ARINC 424 NDB

Draft 4 of Supplement 23
Proposal

Location Virtual Meeting
October 4-7, 2021

UPDATE CHAPTER 8:
424 XML DOCUMENTATION

V1.1

Kevin Wells

SUMMARY
Chapter 8 of the 424 specification provides documentation regarding the 424 XML design and format. (Note that the complete documentation for the XML format is contained in the XSD files themselves.) This proposal is for the updates to Chapter 8 to bring it current with the latest revision of the Supplement 23 XML format.

1.0 INTRODUCTION/ BACK GROUND

Chapter 8 of the ARINC 424 specification documents the 424 XML specification ("424 XML") and provides usages notes for 424 XML. (Note that the complete documentation for the XML format is contained in the XSD files themselves.) 424 XML is evolving, and Chapter 8 should be revised to keep pace.

2.0 DISCUSSION and or ACTION

424 XML is evolving, and Chapter 8 should be revised to keep pace. This proposal updates ARINC 424 Chapter 8 to reflect recent changes in the latest 424 XML supplement 23 draft.

3.0 Changes as depicted (Track Changes is Helpful)

See attachment

Chapter 8 - ARINC 424 XML Specification

ARINC 424 supplement 23 revision

Version 9.41 – October 2021

8.1 Table of Contents

8.2 Introduction

8.2.1 XML

In computing, Extensible Markup Language (XML) is a markup language that defines a set of rules for representing data.

The XML implementation of ARINC 424 data ("424 XML") was created to enable greater flexibility for the increasing complexity of aeronautical data, e.g., procedure and route design parameters.

User of the ARINC 424 XML implementation will require the accompanying electronic files:

- Schema definition files (XSD)
- Model documentation (HTML and/or PDF)

8.2.2 XSD

XSD (XML Schema Definition) specifies the formal description of the elements in an XML document. XSD expresses a set of rules to which an XML document must conform in order to be considered "valid" according to that schema.

The ARINC 424 XSD files specify the 424 XML format, and therefore form the actual ARINC 424 XML specification itself.

The model documentation is automatically generated from the 424 XSD files and provides detailed documentation of the XSD schema.

COMMENTARY

Supplement 22 added the XML implementation of the data in Chapters 4 and 5 of ARINC 424.

This initial implementation is a beta version and the NDB Subcommittee designated this initial implementation mature so that business units could start development.

Being a completely different data representation, 424 XML differs from the legacy 424 format in a number of ways. Sections 8.5 and 8.6 of this chapter provides notes as to these differences. Almost all of the current legacy 424 information can be represented in 424 XML. Section 8.7 notes data items that cannot be represented in 424 XML (without the use of the 424 supplemental data capability).

All new ARINC 424 legacy fixed width proposals will include information to allow them to be incorporated into the XML schema. However, the reverse will not be true going forward with XML specific proposals. It will be up those that wish to continue with the fixed-width format to develop a proposal to incorporate any new XML proposal back into the fixed-width format.

8.3 424 XML Schema Definition (XSD) Files

The ARINC 424 XSD schema definitions are organized into the following files:

• Types/DataTypes.xsd	Contains all of the type definitions for the XML records.
• Types/Enumerations.xsd	Contains all of the enumeration constants for the XML records.
• Records/_____.xsd	Contains the individual 424 XML record definitions (e.g. Ports.xsd, SIDSTARApproach.xsd, Legs.xsd, etc.)

The definitions in the first two files roughly correspond to Chapter 5 of the ARINC 424 specification. The definitions in the Records sub-directory roughly correspond to Chapter 4 of the ARINC 424 specification.

8.4 ARINC 424 XML Design

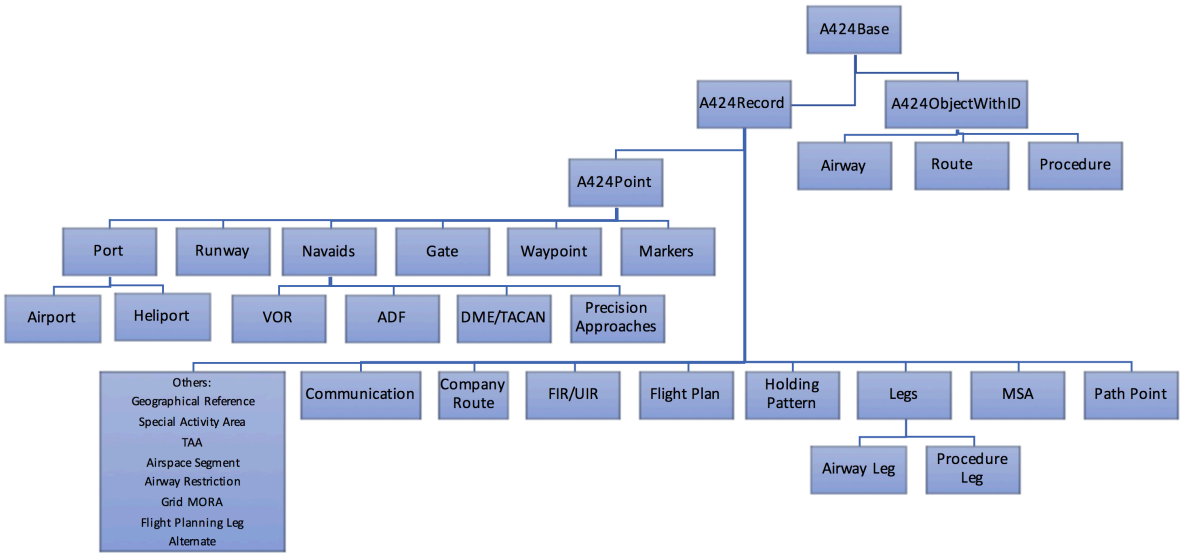
One of the fundamental aspects of the ARINC 424 XML design is that it makes use of two hierarchies:

1. A “class” hierarchy which describes the relationship between types of data elements. The 424 XML hierarchy is an example of a typical class hierarchy found throughout object-oriented software design. This class hierarchy is metadata.
2. A “containment” hierarchy which describes how actual navigation data is grouped into lists or containers.

8.4.1 Class hierarchy

The ARINC 424 XML class hierarchy is specified by the XSD schema definition files. It is not manifested in the XML data itself.

The class hierarchy for ARINC 424 XML shown in Figure 1.



Class	Notes
• A424Base	Contains only user or application specific supplemental data
• A424ObjectWithID	Parent of any class of data which has a unique identifier, but is not a 424 record by itself (for example a terminal procedure, or an airway).
• A424Record	Parent of any class which represents a 424 record (in the sense of the legacy 424 format). Includes record type, cycle date, customer code, area code, and notes.
• A424Point	Parent of any record class which contains a latitude and longitude

Figure 1

Note: Airspace and Cruising Tables are subclasses of A424Base directly because they do not have identifiers, and are not ARINC 424 records in and of themselves.

8.4.2 Containment hierarchy

All of the data normally contained within a single ARINC 424 database is contained in an ARINC 424 XML structure named ***AeroPublication***. An AeroPublication contains the database metadata, such as origin, validity, etc., and a series of lists representing the actual navigation data. The lists structure generally mirrors the record or "file" structure of the ARINC 424 legacy format.

The general list structure is as follows:

AeroPublication
Airports
Heliports
Airways
Enroute Waypoints
VHF Navaids
Enroute NDBs
Company routes
Airspaces
Alternates
Cruising tables
Preferred routes
Enroute Communications
Geographical References
Grid MORAs
Holding Patterns
Aero-Telecom Networks (ATN)

Airport and heliport data elements also contain a series of lists in addition to the data about the airport or heliport itself. The structure of these lists is as follows

Airport / Heliport
○ Runways
○ Helipads
○ Procedures
▪ SIDs
▪ STARs
▪ Approaches
○ Terminal Waypoints
○ Terminal NDBs
○ TAAs
○ Localizers
○ MSAs
○ Communications

8.4.3 Flight Paths

Terminal procedures are composed of a collection of procedure routes. The procedure routes correspond most closely to what are commonly known as “transitions.” For example, approaches generally consist of several approach transitions, a final approach route, and one or more missed approach routes. Each of these is an instance of an approach route.

Airways and procedure routes are composed of sequenced lists of legs.

The other lists in an AeroPublication are generally flat lists of records.

8.4.4 References within XML databases

A key element of the ARINC 424 XML design is the use of references between elements or records. The 424 XML Schema uses the XSD types ID and IDREF to implement these references. These references are effectively links or pointers which allow one element to refer to another. For example, the fix element on a leg record can link directly to the record for that fix (e.g. an airport, navaid, or waypoint).

This design has two benefits:

1. It avoids the need to perform a lookup based on identifier etc. for these referenced elements
2. It reduces the amount of duplicated data in the legacy 424 format

8.4.5 Cycle dates for procedures and routes

If any data at the procedure or route level change (e.g. transition altitude), the cycle date for all legs on that procedure or route must change with it because procedures and routes do not have independent cycle dates. This does not preclude the ability to change cycle dates on individual legs when data specific to that leg is changed.

8.4.6 Ordering of data elements

XML data does not allow for a specific ordering of data elements within a list. For example, the legs in a procedure transition or common route may appear in any order. If data is required to be ordered, it is up to the application to sort it using sequence numbers or other data.

8.4.7 Continuation records

Continuation record data (including flight planning data) is integrated into the primary records in 424 XML such that all data pertaining to a specific element such as an airport, airway, communication facility, etc. is contained within a single XML record.

8.4.8 Data elements which are duplicated at multiple levels

Certain data elements are duplicated at multiple levels in the 424 XML. For example, a data element may appear on a route as well as on individual legs. In this case the route element supports the common use case, and the leg elements would generally only be used in unusual cases as an “override” to the route element.

8.4.8.1 Record Type and Customer Code

The record type data element (*Standard* vs. *Tailored*) appear on Airway and Procedure records as well as on the individual airway and procedure legs. This is to allow tailored data to contain only part of an airway or procedure.

Airway records also have a customer code element at the airway level, as well as at the leg level for the same reason.

8.4.8.2 PBN Navigation Specifications

The RNP and RNAV PBN Navigation Specification data elements (*rnpPbnNavSpec* and *rnavPbnNavSpec*) are available on SID and STAR records and also on the transitions for these

procedures. The use of these data elements on transitions is intended for unusual or exceptional cases only. (For reference see ~~Mitre~~ MITRE Corporation Tracker ARINC-228)

8.4.9 XML version management

As of Supplement 23 the AeroPublication.xsd contains a version number. This is a String added to the metadata attribute group. This version number is starting at 1.0.0 for the release of Supplement 23. This version number follows SEMVER (Semantic Versioning). See Semantic Versioning 2.0.0 | Semantic Versioning (semver.org).

To summarize how SEMVER applies to the ARINC 424 data: A version number consists of MAJOR.MINOR.PATCH.

MAJOR: A change to the XML schema that would be incompatible with either the supplier or receiver of the data. Examples include: a new field was added; a previously mandatory field was removed; an enumeration was added to an existing enumeration.

MINOR: A change to the XML schema that is backward compatible for both the supplier and receiver. Examples include: removing an individual item from an enumeration; removing a previously optional element.

PATCH: A backward compatible bugfix. This change only fixes a bug without adding or removing features. The XML action team will need to agree on what is "broken" and what the state should be (including a consensus that there was a mistake creating the XML). Examples include: fixing the spelling of a variable; adding a variable back in that was deleted by mistake.

In addition, interim versions of the schemas will have -SNAPSHOT added as a suffix to the version number. For example, supplement 23 was issued with the xml version 1.0.0. While the team is working on supplement 23, there will be a X.X.X-SNAPSHOT version (Where the X's are whatever is appropriate for the type of change being worked on). This indicates that the schema is not a released version, and the content is subject to change.

8.4.10XML Development Process

Appendix A of this document describes the XML development process.

8.5 Data elements only in 424 XML

The following data elements only appear in 424 XML and not in the legacy 424 specification or data.

8.5.1 Landing Minimums

Landing minimums for the approach procedure are provided including the altitude (e.g. DA/MDA) and height (e.g. DH/MDH) based on the aircraft category.

~~Instrument approach landing minimums are legally specified weather criteria (generally ceiling and visibility requirements) for landing at an airport under Instrument Flight Rules. Landing minimums are specified for each instrument approach for specific categories and equipage of aircraft.~~

8.6 Notes on specific schema fields/elements

This section contains notes on the design and usage of specific 424 XML data elements, and differences between the legacy and XML 424.

8.6.1 Identifiers

The 424 XML Schema provides a common type – ***CoreIdentifier*** – for all identifiers. This type encompasses the identifier fields described in sections 5.6, 5.8, 5.9, 5.10, and 5.11, 5.13, 5.33, and 5.116 of this document. See those sections for additional specifications for identifiers. ***CoreIdentifier***s can be up to 16 characters long.

8.6.1.1 Procedure identifiers and names

In addition to the identifier inherited from the base ***A424ObjectWithId*** class, Procedures also have an optional alternate (longer) identifier element - ***longIdent***, which is also a ***CoreIdentifier***. This alternate identifier is used by some data suppliers and FMS manufacturers, and may be up to 8 characters long (vs. 6 characters for the basic procedure identifier).

Procedures also have an optional procedure name (***procedureName***) element, which contains the plain text, non-abbreviated full procedure name of the procedure identifier. It facilitates the correct identification and selection of the procedure in the FMS, and avoids the potential confusion caused when only the shorter identifiers are available. (See section 5.139)

Finally, there is a ***procedureDescription*** element in flight planning records. This element contains the textual representation of the procedure name, and may be up to 15 characters long.

8.6.2 Route Identifiers

Routes (procedure transitions, airway segments, etc.) are generally required to have identifiers. The exceptions to this are common segments of SIDs and STARs, and final and missed approach routes on approaches. These fields may have empty strings for identifiers in the 424 XML format. See section 5.11 of this document for more details.

8.6.3 DME and TACAN Navaid records

The 424 XML Schema provides for separate record types for DME and TACAN records. In the 424 legacy format, DME navaids share the record with VORs. If the DME is standalone, the VOR fields in the legacy record are not filled in.

The new XML design explicitly allows for both independent, stand-alone DME and TACAN navaids, as well as DME and TACAN navaids paired with VORs. The dmeRef field in the VOR record is used to indicate paired navaid combinations.

8.6.4 Time Zones

Start and end times are specified in the ***TimesOfOperations*** data type using the XSD type ***xs:time***. A timezone may be set for these fields, in which case the time is specified in the local timezone. If no timezone is set, UTC is assumed.

If a timezone is set indicating a local time, it should be the same for both start and end times. The ***adjustForDST*** field in the enclosing ***TimesOfOperation*** instance may then be set to indicate that the time follows daylight saving time changes in accordance with the local DST calendar.

8.6.5 Airway segments

In the legacy 424 format, airway segments are used to differentiate between separate airways with the same identifier. This concept is not required in 424 XML. The XML will represent different airway segments as multiple airways with the same identifier. This applies both to airway segments which changed at an international border, as well as disjoint airways (i.e geographically split) with the same identifier. (The latter are are represented by separate segments in legacy 424).

To support airway segments, the legacy 424 format uses boundary codes, "End of Continuous Segment" (EOCS) indicators, and duplicated legs. Boundary codes and EOCS indicators are not needed and do not exist in the 424 XML design where different legacy segments are represented by separate airways.

(See ~~Mitre~~ MITRE Corporation Tracker ARINC-185 for more details and historical discussion. See sections 5.12 (Sequence Number) and 5.17 Note 2 regarding End of Continuous Segment for additional details on legacy 424 airway segments.)

8.6.6 Runways and helipads without specific latitude / longitude

The ***isWithoutLocation*** boolean flag indicates that the runway or helipad has no (independent) location. In this case the location field should be populated with the airport or heliport location (i.e. airport or heliport reference point).

8.6.7 Altitude elements

There are a number of altitude related datatypes used in 424 XML. These are widely used and are interrelated.

Altitude – The ***Altitude*** type is the primary type used to represent an altitude. In addition to the numerical value of the altitude, this type also was flags to indicate whether the instance represents a Flight Level, and whether this altitude is undefined or unknown.

AltitudeValue – The ***AltitudeValue*** type represents an actual altitude numerical value. It is used within the other types in this section.

AltitudeLimitation - The ***AltitudeLimitation*** type is used on navaid records in the ***NavaidLimitation*** type. It defines an altitude or range of altitudes at which the limitation applies (Reference ARINC 424 section 5.209)

AltitudeLimit – The ***AltitudeLimit*** type incorporates the Altitude type but ads additional flags to indicate whether the altitude is: at the ground level or mean sea level, is set by NOTAM, is not specified, or is unlimited. (Reference ARINC 424 section 5.121). The ***AltitudeLimit*** data type is currently used on the airspace record types, and to define altitudes on holding pattern and leg records.

8.6.8 Runway and Helipad locations and elevations

In 424 XML, derived runway/helipad coordinates are flagged as derived and separated into the individual location fields; landing threshold, runway end and airport/heliport. Derived runway/helipad elevations are flagged and separated into the individual elevation fields: landing threshold, runway end, touch-down zone and airport/heliport.

The XML format differs from the updated legacy 424-23 specification in that the different coordinate and elevation types are broken out explicitly (rather than using flag fields and implicit rules). The differences are summarized as follows:

Runway end coordinates (***runwayEndLocation***): an additional data element populated only when provided in official government source and there is a displaced threshold.

Runway end elevation (***runwayEndElevation):*** an additional data element populated only when provided in official government source and there is a displaced threshold.

The XML runway touch-down zone elevation element (***touchDownZoneElevation***) differs from the legacy ARINC 424 definition. The legacy field allows for derived touch-down zone elevation values when there is not an elevation provided in official government source. The XML definition only allows officially government source values for ***touchDownZoneElevation***. The reason for this difference is that the full set of values used in the legacy ARINC 424 hierarchy are provided as separate elements in XML: ***landingThresholdElevation***, ***runwayEndElevation***, and the Port's elevation

(See ~~Mitre~~ MITRE Corporation tracker ARINC-237 for more background and details of these data elements)

8.7 Legacy 424 Items which cannot be modeled in XML

This section describes items which appear in the legacy 424 format which can no longer be modeled in the standard 424 XML. These items may be represented with supplemental data for backwards compatibility and testing purposes.

8.7.1 Current legacy format items (as of 424-23)

8.7.1.1 Holding Patterns: Duplicate Indicator

Position 1 of t~~T~~he holding pattern field D*uplicate Indicato*r has been replaced in the XML specification by a collection of boolean elements collectively named *HoldingUses*.

~~This collection does not provide for modeling of the second character position of the duplicate indicator (*Multiple*).~~

There are two *Undefined* values in position 1 of the duplicate indicator (values ***0*** and ***7***). 424 XML no longer distinguishes between these two values.

The *All Altitude* value of position 1 of the duplicate indicator field is modeled in the XML specification by setting both the High and Low Altitude booleans to true.

8.7.1.2 Cycle dates on continuation records:

Because continuation record data has been incorporated into the main 424 XML records, there is no longer a means to record separate cycle dates for that data.

8.7.1.3 Record numbers

Record numbers are necessary for testing purposes, but will not be necessary for production use of 424 XML

8.7.1.4 Airway boundary codes and End of Continuous Segment indicators

See above section on “Airway segments.”

8.7.2 Legacy format items prior to 424-23

This section describes items which appear in 424 supplements prior to -23 but have been removed from the 424-23 (or earlier) revision and 424 XML. These items are enumerated here to enable backwards compatibility testing and interoperability with 424 legacy data prior to ARINC 424-23, and may be represented with supplemental

8.7.2.1 Runway precision approach navaid references

See runway record section 4.1.10

8.7.2.2 Altitude codes

Various altitude codes have been deleted from past versions of 424. See 424 section 5.29.

8.7.2.3 Guard Transmit

See 424 section 5.182.

8.7.2.4 Service Indicators

Various service indicator values have been deleted from past versions of 424. See 424 section 5.106.

8.7.2.5 RNAV usage flag

In the legacy 424 format prior to 424-23, the Waypoint Type field (424 Section 5.42) contains an “RNAV Waypoint” indicator in column 27, and an RNAV usage indicator in the Waypoint Usage field in column 30. These have been combined into a single RNAV boolean in the waypoint type element of 424 XML. It is necessary to capture the Waypoint Usage field data in supplemental data for testing purposes as these two indicators are not always used consistently together.

8.8 Glossary

Class refers to the meta-data description of a type of data, e.g. an airport or waypoint description

Object or ***data element*** refers to an actual piece of data, e.g. a specific airport, latitude, etc.