Aviation Information Data Exchange (AIDX)

XML Implementation Guide

Effective Date
March 2022

Notice

DISCLAIMER. The information contained in this publication is subject to constant review in the light of changing government requirements and regulations. No subscriber or other reader should act on the basis of any such information without referring to applicable laws and regulations and/or without taking appropriate professional advice. Although every effort has been made to ensure accuracy, the International Air Transport Association (IATA), Airlines for America (A4A) and Airports Council International (ACI) shall not be held responsible for any loss or damage caused by errors, omissions, misprints or misinterpretation of the contents hereof. Furthermore, the International Air Transport Association (IATA), Airlines for America (A4A) and Airports Council International (ACI) expressly disclaims any and all liability to any person or entity, whether a purchaser of this publication or not, in respect of anything done or omitted, and the consequences of anything done or omitted, by any such person or entity in reliance on the contents of this publication.

Senior Vice President
Marketing and Commercial Services International Air Transport Association
800 Place Victoria
P.O. Box 113 Montreal, Quebec
CANADA H4Z 1M1

Document Versioning and Change Log

The log describes the actions taken for the document's development.

Version	Date	Change description
1.0	13-Jul-2010	Initial Publication
1.1	11-Nov-2010	Updated RepeatNumber to reflect min of 1 max of 9, Section 4.1.6 and App. D.
1.2	5-Mar-2012	Updated based on AIDX 12.1 changes.
2.0	19-Mar-2013	General update – Significant changes
3.0	21-Feb-2014	Updated based on AIDX 14.1 changes.
15.2	21-Sep-2015	AIDX 15.1 & 15.2 changes. Appendix. C Table clarifications. Appendix H added. Version number aligned with schema version.
16.1	01-Mar-2016	AIDX 16.1 Changes.
17.1	07-Mar-2017	AIDX 16.2 & 17.1 Changes
22.1	01 March 2022	General update, other update to reflect schema changes from version 17.1 and onward. Update of ReasonCode reference to new delay code schema AHM 732

References and Location

AIDX web page on the IATA web-portal
IATA Developer Portal single place to access IATA and open industry innovative resources (e.g., standards, schemas, API's, datasets, etc.) available to the external developer community.
IATA's Airline Industry Data Model (AIDM)
Templates for Change Requests
Latest IATA Data Exchange Standards

Workgroup members or interested parties should send their comments and requests to:
AIDX@iata.org

Table of Contents

1 Introduction

1.1 Purpose and Scope of the Document

This document is the Implementation Guide for AIDX XML messaging. It has been developed by the IATA AIDX Group under the auspices of the Travel Standards Board (TSB). This document adheres to the best practices documented by IATA for the development of XML Schemas. It is designed to help users to communicate on an airport wide basis through standard messaging implemented according to the Technical Standard for AIDX in the Aviation Industry.

This Guide will help relate the components involved in implementation to one another and a typical aviation environment, consistent with the requirements established by the Technical Standard. The components are:

- *IATA XML Best Practices*
- *AIDX Business Requirements Document*
- *AIDX Schemas*
- *XML Change Request Documents*

Taken together, these components provide the means to produce the common message delivery and data needed by the aviation industry in a standardized format.

1.2 Out of Scope Items

This document does not prescribe which underlying transport mechanisms should be used to deliver the XML documents, nor does it prescribe a specific methodology for the integration of communicating systems.

1.3 About this Document

The structure of this document is as follows:

- Section 1: Introduction
  This section introduces the document and describes the purpose and scope of the Implementation Guide.
- Section 2: Why Implement the Technical Standard?
  This section addresses the business rationale and operational issues that are driving the implementation of the Business Requirement Document
- Section 3: Understanding the Business Requirements
- Section 4: AIDX Messages
  Rules and processes associated with the use of AIDX messages
- Section 5: Business Information
  Business information relating to the use of AIDX

- Section 6: Schema Management
- Section 7: Business area contact
- Appendices
  The appendices contain useful reference information for implementers.

1.4 Intended Audience

This Implementation Guide is intended to be used as a resource for airlines, airports and vendors interested in deploying the Aviation Information Data Exchange (AIDX) schema, version 8.1 and newer. This document will present how the schema architects (the IATA XML Working Group; the CUPPS subcommittee; the ACI-NA AIDX team and a host of airline, airport and vendor participants) expected AIDX would be utilized.

A number of topics were identified during the development of the schema which are fundamental to one obtaining a clear understanding of those architects' expectations. Neither the XML notations nor the Recommended Practice document (RP1797a) provided an appropriate place for these instructions.

AIDX, in its purest sense, is simply an XML schema. There is no one set methodology for implementing AIDX. As explained in the Recommended Practice (RP1797a) and Business Requirements Documents, the schema itself may be deployed in communications between an airline and an airport or between an airline and vendor or by an airport strictly within its own infrastructure. Thus, any discussion of an implementation process must be looked at as just one example for deploying this schema.

1.5 History

Airlines rely on the accurate and timely movement of flight data around the world to airports and other stakeholders, whose business relies on the accurate and timely movement of aircraft and passengers the world over. Communications tools and systems have continued to improve in a number of areas of the aviation business. Yet airlines, airports and their system vendors have recognized that the methods of moving flight data between airlines, airports and their respective systems have not evolved as quickly.

Flight data has been sent in the same manner, generally over dedicated telephone circuits, using several similar but nonetheless unique data formats for years. As vendor systems evolved and data exchange requirements changed, airlines found they were supporting an increasing variety of unique data exchange processes and formats. This came at significant expense particularly as systems needed to evolve to new platforms and to meet new business requirements.

The growth of the Internet and its network resources and programming tools presented some intriguing options for airlines and airports in need of moving such amounts of data around the globe quickly, accurately and efficiently. The Aviation Information Data Exchange aims to resolve the inefficiencies of these older data exchange practices by defining a single, open format for the exchange of flight data between any airline, airport and vendor participating in this standard.

Airlines for America (A4A) and Airports Council International (ACI) identified the need for such a standard at the "Seattle Summit" in 2003. It established the FIMS (Flight Information Management System) Subcommittee of ACI-North America to define a new multi-purpose

flight data interface specification. The committee consisted of airlines, airport service providers, government agencies and airport departments. The "FIMS" meetings were held over a period of 18 to 24 months and included all of these stakeholders.

By 2005 the group had identified the content and structure for a multi-purposed data interface utilizing the open Internet "XML" format. The resulting definition file identified 103 elements specific to flight data and another 34 elements for ground statistics. This new standard embodied the hope that one standard would ultimately exist for all IATA/A4A/ACI stakeholders and was adopted by the IATA AIDX sub-group as AIDX version 7.1 in 2007.

During 2008 a Recommended Practice (RP) document and Business Requirements Document (BRD) were crafted to assist with the formulation of a more universally-accepted schema by the IATA XML Working Group. This new AIDX schema (version 8.1) was released in 2008 as a replacement for the older FIMS schema.

Subsequent releases of the AIDX schema have added data elements and attributes in response to requests from members. In particular, the 12.1 schema released in 2012 adds support for the various worldwide initiatives for Collaborative Decision Making (CDM).

1.6 Why Implement the AIDX XML Messages

This section sets the operational context and describes the business drivers and objectives for implementing the AIDX schema as defined through the Business Requirements Document. The Technical Standard has three main benefits for aviation as follows:

1. Cost savings through the use of a single, common, authorized standard:
   - Utilizes common IATA code sets and XML schemas
   - Fewer costly data feed changes will be needed as systems evolve and standardize on AIDX

2. Faster time to market due to:
   - Use of a mature standard
   - Support from product vendors

3. XML technology allows AIDX to join other XML-crafted standards such as:
   - TypeX (as the delivery envelope)
   - BCBP (Passenger Data Exchange)
   - SIDX – Schedule Information Data Exchange (SSM/SSIM/ASM)
   - XML for Slot processes (Chapter 6 in SSIM manual)
   - EDI/XML for PNR Push/AQQ/API

4. Numerous airlines, airports and vendors have deployed AIDX, are planning AIDX efforts or are waiting on others to spur their developments in the future.

2 Understanding the Business Requirements

The Business Requirements Document (BRD) provides the basis for the XML Schema for AIDX developed within the AIDX Standards Sub Group and was subjected to a wide review open to all industry members of the XML Working Group including suppliers. Following the review, it was approved by the XML Steering group and ratified by AIDX sub Group. In addition, the XML development must adhere to the IATA XML Best Practices as well as ensuring that the terminology adheres to the standard Data Dictionary and glossary for the operation.

The BRD will not be updated to reflect changes as the schema evolves to meet the changing requirements of the industry. These changes are documented in various Change Request documents following the IATA change management process. This Implementation Guide will be updated as required in response to these changes.

3 AIDX Messages

3.1 Principles and Considerations

3.1.1 Bi Lateral Agreements

The way in which AIDX is to be used at a particular site should be documented and agreed by all parties in a Bi-Lateral Agreement. Please refer to section 4.1 and Appendix B.

3.1.2 Flight Legs

The principal data structure within AIDX refers to a flight leg, i.e. an aircraft movement covering the departure at the originating airport and the arrival at the destination airport. A flight with points of call en-route therefore consists of multiple flight legs, each with its own set of data. The data associated with an aircraft turnaround at an airport will be contained within two flight leg data structures, one associated with the arriving flight leg and one associated with the departing flight leg.

For a multi-leg flight, i.e. a flight with one or more points of call between its origin and destination which keeps the same flight number throughout, the AssociatedFlightLegSchedule data element identifies previous or subsequent flight legs forming part of the same flight. Similarly for an aircraft turnaround, the AssociatedFlightLegAircraft data element identifies a previous or subsequent flight leg operated by the same aircraft.

For examples of the use of the associated flight leg data elements, please refer to Appendix H.

3.1.3 Multiple Message Destinations

The AIDX XML structure allows for the creation of one message which can be consumed by both departure and arrival airports. This is only applicable should an airline (or other originator of flight data) wish to implement this option. Any change to these flight details, relating to either the origin or destination, must be communicated to both airports.

3.1.4 Multiple Flight Legs in a Message

It shall be permitted to provide more than one flight leg in the same message, this is accommodated by allowing the FlightLeg element to be a repeating element. A Flight Leg Identifier must be provided for each leg sent. There is no limit on the number of flight legs that can be sent within one message.

This is provided to allow more flexible processing and to handle the case where a ‘batch’ of data is to be exchanged, for example once a day.

3.1.5 Code Share Flights

Data associated with code shares is sent in the CodeShareInfo data element within the flight leg data structure associated with the operating airline flight identifier. There shall not be a separate flight leg data element for each code share.

3.1.6 Elements and Attributes

Elements are used to contain data values and attributes are used to describe the data item. A more complete description can be found in the IATA XML Best Practices for Message Development document (see “References and Location” section at the beginning of this Implementation Guide).

3.1.7 Nil Values in Schema

For any particular data element, there is an important distinction between the following cases:

1. The element is missing from a message.
2. The element is present but has a 'nil' attribute assigned to it.

In the first case, this means that the sender is supplying no information about the element. This would typically be because no information is available, or because there has been no change in the value of the element. A recipient would not be expected to take any action as a result of this. In particular, it would not be expected to clear any existing value.

In the second case, this means that the sender has explicitly cleared the element. As a result of this, a recipient would typically clear any value that it had previously stored for this element.

The following table shows an example sequence of messages, and the expected actions taken by the data recipient. It should be noted, however, that the action is up to the recipient. The message from the sender is a notification, and not a command for the recipient to take action.

Message contents	**Expected action by recipient**
PassengerGate element missing	Does not set or change gate value
<PassengerGate>A23</PassengerGate>	Sets gate value to A23
PassengerGate element missing	Does not change the gate value
<PassengerGate xsi:nil="true">	Clears gate value

The sender should not use blank data elements such as <PassengerGate/> or <PassengerGate></PassengerGate> as these may cause validation errors. For example, if the schema specifies a format or a minimum length for an element, then a zero-length element will be invalid. This problem does not occur if using the nil attribute.

Note that where a field is defined as mandatory in the schema, it must not contain a nil value.

3.1.8 Character Encoding

AIDX should be used with UTF-8.

3.1.9 Repeating Elements

A number of elements in the schema are permitted to be provided more than once i.e. repeated. This is to permit more than one value or set of values to be provided. In general the maximum number of times an element can be provided (i.e. repeated) is set within the schema.

Once a repeating element is used the data sender should keep the data in the same order in any subsequent data transfers, and if one element has changed then all the other unchanged elements must also be included.

In many cases a RepeatIndex attribute is defined for use with a repeating element. The RepeatIndex is used to specify the order of elements within a list. The first element in the sequence should be allocated a RepeatIndex of 1, with following elements having a RepeatIndex of 2, 3, 4 etc. If a specific element within the list is used in subsequent messages, the same RepeatIndex attribute as defined in the original message should be used, so that an element can be tracked between messages.

3.1.10 Date and Time References

- The date/time reference "xsd:DateTime" is used throughout AIDX.
- All date and time references in AIDX must be in UTC and must be explicitly shown as UTC using a trailing Z. For example: 2012-08-25T11:35:00Z.
- The Receiver shall recalculate to local time (as needed).

3.1.11 Unique Flight Leg Identifier

A consistent mechanism for uniquely identifying a flight is critical to the operation of AIDX messaging. For a full discussion of this topic please refer to appendix E.

3.2 Message Types

AIDX uses the following three message types:

IATA_AIDX_FlightLegNotifRQ – contains one or more flight leg records

IATA_AIDX_FlightLegRQ – a request for flight leg records

IATA_AIDX_FlightLegRS – a response to either of the other two messages

Each message is defined by a separate schema file, each of which also includes other AIDX XML schemas and standard IATA XML schemas.

3.3 Message Control

The AIDX schemas include message types (FlightLegRQ, FlightLegRS) which can be used as part of an overall mechanism to control the flow of messages, by implementing a request-response sequence. Details can be found in section 3.4. The use of these message types is optional, but their inclusion in the AIDX schema set gives implementers the flexibility to specify a message control mechanism which best fits the characteristics of the systems exchanging information. Use of these message types forms an important part of the bilateral agreement which defines the interface between systems (see Appendix B).

3.3.1 IATA_AIDX_FlightLegNotifRQ

IATA_AIDX_FlightLegNotifRQ is used to transfer unsolicited flight records between entities like airlines, airports, data aggregators and vendors. The message can be sent to downstream consumers based on a received update trigger or on a schedule or time interval.

A single instance of an IATA_AIDX_FlightLegNotifRQ message may include a single flight leg record, or multiple flight leg records.

3.3.2 IATA_AIDX_FlightLegRQ

IATA_AIDX_FlightLegRQ is used to request flight data records from a partner (i.e. airline or data aggregator).

This message type currently allows the requestor to specify a carrier code as the only parameter for the request. If the carrier code is not provided, all relevant carrier flights should be provided. Partners may agree to other implicit criteria for the response records

(e.g. only data for the current day – or the next 24 hours) as part of the bi-lateral interface agreement.

The response to this request should be either:

a) An immediate, synchronous response (of type IATA_AIDX_FlightLegRS) containing the flight data records requested, or
b) A simple, synchronous acknowledgement (of type IATA_AIDX_FlightLegRS), followed by an asynchronous transmission with the requested data (of type IATA_AIDX_FlightLegNotifRQ).

3.3.3 IATA_AIDX_FlightLegRS

IATA_AIDX_FlightLegRS is used as an acknowledgement message to be returned as a response to a notification (IATA_AIDX_FlightLegNotifRQ) or a synchronous response to a flight data request (IATA_AIDX_FlightLegRQ)

The message should indicate either the successful processing of the initial message – or provide informative error messages if errors occurred in the parsing or processing of the data.

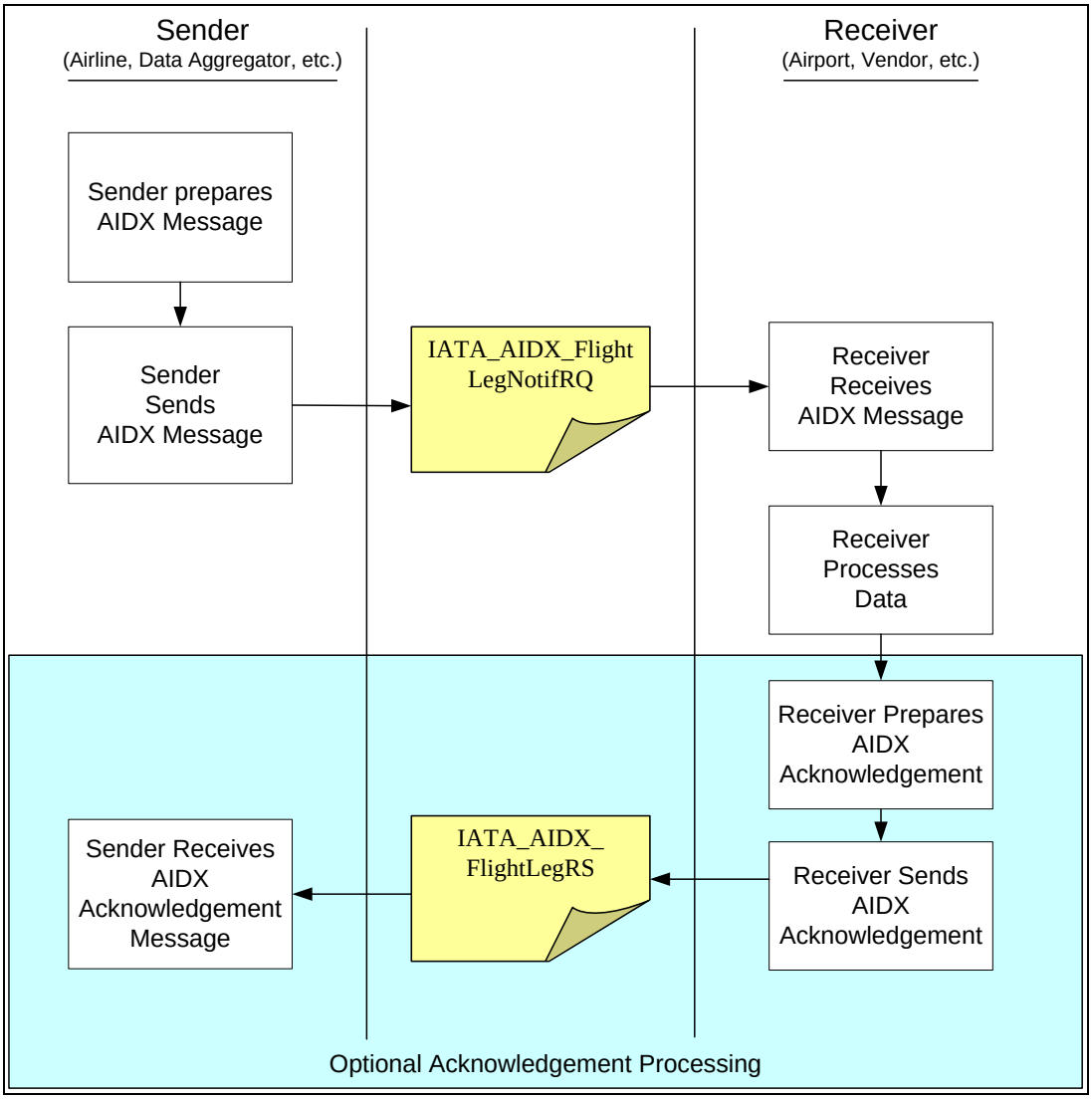
3.4 Data flows

Below are several simple use cases – and the associated data flows and message types used. These are meant as basic examples – and are certainly not the only integration patterns possible. In fact, by combining these integration flow patterns in different ways, and by utilizing other transport methods – like brokers and queues – there are a very large number of possible solutions.

3.4.1 Unsolicited Notification with Optional Synchronous Acknowledgement

In this use case, the sender transmits flight data to the receiver based on either a trigger of some kind, a set time of day – or after a set time interval. The receiver must have an open channel or listener waiting for the incoming AIDX messages. Once the receiver processes

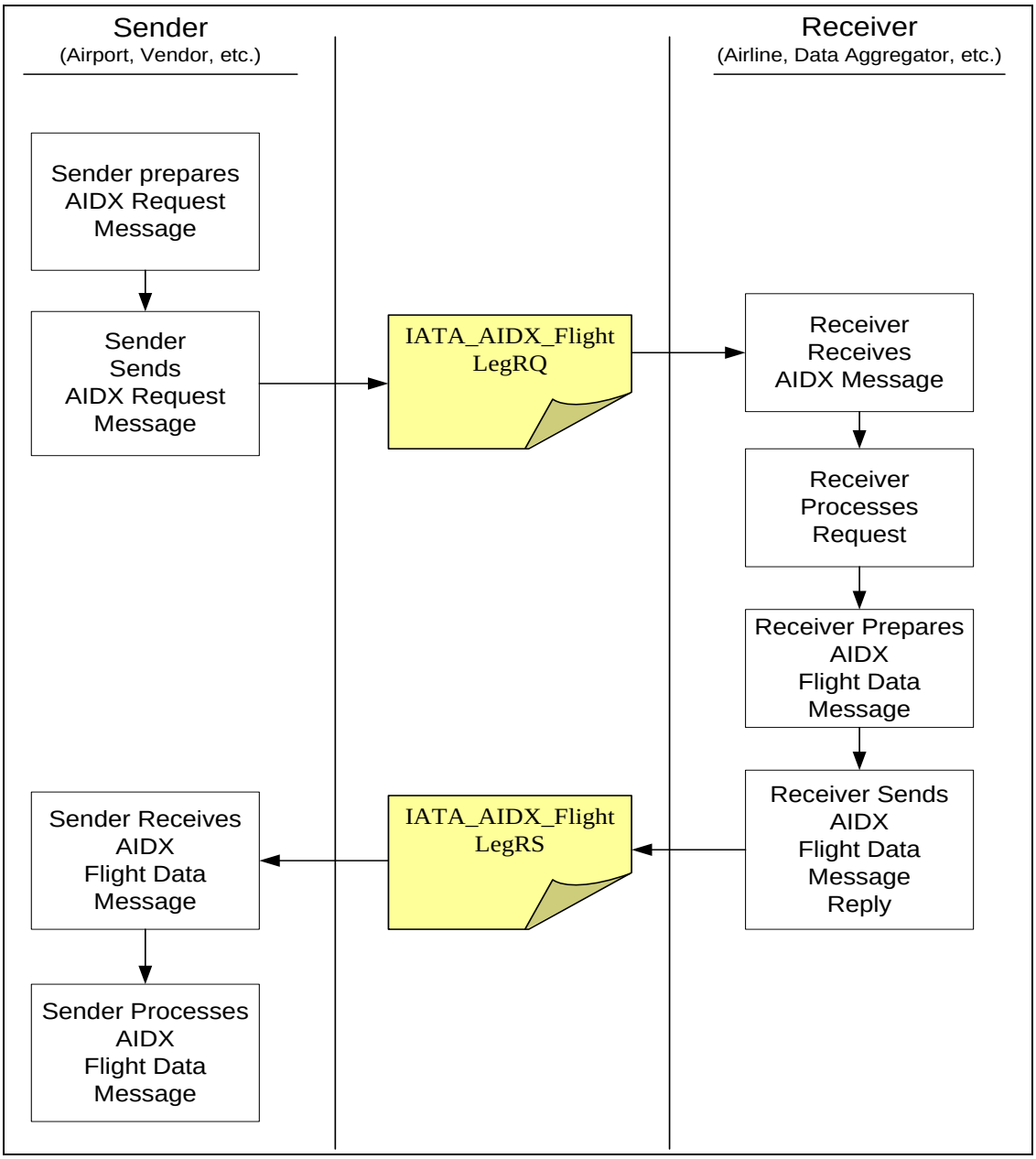
the AIDX XML message, it may send a synchronous acknowledgement indicating either success or failure – along with any relevant error messages.



3.4.2 Request with Synchronous Reply

In the following two use cases, the "Sender" represents an entity that wishes to receive flight data back from the other partner (here called the "Receiver"). In this use case, the receiver of

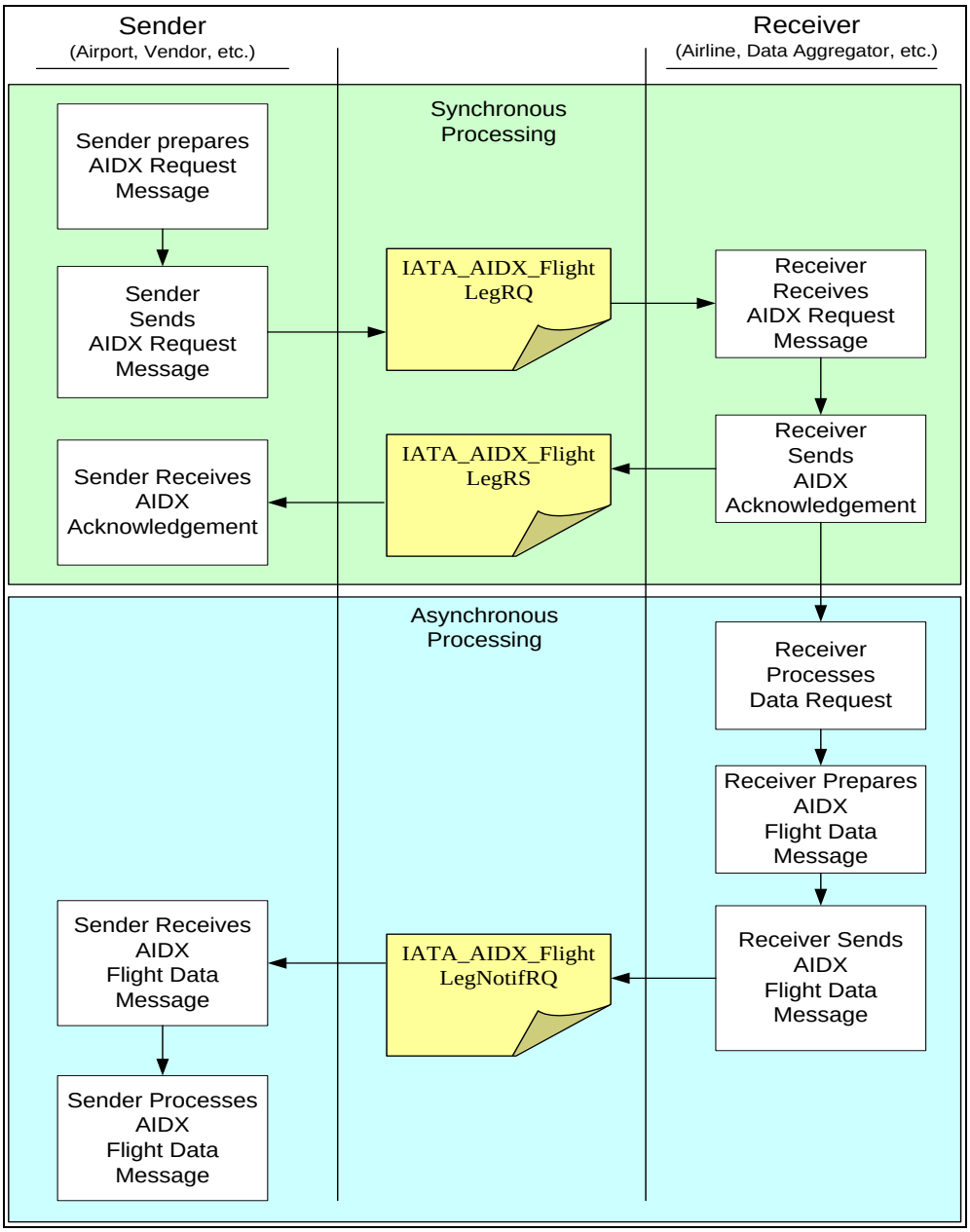
the request retrieves the requested data and sends it back to the initial "Sender" in a synchronous reply message.



3.4.3 Request with Synchronous Acknowledgement and Asynchronous Reply

This use case is similar to the previous case – except that here, the data reply is sent back at a later time – using an asynchronous model. This integration pattern is sometimes preferrable to the previous, synchronous case, because the retrieval and packaging of the requested data may take a significant amount of time. It is usually best practice to keep these types of data connections open for no more than a few seconds at a time. Also, if the sender (of the

request) is already set up to receive and process asynchronous data message notifications, that same channel can be repurposed to receive the reply data.



3.5 Use of Codesets

Many data elements and attributes within the AIDX schema are populated with data from one or more IATA code-sets. The intent is to provide a set of data values which are understood by all users of the AIDX message and to avoid the use of ad-hoc codes which would compromise the AIDX standard. These CodeSets are maintained by IATA and posted on the IATA Developer Portal site for public use. There is no charge for use of these codesets.

The codesets have evolved over a number of years to satisfy many different applications, consequently not all of the codes in a given codeset are relevant for use with AIDX.

Appendix D of this document lists the preferred subsets of codes for use within an AIDX message.

Unmanaged changes to codesets can result in significant deterioration of the AIDX standard and should be avoided.

4 Business Information

4.1 Bilateral Agreements

Each time the AIDX schema is used it is recommended and encouraged that the two parties involved set up and use a bilateral agreement. The objective of the bilateral agreement is to define the expectation the two parties have from the interface, in terms of data, performance, protocol, etc.

It will be an important tool when testing the implementation and in the maintenance of the AIDX interface. It will enable those designing and implementing the AIDX to record how it operates in this particular deployment.

The bilateral agreement does not replace the schema but can be used alongside, as part of or instead of an Interface Control or Definition document.

Appendix B sets out the topics that could be included in a bilateral agreement. The actual contents of the agreement may differ for each implementation and will be defined and agreed by the two parties implementing the AIDX interface.

It is suggested that a Confidentiality Agreement is also signed between the parties to protect each other's interests.

4.2 Irregular Operations Procedures

Appendix C has a list of irregular operations that could impact the flight schedule, operational status or assigned resources, and indicates how these scenarios should be handled in terms of AIDX messages.

Some scenarios such as flight diversion may lead to creation of new flight legs that will trigger sending an updated AIDX message. The new message contains information about estimated departure/arrival times for the new flight.

The level of detail in each AIDX message should be defined in the bilateral agreement between the interfacing parties, e.g. include all scheduled, estimated and actual information, if known at the time of sending the message.

4.3 Collaborative Decision Making (CDM)

CDM is more and more important within the air transport industry, with a number of initiatives being launched by national and international air traffic control organisations. AIDX can be used to transfer information between parties involved in CDM. The current release of AIDX is a standard for SESAR A-CDM information exchange, ACI ACRIS A-CDM Web Services and supported as the data exchange standard for A-CDM by ICAO (ICAO A-CDM Implementation Plan, Asia Pacific). Specific information on this topic can be found in Appendix D, sections D2.1 and D3.3.

4.4 Communications

The AIDX standard defines the message data content and high level message control in XML schemas. It does not define the underlying communication protocols used to transport the

message. Selection of the communication protocol depends on the data exchange requirements such as:

- Expected reliability
- Guaranteed message delivery
- Sensitivity to message duplication
- Scalability
- Overall cost and timeframe
- Expected throughput
- Security requirements
- Existing IT infrastructure

The transport mechanism to be used should be fully described in the bilateral agreement between the interfacing parties.

4.5 Message Security Standards

Different security measures can be incorporated in a data exchange. Each measure will cover a certain security concern. Some of the security concerns that should be considered in a data exchange are:

- Authenticity
- Integrity
- Availability
- Confidentiality
- Non repudiation

Message security is outside the scope of the AIDX standard, but should be agreed and documented in the bilateral agreement between the interfacing parties.

4.6 Schema Validation

In general, each time a data sender composes and sends an AIDX message it shall be subject to XML validation before the message is transmitted.

In general, each time a receiver receives an AIDX message it should be subject to schema validation.

This double level of schema validation may be relaxed or reduced only with agreement between the two parties. At least a single level of schema validation should be completed; either the sender or the receiver should perform schema validation for every message exchanged.

Complete removal of schema validation is not recommended or encouraged.

5 Schema Management

5.1 Change Requests

AIDX changes request is to be reviewed by the Change Management & AIDM Integration Group (CMIG) (sub-group of Architecture and Technology Strategy Board ATSB) and according to the approved IATA modeling guidelines

Change requests and codesets updates should be submitted to the IATA AIDX Group for an initial review and approval prior to submission to the CMIG.

The XML WG meets to validate any new developments and also change requests to the schema. A change request template can be found on the IATA extranet site (see "References and Location" section at the beginning of this Implementation Guide).

Once submitted to the XML WG for approval, it is required that a representative be present to explain the requirement either at the meeting or through conference call, and answer any questions that may be raised.

There are up-to 4 releases/year for the various group to schedule changes and get the IATA Standards XML/API messages delivered in the IATA Development portal as official publication and available to the industry.

Recent changes to the schema are listed in Appendix K of this guide.

5.2 Schema and Implementation Issues

5.2.1 Compatibility with Earlier AIDX Schemas

The starting point for the development of AIDX was the FIMS schemas developed under the auspices of ACI, which were adopted by the IATA AIDX Sub Group as AIDX version 7.1. Significant changes were made to the schemas for version 8.1, to align with IATA policies, and these changes were not designed to be backwards compatible.

For subsequent versions of AIDX, there should be very few problems with backward compatibility, since the changes are generally additive.

5.2.2 Use of the TPA_Extensions

Additional data elements can be added to the schema as part of the "TPA_Extensions" type. This provides flexibility in allowing additional data to be included in the schema to satisfy the requirements of a local implementation. Clearly the addition of such elements needs to be agreed as part of the bi-lateral agreement between interfacing parties.

The extension capability should be used only after very careful consideration, since this will make the schema unique to the particular implementation and therefore compromise the ability to share data with other parties in the future.

If a business need is identified for including new elements into the schema, this should be notified to the IATA AIDX Group, using the contact details in section 6, so that consideration can be given to the permanent inclusion of the data elements as part of the standard.

6 AIDX Point of Contact

For information relating to the development and update of the AIDX schema, and to raise an issue for discussion please contact the AIDX Secretary via email at aidx@iata.org .

For more information please visit the IATA AIDX Website

Appendix A – Glossary of Terms

AIDX – Aviation Information Data Exchange (formerly known as FIMS)

Bilateral Interface Agreement – A documented agreement made between the sender and the receiver prior to the live operation of each message interface. This agreement defines a number of features which are mandatory and optional within this specification and may include commercial restrictions concerning the proprietary nature of the data.

Carrier/airline – The term "carrier" is used interchangeably with the term "airline" in this document.

Code Set – A list of required values used to standardize data content and meaning. Existing Code Sets from IATA will be used as the default. Additional necessary codes will be added to the PADIS codeset directory. See references section for location of document.

FIDS - Flight Information Display System

FIMS – Flight Information Management System

Flight – the airborne activity of an aircraft defined by one primary identifier and possibly one or more additional identifiers (i.e. code shares). A flight may comprise from one to many flight legs.

Flight Leg – An aircraft movement comprising the flight between a departure airport and the corresponding arrival airport.

Marketing airline – a carrier with an agreement (with an operating airline) to jointly promote a flight, also known as a code share. A passenger may purchase a ticket from the marketing airline for a flight of the operating airline. The marketing airline may assign their own flight number to the flight and often the marketing airline's name, logo and flight number are displayed to the public.

Multi Sector Flight – A flight comprised of more than one flight leg.

Operating Airline – the airline that carries out the flight, this will be the airline name on the passengers' ticket. In the majority of cases the owner and operating airline are the same, but not all e.g. Air Wisconsin own and fly aircraft for United who are the operating airline. The SSIM definition of the Operating Airline is the "Administrating Carrier".

Operational Window –That period agreed between the relevant parties, in which updates to flights operating in the window are required for distribution. Note, the primary use of the AIDX standard is during the operational window, but wider usage is not excluded.

Owner Airline – the organization that owns and maintains the aircraft. This will be the airline name used in the Air Traffic Control (ATC) filed flight plan. The flight number used by the owner airline may differ from that used by the operating airline.

Single Sector Flight – A flight comprised of a single flight leg.

Scheduling Window – That period agreed between the relevant parties as to the time period before the schedules are confirmed and published. This window will depend upon airline and could be 2 – 30 days prior to the operational window.

Unique Flight Identifier – The data fields which together define a unique flight leg.

For a complete Aviation Industry glossary please refer to Recommended Practice RP1008 as published in the IATA Passenger Services Conference Resolutions Manual.

Appendix B - Sample Agreements

A bilateral agreement between two parties that wish to exchange AIDX Messages to share flight data should include (but not be limited to) the following topics:

- Key assumptions.
- Description of services and processing logic.
- Data flow diagrams.
- Acknowledgements expected.
- Triggering events.
- Lower level communication details (e.g. physical medium, network stack, IP address, port numbers, URL).
- Details of transport mechanism and protocols to be used (e.g. SOAP over TCP/IP, web service, message queue etc.) (see section 4.4).
- Transport security requirements (SSL, authentication credentials, etc.).
- Required and optional data fields to be used.
- Specific data usage options, e.g. whether or not multiple flight legs are included and use of RepeatNumber as part of the flight ID.
- Agreement on handling of irregular operations procedures.
- Agreed upon valid values lists (Utilizing IATA code-sets) (see Appendix D).
- Exception handling and retry logic.
- Resynchronisation and data recovery.
- Service level agreements, including data quality and timeliness.
- Bilateral agreement management process (change processing).
- Sample message data.

It should be noted that data senders may use different operational windows. The resolution of different senders' operational windows is the responsibility of the receiver.

In some cases there could be multiple parties sending the same flight information, for example estimated arrival time sent from an airline or Air Traffic Control authority or third party. There may be differences in the data depending on the senders. It is the responsibility of the receiver to resolve this issue based on the provisions of the Bilateral Interface Agreement. Ideally, at a global level, each data element should have only one owner who is responsible for updating the data to avoid conflicts associated with multiple updates from different sources.

Appendix C – Irregular Operations

This appendix summarises a number of scenarios, highlighting indicative triggers that could cause them, and suggested AIDX data interactions to handle the event. These scenarios should be covered in a data exchange interface contract with description on how each of these conditions is handled; e.g. automated or manual business process.

C1. Single Leg Flight Route Change Scenarios

Considered here are the cases relating to single leg flights. The original route is AAA-BBB. The following scenarios are considered:

After the departure of the given flight leg:

1. Ground Return – where the aircraft never gets airborne, also referenced as Return to Stand or Gate Return. Two different scenarios can occur after the ground return:
   a) The flight leg is not operated with the aircraft never getting airborne.
   b) The original flight leg is operated.
2. Return from Airborne – where the aircraft gets airborne but then returns to the airport it has departed from. Two different scenarios can occur after the return from airborne:
   a) The flight terminates at the origin station. AAA-BBB becomes AAA-AAA only.
   b) The original flight leg is operated. AAA-BBB becomes AAA-AAA then AAA-BBB.
3. Diversion – a change is made to the destination airport subsequent to the aircraft getting airborne. Two different scenarios can occur after the completion of the diverted leg:
   a) The flight terminates at the diversion station. AAA-BBB becomes AAA-CCC.
   b) The flight continues to the original destination. AAA-BBB becomes AAA-CCC-BBB.

Prior to departure of the given flight leg:

4. Flight Leg Planned Re-route – corresponding to an IATA standard schedule change message for an ad-hoc re-route change.
5. Flight Leg cancelled – corresponding to an IATA cancellation message for an ad-hoc or standard change.

For each of the above scenarios a further case to consider is that the original route is reinstated after the decision to change the route has been made. These separate scenarios are now discussed in more depth.

C1.1 Ground Return

A Ground Return event is indicated by:

- Last station in the PlannedArrivalAptHistory list matches the DepartureAirport.
- Off blocks and on blocks times are held but no take off or landed times are recorded because the aircraft never got airborne.
- OperationalStatus set to “GRT”.

The time range between the decision to return and completion of the return is relatively short i.e. in the order of taxi times; and has no intermediary stage (unlike an Airborne Return when both a landed and then an on-blocks time are recorded in separate events before the return is completed). From an airline perspective it is therefore viewed as sufficient to only acknowledge the ground return event when the on-blocks time is recorded. For an airport or

Ground Services system it may be worthwhile knowing about the ground return before the event is completed and therefore more detail will be required.

C1.1.1 Before the Event

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	AAA	Off Blocks	–	BBB

C1.1.2 After the Return, Leg Still to be Operated

May happen for a number of reasons, e.g. a passenger medical emergency or a flight deck technical alert happening before take-off.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	AAA	Off Blocks / On Blocks	GRT	BBB / AAA
AAA	BBB	2	AAA	–	–	BBB

C1.1.3 After the Return, Leg now Cancelled

Happens when the issue cannot be resolved in time for the flight to operate, e.g. the technical alert cannot be resolved.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	AAA	Off Blocks / On Blocks	GRT	BBB / AAA
AAA	BBB	2	Not operational	–	DX	BBB

C1.2 Airborne Return

C1.2.1 Before the Event

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to BBB	Off Blocks / Take Off	–	BBB

C1.2.2 After the Decision, Leg Cancelled

Happens for a number of reasons, e.g. when a passenger medical emergency or a flight deck technical alert happen after take-off. The decision on whether to expect to continue on the original route or not after the return will depend on the circumstances of the event e.g. if the operating hours of AAA mean the leg will be too late to depart again.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to AAA	Off Blocks / Take Off	DV	BBB / AAA

C1.2.3 After the Decision, Leg Still to be Operated after Return

Happens for a number of reasons, e.g. when a passenger medical emergency or a flight deck technical alert happen after take-off. The decision on whether to expect to continue on the original route or not after the return will depend on the circumstances of the event e.g. if the operating hours of AAA mean the leg will be too late to depart again.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	AAA	Off Blocks / Take Off / Landed / On Block	DV	BBB / AAA
AAA	BBB	2	AAA	–	–	BBB

C1.2.4 After the Return, Leg Cancelled

May happen if the circumstances change further either before or after landing e.g. it becomes apparent that the aircraft needs significant maintenance at AAA.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	AAA	Off Blocks / Take Off / Landed / On Blocks	DV	BBB / AAA
AAA	BBB	2	Not operational	–	DX	BBB

C1.2.5 After the Decision, Route Reinstated Before Return

May happen if the cause of the return from airborne is resolved before the aircraft lands e.g. the crew resolves the technical alert.

The PlannedArrivalAptHistory contains AAA for no other reason than to keep a record that a Return From Airborne was considered.

The OperationalStatus has been reset by setting to SQ (see section 3.1.7) from DV because the leg is now back to operating the original route.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to BBB	Off Blocks / Take Off	SQ	BBB / AAA / BBB

C1.3 Diversion to a New Station

C1.3.1 Before the Event

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to BBB	Off Blocks / Take Off	–	BBB

C1.3.2 After the Decision, Divert and Terminate

Diversion is to station CCC.

Happens for a number of reasons, e.g. the weather deteriorates at the arrival airport, BBB. The decision on whether to expect to continue to the original station or not will depend on the circumstances of the event e.g. the practicalities of terminating the service at CCC. It is assumed not necessary to provide a message for the now non-operating AAA-BBB leg because the status of this leg can be derived from BBB being in the PlannedArrivalAptHistory field but with BBB not the last station held in the list.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to CCC	Off Blocks / Take Off	DV	BBB / CCC

C1.3.3 Divert, Land, then Continue to Original Destination

Happens for a number of reasons, e.g. the weather deteriorates at the arrival airport, BBB. The decision on whether to expect to continue to the original station or not will depend on the circumstances of the event e.g. the practicalities of terminating the service at CCC. A new flight leg is created to cover the CCC – BBB leg of the flight.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	CCC	Off Blocks / Take Off / Landed / On Blocks	DV	BBB / CCC
CCC	BBB	1	CCC	–	-	BBB

C1.3.4 Divert, Land then Cancel Continuation to Original Destination

May happen if the circumstances change further either before or after landing at CCC e.g. the weather at BBB will be bad for longer than expected or the delay means BBB will be closed by the time the flight can now reach it. A new flight leg CCC – BBB was created when the decision was made to continue, but now the decision has been reversed the new flight leg is cancelled.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	CCC	Off Blocks / Take Off / Landed / On Blocks	DV	BBB / CCC
CCC	BBB	1	Not operational	–	DX	BBB

C1.3.5 Reinstate Original Route before Divert Completed

May happen if the cause of the diversion is resolved while the aircraft is still in the air e.g. the weather improves at BBB.

The PlannedArrivalAptHistory contains CCC for no other reason than to keep a record that a Diversion was considered at some point. The OperationalStatus is reset by setting to SQ from DV (see section 3.1.7) because the leg is now back to operating the original route.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to BBB	Off Blocks / Take Off	SQ	BBB / CCC / BBB

C1.4 Re-Route

C1.4.1 Re-Route to a New Station

A Re-Route to a new station is essentially equivalent to a diversion to a new station, see section C1.3, except the re-route events happen *before* the off blocks and take off times have been recorded for the flight leg in question. For a re-routed leg the OperationalStatus field would be set to RT (re-routed) rather than DV.

No distinction is made here between a Scheduled Re-Route (SSM based) and an ad-hoc Re-Route (ASM based) event.

The PlannedArrivalAptHistory field is populated for Re-Route cases to provide a history of what the previous routing was, which may have some value.

It is assumed that for a re-route, the original destination BBB will still be held in the PlannedArrivalAptHistory field but the last station in the list is the new destination CCC, which will mean that BBB will know to no longer expect this service to arrive.

Re-routes can also extend a route as well as redirect it e.g. a flight with route AAA-BBB can be extended to have route AAA-BBB-EEE. In this circumstance a new leg would be added to the flight sequence and the PlannedArrivalAptHistory field of the original leg is left unchanged. Note in this case the OperationalStatus remains empty because the re-route has not changed the arrival station of the legs, just a new leg has been added. This case is illustrated here:

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	AAA	–	–	BBB
BBB	EEE	1	AAA	–	–	EEE

There is a working assumption that if a re-route caused the origin station of the first leg of a flight to be changed i.e. the route changes from AAA-BBB to FFF-BBB, then the flight from AAA is first cancelled and a new flight from FFF is then created.

Re-Routes happen for a number of reasons e.g. a multi-leg flight no longer stops at an intermediary station or a decision is made not to night-stop at a given location for a temporary period.

C1.4.2 Special Case: A Diverted Re-Route

Take the example where a leg has been re-routed to a new station and the leg now terminates at the new station. The message for this event, derived from section C1.3.2, is as follows: -

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	AAA	–	RT	BBB / CCC

Now suppose the leg gets airborne from AAA, recorded with the following message

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to CCC	Off Blocks / Take Off	RT	BBB / CCC

An issue causes the leg to be diverted to a new station, DDD, with the flight terminating there. This would be recorded as follows

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to DDD	Off Blocks / Take Off	RT / DV	BBB / CCC / DDD

The above case is used to demonstrate what happens when two separate events happen to the same leg and to justify why PlannedArrivalAptHistory field is required. It also illustrates why more than one OperationalStatus field entry may be required.

If a BBB based system is still concerned with this flight leg then its interest can be derived from BBB still being held in the PlannedArrivalAptHistory field. An example of why BBB may still be interested in this leg is so members of the public waiting for the arrival of the leg can be told where the leg is now arriving instead of BBB.

If a DDD based system needs to know that the leg was originally destined for BBB then this can be derived from the PlannedArrivalAptHistory field. An example for this requirement is that where BBB and DDD are in different countries then there may be security or access issues causing the ground staff at DDD to need to know that the passengers were expecting to arrive at BBB.

C1.5 Cancellation

Although a record of a cancelled flight leg may not be of importance to an airport it can be valuable for an airline to have a record of any cancelled legs and the associated cancellation reason e.g. for EU Passenger Compensation Rules.

C1.5.1 Before the event

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	AAA	–	–	BBB

C1.5.2 After the Cancellation has been Actioned

Note BBB is not removed from the PlannedArrivalAptHistory field even though the cancellation means that the leg does not actually arrive at this station.

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Not operational	–	DX	BBB

C2. Route Change Scenarios for Multi-Leg Flights

Where a flight has multiple legs there are a few special cases that need to be considered. A multi-leg is taken to be a two-leg flight i.e. with route AAA-BBB BBB-CCC. The cases when the route change is made to the last leg of the flight are exactly the same as those covered in section C1 above. Further consideration is required only when the change is made to other legs i.e. the first leg in a two-leg flight. Similarly the behaviour around ground returns, returns from airborne and cancellations for all legs is exactly the same as that outlined in sections C1.1, C1.2 and C1.5 respectively. The behaviour for reinstating the disrupted legs is also covered by the different cases outlined in section C1.2 above. The following scenarios are to be considered for multi-leg flights. For a flight with legs AAA-BBB BBB-CCC there are four specific cases:

1. An over-fly – the flight operates AAA-CCC only.
2. The flight diverts to a new destination and stops there – the route becomes AAA-DDD.
3. The flight diverts to a new destination but then continues to the original destination of the given leg – the route becomes AAA-DDD DDD-BBB BBB-CCC
4. The flight diverts to a new destination but then continues to the final destination, missing the intermediary stop – the route becomes AAA-DDD DDD-CCC i.e. no arrival at BBB.

The examples used below to illustrate these cases are all based on diversions but, as with the single leg flights, a re-route is equivalent to a diversion except it happens before the leg has the off blocks and take off times recorded.

C2.1 Multi-Leg Over-Fly Diversion

This may happen on a flight where the aircraft has the range to reach CCC as a single leg and if, for example, bad weather prevents the arrival at BBB.

C2.1.1 Before the event

The starting point for each of these multi-leg diversion cases is the same, as shown here for a flight leg originally scheduled to operate route AAA-BBB BBB-CCC

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to BBB	Off Blocks / Take Off	–	BBB
BBB	CCC	1	Flying to BBB	–	–	CCC

C2.1.2 After the Diversion Decision

Where the route becomes AAA-CCC, the event would be represented as follows: -

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to CCC	Off Blocks / Take Off	DV	BBB / CCC
BBB	CCC	1	Not operational	–	DX	CCC

In the above it is necessary to set the BBB-CCC leg to have an OperationalStatus of DX because the flight is not operating. It has not been explicitly cancelled, but it is not operating because of a route change.

C2.2 Multi-Leg Divert and Terminate to New Station

This case may happen for the causes of a diversion of the AAA-BBB leg already discussed.

C2.2.1 Before the event

As already depicted above in C2.1.1

C2.2.2 After the Diversion Decision

Where the route becomes AAA-DDD, the event would be represented as follows: -

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to DDD	–	DV	BBB / DDD
BBB	CCC	1	Not operational	–	DX	CCC

It is necessary to set the BBB-CCC leg to have an OperationalStatus field of DX because the flight is not operating. It has not been explicitly cancelled, but it is not operating because of a route change.

The details that the leg is no longer arriving at BBB can be detected from BBB appearing in the PlannedArrivalAptHistory field for the AAA-DDD leg.

C2.3 Multi-Leg Divert to New Station with Continuation

This case may happen for the causes of a diversion of the AAA-BBB leg already discussed.

C2.3.1 Before the event

As already depicted above in C2.1.1

C2.3.2 After the Diversion Decision

Where the route becomes AAA-DDD DDD-BBB BBB-CCC, the event would be represented as follows: -

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to DDD	Off Blocks / Take Off	DV	BBB / DDD
DDD	BBB	1	Flying to DDD	–	–	BBB
BBB	CCC	1	Flying to DDD	–	–	CCC

For this case if for some reason the service was terminated at DDD then both the DDD-BBB and BBB-CCC legs would be cancelled.

C2.4 Multi-Leg Divert to New Station, Continue but Skip Next Destination

This case may happen when the disruption causing the diversion from BBB to DDD e.g. bad local weather, has not cleared before the decision is made to continue on to CCC.

C2.4.1 Before the event

As already depicted above in C2.1.1

C2.4.2 After the Diversion Decision

Where the route becomes AAA-DDD DDD-CCC, the event would be represented as follows:

DEP AIRPORT	ARR AIRPORT	REPEAT NUMBER	ACTUAL CURRENT LOCATION	TIMES RECORDED	OPERATIONAL STATUS	PLANNED ARRIVAL AIRPORT HISTORY
AAA	BBB	1	Flying to DDD	Off Blocks / Take Off	DV	BBB / DDD
DDD	CCC	1	Flying to DDD	–	–	CCC
BBB	CCC	1	Not operational	–	DX	CCC

In the above messages there is an assumption that the fact that the original AAA-BBB leg is not operating can be derived by BBB based systems from the fact that BBB appears in the PlannedArrivalAptHistory for the now AAA-DDD leg.

Appendix D – Preferred Codeset Values

D1. Use of Codesets

Several elements within the schema are populated from various IATA codesets. The codesets have evolved over a number of years to satisfy many different applications, consequently not all of the codes in a given codeset are relevant for use with AIDX. There are also codes in different codesets which have a similar meaning, leading to the possibility of different implementers choosing a different code for the same purpose.

This appendix identifies the reference data that are typically used in an AIDX data exchange, listed as preferred sub-sets of the relevant IATA codesets. The intention is to standardise the codes that are significant and essential for an interoperable AIDX data exchange. Use of consistent codes improves the interoperability of different implementations, leading to increased reusability and speed to market.

The schema indicates which of the IATA codesets are used for populating a particular element. Codes which are listed in the IATA code sets but not addressed here can still be adopted by bi-lateral agreement between the parties exchanging the data, provided the rules defined in the schema are obeyed.

D2. Flight Status Qualifiers

D2.1 Operational Status

Associated data elements:

- LegData/OperationalStatus
- LegData/PublicStatus

These elements target codes that are operationally significant. This provides the current operational status of a flight. At airports where CDM is being used, this can include the CDM status. Note that use of the PublicStatus element is now deprecated, since the RemarkTextCode element can be used for status indications aimed at the public.

Note that flight status can also be inferred from the LegData/OperationTime element – if for example an actual start of boarding time has been sent, it can be inferred that the flight is now boarding.

The LegData/RemarkTextCode element can be used (see sectionD2.2 of this appendix) for status information where no processing is actually required on the data and it is purely for display to staff or public.

Supported codesets: 1245, 2005 and 9750.

Code Value	Meaning	PADIS Codeset Reference
Operational Status		
DV	Flight diverted	2005
DX	Flight cancelled	2005
RT	Re-route	2005
GRT	Ground Return	2005

Code Value	Meaning	PADIS Codeset Reference
SQ	Re-instate a cancelled or diverted flight	1245
NOP	Non-Operational flight: Planned flight that is not actually used - typically part of season planning, but is not running.	1245
OP	Operational Flight : Flight in operation	1245
CDM Status		
SCH	Scheduled	9750
INI	Initiated (Flight Plan activated)	9750
TKO	Airborne / Departed	9750
FIR	Flight entered local Flight Information Region	9750
FIN	Final approach	9750
LAN	Landed	9750
ONB	On block	9750
SEQ	Sequenced (TSAT issued)	9750
BST	Boarding	9750
RDT	Ready for start	9750
OFB	Off block	9750
DIR	Ready for de-icing	9750
DIC	De-icing in progress	9750
Type of route (only for AIDX version 11.1 and earlier)		
7DO	Domestic flight – version 11.1 and earlier	1245
7IN	International flight – version 11.1 and earlier	1245

D2.2 Displayed Status

Associated data element:

- LegData/RemarkTextCode

This element can be used if there is no processing required on the information, i.e. it is for use only for staff or public display.

Supported codesets: 2005 and 9750.

Code Value	Meaning	PADIS Codeset Reference
Operational Status during the flight		
DV	Flight diverted	2005
DX	Flight cancelled	2005
RT	Re-route	2005
GRT	Ground Return	2005
Status of a flight – Airport view, e.g. FIDS or staff information screen		
BST	Boarding	9750

Code Value	Meaning	PADIS Codeset Reference
BEN	Final Boarding	9750
GCL	Gate Closed	9750
FCL	Flight Closed	9750
OFB	Departed	9750
THM	In Range	9750
STE	Stack Entry	9750
STX	Stack Exit	9750
TEN	Approach	9750
LAN	Landed	9750
ONB	Arrived	9750
EAR	Early	9750
SCT	On Time	9750
DEL	Delayed	9750

The "Qualifier" attribute to this data element takes codes as defined below:

Supported codesets: 9932.

Code Value	Meaning	PADIS Codeset Reference
AIR	Air Side	9932
BAG	Baggage Area	9932
CHK	Check-in Area	9932
COU	Checking-in counters	9932
GTE	Gate area	9932
LND	Land Side	9932
LOU	Boarding Lounge	9932
PAR	Parking area, stand or Apron	9932
PUB	Public Area	9932
STF	Staff Area	9932
TER	Terminal – Visible to Public	9932

D3. Operational Time Qualifiers

Associated data element:

- LegData/OperationTime

D3.1 Current Schema

There are two attributes associated with this data element which contain qualifiers defined by the codeset:

- OperationQualifier
  The activity to which the time relates, e.g. touchdown, on-blocks.
- TimeType
  The significance of the time provided, e.g. estimated, actual.

Supported codesets: 2005 and 9750.

Code Value	Meaning	PADIS Codeset Reference
Operation Qualifiers		
CHK	Check-in Open	9750
THM	In Range	9750
TEN	Approach	9750
TDN	Touchdown time	9750
ONB	On Block time - Arrival	9750
CGT	Commence of Ground Handling Time	9750
FBG	First bag unloaded	9750
ABA	Air bridge attach	9750
LBG	Last bag unloaded	9750
CHC	Check-in Closed	9750
GTO	Gate Open	9750
BST	Start Boarding Time	9750
FCT	Final Call Time: Time of final call in lounge before the aircraft gate is closed.	9750
BEN	Final Boarding	9750
GCL	Gate Close	9750
FCL	Flight Closed	9750
ABD	Air bridge detach	9750
RDT	Ready Time. (The time the pilot informs the ATC of being ready for pushback.)	9750
SRT	Start up Request Time. (Time the pilot requests start up clearance.)	9750
SAT	Start up Approval Time. (Time that an aircraft receives its start up approval.)	9750
OFB	Off Block time – Departure	9750

Code Value	Meaning	PADIS Codeset Reference
DIC	De-ice start time	9750
DIE	De-ice end time	9750
TKO	Take Off time	9750
Time Types		
SCT	Scheduled	2005
PLN	Planned	2005
EST	Estimated	2005
TAR	Target	2005
CAL	Calculated	2005
ACT	Actual	2005

D3.2 AIDX Schema version 11.1 and Earlier

For version 11.1 and earlier of the AIDX schema, there was no TimeType attribute. In this case, a single code in the OperationQualifier attribute determined both the operation and the time type. When using these schemas, the following codes should be used:

Code Value	Meaning	PADIS Codeset Reference
CHK	Check-in Open	9750
THM	In Range	9750
TEN	Approach	9750
EA	Estimated arrival touchdown information – Time	2005
TDN	Actual Touchdown time	9750
SCA	Scheduled On Block time - Arrival	2005
EB	Estimated On Block time – Arrival	2005
OB	Actual On Blocks time – Arrival	2005
CGT	Commence of Ground Handling Time	9750
FBG	First bag unloaded	9750
ABA	Air bridge attach	9750
LBG	Last bag unloaded	9750
CHC	Check-in Closed	9750
GTO	Gate Open	9750
BST	Start Boarding Time	9750
FCT	Final Call Time: Time of final call in lounge before the Aircraft gate is closed.	9750
BEN	Final Boarding	9750
GCL	Gate Close	9750
FCL	Flight Closed	9750
SCD	Scheduled Off Block time – Departure	2005

Code Value	Meaning	PADIS Codeset Reference
ED	Estimated Off Block time – Departure	2005
AD	Actual Off Blocks time – Departure	2005
ABD	Air bridge detach	9750
DIC	Actual time of Deice	9750
EO	Estimated take off information – Time	2005
TKO	Actual Take Off time	9750

D3.3 CDM Support

The following table identifies how the various qualifier values can be used to support the milestones defined by Eurocontrol A-CDM.

CDM Acronym	CDM Term	Operation Qualifier	TimeType
ELDT	Estimated Landing Time	TDN	EST
ALDT	Actual Landing Time	TDN	ACT
EIBT	Estimated In-Block Time	ONB	EST
AIBT	Actual In-Block Time	ONB	ACT
ACGT	Actual Commence of Ground Handling Time	CGT	ACT
ASBT	Actual Start Boarding Time	BST	ACT
ARDT	Actual Ready Time	RDT	ACT
TSAT	Target Start Up Approval Time	SAT	TAR
ASRT	Actual Start Up Request Time	SRT	ACT
ASAT	Actual Start Up Approval Time	SAT	ACT
SOBT	Scheduled Off-Block Time	OFB	SCT
TOBT	Target Off-Block Time	OFB	TAR
EOBT	Estimated Off-Block Time	OFB	EST
AOBT	Actual Off-Block Time	OFB	ACT
ECZT	Estimated Commencement of De-icing Time	DIC	EST
ACZT	Actual Commencement of De-icing Time	DIC	ACT
EEZT	Estimated End of De-icing Time	DIE	EST
AEZT	Actual End of De-icing Time	DIE	ACT
TTOT	Target Take Off Time	TKO	TAR
CTOT	Calculated Take Off Time	TKO	CAL
ATOT	Actual Take Off Time	TKO	ACT

D4. Handling Agents

Associated data element:

- LegData/AircraftInfo/AgentInfo

The Qualifier attribute to this data element should be populated using the codes given below. Note that the FLT code should be used for all activities not covered by a dedicated agent, e.g. if all functions performed by the same handler, that handling agent should be identified as type FLT.

Supported codeset: 3035

Code Value	Meaning	IATA Codeset Reference
BAG	Baggage handling	3035
PAX	Passenger handling	3035
CAT	Catering	3035
FUE	Fuel handling	3035
FLT	Flight handling	3035

D5. Cabin Class

Associated data element:

- LegData/CabinClass/@Class
- LegData/AircraftInfo/Baggage/@ServiceClass
- LegData/AirportResources/Resource/CheckInInfo/@Class
- LegData/AircraftInfo/CrewInfo/@Qualifier

Supported codeset: 9873

Code Value	Meaning	IATA Codeset Reference
1	First	9873
2	Business	9873
3	Third Class (All economy)	9873
4	Economy Premium	9873
5	Economy	9873
6	Economy Discounted	9873
7	All	9873

D6. Passenger Numbers

Associated data element:

- LegData/CabinClass/PaxCount

The "Qualifier" attribute associated with this data element can be used to identify specific groups of passengers using the 6353 codeset in agreement with all parties participating in the interface. An example might be the use of the "UM" code to indicate the number of unaccompanied minors.

Numbers of passengers with specific types of reduced mobility can also be specified by using the codes defined in section 2.12.6 of IATA Recommended Practice 1708a.

In all other cases, code "70A" (total number of passengers) should be used. Breakdown of passengers on a per-class basis is performed using the "Class" attribute of the LegData/CabinClass attribute (see section D5).

Supported codeset: 6353

D7. Crew Information

Associated data element:

- LegData/AircraftInfo/CrewInfo

The "Qualifier" attribute specifies the cabin classes associated with the crew, please see section D5.

D8. Error Types

The "Type" attribute to the "Error" data element used in the IATA_AIDX_FlightLegRS schema can contain values as follows:

Supported codeset: 9321

Code Value	Meaning	IATA Codeset Reference	Error Handling
294	Invalid Format	9321	Do not resend the message, until the format issue addressed.
911	Unable to process - system error	9321	Retry and resend the message.

D9. Customs Clearance Agreement

Associated data element:

- LegData/ClearanceAgreement

Supported codeset: 9970

Code Value	Meaning	IATA Codeset Reference
TRB	Transborder	9970
INT	International	9970
DOM	Domestic	9970
SCH	Schengen	9970

D10. Baggage Reclaim Type

Associated data element:

- LegData/AirportResources/Resource/BaggageClaimUnit/@AreaLocation

Supported codeset: 9988

Code Value	Meaning	IATA Codeset Reference
DOM	Domestic	9988
INT	International	9988
TRA	Transit	9988
TRS	Transfer	9988
SCH	Schengen	9988

Appendix E – Unique Flight Identifiers (UFI)

E1. General Principles

The AIDX Unique Flight Identifier was designed to represent the ideal. As such, no consideration was given to the current design of existing airline or airport systems. On this basis, the design does not make any compromise for the status quo, but represents a target that system designs can move towards.

It is understood that some current industry systems will not be able to provide all of the data elements exactly as defined, and some suggestions for adaptations are provided in section E2.8 of this appendix.

The original design was aimed at supporting scheduled commercial flights, and no specific consideration was given to General Aviation flights. This shortfall has been addressed in version 16.2 and later versions of the schema, however a method for supporting general aviation flights using earlier schema versions is provided in sectionE3 of this appendix. Using the latest schema, either the LegIdentifier, or the GeneralAviationLegIdentifier is populated, but not both.

E2. Commercial Flights

The unique flight leg identifier for commercial flights is provided in the LegIdentifier data element, which comprises the following data items:

- Airline
- FlightNumber
- OperationalSuffix (optional)
- OriginDate
- DepartureAirport
- ArrivalAirport
- RepeatNumber (optional)

A discussion of the individual elements follows:

E2.1 Airline Code

The IATA or ICAO code for the airline operating the flight.

E2.2 Flight Number

As defined in the IATA Standard Schedules Information Manual (SSIM) chapters 4, 5, and 6. Flight numbers of two digits or less are padded with leading zeros to a length of three digits.

E2.3 Operational Suffix

Sometimes omitted in industry UFI's, and sometimes misunderstood. IATA allows any single character A-Z to appear in the Suffix. The only recommended use is for the 'Z' character. The 'Z' character is used to distinguish between two flights with the same airline code and flight number that are scheduled for departure in the same UTC date. One of the flights will carry a 'Z' suffix**.** For a complete discussion see Appendix 'H' of the IATA Standard Schedules Information Manual (SSIM) under "Time Mode".

There seems to be a misunderstanding regarding use of the suffix in some industry systems where it is modified when a flight activity is delayed from a prior day. This is not the correct use of the suffix particularly as a component of a UFI. The suffix value is static for the history of the given flight. Once the flight has been created the suffix value cannot be altered. To add or remove a suffix the current flight must be cancelled and a fresh flight created with the required suffix value.

The description of the 'Z' suffix in the IATA SSIM manual is clear, however another possible cause for misunderstanding of the 'Z' suffix is identified in Appendix H of the SSIM manual. This correctly points out that the 'Z' suffix may be suppressed in systems that work in local time; the requirement to use the suffix to distinguish between two flights on the same UTC date may not occur when the date is converted to local time. However where scheduling would cause a local time duplicate the mandated procedure is to create one of the two duplicate flights with an entirely different flight number.

E2.4 Origin Date

The origin date is the UTC scheduled date of departure of a flight. When this is a single flight leg, the origin date is the same as the scheduled date of departure of the flight leg. However, for a multi-leg flight, the origin date for each flight leg in the flight is the scheduled date of departure of the first flight leg in the series. For example a flight SFO-DEN-LHR will have 2 flight legs SFO-DEN and DEN-LHR both of which will have the origin date of the departure from SFO.

The origin date ties all the flight legs in a flight together, and without this link, there would be no way of identifying the relationship between all flight legs in a multi-leg flight; the start of the flight's first leg and finish of its last leg may span more than one calendar day or indeed 24 hours.

Airport oriented industry systems may use the scheduled date of an arrival or scheduled date of a departure in place of the origin date. These systems are not in a great position to communicate in a complete manner with airlines. Typically this results in systems having to exchange additional 'contextual' data, when available, in order to fill in the requirements of each other's flight key. This can be a cause of an ongoing IT support burden when certain data items are not immediately available, such as during operational disruption.

The origin date is a static value for the history of a given flight. Note especially that this is true even if the flight's first flight leg is rescheduled such that the STD is a different UTC date. The only way to change the origin date is to cancel the original flight leg and create a new one with the required value.

E2.5 Departure Airport Code

The IATA or ICAO code for the airport from which the flight leg originates.

E2.6 Arrival Airport Code

This is the originally scheduled arrival airport, and does not change if the flight is diverted or re-routed. The originally scheduled arrival airport is generally available to industry systems, but outside of AIDX is often not included as part of the flight identifier, and even overwritten in the case of diversions and re-routes. This can cause a good deal of confusion where multiple diversions take place, and the originally intended arrival airport is obscured.

From an airport perspective it is important to understand both scheduled flights that are expected to arrive, and flights that were scheduled to arrive and that subsequently will not. AIDX maintains a Planned Arrival Airport History, to track changes to the arrival airport (Please see Appendix C for further information).

E2.7 Repeat Number

There are circumstances that can cause aircraft operators to conduct more than a single attempt to operate a flight leg; as problems may occur on the ground or in the air. The first attempt to operate a flight has a Repeat Number of '1', and each subsequent attempt increments the Repeat Number. Where Repeat Number is part of the UFI, each repeat will create an additional unique flight leg. (For details of the various possible scenarios and the usage of repeat number for each case, please see Appendix C).

The repeat number is often not present in industry systems. The repeat number is an optional element in the AIDX UFI, as it is accepted that not all systems need to expose the level of granularity associated with a repeat number.

Some systems may need to know what repeat number a particular passenger was on – particularly if there was any disembarkation between attempts, or to understand the timing of events that might be of concern for engineering or flight training purposes. Repeat Number can be key for recording certain events e.g. the number of 'landings' performed by an aircraft. It is also worth noting that different aircraft may operate two legs distinguished only upon the basis of the repeat number.

Other systems may just be concerned with the latest repeat number that actually arrived at their airport, with no interest in the number of attempts that occurred.

E2.8 Suggested Adaptation for Systems with non Compliant UFI

A typical scenario is where an industry system has no visibility of the origin date, but does have the scheduled date of departure or the scheduled date of arrival.

In this circumstance, the industry system would be advised to make a Bilateral Interface Agreement, which would stipulate that the scheduled date of departure and/or the scheduled date of arrival become mandatory in the OperationTime section of the message. To indicate that the OriginDate field is not being used, this field should be populated with a dummy date of Jan 1st 2000. This approach should be used with extreme caution, since it deviates from the standard and will therefore compromise the ability to exchange data with other systems using the same implementation.

E3. General Aviation (GA) Flights

E3.1 Current Schema

In version 16.2 and later versions of the schema, GA flights are handled by using the "GeneralAviationLegIdentifier" part of the schema. This allows the ATC callsign, aircraft registration or other value to be used as the primary identifier, with the departure airport and planned departure time added to make the identifier unique. Other optional elements are included for use if the GA flight is being operated by a commercial airline.

The GeneralAviationLegIdentifier data element comprises the following data items:

- Airline
- GeneralAviationIdentifier
- DepartureAirport
- PlannedDepartureDateTime
- FlightNumber (optional)
- OperationalSuffix (optional)
- ArrivalAirport (optional)
- RepeatNumber (optional)

The optional elements are only used if the GA flight is being operated by a commercial airline, in which case they are populated according the rules described in sections E2.2, E2.3, E2.6 and E2.7. A discussion of the remaining elements follows:

E3.1.1 Airline

Generally, the "Airline" element should be populated with a carrier code of "GN" (and "CodeContext" attribute of "3"), unless a commercial airline is operating the flight, in which case the rules outlined in section E2.1 apply.

E3.1.2 General Aviation Identifier

Typically this element will be populated with either the ATC callsign for the aircraft, or the aircraft registration, but could be any item agreed with all the interfacing parties which uniquely identifies the flight when taken together with other elements of the "GeneralAviationLegIdentifier". The mandatory "Category" attribute must be populated with values of "Callsign", "Registration" or "Other", as appropriate.

E3.1.3 Departure Airport Code

The IATA or ICAO code for the airport from which the flight leg originates. If the departure airport has neither an IATA code nor an ICAO code, then an identifier mutually agreed between the interfacing parties can be used, with the "CodeContext" attribute set to a value of "ZZZ" if there is no code defined in IATA Codeset 3055 for the type of code used.

E3.1.4 Planned Departure Date Time

This element is used to differentiate between multiple flights using the same aircraft from the same airport. Will normally be the UTC time of the planned departure. Once set, the element does not change to reflect the estimated or actual time since the identifier must be static. A different value of PlannedDepatureDateTime would represent a different flight.

E3.2 Schema Version 16.1 and Earlier

For versions of the schema prior to 16.2, GA flights should be handled as follows:

An IATA airline (carrier) code of "GN" should be used. This code is currently reserved for use by GA flights for slot co-ordination.

The flight number should be populated with the scheduled local time of departure of the flight expressed as a four digit quantity HHMM using the 24 hour clock, for example a flight departing at 3:25 pm would have a flight number value of 1525. This is provided so that

flights operating on the same day from the same airfield using the same aircraft can be uniquely identified.

The origin date should be the scheduled date of departure for this flight. All GA flights are considered to comprise of a single flight leg.

No operational suffix should be provided. All other elements of the flight leg identifier are provided in the same manner as a commercial flight, however the aircraft registration (LegData/AircraftInfo/Registration) should be considered mandatory information for a GA flight, and will effectively form part of the flight identifier.

Appendix F – Frequently Asked Questions

The following questions have been asked by implementers using AIDX for the first time, and the answers given may prove helpful.

Q – *If a flight is cancelled 2-3 days out, should it be flagged as a cancellation or as a deletion?*

A – The flight should be cancelled by setting OperationalStatus to "DX". Deletions should only be used to remove a flight which was sent in error.

Q – *Are "through flights" specified using the Associated Flight Leg field?*

A – Yes, the AssociatedFlightLegSchedule is used to identify onward flight legs which have the same flight number. A FIDS system would use AssociatedFlightLegSchedule to find the ports of call and final destination or origin of a flight. See Appendix H of this Guide.

Q – *Codeset 2005 includes codes such as "EB" for estimated on block time, but the schema specifies two attributes to describe the time value supplied in OperationTime, how should the attributes be populated?*

A – The latest version of the schema uses two attributes. One to determine what the event is and the other to determine if the time is an actual time or is estimated, scheduled etc. So to specify the estimated on blocks time, the OperationQualifier attribute should be set to "ONB", and the TimeType attribute set to "EST". Earlier versions of the schema (prior to 12.1) did not have a TimeType attribute, so codes describing both aspects had to be used. Only when working with AIDX versions earlier than 12.1 should the "EB" and similar codes be used.

Q – *Are flight times specified in local time or Zulu (UTC) time?*

A – Times are always specified in UTC in AIDX, and a "Z" appended to the time value, e.g. "2012-04-13T13:32:50Z"

Q – *Most of the fields in the LegData element are optional. Surely some of these should be mandatory?*

A – The schema is designed to operate with many different types of system, which will have different requirements for mandatory data items. The bilateral interface agreement between the interfacing parties should identify any data items which are mandatory. Typically, some data items will be mandatory when data for a flight is sent for the first time, but may not be mandatory for subsequent update messages.

Q – *For codeshare flights, are separate messages sent for each codeshare?*

A – No, all data for a flight leg, including any codeshares, are sent in the same LegData element.

Q – *I want to set passenger counts separately for business class, first class and economy passengers, but the qualifier attribute associated with the PaxCount data element is populated from codeset 6353 which doesn't have codes for economy, business etc, How do I specify the cabin class associated with the given pax count?*

A – The CabinClass data element has a Class attribute populated from codeset 9873. This is where the cabin class should be specified. The DestinationType attribute on PaxCount can be used to specify transit, transfer, local passengers etc. – either on a per cabin class basis, or overall by setting the Class attribute to "7" (meaning "all classes"). It is recommended that

the Qualifier attribute on PaxCount is set to “70A” (meaning “total”). For example, to specify 25 business class passengers:

```
<CabinClass Class="2">
    <PaxCount Qualifier="70A">25</PaxCount>
</CabinClass>
```

And to specify 42 transfer passengers across all classes:

```
<CabinClass Class="7">
    <PaxCount Qualifier="70A" DestinationType="Transfer">42</PaxCount>
</CabinClass>
```

Q – *I am filtering messages for flights which have either ArrivalAirport or DepartureAirport set to my airport. Flights which were originally landing elsewhere but have been diverted to my airport are being blocked by the filter. Should the ArrivalAirport for a flight be amended when a flight is diverted?*

A – No. The ArrivalAirport element contains the originally scheduled destination for the flight leg. It forms part of the unique identifier for the flight leg, so cannot be changed, since it would then refer to a new flight leg rather than an existing one which has been diverted. For a diverted flight, the new arrival airport is added to the list of PlannedArrivalAptHistory data elements, and the ArrivalAirport remains set to the original intended destination. When filtering for flight legs relevant to a particular airport, the PlannedArrivalAptHistory must be considered.

Appendix G – Sample Messages and Instances

The examples in this appendix relate to simple cases of normally operating flights. A more comprehensive set of examples, including disruption scenarios, can be found at https://www.iata.org/publications/Pages/info-data-exchange.aspx.

AIDX Flight Data Request

Request made to United Airlines (UA) from Las Vegas Airport (LAS)

```
<?xml version="1.0" encoding="UTF-8"?>
<IATA_AIDX_FlightLegRQ TimeStamp="2012-07-01T19:56:09Z" Target="Test"
Version="12.1" TransactionIdentifier="575268690" TransactionStatusCode="Start"
RetransmissionIndicator="false" PrimaryLangID="US" AltLangID="US"
xmlns="http://www.iata.org/IATA/2007/00"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
     <Airline Code="UA" CodeContext="3"/>
</IATA_AIDX_FlightLegRQ>
```

AIDX Flight Data Response

Information response from United Airlines (UA) for flights departing from LAS to IAH, EWR to LAS, and CLE to LAS.

```
<?xml version="1.0" encoding="UTF-8"?>
<IATA_AIDX_FlightLegRS CodeContext="3" TimeStamp="2012-07-01T19:56:09Z"
Target="Test" Version="12.1" TransactionIdentifier="575268689"
TransactionStatusCode="Start" RetransmissionIndicator="false" PrimaryLangID="US"
AltLangID="US" xmlns="http://www.iata.org/IATA/2007/00"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <Success/>
     <FlightLeg>
          <LegIdentifier>
               <Airline CodeContext="3">UA</Airline>
               <FlightNumber>396</FlightNumber>
               <DepartureAirport CodeContext="3">LAS</DepartureAirport>
               <ArrivalAirport CodeContext="3">IAH</ArrivalAirport>
               <OriginDate>2012-05-19</OriginDate>
          </LegIdentifier>
          <LegData>
               <AirportResources Usage="Planned">
                    <Resource DepartureOrArrival="Departure">
                         <PassengerGate RepeatIndex="1" xsi:nil="true"/>
                         <AircraftTerminal>D1</AircraftTerminal>
                    </Resource>
               </AirportResources>
               <OperationTime OperationQualifier="OFB" CodeContext="9750"
RepeatIndex="1" TimeType="EST">2012-05-19T16:38:00Z</OperationTime>
               <OperationTime OperationQualifier="OFB" CodeContext="9750"
RepeatIndex="2" TimeType="SCT">2012-05-19T16:38:00Z</OperationTime>
          </LegData>
     </FlightLeg>
     <FlightLeg>
          <LegIdentifier>
               <Airline CodeContext="3">UA</Airline>
               <FlightNumber>868</FlightNumber>
```

```
               <DepartureAirport CodeContext="3">EWR</DepartureAirport>
               <ArrivalAirport CodeContext="3">LAS</ArrivalAirport>
               <OriginDate>2012-05-19</OriginDate>
          </LegIdentifier>
          <LegData>
               <AirportResources Usage="Planned">
                    <Resource DepartureOrArrival="Arrival">
                         <PassengerGate RepeatIndex="1">D22</PassengerGate>
                         <AircraftTerminal>D1</AircraftTerminal>
                         <BaggageClaimUnit RepeatIndex="1">10</BaggageClaimUnit>
                    </Resource>
               </AirportResources>
               <OperationTime OperationQualifier="ONB" CodeContext="9750"
RepeatIndex="1" TimeType="EST">2012-05-19T16:45:00Z</OperationTime>
               <OperationTime OperationQualifier="ONB" CodeContext="9750"
RepeatIndex="2" TimeType="SCT">2012-05-19T17:09:00Z</OperationTime>
          </LegData>
     </FlightLeg>
     <FlightLeg>
          <LegIdentifier>
               <Airline CodeContext="3">UA</Airline>
               <FlightNumber>581</FlightNumber>
               <DepartureAirport CodeContext="3">CLE</DepartureAirport>
               <ArrivalAirport CodeContext="3">LAS</ArrivalAirport>
               <OriginDate>2012-05-19</OriginDate>
          </LegIdentifier>
          <LegData>
               <AirportResources Usage="Planned">
                    <Resource DepartureOrArrival="Arrival">
                         <PassengerGate RepeatIndex="1">D19</PassengerGate>
                         <AircraftTerminal>D1</AircraftTerminal>
                         <BaggageClaimUnit RepeatIndex="1">8</BaggageClaimUnit>
                    </Resource>
               </AirportResources>
               <OperationTime OperationQualifier="ONB" CodeContext="9750"
RepeatIndex="1" TimeType="EST">2012-05-19T16:59:00Z</OperationTime>
               <OperationTime OperationQualifier="ONB" CodeContext="9750"
RepeatIndex="2" TimeType="SCT">2012-05-19T17:18:00Z</OperationTime>
          </LegData>
     </FlightLeg>
</IATA_AIDX_FlightLegRS>
```

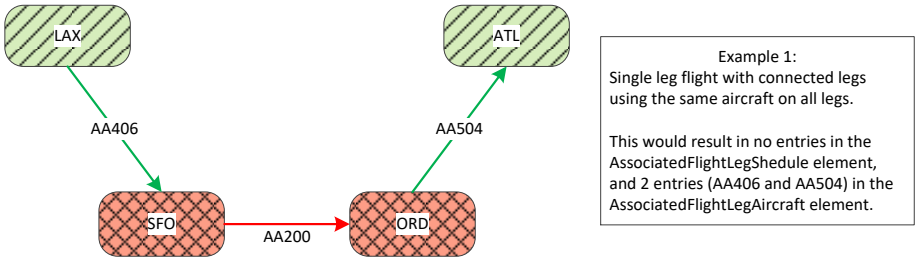
Flight Leg Notification Request

Notification of flight leg information from UA to DEN airport for an arriving flight ZK 5101 from BEF on the 02-07-2010 with GATE 61 and using bag claim area 9

```
<?xml version="1.0" encoding="UTF-8"?>
<IATA_AIDX_FlightLegNotifRQ CodeContext="3" TimeStamp="2012-07-01T19:56:09Z"
Target="Test" Version="12.1" TransactionIdentifier="575268688"
TransactionStatusCode="Start" RetransmissionIndicator="false" PrimaryLangID="US"
AltLangID="US" xmlns="http://www.iata.org/IATA/2007/00"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <Originator CompanyShortName="DIAAIHFBTST01.ua" TravelSector="A" Code="UA"
CodeContext="3"/>
     <DeliveringSystem CompanyShortName="DIAAIHSOADEV01.dia.dnvr"
TravelSector="C" Code="DEN" CodeContext="3"/>
     <FlightLeg>
```

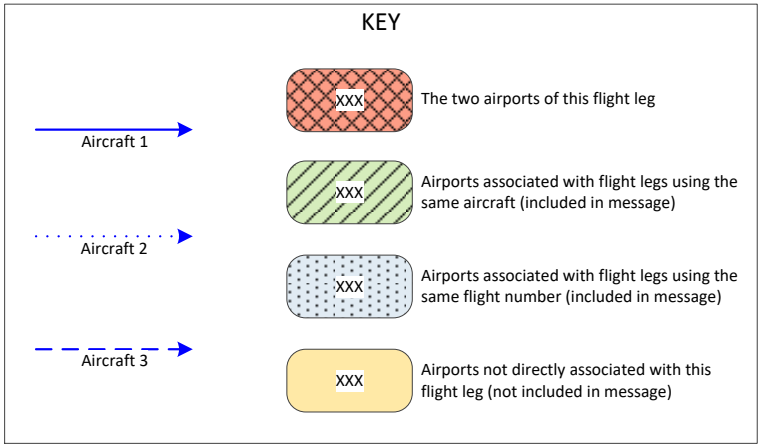
```
          <LegIdentifier>
               <Airline CodeContext="3">ZK</Airline>
               <FlightNumber>5101</FlightNumber>
               <DepartureAirport CodeContext="3">BFF</DepartureAirport>
               <ArrivalAirport CodeContext="3">DEN</ArrivalAirport>
               <OriginDate>2012-07-02</OriginDate>
          </LegIdentifier>
          <LegData>
               <RemarkTextCode Qualifier="GTE" CodeContext="9750"
RepeatIndex="1">SCT</RemarkTextCode>
               <AirportResources Usage="Planned">
                    <Resource DepartureOrArrival="Arrival">
                         <AirportZone>A</AirportZone>
                         <PassengerGate RepeatIndex="1">A61</PassengerGate>
                         <BaggageClaimUnit RepeatIndex="1">9</BaggageClaimUnit>
                    </Resource>
               </AirportResources>
               <OperationTime OperationQualifier="ONB" CodeContext="9750"
RepeatIndex="1" TimeType="EST">2012-07-02T23:28:00Z</OperationTime>
               <OperationTime OperationQualifier="ONB" CodeContext="9750"
RepeatIndex="2" TimeType="SCT">2012-07-02T23:28:00Z</OperationTime>
               <AircraftInfo/>
          </LegData>
     </FlightLeg>
</IATA_AIDX_FlightLegNotifRQ>
```

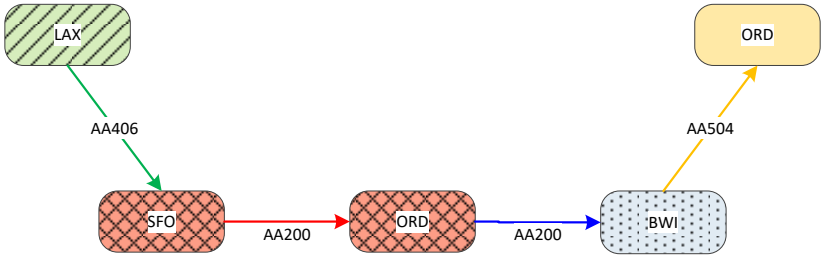
Appendix H - Associated Flights



Example 1:
Single leg flight with connected legs using the same aircraft on all legs.

This would result in no entries in the AssociatedFlightLegShedule element, and 2 entries (AA406 and AA504) in the AssociatedFlightLegAircraft element.

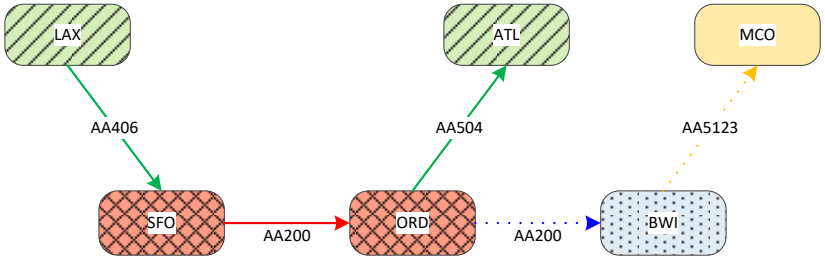




Example 2:
Multi-leg flight with no aircraft change.

In this example the same aircraft is used on all legs.

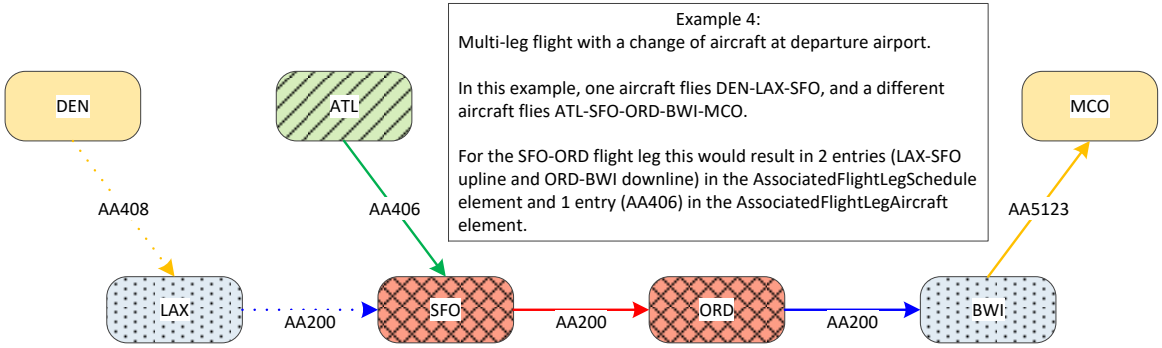
For the SFO-ORD flight leg this would result in 1 entry (ORD-BWI) in the AssociatedFlightLegSchedule element and 1 entry (AA406) in the AssociatedFlightLegAircraft element.

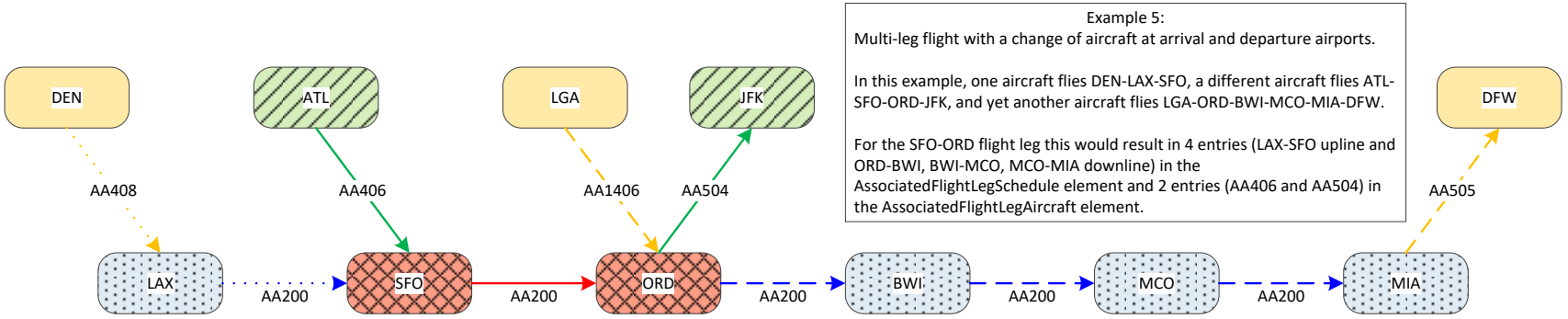


Example 3:
Multi-leg flight with a change of aircraft at Arrival airport.

In this example one aircraft flies LAX-SFO-ORD-ATL and a different aircraft flies ORD-BWI-MCO.

For the SFO-ORD flight leg this would result in 1 entry (ORD-BWI) in the AssociatedFlightLegSchedule element and 2 entries (AA406 and AA504) in the AssociatedFlightLegAircraft element.





Appendix J – Data Description Table

J1. Flight Data Elements

XML TAG	Path	Description	Codeset	Note	Example
AgentInfo	LegData/AircraftInfo	Identifier or company / name of Handling Agent for Flight	AGT	ID only included if other than the airline.	“OGD”
AgentInfo/@Qualifier	LegData/AircraftInfo	Handling Agent type for arrival or departure – repeating group	3035	Must be provided if AgentInfo populated	“BAG”
AgentInfo/ @DepartureOrArrival	LegData/AircraftInfo	A flag to indicate if the agent details are for the arrival or departure end of the flight leg	enumeration	Must be provided if AgentInfo/@Qualifier populated. Possible values “Arrival”, “Departure”	“Arrival”
AircraftParkingPosition	LegData/AirportResources/ Resource	Gate or hard stand where the aircraft is located.		When qualifier set as public this will be the same as PassengerGate	“C102”
AircraftParkingPosition/ @Qualifier	LegData/AirportResources/ Resource	A flag to state the type of parking stand.		Possible values “Gate”, “Public”, “Remote” or “Other”	“Remote”
AircraftSubType	LegData/AircraftInfo	Aircraft IATA Sub-Type	7800	Use SSIM code list – Appendix A (aircraft type)	“M83”
AircraftTerminal	LegData/AirportResources/ Resource	Terminal where the aircraft is located.	3223 and 3233	See SSIM for details about standard terminal information.	
AircraftType	LegData/AircraftInfo	Aircraft Type.		CodeContext attribute will determine if an IATA code (CodeContext=3) or ICAO code (CodeContext=13) is used. Use CodeContext=ZZZ if neither an IATA nor an ICAO code exists and a mutually defined code is being used. If IATA code specified, use SSIM code list – Appendix A (aircraft group)	“DC9”
Airline	GeneralAviationLegIdentifier	Carrier code for the operating airline	IATA, ICAO, or Other code	Typically will be set to “GN” to indicate a GA flight, but if the GA flight is being operated by a commercial airline should contain the carrier code for that airline.	“GN”
Airline	LegIdentifier	Carrier code for the operating airline.	IATA, ICAO, or Other code	The operating carrier which may differ from the aircraft owner	

XML TAG	Path	Description	Codeset	Note	Example
Airline	LegData/ AssociatedFlightLegAircraft	Carrier code for the operating airline of associated flights serviced by this aircraft (e.g. next departure flight to be serviced by this aircraft at the arrival airport).	IATA, ICAO, or Other code	See Appendix H for examples of usage of the associated flight leg data elements.	
Airline	LegData/CodeShareInfo	Carrier code for the airline marketing this flight as a code share.	IATA, ICAO, or Other code		
Airline	LegData/OwnerAirline	Aircraft owner code	IATA, ICAO, or Other code	The aircraft owner if different from operating carrier.	“UAL” or “UA”
AirlineType	LegData/PublicFlightDisplay	The Carrier code to be used on public displays, if different from LegIdentifier/Airline	IATA, ICAO, or Other code		
AirportResources/ @Usage	LegData	A flag to indicate if this resource assignment is the intended (i.e. the planned) one or is it the resource that was truly used (i.e. the actual usage)	enumeration	Must be provided for each resource assigned. Possible values “Planned”, “Actual”.	“Actual”
AirportZone	LegData/AirportResources/ Resource	The area in the airport which the flight uses			“Concourse C” or “Charter” or “GA”
ArrSecurityCheckInd	LegData	TRUE if additional security checks are required for the arrival part of the flight leg		Boolean value.	“true”
ArrivalAirport	GeneralAviationLegIdentifier	Code of scheduled arrival airport	IATA or ICAO	Optional for GA flights	
ArrivalAirport	LegIdentifier	Code of scheduled arrival airport	IATA or ICAO	This will not change, even in the case of a diversion or other re-routing. If the arrival station changes then this is reflected in the PlannedArrivalAptHistory field.	“DEN”
ArrivalAirport	LegData/ AssociatedFlightLegAircraft	Code of scheduled arrival airport of another flight associated with this aircraft, (e.g. the next departure flight to be serviced by this aircraft at the arrival airport).	IATA or ICAO	See Appendix H for examples of usage of the associated flight leg data elements.	
ArrivalAirport	LegData/ AssociatedFlightLegSchedule	Code of scheduled arrival airport of another flight leg associated with this flight (e.g. the arrival airport of the next flight leg in sequence for this flight number)	IATA or ICAO	See Appendix H for examples of usage of the associated flight leg data elements.	

XML TAG	Path	Description	Codeset	Note	Example
AssociatedFlightLegAircraft/@FlightSequence	LegData	Determines whether the specified flight leg occurs before (upline) or after (downline) the current one	enumeration	Possible values “upline”, “downline”. See Appendix H for examples of usage of the associated flight leg data elements.	“downline”
AssociatedFlightLegSchedule/@FlightSequence	LegData	Determines whether the specified flight leg occurs before (upline) or after (downline) the current one	enumeration	Possible values “upline”, “downline”. See Appendix H for examples of usage of the associated flight leg data elements.	“upline”
BagCount	LegData/AircraftInfo/Baggage	Number of bags for a destination type and cabin class in a specified location. Repeating group to cover the different ULDs.		To provide summary level information (with sufficient breakdown) to enable baggage handling to schedule appropriate resources	
BagCount/@Location	LegData/AircraftInfo/Baggage	The location on the aircraft of the baggage included in the BagCount, e.g. Bin name or ULD ID.			“UKE1234UA”
Baggage/ @DestinationType	LegData/AircraftInfo/	Used to specify the onward routing status.	enumeration	Possible values “Local”, “Transit”, “Transfer”	“Transfer”
Baggage/ @ServiceClass	LegData/AircraftInfo/	Specifies a class of service for the baggage loaded information.	9873	See Appendix D, section D5.	“2” (to indicate business class) or “7” (to indicate total for all classes)
BaggageClaimUnit	LegData/AirportResources/ Resource	The name or number of the assigned Baggage reclaim unit		Repeating group with type and areas to provide for more than one assignment and assignments of different types and different locations.	“T1A”
BaggageClaimUnit/ @AreaLocation	LegData/AirportResources/ Resource	Defines the location of the assigned Baggage reclaim device	9988	Must be provided for each bag claim unit (default is none) See Appendix D Section D10.	“INT”
BaggageClaimUnit/ @BaggageProcess	LegData/AirportResources/ Resource	The mutually agreed name of the baggage process that is planned on this baggage claim unit.			“Reclaim” or “VIP Reclaim”
BaggageClaimUnit/ @CloseTime	LegData/AirportResources/ Resource	The close date/time of the baggage claim unit in the baggage planning			“2021-12-05T15:40Z”
BaggageClaimUnit/ @OpenTime	LegData/AirportResources/ Resource	The open date/time of the baggage claim unit in the baggage planning			“2021-12-05T14:40Z”
BaggageClaimUnit/ @Qualifier	LegData/AirportResources/ Resource	Defines the type of the Baggage claim device assigned	BAG	Must be provided for each bag claim unit (default is standard bags)	“REG”

XML TAG	Path	Description	Codeset	Note	Example
BaggageClaimUnit/ @SegregationName	LegData/AirportResources/ Resource	The mutually agreed name of the segregation (sub-sortation) that is planned on this baggage claim unit.			"Y"
BaggageClaimUnit/ @ServiceClass	LegData/AirportResources/ Resource	Specifies a class of service for the baggage loaded information.	9873	See Appendix D, section D5.	"2" (to indicate business class)
BaggageMakeupBelt	LegData/AirportResources/ Resource	The baggage makeup belt(s) assigned for outgoing bags – a repeating group (up to 100 items)			"E4"
BaggageMakeupBelt/ @BaggageProcess	LegData/AirportResources/ Resource	The mutually agreed name of the baggage process that is planned on this baggage makeup belt.			"In-time Build" or "Last-minute Build"
BaggageMakeupBelt/ @CloseTime	LegData/AirportResources/ Resource	The close date/time of the baggage makeup belt in the baggage planning			"2021-12-05T15:40Z"
BaggageMakeupBelt/ @OpenTime	LegData/AirportResources/ Resource	The open date/time of the baggage makeup belt in the baggage planning			"2021-12-05T14:40Z"
BaggageMakeupBelt/ @SegregationName	LegData/AirportResources/ Resource	The mutually agreed name of the segregation (sub-sortation) that is planned on this baggage makeup belt.			"Y/ONW/SYD"
CabinClass/@Class	LegData	Cabin type to be used with the seat capacity and pax count values in the repeating group.	9873	Must be provided when SeatCapacity or PaxCount provided. See Appendix D, section D5.	"3" (to indicate economy class)
CabinCrewAirline	LegData	Airline providing the cabin crew, if this differs from the operating airline.	IATA, ICAO or other code		"BA"
CallSign	LegData/AircraftInfo	Defined in Flight Plan.			"AAL1234"
CheckInInfo/@Class	LegData/AirportResources/ Resource	The passenger class of the allocated range of positions for check-in activities. This is a repeating group to allow for the different types, locations and non-contiguous ranges.	9873	See Appendix D, section D5.	
CheckInInfo/@Location	LegData/AirportResources/ Resource	Where within the passenger terminal the allocated range of positions for check-in activities is located This is a repeating group to allow for the different types, locations and non-contiguous ranges.	9932	Only uses appropriate 3 letter codes from IATA code set 9932.	"AIR" (to indicate Airside [transfer] check-in)

XML TAG	Path	Description	Codeset	Note	Example
CheckInInfo/@Qualifier	LegData/AirportResources/ Resource	The type of the allocated range of positions for check-in activities. This is a repeating group to allow for the different types, locations and non-contiguous ranges.	CHK		"ODD" (to indicate out-of-gauge check-in)
ClearanceAgreement	LegData	Identifies the customs clearance arrangements for the flight.	9970	See Appendix D, section D9.	
CrewBusInd	LegData/AirportResources/ Resource	TRUE if airside bus used for the crew.			"true"
CrewInfo	LegData/AircraftInfo	Number of Crew Members (cockpit & cabin, jump seat).		Can repeat.	
CrewInfo/@Airline	LegData/AircraftInfo	The airline associated with the crew for which information is being provided.	IATA or ICAO		
CrewInfo/@Qualifier	LegData/AircraftInfo	Cabin class associated with the crew for which information is being provided.	9873	Must be provided when CrewInfo provided. See Appendix D, section D5.	
DeadLoad/ @DestinationType	LegData/AircraftInfo	Used to specify the onward routing status for the cargo, mail etc. defined in DeadLoad/Type.	enumeration	Possible values "Local", "Transit", "Transfer"	"Transfer"
DeIceLocation	LegData/AirportResources/Resource	Specifies the location at which the aircraft is to be de-iced.			"S34"
DeIceLocation/@Qualifier	LegData/AirportResources/Resource	Whether the de-ice location is a parking stand or a dedicated de-ice pan.	9932	Expected values "PAR" (Parking stand) or "PAN" (De-ice pan).	"PAR"
DepartureAirport	GeneralAviationLegIdentifier	Code of scheduled departure airport	IATA or ICAO		
DepartureAirport	LegIdentifier	Code of scheduled departure airport	IATA or ICAO	This will not change, even in the case of a diversion or other re-routing. If the departure station changes then the leg would be cancelled and a new leg created.	"STL"
DepartureAirport	LegData/ AssociatedFlightLegAircraft	Code of scheduled departure airport of another flight associated with this aircraft, (e.g. the previous flight to be serviced by this aircraft at the departure airport).	IATA or ICAO	See Appendix H for examples of usage of the associated flight leg data elements.	

XML TAG	Path	Description	Codeset	Note	Example
DepartureAirport	LegData/ AssociatedFlightLegSchedule	Code of scheduled departure airport of another flight leg associated with this flight (e.g. the departure airport of the previous flight leg in sequence for this flight number).	IATA or ICAO	See Appendix H for examples of usage of the associated flight leg data elements.	
DepSecurityCheckInd	LegData	TRUE if additional security checks are required for the departure part of the flight leg		Boolean value.	“true”
Duration	LegData/IrregularityDelay	Actual delay associated with the irregularity.		Measurements won't use days/years/months, so this field will always begin with PT	“PT2H15M” (=2 hours 15 min)
EstFlightDuration	LegData	Estimated Flight Duration Time, i.e. the time from off blocks to on blocks.		Measurements won't use days/years/months, so this field will always begin with PT	“PT11H45M” (=11 hours 45 min)
FlightCrewAirline	LegData	Airline providing the flight crew, if this differs from the operating airline.	IATA, ICAO or other code		“BA”
FirstPosition	LegData/AirportResources/ Resource/CheckInInfo	The start of an allocated range of desk positions for check-in activities. This is a repeating group to allow for the different types, locations and non-contiguous ranges.		If provided, last position, type and location must also be provided. If only a single position is allocated then the first and last position will be the same	
FleetNumber	LegData/AircraftInfo	Airline ship / fleet number – as assigned by the airline			
FlightNumber	GeneralAviationLegIdentifier	Flight number, if one is allocated		Typically GA flights will not have a flight number.	
FlightNumber	LegIdentifier	Actual flight number		Normally 4 digits without leading zeros, or 3 digits padded with leading zeros.	“009”
FlightNumber	LegData/ AssociatedFlightLegAircraft	Flight number of another flight associated with this aircraft, (e.g. the previous flight to be serviced by this aircraft at the departure airport).		Normally 4 digits without leading zeros, or 3 digits padded with leading zeros. See Appendix H for examples of usage of the associated flight leg data elements.	
FlightNumber	LegData/CodeShareInfo	Flight number of the airline marketing this flight as a code share.		Normally 4 digits without leading zeros, or 3 digits padded with leading zeros.	”1245”
FlightNumber	LegData/OwnerAirline	Aircraft owner flight number		The aircraft owner’s flight number if different from operating flight number. Normally 4 digits without leading zeros, or 3 digits padded with leading zeros.	“016”
FlightNumber	LegData/PublicFlightDisplay	The flight number to be used on public displays, if different from LegIdentifier/FlightNumber.		Normally 4 digits without leading zeros, or 3 digits padded with leading zeros.	

XML TAG	Path	Description	Codeset	Note	Example
GeneralAviationIdentifier	GeneralAviationLegIdentifier	The unique identifier for the GA flight		Typically will be the ATC callsign or the aircraft registration.	
GeneralAviationIdentifier/ @Category	GeneralAviationLegIdentifier	Identifies the type of identifier for a GA flight	enumeration	Possible values “Callsign”, “Registration”, “Other”.	“Callsign”
InflightService	LegData/CabinClass	List of the facilities offered during this flight leg. This is a repeating group of up to 10 to list all the services for each cabin	9932	Only using the numeric part of the IATA code set.	
InflightMealService	LegData/CabinClass	List of the refreshment(s) offered during this flight leg. Defined for each cabin and can be more than one for each cabin (Repeating group).	7161		
IrregularityDelay/ @DepartureOrArrival	LegData	Determines whether the delay is associated with the departure or arrival part of the flight leg.	enumeration	Possible values “Arrival”, “Departure”	
LastPosition	LegData/AirportResources/ Resource/CheckInInfo	The last of an allocated range of desk positions for check-in activities. This is a repeating group to allow for the different types, locations and non-contiguous ranges.		If only a single position is allocated then the first and last position will be the same.	
LegData/ @FlightClassification		Commercial name for express or other sub-carriers for the operating flight.	Free text	Commercial name.	“AmE”
LegData/ @InternationalStatus		Classifies flight as international or domestic. Used to determine whether the flight uses an international or domestic gate at the airport.	enumeration	Possible values “International”, “Domestic”	
OperatingAlliance	LegData	Airline alliance associated with the operating carrier.	9906		“701” (to indicate One World Alliance)
OperationalStatus	LegData	Defines status or details about the flight leg that should be used to inform the airline and airport operational staff. This in addition to the remarks data	1245 and 2005	Note that the operational status is needed as an airline may inform the staff of a cancellation before the passengers need to be informed (enabling time to prepare re-routing details etc.) See Appendix D section D2.1. Note that if a flight has been cancelled using the DX code, it can only be re-instated by explicitly doing so using the SQ code. Use of any other code will not implicitly reinstate the flight.	

XML TAG	Path	Description	Codeset	Note	Example
OperationalStatus/ @FlightLegScope	LegData	Determines whether the OperationalStatus element applies to the arrival flight, the departure flight or both.	enumeration	Possible values “Arrival”, “Departure”, “FlightLeg”	“FlightLeg”
OperationalSuffix	GeneralAviationLegIdentifier	Flight number suffix for a GA flight		Typically, GA flights will not have a flight number or a suffix.	
OperationalSuffix	LegIdentifier	Flight number suffix		Should be upper case only.	
OperationalSuffix	LegData/ AssociatedFlightLegAircraft	Flight number suffix of another flight associated with this aircraft, (e.g. the previous flight to be serviced by this aircraft at the departure airport).		Should be upper case only. See Appendix H for examples of usage of the associated flight leg data elements.	
OperationalSuffix	LegData/OwnerAirline	Aircraft owner flight number suffix		Should be upper case only.	
OperationDuration	LegData	Durations of various flight events as described by the OperationQualifier and TimeType attributes			
OperationDuration/ @OperationQualifie	LegData	The flight event to which the OperationDuration refers.	9750		
OperationDuration/ @TimeType	LegData	Used to specify the type of operation time.	2005	Typical types are estimated, actual etc. See Appendix D section D3.1 for allowed codes.	“ACT” (to indicate actual)
OperationTime	LegData	Times of various flight events as described by the OperationQualifier and TimeType attributes		See Appendix D section D3.	“2012-09-28T14:46Z”
OperationTime/ @OperationQualifier	LegData	The flight event to which the OperationTime refers.	2005 and 9750	Typical events are On Block, Off Block, Boarding , etc. See Appendix D section D3.	“ONB” (to indicate on blocks)
OperationTime/ @TimeType	LegData	Used to specify the type of operation time.	2005	Typical types are estimated, actual etc. See Appendix D section D3.	“SCT” (to indicate scheduled)

XML TAG	Path	Description	Codeset	Note	Example
OriginationDate	LegData/CodeShareInfo	Scheduled flight origin date of this code share flight		Date expressed in UTC. Time is not included. This date will not change once initialized in AIDX message. Refers to the UTC date of departure of the first sector of this Code share Flight (Code share flights may be single sector or multi-sector). Note that for a multi-sector operating flight the Code share OriginDate and the operating flight leg OriginDate may differ - specifically if the code share starts after the initial sector of the associated operating flight leg, and the initial operating sector goes over a date boundary.	2012-11-27
OriginDate	LegIdentifier	Scheduled flight origin date based on the flight not the flight leg.		Date expressed in UTC. Time is not included. This date MUST not change once initialized in AIDX message. For a flight SFO-DEN-LHR both flight legs SFO-DEN and DEN-LHR will have the OriginDate of the SFO departure date. See Appendix E.	2001-11-27
OriginDate	LegData/ AssociatedFlightLegAircraft	Scheduled flight origin date of another flight associated with this aircraft, (e.g. the previous flight to be serviced by this aircraft at the departure airport).		See note above relating to OriginDate. See Appendix H for examples of usage of the associated flight leg data elements.	
PassengerGate	LegData/AirportResources/ Resource	Public Gate which the passengers will use to board or disembark.		Repeating 3 times to allow for more than one for the same arrival / departure	“A5s”
PaxBusInd	LegData/AirportResources/ Resource	TRUE if an Airside Bus to be used for the passengers			“true”
PaxCount	LegData/CabinClass	The number of passengers of a specified passenger class – repeating group to cover the different classes and planned and actual			
PaxCount/ @DCS_Usage	LegData/CabinClass	Flag to indicate if the passenger count data is booked, accepted or boarded – repeating group to cover the different types	enumeration	Typically sourced from the airline’s DCS. Possible values “Booked”, “Accepted”, “Boarded”	“Booked”
PaxCount/ @DestinationType	LegData/CabinClass	Used to specify the onward routing status associated with the passenger count at the destination (i.e. the arrival flight) such as local, transit or transfer.	enumeration	Possible values “Local”, “Transit”, “Transfer”	“Transit”

XML TAG	Path	Description	Codeset	Note	Example
PaxCount/ @OriginationType	LegData/CabinClass	Used to specify the routing status associated with the passenger count at the origin (i.e. departure flight) such as local, transit or transfer.	enumeration	Possible values “Local”, “Transit”, “Transfer.	“Local”
PaxCount/@Qualifier	LegData/CabinClass	The type of the passenger count data being provided. – repeating group to cover the different types and planned and actual	Codeset 6353	In practice will usually be the code for all pax. See Appendix D section D6, however can also be used to specify numbers of passengers with reduced mobility, in which case Special Service Requirement (SSR) codes defined in the IATA Passenger Services Conference Resolutions Manual Recommended Practice 1708a paragraph 2.12.6 can be used.	“70A”
PaxCount/@Usage	LegData/CabinClass	Flag to indicate if the passenger count data is planned or actual – repeating group to cover the different types and planned and actual	enumeration	Must be provided if PaxCount provided. Possible values “Planned”, “Actual”.	“Planned”
PlannedArrivalAptHistory	LegData	Ordered list of stations	IATA or ICAO	Airports that the leg has previously and now been planned to arrive at. The last airport in the list is the currently planned destination. Used to determine the history of the flight, particularly following a return or other irregular operation. See Appendix C.	“ORD”
PlannedDepartureDateTime	GeneralAviationLegIdentifier	Planned time and date of departure for a GA flight.		Part of the flight identifier, should not change even if the planned departure time changes, unless the flight is cancelled and a new flight leg created. This is used to differentiate flights where the same aircraft departs from the same airfield multiple times during the day.	
PreClearedGateInd	LegData/AirportResources/ Resource	TRUE if the departure gate used for this flight leg is an immigration ‘pre-cleared’ gate (also known as a Schengen or trans-border gate).	-		
PublicStatus	LegData	Defines status or details about the flight leg that should be used to inform the public. This in addition to the remarks data	1245 and 2005	See Appendix D section D2.1. Use of this element is now deprecated.	
PublicStatus/ @FlightLegScope	LegData	Determines whether the PublicStatus element applies to the arrival flight, the departure flight or both.	enumeration	Possible values “Arrival”, “Departure”, “FlightLeg”	“FlightLeg”

XML TAG	Path	Description	Codeset	Note	Example
PublicTerminal	LegData/AirportResources/ Resource	Terminal where the passengers will be processed.	3223 and 3233	Repeating 3 times to allow for more than one for the same arrival / departure.	
Quantity	LegData/AircraftInfo/Fuel	The quantity of fuel			
Quantity/ @MeasurementUnit	LegData/AircraftInfo/Fuel	Unit of weight measurement	enumeration	Possible values: “Kilogram”, “Pound”, “Ton”, “Tonne”, “Litre”, ”USGallon”, “ImperialGallon”	“ImperialGallon”
ReasonCode	LegData/IrregularityDelay	This is part of a repeating group of up to 8 irregularity delays. Use IATA Irregularity/Delay Code for Departure. See IATA Airport Handling Manual (AHM) for delay codes and detailed format.	IRR	The code is an IATA standard code based on the Airport Handling Manual. Format is numeric or 2 character alphabetic code and one char sub-code. A new delay code schema is introduced as of the publication of AHM 732. The AHM 730 and 731 delay codes will continue to be published for 2 more editions (42nd and 43rd) to give the industry time to transition to the new delay codes. As of the 44th edition both AHM 730 and 731 will be phased-out. The new delay codes are available in a handy, easy to use app. that can be used free of charge at https://iata-ahm732.azurewebsites.net	
Registration	LegData/AircraftInfo	Aircraft Registration Number as assigned by aircraft manufacturer.		As per SSIM manual, no hyphen or other special character is permitted.	“N651UA”
RemarkTextCode	LegData	Remark text related to remark type using fixed data	2005 and 9750	The sender will provide the remark using the defined code sets. Senders will not define the text or words of the remark(s). See Appendix D section D2.2.	
RemarkTextCode/ @FlightLegScope	LegData	Determines whether the RemarkTextCode element applies to the arrival flight, the departure flight or both.	enumeration	Possible values “Arrival”, “Departure”, “FlightLeg”	“FlightLeg”
RemarkTextCode/ @Qualifier	LegData	Defines the area of the airport where the remark is to be displayed.	9932	Used by the receiver to determine where to display the remark data provided. When a remark is provided this field MUST be populated. Only uses the alphabetic part of the IATA code set. For public remarks use TER and for apron remarks use PAR. See Appendix D section D2.2.	
RemarkFreeText	LegData	Supplementary info for staff.	Free text		

XML TAG	Path	Description	Codeset	Note	Example
RemarkFreeText/ @FlightLegScope	LegData	Determines whether the RemarkFreeText element applies to the arrival flight, the departure flight or both.	enumeration	Possible values “Arrival”, “Departure”, “FlightLeg”	“FlightLeg”
RemoteOperationalGate	LegData/AirportResources/ Resource	An additional location used to transfer passengers to or from a remote parking position.		Only used if different from passenger gate. Repeating 3 times to allow for more than one for the same arrival / departure	
RepeatNumber	GeneralAviationLegIdentifier	Used to distinguish multiple departures, or attempted departures, of the same GA flight.			
RepeatNumber/ @CurrentInd	GeneralAviationLegIdentifier	TRUE if this repeat number is the current operating flight leg. FALSE if the flight leg has been replaced by one with a later (higher) RepeatNumber.		Boolean value	“true”
RepeatNumber/ @AirborneReturnNumber	GeneralAviationLegIdentifier	Maintains a count of the number of airborne returns in the overall number of repeated departure attempts.			
RepeatNumber	LegIdentifier	Used to distinguish multiple departures, or attempted departures, of the same flight.		See Appendix C.	
RepeatNumber/ @AirborneReturnNumber	Legidentifier	Maintains a count of the number of airborne returns in the overall number of repeated departure attempts.			
RepeatNumber/ @CurrentInd	LegIdentifier	TRUE if this repeat number is the current operating flight leg. FALSE if the flight leg has been replaced by one with a later (higher) RepeatNumber.		Boolean value.	“false”
RepeatNumber	LegData/ AssociatedFlightLegAircraft	Used to distinguish multiple departures, or attempted departures, of the same flight.			
RepeatNumber/ @AirborneReturnNumber	LegData/ AssociatedFlightLegAircraft	Maintains a count of the number of airborne returns in the overall number of repeated departure attempts.			
RepeatNumber/ @CurrentInd	LegData/ AssociatedFlightLegAircraft	TRUE if this repeat number, relating to another flight associated with this aircraft, (e.g. the previous flight to be serviced by this aircraft at the departure airport), is an operating flight leg. FALSE if the flight leg has been replaced by one with a later (higher) RepeatNumber.		Boolean value. See Appendix H for examples of usage of the associated flight leg data elements.	“false”

XML TAG	Path	Description	Codeset	Note	Example
Resource/ @ChargeType	LegData/AirportResources	Used to specify how the airline / aircraft operator pays for the associated resource	5903		
Resource/ @DepartureOrArrival	LegData/AirportResources	Determines whether the resource is associated with the departure or arrival part of the flight leg.	enumeration	Possible values “Arrival”, “Departure”	
Runway	LegData/AirportResources/ Resource	Runway used for take-off or landing.		Note that although this is a four character field, runway designators should be either two or three characters.	“19R”
SeatCapacity	LegData/CabinClass	Seating Capacity in each cabin.		Use a repeating group with cabin type	
ServiceType	LegData	IATA Flight Service Type of the operating flight.	IATA codeset	Refer to IATA SSIM appendix C - service type –	
SharedAlliance	LegData/CodeShareInfo	The alliance partner associated with each codeshare partner.	9906		
SpecialAction		To indicate the action needed for a flight leg record: delete, lock down, no display, empty.	enumeration	**Delete:** Delete the record. Delete is used only to delete flight legs which were created in error. Otherwise if a flight leg is not to operate the OperationalStatus should be set to cancelled **LockDown:** Lock down the record. Data lock down to be used if there is an operational incident when all information about the flight leg must be protected and access restricted. When the receiver is sent the “LockDown” flag then access to the flight leg data should be restricted to admin level access only. **DoNotDisplay:** Do not display this flight leg on public displays.	“Delete”
SpecialCargo	LegData	Details of any special cargo onboard	CAR	Live animals, Hazardous Material, Human remains, etc.	“2” (to indicate hazardous material)
SpecialEmphasis	LegData	To flag the flight for special handling. This is a repeating group of up to 3 codes (to allow multiple codes to be provided)	EMP	Used to flag that the flight requires particular attention / handling e.g. VIP on board. or first flight	“VP” (to indicate VIP on board)
TailNumber	LegData/AircraftInfo	Tail number as painted in the tail – used by some airlines as the aircraft identifier. Often the last 3 characters of the aircraft registration.			“1UA”

XML TAG	Path	Description	Codeset	Note	Example
TechnicalStopInd	LegData	TRUE if this stop is a technical stop.		Defined to be where an aircraft arrives or departs but does not enplane or deplane passengers, cargo or baggage. May conduct fuel, catering, crew change, customs or similar operations	“true”
TechnicalStopInd/@DepartureOrArrival	LegData	Determines whether it is the arrival or departure element of the flight leg to which the TechnicalStopInd data element applies.	enumeration	Possible values “Arrival”, “Departure”	“Departure”
TPA_Extension		Provided to allow extensions to be added by trading partner agreement.		To enable the provision of data not covered elsewhere in the existing schema required by a specific implementation. Please see section 5.2.2..	
Type	LegData/AircraftInfo/DeadLoad	Type of dead load.	7085		“D” (to indicate crew bags)
Type	LegData/AircraftInfo/Fuel	Type of fuel data.	enumeration	**FuelUplift**: the amount of fuel the fuelling company should load on the aircraft. **FuelOnboard**: The amount of fuel the aircraft has in its tanks while on the ramp/stand. **TripFuel**: The amount of fuel the flight planning system predicts the aircraft will burn in flight. **TakeoffFuel**: The amount of fuel the aircraft has in its tanks at takeoff. (FuelOnboard less the amount burned to get to start of runway).	“FuelOnboard”
Type/@extension	LegData/AircraftInfo/Fuel	A type of fuel not covered by the enumerated values allowed in Fuel/Type		Allows an extension to the enumerated values list, by mutual agreement with the users of the message.	
Weight	LegData/AircraftInfo/DeadLoad	Repeating weight elements to record dead load weight data for an aircraft. Load can be cargo, mail etc, as defined in DeadLoad/@Type.		Provide summary level information to enable ground handling agents to schedule appropriate resources.	
Weight/ @MeasurementUnit	LegData/AircraftInfo/DeadLoad	Unit of weight measurement	enumeration	Possible values: “Kilogram”, “Pound”, “Ton”, “Tonne”, “Litre”, ”USGallon”, “ImperialGallon” Only “Kilogram”, “Pound”, “Ton”, and “Tonne” valid for DeadLoad.	“Tonne”
Weight	LegData/AircraftInfo/Baggage	Weight of baggage loaded on the aircraft			

XML TAG	Path	Description	Codeset	Note	Example
Weight/ @MeasurementUnit	LegData/AircraftInfo/Baggage	Unit of weight measurement	enumeration	Possible values: “Kilogram”, “Pound”, “Ton”, “Tonne”, “Litre”, ”USGallon”, “ImperialGallon” Only “Kilogram”, “Pound”, “Ton”, and “Tonne” valid for baggage.	“Ton”

J2. Message Control Elements

Reference should be made to the documentation fields within the following schema:

- IATA_AIDX_FlightLegRQ
- IATA_AIDX_FlightLegNotifRQ
- IATA_AIDX_FlightLegRS

J3. Generic Attributes

XML TAG	Path	Description	Codeset	Note	Example
RepeatIndex		Identifies an order for a repeating item		See section 3.1.9	
CodeContext		Identifies the IATA codeset in which the code used to populate the associated element can be found. For some elements this attribute is populated from codeset 3055, as indicated in the schema.			

Appendix K – Schema Changes

This guide is based on version 22.1 of the AIDX schema. The following changes have been made to the schema since version 12.2. Note that schema changes are in general designed to be backwards compatible with earlier versions so that existing implementations can operate with the new schemas.

Changes to the various codesets which are used to populate certain data elements are also made from time to time, these changes are not recorded here. Codeset changes are generally restricted to the addition of new codes, so that backwards compatibility is maintained.

K1. Version 13.1

No changes were introduced at version 13.1.

K2. Version 13.2

Schema Name	Change	Background
IATA_SimpleTypes/ServiceType	Replace current restriction on ServiceType with an AlphaLength1, to allow any single alphabetic character to be used.	The values which are allowed in the ServiceType element are currently restricted to [J|S|U|F|V|M|Q|G|B|A|R|C|O|H|L|P|T|K|D|E|W|X|N|I]. According to the SSIM Manual, codes "Y" and "Z" are for special internal company purposes, but may later be assigned for specific purposes. Some airlines use code "Y" and wish to send details of flights with this service type in an AIDX message.
IATA_AIDX_CommonTypes	Update the description of /LegData/AircraftInfo/AgentInfo@Qualifier to indicate codeset 3035	The description of the field associated with the attribute AgentInfo Qualifier is currently suggesting that the attribute should be populated from the IATA IATACode List "AGT". However, the attribute should be populated from code set 3035.

K3. Version 14.1

Schema Name	Change	Background
IATA_AIDX_CommonTypes	Add a new attribute “FlightSequence “ to /LegData/AssociatedFlightSchedule, with enumerated values “upline” and “downline”.	There is a business requirement to differentiate between associated upline flight legs (i.e. those which occur before the current leg) and associated downline flight legs (i.e. those which occur after the current leg).
IATA_AIDX_CommonTypes	Add a new attribute “DepartureOrArrival” to /FlightLegType/LegData/ TechnicalStopInd. Make the TechnicalStopInd element a repeating element with max occurrences = 2 to allow for the case where both ends of the flight leg relate to a technical stop.	The TechnicalStopInd is set to indicate a technical stop – but there is no way to tell if this relates to the departure or arrival end of the flight leg.
IATA_AIDX_CommonTypes	For /LegData/CabinClass/PaxCount, change maxOccurs value from 3 to 20	Allows more than three different passenger counts to be defined, with different combinations of the attributes DestinationType and Qualifier. The qualifier can be used to specify numbers of unaccompanied minors, infants, children, total, etc.
IATA_AIDX_CommonTypes	Remove the extra sequence construct in /FlightLegType/LegData/AssociatedFlight LegSchedule.	Spurious sequence construct in the schema - there is no operational detriment, but it is untidy.
IATA_AIDX_CommonTypes	For /LegData/AircraftInfo/Baggage/ BagCount, change from: <xs:element name="BagCount" minOccurs="0" maxOccurs="50"> to <xs:element name="BagCount" nillable="true" minOccurs="0" maxOccurs="50">	The BagCount element was specified as not nillable. If a baggage location was specified in error, or if bags moved from one ULD to another and the original ULD removed from the flight, it was impossible to correct the situation by setting the BagCount for the erroneous location or unused ULD to nil.

K4. Version 14.2

Schema Name	Change	Background
IATA_AIDX_CommonTypes	For elements /LegData/AircraftInfo/Baggage/Weight, /LegData/AircraftInfo/DeadLoad/Weight, /LegData/AircraftInfo/Fuel/Quantity add nillable="true"	Various weights (for baggage, deadload and fuel) are specified as not nillable. This means that if a value is specified in error, it is impossible to correct the situation by setting the weight for the erroneous location to nil. Setting the value to zero is misleading, since it would imply that for example, no cargo has been loaded, rather than the weight of cargo being unknown.

K5. Version 15.1

Schema Name	Change	Background
IATA_AIDX_CommonTypes	Correction of spelling mistake in the enumerated list associated with attribute “FlightSequence” in data element /LegData/AssociatedFlightSchedule”.	Enumerated list included “downlilne” which should have been “downline”.

Note: Additional enumeration values were added to AIDX Common Type “MeasurementUnitType” in version 15.1, these are used in the fuel pre-transaction schemas which use AIDX types, but are not relevant to the AIDX schemas described in this Implementation Guide.

K6. Version 15.2

Schema Name	Change	Background
IATA_CommonTypes	In WarningsType and ErrorsType, "Type" attribute made optional, and "Owner" attribute added. These types are used in the IATA_AIDX_FlightLegRS schema. Reference to codeset 9321 removed from the "Type" attribute documentation text.	Generic change, not specific to AIDX. During a IATA review it was noted that the IATA standard error/warning XML schema contains an incorrect reference to IATA codelist in the "Type" attribute definition, and that the "Type" attribute is mandatory but should be optional. An optional attribute "owner" included to indicate the party whose error code is being used. If no indication is provided, IATA codes are in use.

K7. Version 16.1

Schema Name	Change	Background
IATA_AIDX_CommonTypes	DeIceLocation element added to /LegData/AirportResources/Resource. The element has an attribute "Qualifier"	Added to support A-CDM. Holds the location at which de-icing is performed An aircraft may be de-iced on the parking stand or on a dedicated de-icing pan, as indicated by the qualifier attribute.
IATA_AIDX_CommonTypes	Attribute "CodeContext" added to AircraftType element in /LegData/AircraftInfo, populated from codeset 3055. Constraint removed on AircraftType to allow a four character string to be used, and up to two instances of the data element allowed (previously was restricted to one).	Allows aircraft type codes other that IATA to be used. Typically IATA or ICAO codes would be specified, but code "ZZZ" can be used for aircraft which do not have a defined IATA or ICAO type. Multiple instances allowed so that both IATA and ICAO codes can be specified in the same message.
IATA_AIDX_CommonTypes	Addition of FlightCrewAirline and CabinCrewAirline to LegData.	Flight crew and cabin crew may be resourced from a different airline to that of the operating airline.

Schema Name	Change	Background
IATA_AIDX_CommonTypes	Attribute "AirborneReturnNumber" added to LegIdentifier/RepeatNumber	The RepeatNumber element is incremented for both ground returns and airborne returns. This attribute allows a separate count to be maintained for airborne returns, for those clients who are not interested in ground returns.
IATA_AIDX_CommonTypes	FlightSequence attribute added to Legdata/AssociatedFlightLegAircraft, with enumerated values "upline and "downline".	Distinguishes between flight legs occurring before or after the current leg. (Similar attribute was added to AssociatedFlightLegSchedule in schema version 14.1)

K8. Version 16.2

Schema Name	Change	Background
IATA_AIDX_CommonTypes	Complex element GeneralAviationLegIdentifier" added.	Allows AIDX to be used for non-commercial (general aviation) flights, which do not have a flight number. Can also be used by ATC systems which typically use callsign rather than flight number to identify flights, provided that the data elements are understood and can be used to identify the flight by the message recipient.
IATA_AIDX_CommonTypes	Complex element "OperationDuration" added to Legdata.	This allows time durations rather than a specific time of day to be defined for items such as estimated taxi times, mean turnaround times and de-icing times. The attributes define which duration is being specified.

Schema Name	Change	Background
IATA_AIDX_CommonTypes	DCS_Usage attribute added to Legdata/CabinClass/PaxCount, and existing "Usage" attribute made optional	Allows counts based on the number of passengers "Booked", "Accepted" and "Boarded" to be specified.
IATA_AIDX_CommonTypes	OriginationType attribute added to Legdata/CabinClass/PaxCount	The passenger count can now be broken down into local, transit and transfer passengers at the departure airport. Previously this breakdown was only possible for the arrival airport.
IATA_AIDX_CommonTypes	Constraint on Legdata/CabinClass/Paxcount@Qualifier changed to allow a 4 character alphanumeric string	Allows codes from IATA RP 1708a specifying types of passengers with reduced mobility to be used in addition to codes in IATA codeset 6353.
IATA_AIDX_CommonTypes	maxOccurs parameter for Legdata/CabinClass/PaxCount changed from 20 to 99	Allows for the larger number of possible passenger counts to be accommodated.
IATA_AIDX_CommonTypes	Attribute "FlightLegScopeType" added to elements Legdata/OperationalStatus, Legdata/PublicStatus, Legdata/RemarkTextCode, and Legdata/RemarkFreeText Maximum occurrences of PublicStatus and RemarkFreeText changed from 1 to 2.	The listed statuses and remarks can now be explicitly associated with the departure airport, arrival airport or both.

K9. Version 17.1

No changes relevant to AIDX were introduced at version 17.1

K10. Version 17.2

No changes relevant to AIDX were introduced at version 17.2

K11. Version 18.1

No changes relevant to AIDX were introduced at version 18.1

K12. Version 18.2

No changes relevant to AIDX were introduced at version 18.2

K13. Version 19.1

No changes relevant to AIDX were introduced at version 19.1

K14. Version 19.2

No changes relevant to AIDX were introduced at version 19.2

K15. Version 20.1

No changes relevant to AIDX were introduced at version 20.1

K16. Version 20.2

Schema Name	Change	Background
IATA_AIDX_CommonTypes	Added element OperationalSuffix to LegData/CodeShareInfo.	Allow code shares with a different operational suffix than the main flight.
IATA_AIDX_CommonTypes	Changed FlightLegScope attribute to FlightLegScope in LegData/RemarkFreeText	Fix typo.
IATA_AIDX_CommonTypes	Limit LegData/AirportResources/Resource/AircraftTerminal and PublicTerminal to alphanumeric characters, and don’t allow zero-length strings.	

Schema Name	Change	Background
IATA_AIDX_CommonTypes	Increase maximum occurrences of LegData/AirportResources/Resources/BaggageClaimUnit and BaggageMakeupBelt from 5 to 100.	5 occurrences turned out to be insufficient in practice in some cases.
IATA_AIDX_CommonTypes	Added the following attributes to LegData/AirportResources/Resource/BaggageClaimUnit and BaggageMakeupBelt: • BaggageProcess • SegregationName • OpenTime • CloseTime	Allows for extra information per ClaimUnit or MakeupBelt.
Code set directory	Added the following values to code set 9750 for use in OperationQualifier: • FBA (First Bag Arrived) • LBA (Last Bag Arrived)	Support First Bag Arrived and Last Bag Arrived, in addition to the existing First Bag Unloaded and Last Bag Unloaded.

K17. Version 21.1

Schema Name	Change	Background
IATA_AIDX_CommonTypes	Added AssociatedGeneralAviationFlightLegAircraft element to LegData	Support general aviation leg identifier.

K18. Version 21.2

ReasonCode reference to relevant to AIDX were introduced at version 21.2

K19. Version 21.3

No changes relevant to AIDX were introduced at version 21.3

K20. Version 21.4

No changes relevant to AIDX were introduced at version 21.4

K21. Version 22.1

Update of ReasonCode “note” with reference to the new delay code schema AHM732

- End -