FIS(ATIS) Application described in the FIS(ATIS) SARPs allows a pilot to request and receive ATIS information from ground FIS systems via data link. It provides both air and ground users with the FIS Data Link Service limited to the ATIS information. The ATIS data link service supplements the existing availability of ATIS as a voice broadcast service, provided at aerodromes world-wide. All types of ATIS currently in use are encompassed (i.e. arrival, departure and combined).

1.5.6 The ATIS data link service has been specified to meet the requirements of the Civil Aviation Community and gives the following technical benefits to its users:

- reduced flight crew workload: ATIS information does not need to be copied by the flight crew if the message is printed on a cockpit printer or is recallable on a data link display. Also, the flight crew does not have to divert attention from ATS operational channels to listen to the ATIS broadcast;
- reduced ambiguity in the transmitted information: data link implementation eliminates potential misinterpretation resulting from poor transmission quality; and
- potential for increased accessibility to ATIS information: flight crew are able to request timely ATIS information from any aerodrome capable of providing data link information.

1.5.7 The aircraft (pilot and/or avionics) requests the service by generating a request message for transmission to a FIS ground system. A FIS contract is then established by the FIS service provider which could take one of the two following forms:

a) the *FIS Demand Contract* where the ground FIS system provides the information immediately and once only, and

b) the *FIS Update Contract* where the ground FIS system provides the information and any subsequent update of this information.

1.5.8 These two types of FIS contract have been identified based on the analysis of the ATIS and METAR services. It is likely that additional type of contracts (e.g. Periodic Contract) will be identified to support other DFIS Services.

1.6 Inter-relationships with Other SARPs

1.6.1 The pre-requisite for the establishment of any FIS contract is the knowledge by the airborne system initiating the dialogue of the ATN name and address of the contacted ground FIS system. The Context Management (CM) application is the most natural way to retrieve this addressing information. A CM-logon exchange is therefore required between the aircraft and the ground CM system administrating the contacted FIS system. There is no direct interaction between then CM application and the FIS(ATIS) application (in fact there are not running at the same time), but the addressing information exchanged by CM is required to establish a FIS contract.

1.6.2 The FIS SARPs also make use of the Upper Layer Application SARPs (ref xx) to perform dialogue service functions required by the FIS(ATIS) application.

1.7 SARPs Chapter Overview

1.7.1 All the air-ground SARPs are produced to a standard format. This has greatly helped the maintenance of document stability, commonality and presentation. FIS(ATIS) SARPs are no different in basic layout from all other air-ground applications SARPs.

1.7.2 The FIS(ATIS) SARPs constitute the fourth part of sub-volume 2.

1.7.3 Chapter 2.4.1 - INTRODUCTION - gives a very brief, high level description of FIS, as an application enabling FIS services to be provided to a pilot via the exchange of messages between aircraft avionics and ground FIS systems. Since this overview contains no information directly related to the stipulation of specific standards, it is almost entirely written as series of informative notes.

1.7.4 Chapter 2.4.2 - GENERAL REQUIREMENTS - contains information and high level requirements for the maintenance of backward compatibility and error processing.

1.7.5 Chapter 2.4.3 - ABSTRACT SERVICE - defines the abstract service interface for the FIS Application. The FIS Application Service Element (FIS-ASE) abstract service is described from the viewpoint of the FIS-air-user, the FIS-ground-user and the FIS-service-provider.

1.7.6 Chapter 2.4.4 - FORMAL DEFINITION OF MESSAGES - describes the contents of all permissible FIS messages through definition of the FIS ASN.1 abstract syntax. All possible combinations of message parameters and their range of values are detailed.

1.7.7 Chapter 2.4.5 - PROTOCOL DEFINITION - splits up the specification of the FIS(ATIS) protocol into three parts: sequence diagrams for the services covered by the abstract service, protocol descriptions and error handling for the FIS-Air- and Ground-ASEs, and State Tables.

1.7.8 Chapter 2.4.6 - COMMUNICATION REQUIREMENTS - specify the use of Packed Encoding Rules (PER) to encode/decode the ASN.1 message structure and stipulates the Dialogue Service requirements, including Quality of Service (QoS).

1.7.9 Chapter 2.4.7 - FIS USER REQUIREMENTS - describes the requirements imposed on the FIS-users concerning FIS messages and interfacing with the FIS-ASEs.

1.7.10 Chapter 2.4.8 - SUBSETTING RULES - specifies conformance requirements which all implementations of the FIS protocol obey. The protocol options are tabulated, and indication is given as to whether mandatory, optional or conditional support is required to ensure conformance to the SARPs. These subsetting rules will permit applications to be tailored to suit individual ground implementations, commensurate with the underlying task, while still maintaining an acceptable level of interoperability.

1.8 References

<Editor's note: add references to ADSP Manual, ULA SARPs and CM SARPs>

2. OVERALL GENERAL FUNCTIONALITY

2.1 General

2.1.1 There is not one FIS Data Link Application but several FIS(DFIS Service) Data Link Applications. There are as many FIS Applications as there are DFIS services. The FIS(ATIS) SARPs describe one of these applications: the FIS(ATIS) Data Link Application[1].

2.2 Topology

2.2.1 The FIS Application functions on a client/server schema. FIS ground systems are servers providing FIS services to any client being the data link equipped aircraft hosting the FIS application.

2.2.2 A FIS ground system maintains FIS information related to one or several DFIS services. For instance, a FIS ground system may be specialized for providing ATIS information only, or both ATIS and TAF/METAR information. A FIS ground system hosts as many types of FIS applications as different DFIS services are provided by this system. Each FIS application is identified by an Application Name and an ATN address (both standardized and attributed by ICAO).

2.2.3 A FIS(ATIS) ground system maintains ATIS information related to one or several airports. For instance, a FIS(ATIS) system may be implemented in one airport to provide the ATIS of this airport or a centralized FIS(ATIS) system may provide the ATIS for several airports.

2.2.4 This flexible architecture permits multiple implementation choices concerning the ground topology definition. This choice is made by the authorities providing the FIS information based on their own criteria: accessibility of the FIS information, existing communication infrastructure, distribution of the FIS addressing access points, etc...

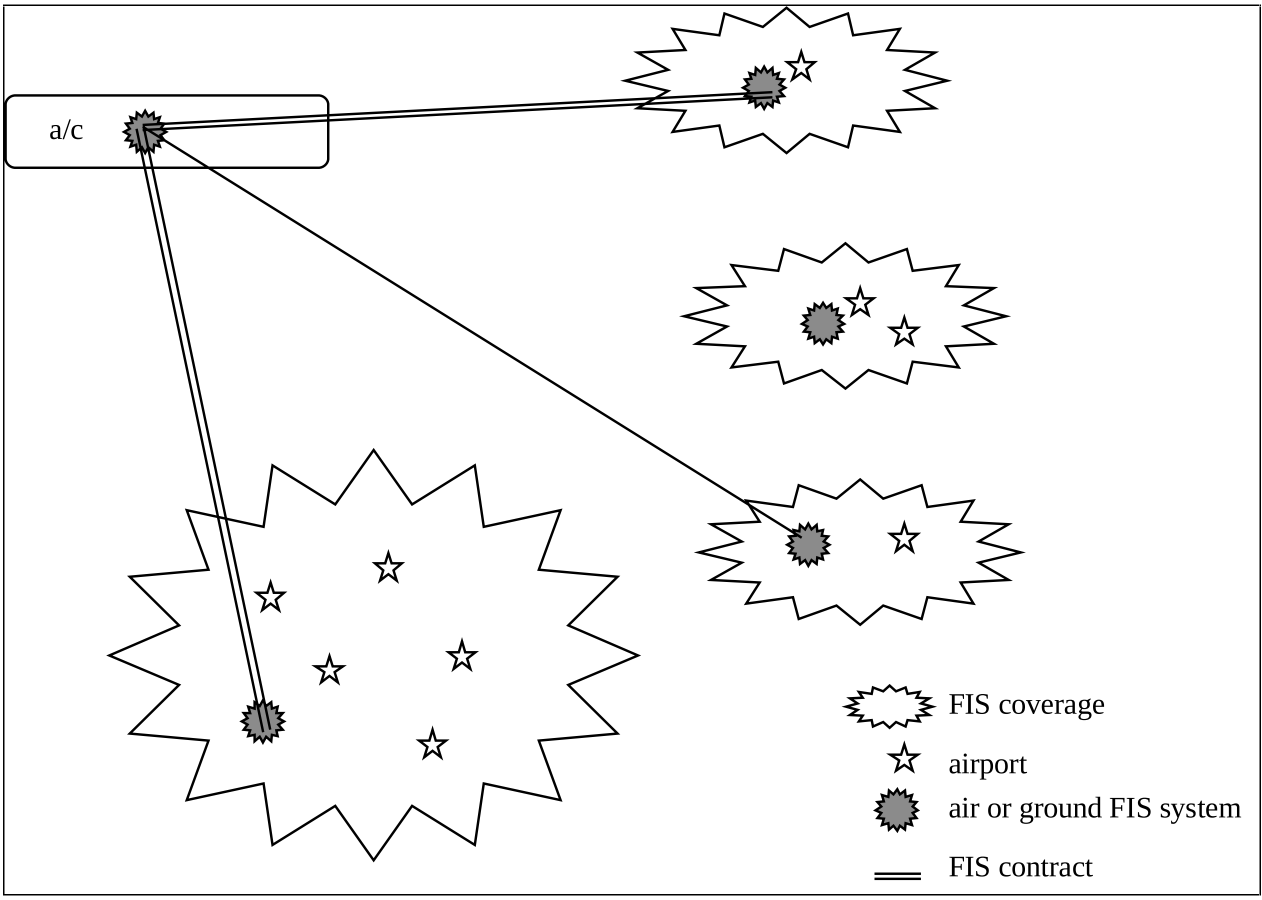
2.2.5 The selection of a particular ground topology does not impact the operational procedures for FIS. A FIS equipped aircraft inter-operate with centralized and non-centralized FIS ground systems. However, as the airborne system must know the ATN address of the contacted FIS ground systems, the number of CM-logon exchanges before establishing a FIS contract is tied with the type of these systems: one CM-logon is needed for a FIS ground system centralized at the country scale whereas several CM-logon may be required when several FIS ground systems cover the country.

2.2.6 The following topologies may be envisaged:

1. the FIS(ATIS) ground system is tied to one airport (i.e. it contains the ATIS information of that airport only),
2. a centralized FIS(ATIS) ground system is tied to several airports (of a country or a region),
3. the ground resources used for providing the ATIS service can be used for providing other types of FIS information.

---

[1] In this document, "FIS Application" is a generic term referring to all FIS(XXX) Data Link Applications whereas "FIS(ATIS)" Application refers to the specific instance of the FIS Application developed for providing the ATIS service.

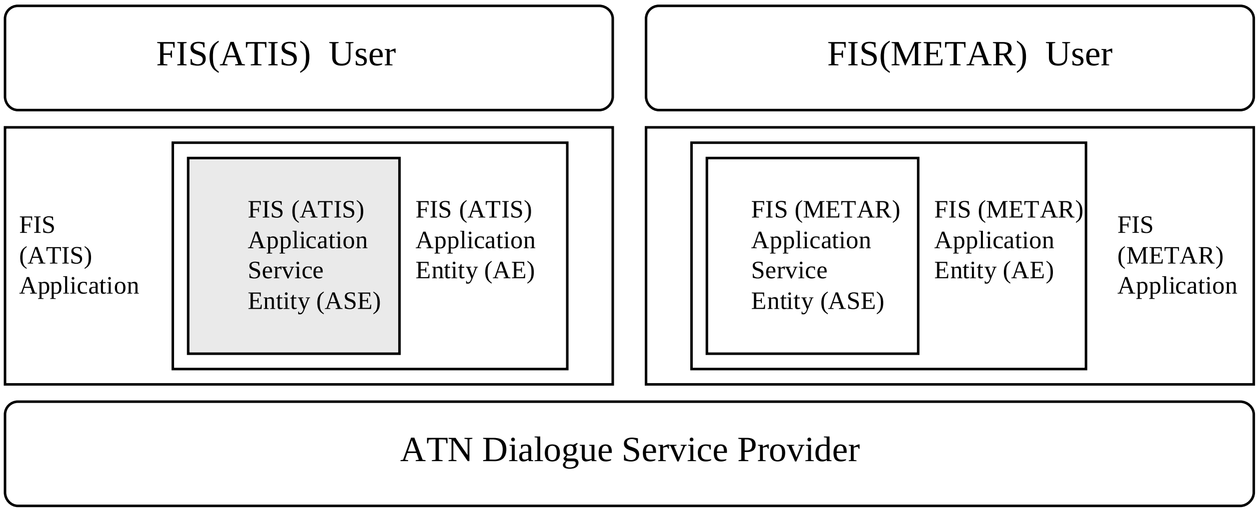


2.2.7 How the information is conveyed and updated between the FIS ground systems and the actual sources of the FIS information (airports for ATIS) is not addressed by the SARPs. The ground exchanges are of the responsibility of the administrative authority producing the FIS information and are defined locally or by bilateral agreements between authorities.

2.3 Internal Architecture

2.3.1 The architectural model for the FIS Application conforms to the Upper Layer Architecture (ULA) SARPs (ref. xx):

- one Application Entity (AE) is defined per FIS Application. The FIS(ATIS) SARPs specify the FIS(ATIS) Application Entity.
- each AE contains an ATN Application Service Element (ASE), which is the communication element responsible for the ATN Application. The FIS(ATIS) SARPs specify the FIS(ATIS) Application Service Element. There is no architectural capability for multiple instances of the FIS(ATIS) ASE within the same FIS(ATIS) AE. This implies that the FIS(ATIS) ASE will generate and manage only one dialogue over the lifetime of the ASE.
- the internal structure of the ATN ASE may be of arbitrary complexity. The FIS(ATIS) ASE contains four modules which are described later.
- the dialogue service is the ATN ASE's view of the ATN Upper Layer Architecture. The modules mentioned above use the Dialogue Service for communication with the peer ASE through the ATN.



2.3.2 The FIS(ATIS) SARPs describe FIS(ATIS) Application Service Entity. The service provided by the FIS(ATIS) ASE at its top interface is provided as such to the FIS(ATIS)-users.

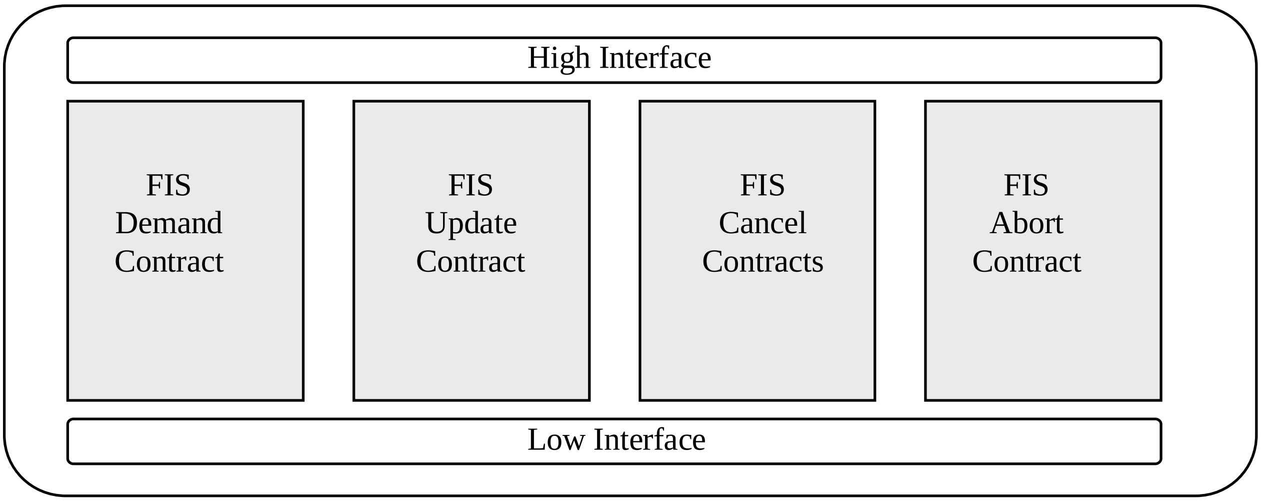
2.3.3 The FIS(ATIS)-air-user is responsible for initiating the FIS contracts and making the FIS-reports available to the end user, i.e. the pilot or the crew members. The FIS(ATIS)-ground-user is responsible for the operation of the FIS contracts. On receipt of the contract request, it is responsible for responding to the request and then creating and submitting ADS reports in line with the contract. The functionality of the FIS(ATIS)-users is described in chapter 2.4.7 of the FIS(ATIS) SARPs.

2.3.4 The internal structure of the FIS(ATIS) ASE has been defined based on the following principle: the basic procedures used to support the operation of the FIS contracts are common to most DFIS services. The major difference amongst FIS contracts is the nature of the operational data exchanged during the contract lifetime, not the nature of the procedures. It is therefore possible to define generic ASE modules common to several FIS Applications. Those generic FIS ASE modules are integrated in a specific FIS Application and are slightly customised to support the specific DFIS service.

2.3.5 As far as the FIS(ATIS) Application is concerned, four generic FIS ASE modules have been selected for handling the ATIS protocol:

- the FIS Demand Contract (DC) module,
- the FIS Update Contract (UC) module,
- the FIS Cancel Contract (CL) module, and
- the FIS Abort Contract (AB) module.

2.3.6 The internal structure of the FIS(ATIS) ASE is illustrated in figure x.x. Two interface modules have been defined to cope with the interface with the ASE-users (High Interface (HI) module) and with the Dialogue Service Interface (Low Interface (LI) module).



2.3.7 This flexible architecture allows for the reuse of common FIS ASE modules in FIS Applications that will be specified later and the easy incorporation of additional services and service negotiation mechanisms in future versions of the FIS(ATIS) application.

2.4 Implementation Dependent Functionality

2.4.1 The FIS SARPs specify some of the requirements for the user, but leave a lot up to the implementor. There are no requirements that state how the user interface appears, how FIS interacts with the systems generating the ATIS information, how FIS interacts with higher level functionality and with other applications such as CM. All this is implementation dependent.

2.4.2 The definition of the ATIS reports in the SARPs is quite open. It has not been possible to specify with the ASN.1 notation all the constraints and relationships between ATIS fields. The rules for building a consistent ATIS report are not checked by the FIS ASEs and have to be implemented by the FIS-users.

2.5 Rationale for ASE / Users Split

2.5.1 The rationale for the split in functionality between the FIS ASEs and the FIS-users is as follows:

2.5.1.1 The ASE contains all functionality that is necessary to ensure the interoperability at the syntactic level. That is, two valid implementations of ASEs will be able to interact, passing data to each other in the correct order; they will be able to check the format of the data, ensure that it has been sent with the appropriate dialogue service primitive and ensure that the peer ASE is behaving according to the requirements in the SARPs also. The ASE thus ensures interoperability.

2.5.1.2 The SARPs define some requirements for the users. These are minimum that are required to ensure the semantic interoperability of the two peers. That is, it explains how the data that is transported by the ASE should be interpreted. Thus it explains how each type of FIS contract is interpreted and how it should operate.

2.5.1.3 Some care has been taken to ensure that the requirements are not over-specified. That is, they do not specify rules which are not absolutely essential to the syntactic ans semantic interoperability of the FIS function.

This implementation dependent part can be built by different manufacturers in different ways, without effecting the interoperability between different implementations. This implementation dependent part has not been specified in the SARPs, but should be specified by individual product manufacturers or regional standards.

2.6 Inter-relationship with other ATN Applications

2.6.1 There is no interaction required between the FIS(ATIS) Application and the other CNS/ATM-1 Data Link applications. The FIS(ATIS) application is a stand-alone application which can be developed, certified, installed and operated completely independently from the other ATN Applications.

2.6.2 However, in order to be able to initiate a FIS(ATIS) dialogue, some naming and addressing information have to be known from the dialogue initiator:

- the ATN address of the ground FIS(ATIS) system must be present somewhere in the data link communication system. Without this information, the dialogue with the ground FIS system can not be established.
- the version number of the FIS(ATIS) protocol run in the ground FIS system must be available to the FIS-air-user. The protocol version negotiation is not performed by the FIS(ATIS) Application Entities and the FIS-air-user is responsible for checking the version compatibility before establishing a FIS Contract.

2.6.3 The Context Management (CM) Application is the mean defined in CNS/ATM-1 to exchange this information. Other solutions could be chosen, like the hard-coding of the addressing data bases or the loading of this information at the gate before the take-off.

2.6.4 In the hypothesis the CM Application is run, the definition of the ground topology for CM and FIS ground systems must follow the following rules:

1. a CM area can not contain more than one FIS ground system of a same version number,
2. a FIS ground system can belong to more than one CM area.

2.7 Ground FIS Exchanges

2.7.1 No operational requirement related to the ground FIS exchanges have been specified in the ADSP Manual [ref. xx]. As a consequence, there is no interaction defined between FIS-ground-users in the first version of the FIS(ATIS) Application.

2.7.2 As a consequence of the non availability of ground communications between FIS ground systems, FIS(ATIS) requests must be sent to the appropriate ground FIS system, i.e. the one which has the requested information resident. If not, the request is rejected by the ground FIS user.

2.7.3 Two pieces of information must be available before a FIS Contract can be established:

1. the identification of the ground FIS system which hold the desired information (e.g. "the centralised ground FIS system in Paris stores the ATIS information of all French airports"),
2. the identification and the address of the CM Application attached to the ground FIS system identified in 1/. The CM Logon procedure with this ground CM system is run to retrieve the address of the ground FIS system.

2.8 Dialogue Management

2.8.1 General

2.8.1.1 The dialogue is always initiated by the airborne system, by the process of setting up a FIS contract. In like manner, the airborne system is always responsible for closing the dialogue.

2.8.1.2 The dialogue service provider is used by the FIS-ASEs to provide the following services:

- establishment, graceful release and abort of a dialogue,
- unstructured data transfer,
- support for quality of service and version number negotiation,
- application naming.

2.8.2 Optimisation of the use of dialogues

2.8.2.1 This FIS service hides the use of the underlying communication service to the FIS-users: the FIS service users do not realise when the dialogue is being established or released.

2.8.2.2 Multiplexing over a single dialogue of FIS contracts set up between an aircraft and a ground FIS system is supported by the FIS protocol. Up to 256 FIS contracts can be multiplexed in parallel on a single dialogue. The time required to establish the dialogue before any operational data can be exchanged penalises therefore only the first FIS contract. The dialogue multiplexing is performed by the Low Interface Module.

2.8.3 Dialogue Establishment

2.8.3.1 If no dialogue was established with the ground FIS system (i.e. if no connection has already been established), a dialogue is established first, then data related to the FIS contract are then exchanged. During the time of the dialogue establishment, no new request can be accepted by the FIS-service provider.

2.8.3.2 On receipt of a FIS contract establishment request, if a dialogue is still in place and was maintained for a future use, it is used immediately for data transmission.

2.8.4 Dialogue Release

2.8.4.1 The dialogue is closed when no activity takes place during a certain period of time. Without explicit action from the pilot or a ground operator, the air FIS system trigger the release of the dialogue in place. The value for the inactivity timer is not standardised in the FIS(ATIS) SARPs. It could be customised for each airborne system.

2.9 Protocol Monitoring

2.9.1 In case of confirmed service, the generation and the transmission of the response expected to be issued by the remote FIS-user are monitored by the FIS ASEs.

2.9.2 If the FIS-demand-contract response or the FIS-update-contract response is not invoked by the FIS-ground-user within a period of time locally specified in the aircraft, the dialogue in place with the ground FIS system is aborted and all contracts in place or being established are cancelled. FIS-users are informed of this situation by a FIS-provider-abort indication received at both side.

2.10 Version Number Negotiation

2.10.1 The FIS(ATIS) SARPs specify the operation of version 1 of the FIS(ATIS) application. The version number is a value inherent to the FIS-ASE and is not provided by the FIS users.

2.10.2 The version numbers supported by the air and ground systems for the FIS(ATIS) application are exchanged during the CM procedures. The FIS-air-user must check whether the ASE implemented onboard is compatible with one of the versions implemented on the ground.

2.10.3 The version number negotiation is implicitly performed by the upper layers when establishing the application-association. The application context name proposed by the initiator must be supported by the receptor.

2.11 Number of Concurrent FIS(ATIS) Contracts

2.11.1 In theory, there is no lower and upper limits specified in the SARPs for the number of concurrent FIS(ATIS) contracts.

2.11.2 However, dimensioning and cost consideration should be given when allowing an airborne system to run multiple FIS contracts with several ground systems. There will be as many transport connections set up in parallel as there are ground systems. When parallel contracts are set up with the same ground system, only one transport connection is used and all contracts are multiplexed over this dialogue.

2.12 Application Naming

2.12.1 The ULCS SARPs (ref x.x) have assigned a unique and unambiguous application entity title to the air and ground FIS application entities, as follows:

{ iso(1) identified-organisation (3) icao (27) atn-end-system-air (1) <end-system-id> (n) operational (0) ATI (3) }

{ iso(1) identified-organisation (3) icao (27) atn-end-system-ground (2) <end-system-id> (n) operational (0) ATI (3) }

3. FUNCTIONALITY OF SERVICES

3.1 Concepts

3.1.1 Users of the FIS service are termed *FIS-ground-user* and *FIS-air-user* or *FIS-user* when the physical location of the service-user is meaningless. The FIS-user represents the operational part of the FIS system. It is either the final end-user (e.g. a crew member) or a automated system. The FIS-air-user that initiates a FIS Contract is termed the *calling* FIS-user or *initiator*. The FIS-ground-user that the initiator is trying to contact is termed the *called* FIS-user or *responder*.

3.1.2 This section describes first the information required by the ASEs from the FIS-users. Then, it considers FIS Service in turn and provides a small overview of the data flow within the ASE to handle the service primitives. The primitives are grouped according to the services they provide: setting a FIS Demand Contract, setting a FIS Update Contract, cancelling FIS contracts and Aborting FIS contracts.

3.2 Information Provided by the Service Users

3.2.1 FIS Contract Number

3.2.1.1 During FIS contract establishment (demand or update), the contract initiator defines a *FIS Contract Number* to name the FIS Contract uniquely for the application.

3.2.1.2 The FIS Contract Number permits to identify unambiguously any contract in place between a peer of FIS-users. This identifier must be supplied as parameter of any request or response primitives invoked by the FIS-users. Unallocated number must be used when invoking a FIS-demand-contract request. The exchange of terminal primitives (such as FIS-demand/update-contract confirmation (accepted) or FIS-user-abort request/indication) causes the desallocation of the current contract number(s) which could be used for a new FIS contract.

3.2.1.3 The FIS contract number supplied by the FIS-ground-ASE when invoking the FIS-report service or the FIS-cancel-update-contract service must have been assigned previously to an existing FIS contract.

3.2.2 ICAO Facility Designation Selection

3.2.2.1 The contacted ground FIS system must be retrievable from the addressing data base of the aircraft. If the CM application is used, an exchange of naming and addressing information (CM-Logon or CM-Update) must take place successfully before the FIS-demand-contract service is invoked. The ground system must support the FIS application. Depending on the version of the FIS applications supported onboard and on the ground, the FIS-air-user must identify in the initial request the appropriate ground component.

3.2.3 Class Of Communication Selection

3.2.3.1 The Class of Communication provided by the FIS-air-user is supplied to the Transport Service Provider (TSP) when the connection is established. The Class of Communication is a mean for the user to indicate approximately the expected Quality of Service in terms of end-to-end transit delay. Values for these transit delays are given in Sub-Volume 1 of the ATN SARPs. The Class of Communication is not guaranteed nor a degradation of the provided class is indicated to the users.

3.2.3.2 There is no negotiation of the Class of Communication between air and ground FIS-users. Thus, the value of the Class of Communication requested by a FIS-air-user for a dialogue is not transmitted to the FIS-ground-user.

3.2.3.3 If the FIS-air-user does not require a particular Class of Communication, the Class of Communication parameter may be left blank and the dialogue service uses the default value "xx".

3.2.4 Application Priority and Residual Error Rate

3.2.4.1 In addition to the Class Of Communication provided by the contract initiator, other Quality of Service (QOS) parameters are attached by the ASE to the dialogue supporting the communication. Values for the Application Priority and the Residual Error Rate are constant for the FIS(ATIS) application and therefore are not requested to the users.

3.2.4.2 The Application Priority has the value "Aeronautical Information Service messages".

3.2.4.3 The Residual Error Rate has the value "low".

3.2.5 User Data

3.2.5.1 The user data parameter contains the operational data to be exchanged on the FIS Contract.

3.2.5.2 These User Data pass transparently through the FIS Service provider. The FIS service provider will attach a small header to the user data used for the co-ordination of the two FIS ASEs. This header is termed the Protocol Control Information (PCI).

3.3 Setting a FIS Demand Contract

3.3.1 The *FIS-demand-contract* service is used to set up a FIS Demand Contract between a FIS-air-user and a FIS-ground-user. It is a confirmed service, initiated by the FIS-air-user.

- The FIS-demand-contract request is passed to the air HI module which examines it to see which other module to pass it on to. The request is passed to the air DC module. The FIS Contract Number attached to the new demand contract is stored by the air HI module.
- The air DC module generates a FISDownlinkAPDU [FISRequest] APDU. It passes the APDU to the air LI module with the ground system identifier and the class of communication supplied by the user. Timers t-DC-1 and t-DC-2 are started to monitor the reception of the reply from the ground ([FISAccept] APDU or [FISReject] APDU) and the reception of the FIS report when the response is postponed.
- The air LI module decides how to use the dialogue service. If a dialogue exists already, it makes use of that dialogue and uses the D-DATA service to pass the APDU to the ground system. If no dialogue exist, it uses the D-START service to pass the APDU to the ground system. The APDU is passed to the ground system by way of the upper and lower layers and emerges in the ground LI module.
- The ground LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the ground DC module.
- The ground DC module recognises the APDU as FISDownlinkAPDU [FISRequest] APDU and passes a FIS-demand-contract indication to the ground HI module.
- The ground HI module passes it to the FIS-ground-user.

3.3.2 The FIS demand contract may be either accepted (in that case, a FIS report is enclosed in the response message), rejected (a reject reason is provided) or postponed (the FIS report is sent later using the FIS-report service) by the ground FIS system.

- The FIS-demand-contract response is passed to the ground HI module which examines it to see which other module to pass it on to. The request is passed to the ground DC module.
- The ground DC module generates a FISUplinkAPDU [FISAccept] or [FISReject] APDU based on the reply supplied by the FIS-ground-user. It passes the APDU to the ground LI module.
- The ground LI module then passes the APDU to the air using a D-DATA request (if a dialogue is already open) or a D-START response (if no D-START response has yet been given). The APDU is passed to the airborne system by way of the upper and lower layers and emerges in the air LI module.
- The air LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the air DC module.
- The air DC module recognises the APDU as FISUplinkAPDU [FISAccept] or [FISReject] APDU and accordingly passes a positive or negative FIS-demand-contract confirmation to the air HI module. If the response is not postponed and if there is no FIS contract in place, the timer t-inactivity is started.
- The air HI module passes it to the FIS-air-user.

3.3.3 The *FIS-report* service is then invoked by the FIS-ground-user to both send a FIS report and close a previously postponed FIS Demand Contract. It is an unconfirmed service initiated by the FIS-ground-user.

- The FIS-report request is passed to the ground HI module which examines it to see which other module to pass it on to. The decision is taken based on the FIS contract number supplied by the user and the contracts already managed and identified by the FIS ASE. The request is passed to the ground DC module.
- The ground DC module generates a FISUplinkAPDU [FISReport]. It passes the APDU to the ground LI module.
- The ground LI module then passes the APDU to the air using a D-DATA request. The APDU is passed to the airborne system by way of the upper and lower layers and emerges in the air LI module.
- The air LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the air DC module.
- The air DC module recognises the APDU as FISUplinkAPDU [FISReport] APDU and passes a FIS-report indication to the air HI module. If there is no FIS contract in place, the timer t-inactivity is started.
- The air HI module passes it to the FIS-air-user.

3.4 Setting a FIS Update Contract

3.4.1 The *FIS-update-contract* service sets up a FIS Update Contract between a FIS-air-user and a FIS-ground-user. It is a confirmed service, initiated by the FIS-air-user.

- The FIS-update-contract request is passed to the air HI module which examines it to see which other module to pass it on to. The request is passed to the air UC module. The FIS Contract Number attached to the new demand contract is stored by the air HI module.

- The air UC module generates a FISDownlinkAPDU [FISRequest] APDU. It passes the APDU to the air LI module with the ground system identifier and the class of communication supplied by the user. Timers t-UC-1 and t-UC-2 are started to monitor the reception of the reply from the ground ([FISAccept] APDU or [FISReject] APDU) and the reception of the FIS report when the response is postponed.

- The air LI module decides how to use the dialogue service. If a dialogue exists already, it makes use of that dialogue and uses the D-DATA service to pass the APDU to the ground system. If no dialogue exist, it uses the D-START service to pass the APDU to the ground system. The APDU is passed to the ground system by way of the upper and lower layers and emerges in the ground LI module.

- The ground LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the ground UC module.

- The ground UC module recognises the APDU as FISDownlinkAPDU [FISRequest] APDU and passes a FIS-update-contract indication to the ground HI module.

- The ground HI module passes it to the FIS-ground-user.

3.4.2 The update contract may be either accepted (in that case, the FIS report is enclosed in the response message), rejected (a reject reason is provided, optionally the FIS report is enclosed in the response message) or postponed (the FIS report is sent later) by the ground FIS system. A FIS Update Contract must be explicitly closed by the FIS-users.

- The FIS-update-contract response is passed to the ground HI module which examines it to see which other module to pass it on to. The request is passed to the ground UC module.

- The ground UC module generates a FISUplinkAPDU [FISAccept] or [FISReject] APDU based on the reply supplied by the FIS-ground-user. It passes the APDU to the ground LI module.

- The ground LI module then passes the APDU to the air using a D-DATA request (if a dialogue is already open) or a D-START response (if no D-START response has yet been given). The APDU is passed to the airborne system by way of the upper and lower layers and emerges in the air LI module.

- The air LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the air UC module.

- The air UC module recognises the APDU as FISUplinkAPDU [FISAccept] or [FISReject] APDU and accordingly passes a positive or negative FIS-update-contract confirmation to the air HI module.

- The air HI module passes it to the FIS-air-user.

3.4.3 The *FIS-report* service is then invoked by the FIS-ground-user to activate a postponed FIS Update Contract and to uplink any update of the FIS information. It is an unconfirmed service initiated by the FIS-ground-user. FIS reports are delivered to the FIS-air-user in the order they have been supplied by the FIS-ground-user.

- The FIS-report request is passed to the ground HI module which examines it to see which other module to pass it on to. The decision is taken based on the FIS contract number supplied by the user and the contracts already managed and identified by the FIS ASE. The request is passed to the ground UC module.

- The ground UC module generates a FISUplinkAPDU [FISReport]. It passes the APDU to the ground LI module.
- The ground LI module then passes the APDU to the air using a D-DATA request. The APDU is passed to the airborne system by way of the upper and lower layers and emerges in the air LI module.
- The air LI module examines the APDU in order to decide which module to pass it to based on the FIs contract number. The APDU is passed to the air UC module.
- The air UC module recognises the APDU as FISUplinkAPDU [FISReport] APDU and passes a FIS-report indication to the air HI module.
- The air HI module passes it to the FIS-air-user.

3.5 Cancelling FIS Contracts

3.5.1.1.1 Cancelling several contracts

3.5.1.1.1.1 The *FIS-cancel-contracts* service allows the FIS-air-user to cancel all FIS demand and update contracts of the same type with a particular FIS-ground-user. This service does not affect the FIS contracts of other types still in place.

- The FIS-cancel-contracts request is passed to the air HI module which examines it to see which other module to pass it on to. The request is passed to the air CL module.
- The air CL module generates a FISDownlinkAPDU [FISCancelContracts] APDU. It requests the air DC and UC modules to stop operation. It passes the APDU to the air LI module. Timer t-CL-1 is started to monitor the reception of the reply from the ground ([FISCancelContractsAccept] APDU.
- The air LI module then passes the APDU to the ground using a D-DATA request. The APDU is passed to the ground system by way of the upper and lower layers and emerges in the ground LI module.
- The ground LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the ground CL module.
- The ground CL module recognises the APDU as FISDownlinkAPDU [FISCancelContracts] APDU and passes a FIS-cancel-contracts indication to the ground HI module. It requests the ground DC and UC modules to stop operation.
- The ground HI module passes the FIS-cancel-contracts indication to the FIS-ground-user.

3.5.1.1.1.2 The FIS-cancel-contracts service is a confirmed service initiated by the FIS-air-user. As the cancellation can not be rejected by the FIS-ground-user, no response is requested. The confirmation APDU is sent automatically by the FIS-ground-ASE.

- The ground CL module generates a FISUplinkAPDU [FISCancelContractsAccept] APDU. It passes the APDU to the ground LI module.

- The ground LI module then passes the APDU to the air using the D-DATA request. The APDU is passed to the airborne system by way of the upper and lower layers and emerges in the air LI module.

- The air LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the air CL module.

- The air CL module recognises the APDU as FISDownlinkAPDU [FISCancelContractsAccept] APDU and passes a FIS-cancel-contracts confirmation to the air HI module. The t-inactivity timer is started.

- The air HI module passes the FIS-cancel-contracts confirmation to the FIS-air-user.

3.5.1.1.1.3 The FIS-air-user specifies the type of FIS contracts it wants to cancel.

3.5.1.1.2 Cancelling a particular Update Contract

3.5.1.1.2.1 The *FIS-cancel-update-contract* service allows the FIS-air-user or the FIS-ground-user to cancel in an orderly manner an active FIS update contract. The FIS reports in transit in the communication system are delivered before the FIS update contract is cancelled. This service does not affect the other FIS contracts in place.

- The FIS-cancel-update-contract request is passed to the local HI module which examines it to see which other module to pass it on to. The request is passed to the local UC module.

- The local UC module generates a FISDownlinkAPDU or FISUplinkAPDU [FISCancelUpdateContract] APDU. It passes the APDU to the local LI module. Timer t-UC-3 is started to monitor the reception of the reply from the peer ([FISCancelUpdateContractAccept] APDU).

- The local LI module then passes the APDU to the peer using a D-DATA request. The APDU is passed to the peer system by way of the upper and lower layers and emerges in the remote LI module.

- The remote LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the remote UC module.

- The remote UC module recognises the APDU as FISDownlinkAPDU or FISUplinkAPDU [FISCancelUpdateContract] APDU and passes a FIS-cancel-update-contract indication to the remote HI module. If the remote system is air based, the t-inactivity timer is started.

- The remote HI module passes the FIS-cancel-update-contract indication to the remote FIS-user.

3.5.1.1.2.2 The FIS-cancel-update-contract service is a confirmed service, initiated by the FIS-air-user or the FIS-ground-user. As the cancellation can not be rejected by the FIS-ground-user, no response is requested. The confirmation APDU is sent automatically by the FIS -ASE.

- The remote UC module generates a FISUplinkAPDU or FISDownlink [FISCancelUpdateContractAccept] APDU. It passes the APDU to the remote LI module.

- The remote LI module then passes the APDU to the peer using the D-DATA request. The APDU is passed to the peer system by way of the upper and lower layers and emerges in the local LI module.

- The local LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the local CL module.

- The local CL module recognises the APDU as FISDownlinkAPDU or FISUplinkAPDU [FISCancelContractsAccept] APDU and passes a FIS-cancel-contracts confirmation to the local HI module. If the local system is air based, the t-inactivity timer is started.

- The local HI module passes the FIS-cancel-contracts confirmation to the local FIS-user.

3.6 Aborting a FIS Contract

3.6.1.1.1 User-Initiated Abort

3.6.1.1.1.1 When a FIS-user detects an error it views as catastrophic, the *FIS-user-abort* service is invoked. All active FIS demand and update contracts processed by the ASE are cancelled. Messages in transit may be lost during this operation. This is an unconfirmed service without parameter.

3.6.1.1.1.2 The FIS-user-abort service can be invoked at any time that the FIS-user is aware that any FIS service is in operation. After this primitive is invoked, no further primitives may be invoked for the current FIS contracts.

- The FIS-user-abort request is passed to the local HI module which examines it to see which other module to pass it on to. The request is passed to the local AB module.
- The local AB module requests all running local ASE modules to stop their current activities. It request the local LI module to abort the dialogue.
- The local LI module abort the dialogue using a D-ABORT request if the dialogue is still open. The abort originator parameter is set to "user". A D-ABORT indication emerges in the remote LI module.
- The remote LI module passes the D-ABORT indication to the remote AB module.
- The remote AB module recognises a user abort situation (the abort originator of the D-ABORT indication is set to "user"). It requests all running remote ASE modules to stop their current activities. It passes a FIS-user-abort indication to the remote HI module.
- The remote HI module passes it to the remote FIS-user if still active.

3.6.1.1.2 Provider-Initiated Abort

3.6.1.1.2.1 When the FIS ASE detects an error from which it cannot recover, the *FIS-provider-abort* service is invoked. This is a FIS provider-initiated service with a single, mandatory parameter telling why the FIS Contracts in progress with a ground FIS system were lost. Messages in transit may be lost during this operation.

- The local ASE module detecting the error situation request the local AB module to abort the communication with a reason value.
- The local AB module requests all running local ASE modules to stop their current activities. It passes a FIS-provider-abort indication to the local HI module. It generates a FISUplinkAPDU or FISDownlinkAPDU [FISAbort]. It passes the APDU to the local LI module.
- The local HI module passes the FIS-provider-abort indication to the local FIS-user if he is still active,
- The local LI module passes the APDU to the remote system using a D-ABORT request if the dialogue is still open. The abort originator parameter is set to "provider". The APDU is passed to the remote system by way of the upper and lower layers and emerges in the remote LI module.
- The remote LI module examines the APDU in order to decide which module to pass it to. The APDU is passed to the remote AB module.

- The remote AB module recognises a provider abort situation (the abort originator of the D-ABORT indication is set to "provider"). It requests all running remote ASE modules to stop their current activities. It decodes the FISUplinkAPDU [FISAbort] APDU and passes a FIS-provider-abort indication to the remote HI module.

- The remote HI module passes it to the remote FIS-user if still active.

3.6.1.1.2.2 Errors detected by the ASE are the following:

- *timer expiration*: a timer set previously by the ASE elapses.
- *protocol error*: the ASE detects that the peer ASE does not respect the rules of the protocol.
- *sequence error*: a not expected dialogue service primitive has been received by the ASE.
- *decoding error*: the ASE is not able to decode the data packet received in the user data parameter of a dialogue service primitive.
- *unrecoverable internal error*: a system function call has failed, for instance a memory allocation/release function, an I/O operation, etc...
- *invalid contract number*: the contract number received from the peer does not correspond to an existing contract.
- *dialogue end not supported*: A D-END response negative has been received, but this is not allowed by the FIS protocol.
- *undefined error*: any other error.

4. FIS(ATIS) CHAPTER DESCRIPTION

4.1 Chapter 2.4.2: GENERAL REQUIREMENTS

4.1.1 Purpose of Chapter 2

<Editor’s note: tbd>

4.1.2 Version Number

4.1.2.1 This section is included to allow the Context Management (CM) application to exchange version numbers of the FIS application. It is necessary to allow for future versions of the protocol to be negotiated by CM. It has no effect on the FIS functionality.

4.1.3 Error Processing Requirements

4.1.3.1 In the abstract service definition, each service has a set of parameters and the abstract syntax of those parameters specified. Thus information which is not a valid syntax is not allowed to be input.

4.1.3.2 In the protocol definition, there is a requirement that no service is permitted to be called when in an inappropriate state. Thus making use of the abstract services is not permitted at these times.

4.1.3.3 A sensible method of implementation would not allow the user to take such invalid actions; however, there is no requirement to prevent an implementation from allowing this. The error processing requirements section thus says that *if* the implementation allows the user to enter invalid information, the system must inform the user that they have made a mistake. In that case, the error is locally detected and it is not required to abort the dialogue.

4.2 Chapter 2.4.3: ABSTRACT SERVICE DEFINITION

4.2.1 Purpose of Chapter 3

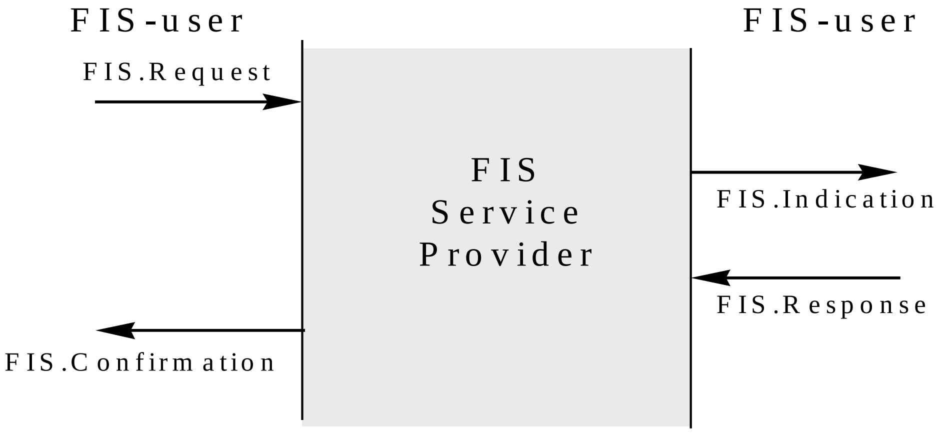
4.2.1.1 The FIS Abstract Service represents a set of functions offered to the FIS-user by the FIS-service provider.

4.2.2 Conventions

4.2.2.1 Service Primitives

4.2.2.1.1 A FIS service may consist of one to four primitives. Each primitive consists of the name of the FIS service (FIS-demand-contract, FIS-update-contract, FIS-cancel-contracts, FIS-cancel-update-contract, FIS-user-abort and FIS-provider-abort) and a suffix that indicates at what point in the service the primitive occurs (request, indication, response, confirmation).

4.2.2.1.2 A *confirmed FIS service* is one that involves a handshake between the user that requests the service and the user that is informed that the service has been requested:



4.2.2.1.3 An *unconfirmed FIS service* involves no handshake.

4.2.2.1.4 A *provider-initiated service* is generated by the service provider in response to some internal condition. They consist of one indication primitive which is given to both users by the FIS service provider.

4.2.2.2 Service Parameters

4.2.2.2.1 Some primitive parameters have the same contents than APDU fields in the ASN.1 description. In most cases, the FIS-ASE is tasked to copy the parameter value within the APDU field. In order to avoid confusion by defining identical data structures twice, the type of those primitive parameters is specified by simply referring to the corresponding ASN.1 type in the APDU. The ASN.1 is used in the service definition as a syntax notation only and does not implicitly imply any local encoding of these parameters. The local implementation of these parameters remains a local implementation issue.

4.2.2.2.2 For the other parameters, the syntax is described by enumerating the authorised abstract values.

4.2.2.2.3 These service conventions have no impact on implementations. For instance a service primitive may be implemented through a function call or through an exchange via a mailbox (IPC mechanism).

4.2.2.2.4 There is no requirement for a SARPs compliant FIS system to be conformed to the abstract service specified in the SARPs.

4.3 Chapter 2.4.4: FORMAL DEFINITION OF MESSAGES

4.3.1 Purpose of Chapter 4

<Editor's note: tbd>

4.3.2 ASN.1

4.3.2.1 Data types exchanged by FIS ASEs are described in the FIS SARPs by using a machine-independent and language-independent syntax. There is no constraint put on the implementors concerning the machine nor the development language to be selected for implementing the protocol.

4.3.2.2 The ASN.1 module *MessageSetVersion1* contains the data types of the protocol data units handled by the FIS(ATIS) ASEs. Unlike common OSI ASEs (e.g. ACSE), no object identifier has been attached to the FIS(ATIS) ASN.1 specification. Indeed, the ULCS architecture releases the applications from negotiating during the dialogue establishment the applicable abstract syntax. Object identifiers related to FIS applications (application context name and version number) are defined in the ULCS SARPs.

4.3.2.3 ASN.1 Tags

4.3.2.3.1 Tags are used in ASN.1 to allow to distinguish data types when confusion is possible. For instance, when a data type contains two optional elements of the same type, if only one is encoded there is no means for the decoder to know which element the decoded value is attached to.

4.3.2.3.2 Even if tag values are not used by the Packed Encoding Rules, the ASN.1 grammar mandate the use of tags in some cases. When specifying the FIS data types, the following rules have been used:

- tags are always used within CHOICE data type, starting at 0 and then incremented by 1 for each entry.
- tags are not used at all in SEQUENCE data type when no confusion is possible. When an optional element is defined, all elements in the sequence are tagged.

4.3.2.4 Extensibility Markers

<Editor's note: definition of extensibility marker tbd>

4.3.2.4.1 In order to allow the upgrade of the ASN.1 specification when new FIS service are made available, the extensibility ASN.1 feature (ellipse) has been used in each data type identifying the DFIS service.

4.3.2.5 Data type specification when unit, range and resolution are defined

4.3.2.5.1 Some operational data are real. REAL data types exist in ASN.1 but the associated encoding procedures are not optimized in term of data size (PER do not specify specific rules for real but import the BER ones which requires the encoding of a mantissa, a base and a exponent taking several octets). Real values are therefore specified via INTEGER data type.

4.3.2.5.2 ASN.1 constraints are attached to the INTEGER data type to take advantage of both range and resolution. Lower bound is equal to the operational minimal value devised by the resolution and the upper bound is equal to the operational maximal value devised by the resolution.

4.3.2.5.3 For instance:

```
PressureMeasure ::= CHOICE
{
hPa             [0] INTEGER (7500..12500)
-- units = hPa, range (750.0..1250.0), resolution 0.1
inches [1] INTEGER (2200..3200)
-- units = inches of Mercury, range (22.00..32..00), resolution = 0.01
}
```

4.3.2.6 Entry Points

4.3.2.6.1 The ASN.1 description does not specify what data type a ground system or a air system is required to be able to encode and to decode. Two problems exist:

1. On reception of a APDU, the FIS system does not how to decode it since there is no 'top level choice' in the description indicating whether the APDU is a Downlink APDU or an Uplink APDU,
2. A system is not required to be able to encode or decode all messages specified in the ASN.1 description. Typically, a ground system is not required to be able to encode a Downlink APDU nor to decode an Uplink APDU.

4.3.2.6.2 The SARPs requirements attached to the ASN.1 description are stated to guide implementors in the resolution of these problems.

4.3.2.7 Data types common to several Package 1 Applications

<Editor's note: tbd>

4.3.2.8 Time Representation

4.3.2.8.1 Data types have been specified for containing time indication (Date, DateTimeGroup, Year, Month, Hours, Minuts, Seconds). This way to represent time has been preferred over the pre-defined ASN.1 representations (GeneralizedTime and UTCTime) for optimization of the PER encoding.

4.3.3 Transfer Syntax

4.3.3.1 The transfer syntax represents unambiguously the values of data types as they are transmitted on the network. The ULCS SARPs mandate the use of the Packet Encoding Rules for application data.

4.3.4 ASN.1 Glossary

4.3.4.1 Two sets of data types are defined in the FIS(ATIS) ASN.1 description. The first set (FIS level) contains information common to all type of DFIS services. The second set is contains information related to a specific DFIS service, here the ATIS service.

4.3.4.2 FIS Protocol data

4.3.4.2.1 These data are used to co-ordinate the processing of remote FIS ASEs. They are Protocol Control Information which do not carry operational data but are used for protocol processing purpose. They are used:

1. to identify the type of the APDU transmitted on the line (e.g. FISDemandContract Downlink APDU or FISAbort Uplink APDU),
2. to identify the type of DFIS service the contents of the APDU is related to (e.g. ATIS, METAR, etc...), and
3. to identify the contract the APDU is related to.

4.3.4.2.2 The following data are used as the FIS message variables, or component of the variables, and are shown here in the alphabetic order:

ContractNumber	Identifies the contract to which the APDU is related to.
DownlinkAPDU	Contains the identification and the contents of the following downlink FIS messages: *FISRequest*, *FISCancelUpdateContract*, *FISCancelUpdateAccept*, *FISCancelContracts* or *FISAbort*.
UplinkAPDU	Contains the identification and the contents of the following uplink FIS messages: *FISAccept*, *FISReject*, *FISReport*, *FISCancelUpdateContract*, *FISCancelUpdateAccept*, *FISCancelContractsAccept* or *FISAbort*.
FISAbort	Indicates the type of FIS information being at the origin of the communication abort (i.e. "ATIS") and any additional information explaining the cause of the abort. This message is created when an unrecoverable error in detected by either of the FIS-ASEs by using the *FISProtocolErrorDiag* data.
FISAccept	Identifies the FIS contract being accepted by this message by using the *ContractNumber* data. Contains in the *FISAcceptData* data the contract response information supplied by the FIS-ground-user.
FISAcceptData	Contains either the indication of a positive acknowledgement or details of the FIS report when the contract is accepted by using the *FISReportData* data.
FISCancelContracts	tbd
FISCancelContracts Accept	tbd
FISCancelAcceptData	tbd
FISCancelUpdateAccept	tbd
FISCancelUpdate Contract	tbd
FISCancelUpdateData	tbd
FISDownlinkAPDU	This is the message unit formatted by the FIS-air-ASE and downlinked to the FIS-ground-ASE. It specifies the time the message has been received for transmission and the downlink FIS message itself by using the *DateTimeGroup* and *DownlinkAPDU* data.
FISProtocolErrorDiag	Specifies the reason of the abort by either of the FIS-ASEs.

FISReject	tbd
FISRejectData	tbd
FISRejectReason	tbd
FISReport	Specifies the contract number the FIS report is related to by using the *ContractNumber* data. Contains in the *FISReportData* data the report information supplied by the FIS-ground-user.
FISReportData	Identifies the type of the FIS report transmitted (i.e. "ATIS) and contains the user data describing the report (*ATISReport* data for ATIS).
FISRequest	Specifies the contract type and the contract number by using the *ContractNumber* and *ContractType* data. Contains in the *FISRequestData* data the contract request information supplied by the FIS-air-user.
FISRequestData	Identifies the type of the FIS Information requested (i.e. "ATIS") and contains the user data describing the request (*ATISRequest* data for ATIS).
FISServiceType	tbd
FISUplinkAPDU	This is the message unit formatted by the FIS-ground-ASE and uplinked to the FIS-air-ASE. It specifies the time the message has been received for transmission and the uplink FIS message itself by using the *DateTimeGroup* and *UplinkAPDU* data.

4.3.4.3 ATIS Protocol data

4.3.4.3.1 These data are used to co-ordinate the processing between FIS ASEs providing the ATIS service. They are used to convey the operational information likely to be generated by ATIS users.

4.3.4.3.2 Some types are common to several data link applications. However, even if the semantic is the same, the range and resolution required by each application may be specific. Therefore it has been decided to specify stand-alone ASN.1 descriptions completely independent from the others.

4.3.4.3.3 The following data are used as the ATIS message variables, or component of the variables, and are shown here in the alphabetic order:

Airport	Four characters that specifies the ICAO four-letter identifier of the airport.
AltimeterSetting	tbd
AltitudeFlightLevel	tbd
AltitudeFlightLevel Metric	tbd
Approach	tbd
ApproachType	tbd
ArrivalAndDeparture ATIS	tbd
ArrivalDeparture Indicator	tbd
ArrivalATIS	tbd
ArrivalRunway	tbd

ATISCode	Specifies the alphanumeric value for the current version of the automatic terminal information service (ATIS) in effect at a given location.
BrakingAction	tbd
BrakingAction Description	tbd
BrakingActionQuality	tbd
CloundAmount	tbd
CloudHeight	tbd
CloudInformation	tbd
CloudSkyCoverGroup	tbd
CloudType	tbd
ATISInformation	tbd
ATISReport	tbd
ATISRequest	tbd
ContractType	tbd
CombinedATIS	tbd
CommonATIS Information	tbd
Date	Gives the date using *Year*, *Month* and *Day* data.
DateTimeGroup	Gives date and time using *Date* and *HHMMSS* data.
Day	Day of the month.
DepartureATIS	tbd
DepartureRunway	tbd
FreeText	Used to convey unstructured information.
HHMMSS	tbd
Level	tbd
MeasuredWind Variations	tbd
Month	Month of the year.
PressureMeasure	tbd
RunwayId	tbd
Runway	tbd
RynwayLetter	tbd
RunwayNumber	tbd
RunwayQFE	tbd

RunwayType	tbd
RunwayVisibility	tbd
RVR	tbd
SpecificATISArrival Info	tbd
SignificantMetInfo	tbd
SignificantMet Phenomena	tbd
SurfaceWinds	tbd
Time	Provides the time using *TimeHours* and *TimeMinutes* data.
TimeHours	Specifies time in hours of a day.
TimeMinutes	Specifies time in minutes of an hour.
TimeSeconds	Specifies time in seconds of a minute.
Temperature	Indicates temperature expressed in either degrees Celcius or degrees Farenheit.
VerticalVisibility	tbd
Visibility	tbd
VisibilityDirection	tbd
VisibilityNautMiles	tbd
VisibilityValue	tbd
VisualVisibility	tbd
Winds	Provides wind direction, wind variation, wind speed and gust speed using *WindDirection*, *WindVariations* and *WindSpeed* data.
WindDirection	Specifies the direction of the wind expressed in degrees.
WindSpeed	Specifies the speed of the wind expressed in kilometre per hour mile per hour or knots.
WindVariations	tbd
Year	Provides the year between 1996 and 2095.

4.4 Chapter 2.4.5: PROTOCOL DEFINITION

4.4.1 Purpose of Chapter 5

<Editor's note: tbd>

4.4.2 Message Sequence Diagrams

4.4.2.1 Time sequence diagrams or message sequence diagrams are used to denote the relationship between the primitives that form a FIS service and the order in which they occur.

4.4.2.2 Implicitly, the concept of asynchrony is given through these Message Sequence Diagrams, e.g. the indication/confirmation primitives occurs some time after the request/response primitives.

4.4.2.3 Inherent to the service model is the notion of queuing. The FIS-service indications and confirmations are delivered to the FIS-users in the order that the corresponding FIS-service requests and responses were issued. The users can therefore initiate several services in parallel without having to wait between each invocation. One exception to the notion of queuing is the abortive services (FIS-user-abort and FIS-provider-abort services) which may overtake other primitives by empty the primitives in the queues.

4.4.3 Technical Timers

<Editor's note: define what technical timers are>

<Editor's note: explain derivation of technical timers (as opposed to operational)>

4.4.3.1 The assignment of values for timers must be optimised based on operational testing of the application. In such testing, incompatible timer values and optimum combinations can be identified. Implementations of FIS protocol are required to support configurable values for all timers and protocol parameters, rather than having fixed values. This allows modification as operational experience is gained.

4.4.4 State Machines

4.4.4.1 The FIS service provider is described in the SARPs as finite state machines or protocol machines (PM). The protocol machine for a particular service starts in an initial state (IDLE). Events, which are service primitives received from the FIS-users above or the Dialogue Service provider below, as they occur, trigger activity on the part of the PM. As part of this activity, actions may be required (service primitives issued to the FIS-users and/or the underlying Dialogue Service provider).

4.4.4.2 The protocol is described under two forms: the textual and the tabular descriptions. The textual description takes precedence over the tabular description.

4.4.4.3 Functional Model

4.4.4.3.1 The protocol activity is composed of three main tasks: the management of FIS contracts, the encoding of the Protocol Data Unit (PDU) and the management of the underlying dialogue supporting the communication.

Function	**Module**
Demand Contract Function	DC Module
Update Contract Function	UC Module
Contract Cancellation	CL Module
Automatic Termination of Dialogue	LI Module
Multiplexing FIS Contracts over a single Dialogue	LI Module
Exception Handling	AB Module
Interface with Dialogue Service	LI Module

4.4.4.4 Textual Description of the Protocol

4.4.4.4.1 Protocol is explained based on incoming events. An incoming event may be issued by the local FIS-service user or the Dialogue service-provider. For each state of the PM allowing the reception of the incoming event is listed the actions to be performed by the PM.

4.4.4.4.2 Actions are enumerated in the order they have to be carried out by the protocol machine.

4.4.4.4.3 Actions described must be implemented. What is not described is not allowed and must be covered by the Exception Handling procedures.

4.4.4.5 Tabular Description of the Protocol

<Editor’s note: "Cannot occur" is used when...>

<Editor’s note: "Not permitted" is used when...>

4.5 Chapter 2.4.6: COMMUNICATION REQUIREMENTS

4.5.1 Purpose of Chapter 6

<Editor’s note: tbd>

4.5.2 Encoding Rules

<Editor’s note: PER - choice and implementation>

4.5.3 Dialogue Service Requirements

<Editor’s note: QOS priority: where it comes from (ADSP), how they relate to other application, use by the ATN (in case of congestion), benefits and effets>

<Editor’s note: QOS RER High/low explanation>

4.6 Chapter 2.4.7: FIS USER REQUIREMENTS

4.6.1 Purpose of Chapter 7

<Editor’s note: justify chapter 7. Chapters 3 and 5 guarantee interoperability but not operational acceptability (use of service primitives, use of optional ASN.1 data, required relationships between parameters in the request and in the response (e.g. ATIS report when FIS demand report contains ATIS, definition of event for ADS,...).>

<Range and Resolution - from Manual - Distance units.>

<Answers to some "whys". Stand alone Elements (justify rules beyond the ADSP instructions)>

<What is conveyed in a FIS message.>

<Messages inserted / ordered to preserve an element of compatibility with previous systems.>

4.7 Chapter 2.4.8: SUBSETTING RULES

4.7.1 Purpose of Chapter 8

<Editor’s note: Any ground configuration works with any air configuration>

<a system will implement a configuration at a time>

<no correlation between version number and configuration>

<ASE implementation drives configuration>

<"do not use" options specify predicates>

4.7.1.1 An implementation of either the FIS ground based service or the FIS air based service claiming conformance to 2.4. must support the FIS protocol features as described below.

4.7.1.2 Only version 1 of the FIS(ATIS) protocol is defined. Any CNS/ATM-1 compliant FIS system must support this version.

4.7.1.3 Functionality supported by an air based FIS system is different from the functionality supported by a ground based FIS system. The air based FIS system acts as the originator of the FIS contracts whereas the ground based FIS system is the provider of the FIS information.

4.7.1.4 The FIS demand contract is always supported by both type of systems. The ground system can delay the uplink of the FIS report by sending first a positive acknowledgement and subsequently the FIS report itself. These functions constitutes the core functionality of the FIS protocol. The FIS-demand-contract and the FIS-report services are therefore always offered to air and ground users.

4.7.1.5 The FIS update contract is optionally supported by the FIS systems:

- the FIS update contract is a facility provided to the pilots in the aircraft upon the choice of the airline. When the update contract service is not available to the pilots, they must request the update of the FIS information by invoking the FIS-demand-contract service several times. They are not automatically warned of the update of the information they have received previously. The dialogue is not maintained between the aircraft and the FIS ground system and each new request causes the establishment of a new dialogue.
- the ground FIS systems provide the FIS update contract service upon the choice of the FIS service provider. When the FIS contract service is not available, the FIS ground system is able to indicate to a pilots having issued an FIS-update-contract request that

4.7.1.6 In case the update contract service is not provided by the ground FIS system, upon receipt of an FIS-update-contract request from an aircraft, the ground ASE forwards the request to the FIS-ground-user. The FIS-update-contract response primitive allows the FIS-ground-user to indicate that it can not provide updates of the FIS information and optionally to uplink a single FIS report. The FIS-update-contract service is therefore optionally supported onboard and mandatory supported on the ground even if the update contract service is not supported.

4.7.1.7 The FIS-cancel-contracts service is optionally offered to the pilots ??. The FIS ground system supporting the FIS update contract or not is able to receive a request of the cancellation of all contracts.

4.7.1.8 Only when the update contract is supported the FIS-cancel-update service is supported.

4.7.1.9 The FIS-demand-contract service is part of a core functionality of the FIS systems. It must be supported by any configuration of airborne and ground FIS systems.

4.7.1.10 The FIS-report service is part of a core functionality of the FIS systems. It must be supported by any configuration of airborne and ground FIS systems.

4.7.1.11 The FIS-update-contract service is part of a core functionality of the FIS systems. It must be supported by any configuration of airborne and ground FIS systems. However, the FIS-ground-user is allowed to not support the update service. In such a case, the FIS-ground-user invokes the negative FIS-contract-response to inform the FIS-air-user that the requested service is not available.

4.7.1.12 This service is provided by any configuration supporting the FIS-contract-update in full.

5. DIMENSIONS

5.1 PDU Size

<Editor's note: tbd>

5.2 Rate of Message Transmission

<Editor's note: tbd>

5.3 Number of FIS connections

<Editor's note: tbd>

6. ASN.1 INDEX

<Editor's note: tbd>

7. EXAMPLE OF OPERATIONAL SCENARIOS

7.1 Introduction

7.1.1 This section contains a set of example scenarios of use. The purpose of this section is to demonstrate different scenarios that are theoretically possible using FIS. It is not meant to indicate what is required from an operational point of view.

7.2 Scenario #1

7.2.1 The aircraft is equipped with CM, ADS, CPDLC and FIS(ATIS) applications.

7.2.2 Before take-off.

1. The pilot initiate the CM-Logon process with the ground CM covering the departure airport. The CM's response includes the address information for the ground CPDLC and FIS(ATIS) applications.
2. Due the potential for last minute weather changes, the pilot initiates a request for an FIS(ATIS) Update Contract for the departure airport. The ground FIS system accepts the contract and sends the current ATIS to the aircraft.

3. An indication alerts the flight crew to an updated ATIS. This ATIS indicates that the forecasted weather has arrived in the airport environment, and warns the crew of the approaching thunderstorms.
4. The aircraft The rain has arrived at the airport now, and the flight crew receives another update from the FIS(ATIS) contract. Although conditions are deteriorating, the RVR is acceptable for departure.

7.2.3 After take-off

1. The aircraft is now leaving the TMA environment and the pilot elects to cancel the FIS(ATIS) contract for the departure airport.
2. The pilot initiate the CM-Logon process with the ground CM covering the destination airport.
3. The pilot initiates a FIS(ATIS) Demand Contract for the destination airport. He informs the passengers about the weather conditions at the arrival airport.
4. Times to times, the pilot initiates a FIS(ATIS) Demand Contract for the arrival airport.
5. One hour before arrival time, the pilots establishes a FIS(ATIS) Update Contract for the destination airport. The ATIS information indicates that conditions at the airport are conducive to an approach to the alternate arrival runway.

7.2.4 After landing

1. The pilot request the cancellation of all FIS contracts still pending.

8. EXAMPLE ENCODING

8.1 Encoding/Decoding Rules

8.1.1 Samples of PER encoded FIS messages are provided in this section. For each APDU, the value tree containing the values assigned to the APDU components is provided. Then, the octet string resulting of the PER encoding is dumped in hexadecimal.

8.1.2 FIS Downlink APDU [FISRequest]

```
********************************************************************************
*                              FIS Request APDU                                *
********************************************************************************
FISDownlinkAPDU SEQUENCE COMPLETE
 {
    time SEQUENCE
     {
        date SEQUENCE
         {
            year INTEGER  = 1996
            month INTEGER  = 12
            day INTEGER  = 12
         }
        time SEQUENCE
         {
            timeHours INTEGER  = 11
            timeMinutes INTEGER  = 29
            timeSeconds INTEGER  = 0
         }
     }
    fisDownlinkAPDU CHOICE
     {
        fISRequest SEQUENCE
         {
```

```
            contractNumber INTEGER = 1
            contractType ENUMERATED VAL_DEF = demandContract(0)
            fISRequestdata CHOICE
             {
                aTISRequest SEQUENCE
                 {
                    airportID OCTET STRING = 0x4C46424F
                    arrivalDepartureIndicator ENUMERATED VAL_DEF = arrival(0)
                 }
             }
         }
     }
 }


********************************************************************************
*                                 PER STREAM                                   *
*                                ( 10 octets )                                 *
********************************************************************************
0000: 01 6b 5b a0 04 00 cc 8d 0a 78                   .k[......x
```

8.1.3 FIS Downlink APDU [FISAbort]

```
********************************************************************************
*                             FIS Downlink ABORT APDU                          *
********************************************************************************
FISDownlinkAPDU SEQUENCE COMPLETE
 {
    time SEQUENCE
     {
        date SEQUENCE
         {
            year INTEGER  = 1996
            month INTEGER  = 12
            day INTEGER  = 12
         }
        time SEQUENCE
         {
            timeHours INTEGER  = 11
            timeMinutes INTEGER  = 29
            timeSeconds INTEGER  = 0
         }
     }
    fisDownlinkAPDU CHOICE
     {
        fISAbort CHOICE
         {
            atis ENUMERATED = protocolError(1)
         }
     }
 }


********************************************************************************
*                                  PER STREAM                                  *
*                                ( 6 octets )                                  *
********************************************************************************
0000: 01 6b 5b a0 20 40                                      .k[. @
```

8.1.4 FIS Uplink APDU [FISAccept(positive-acknowledgement)]

```
*****************************************************************************
*                    FIS Accept - Positive Acknowledgement                  *
*****************************************************************************
FISUplinkAPDU SEQUENCE COMPLETE
 {
    time SEQUENCE
     {
        date SEQUENCE
         {
            year INTEGER  = 1996
            month INTEGER  = 12
            day INTEGER  = 12
         }
        time SEQUENCE
         {
            timeHours INTEGER  = 11
            timeMinutes INTEGER  = 29
            timeSeconds INTEGER  = 0
         }
     }
    fisUplinkAPDU CHOICE
     {
        fISAccept SEQUENCE
         {
            contractNumber INTEGER = 1
            fISAcceptData CHOICE
             {
                positiveAcknowledgement NULL = nil
             }
         }
     }
 }

*****************************************************************************
*                                 PER STREAM                                *
*                                ( 6 octets )                               *
*****************************************************************************
0000: 01 6b 5b a0 00 04                                .k[...
```

8.1.5 FIS Uplink APDU [FISReport]

```
********************************************************************************
*                              FIS Uplink Report APDU                          *
********************************************************************************
FISUplinkAPDU SEQUENCE COMPLETE
 {
    time SEQUENCE
     {
        date SEQUENCE
         {
            year INTEGER  = 1996
            month INTEGER  = 12
            day INTEGER  = 12
         }
        time SEQUENCE
         {
            timeHours INTEGER  = 11
            timeMinutes INTEGER  = 29
            timeSeconds INTEGER  = 0
         }
     }
    fisUplinkAPDU CHOICE
     {
        fISReport SEQUENCE
         {
            contractNumber INTEGER = 1
            fISReportData CHOICE
             {
                atis SEQUENCE
                 {
                    airportId OCTET STRING = 0x4C46424F
                    aTISInformation CHOICE
                     {
                        arrivalATIS SEQUENCE
                         {
                            aTISCode OCTET STRING = 0x41
                            aTISTimeofObservation SEQUENCE OPT
                             {
                                timeHours INTEGER = 14
                                timeMinutes INTEGER = 13
                             }
                            arrivalRunwaysinUse SEQUENCE OF  SEQUENCE (1)
                             {
                                Runway SEQUENCE
                                 {
                                    runwayId SEQUENCE
                                     {
                                        runwayNumber INTEGER = 3
                                        runwayLetter ENUMERATED OPT = runwayleft(1)
                                     }
                                    brakingAction SEQUENCE OPT
                                     {
                                        brakingActionFirstHalf CHOICE
                                         {
                                            brakingActionQuality ENUMERATED = good(2)
                                         }
                                        brakingActionSecondHalf CHOICE OPT
                                         {
                                            brakingActionQuality ENUMERATED = poor(0)
                                         }
                                     }
                                 }
                             }
                            commonATISInfo SEQUENCE
                             {
                                surfaceWinds CHOICE
                                 {
                                    calmIndicator NULL = nil
                                 }
                                visibility SEQUENCE
                                 {
                                    value CHOICE
                                     {
                                        meters INTEGER = 30
                                     }
                                 }
                                cloudSkyCoverGroup SEQUENCE OF  SEQUENCE (1)
```

```
                                 {
                                    -------- SEQUENCE
                                     {
                                        cloudAmount ENUMERATED = skyclear(0)
                                        cloudHeight CHOICE
                                         {
                                            cloudHeightMeters INTEGER = 1000
                                         }
                                     }
                                 }
                                airTemperature CHOICE
                                 {
                                    temperatureC INTEGER = 10
                                 }
                                dewPointTemperature CHOICE
                                 {
                                    temperatureC INTEGER = 0
                                 }
                                altimeterSetting SEQUENCE
                                 {
                                    qNH CHOICE
                                     {
                                        hPa INTEGER = 10000
                                     }
                                 }
                                significantMetInformation CHOICE
                                 {
                                    nosig NULL = nil
                                 }
                                transitionLevel CHOICE OPT
                                 {
                                    altitudeFlightLevel INTEGER = 350
                                 }
                             }
                            arrivalATISInfo SEQUENCE
                             {
                                approachType CHOICE OPT
                                 {
                                    approachType ENUMERATED = ils(0)
                                 }
                             }
                         }
                     }
                 }
             }
         }
     }
 }
```

```
********************************************************************************
*                                 PER STREAM                                   *
*                               ( 26 octets )                                  *
********************************************************************************
0000: 01 6b 5b a0 10 02 64 68 53 cc 17 1a 02 84 68 02 .k[...dhS.....h.
0010: 2f 01 f4 16 8a 02 71 05 02 00                   /.....q...
```