

# Update GUFU to align with new FF-ICE guidance

<b>CR Number (ID): 163</b>		<b>Submission Date: 5/6/2022</b>	
<b>GENERAL INFORMATION (AUTHOR TO COMPLETE)</b>			
Author	FAA FIXM Development Team		
Brief Description (Subject)	FIXM needs an update to align with the new composition of the GUFU, which includes a namespace and timestamp in addition to the UUID.		
Priority*	<input type="checkbox"/> Immediate	<input checked="" type="checkbox"/> High	<input type="checkbox"/> Medium <input type="checkbox"/> Low
Scale*	<input type="checkbox"/> Major	<input type="checkbox"/> Medium	<input checked="" type="checkbox"/> Minor
Expected Impact to Implementers*	<input checked="" type="checkbox"/> High	<input type="checkbox"/> Medium	<input type="checkbox"/> Low <input type="checkbox"/> None
Target FIXM Component(s)	<input checked="" type="checkbox"/> FIXM Logical Model <input type="checkbox"/> FIXM Application <input checked="" type="checkbox"/> FIXM XML Schemas <input type="checkbox"/> Other Specify...		
Target FIXM Release	<input checked="" type="checkbox"/> FIXM Core Version 4.3.0 <input type="checkbox"/> Application Version		
Related FIXM CRs			
Motivation / Change Reason	<input checked="" type="checkbox"/> Requirement <input type="checkbox"/> Functionality/Operability <input type="checkbox"/> Maintenance <input type="checkbox"/> Defect <input type="checkbox"/> Other Specify ...		

\* Optional fields; FIXM Secretariat may update during review

## MOTIVATION / CHANGE REASON DESCRIPTION (AUTHOR TO COMPLETE)

Support new requirement from the FF-ICE IG v0.99 section 3.7.5:

### 3.7.5 GUFU Composition

3.7.5.1 A GUFU shall include a version 4 "Universally Unique Identifier" (UUID), as standardised by IETF RFC 4122 of the Open Software Foundation (OSF) and documented by the International Standards Office (ISO/IEC 9834-8: 2014) and the International Telecommunications Union.

*3.7.5.2 The GUFIs' UUID shall be supplemented with the following additional information:*

- 1. A namespace identifier to record the originator of the GUFIs. Namespaces are used in a variety of settings to both organize content as well as help ensure that content is uniquely identifiable. In FF-ICE, the originator of each GUFIs can construct their own namespace as described in Appendix G-1 to ensure that no GUFIs generated with a given namespace can be a duplicate of a GUFIs generated with any other namespace.*
- 2. A GUFIs creation timestamp to further decrease the chances of a GUFIs collision. Timestamps will also ensure the GUFIs is perpetually unique over any period of interest.*

#### PROPOSED CHANGE (AUTHOR TO COMPLETE)

In the Types package, create a new class named GloballyUniqueFlightIdentifier that inherits from the UniversallyUniqueIdentifier class and has the following definition: "An immutable identifier associated with a flight that allows all eligible members of the ATM community to unambiguously refer to information pertaining to the flight."

Create a new enumeration named NamespaceDomain with the following definition: "Identifies the particular type of namespace used by the originator of a GUFIs." Include the following enumerated values:

- Name: AIRCRAFT\_ADDRESS
- Definition: A namespace can be derived from an aircraft's ICAO address. The 24-bit ICAO address information should be represented via six hexadecimal digits. For example: ABC123.
- Name: AIRCRAFT\_REGISTRATION
- Definition: An individual aircraft is to be identified by its aircraft registration. For example, the aircraft with registration N1234 might use a namespace of: N1234.
- Name: ATM\_UNIT
- Definition: An ATM unit is to be identified by its corresponding four letter Location Indicator (LOCID as determined in ICAO Doc. 7910). A LOCID is available for each FIR or ACC, as well as for airport locations that operators are tied into. For example, the ATM unit "Washington ATC Center", LOCID KZDC, assuming just one system generates GUFIs, might use the following namespace "KZDC".

It does not seem likely that any ATM unit would have more than one source of GUFIs, but if that were the case a numeric station identifier can be added as well. Example: KZDC-2.
- Name: MAC\_ADDRESS

- Definition: A MAC address is a unique identifier assigned to a network interface controller (NIC) of the device by which the flight plan is submitted, formed according to the Institute of Electrical and Electronics Engineers (IEEE) Guidelines for 48-Bit Global Identifier (EUI-48).

If an originator generates GUFIs from several stations, it might be appropriate to use the MAC addresses of the devices submitting the flight plan information to properly distinguish the stations. The 48-bit MAC address should be represented via twelve hexadecimal digits. For example: ABCDEF123456.

Note: Use of the MAC address of the originating device will leak information about the hardware of the originating device (at least the manufacturer of the NIC and possibly the manufacturer and model of the device itself) outside the originator's organisation, and may thus increase security risk.

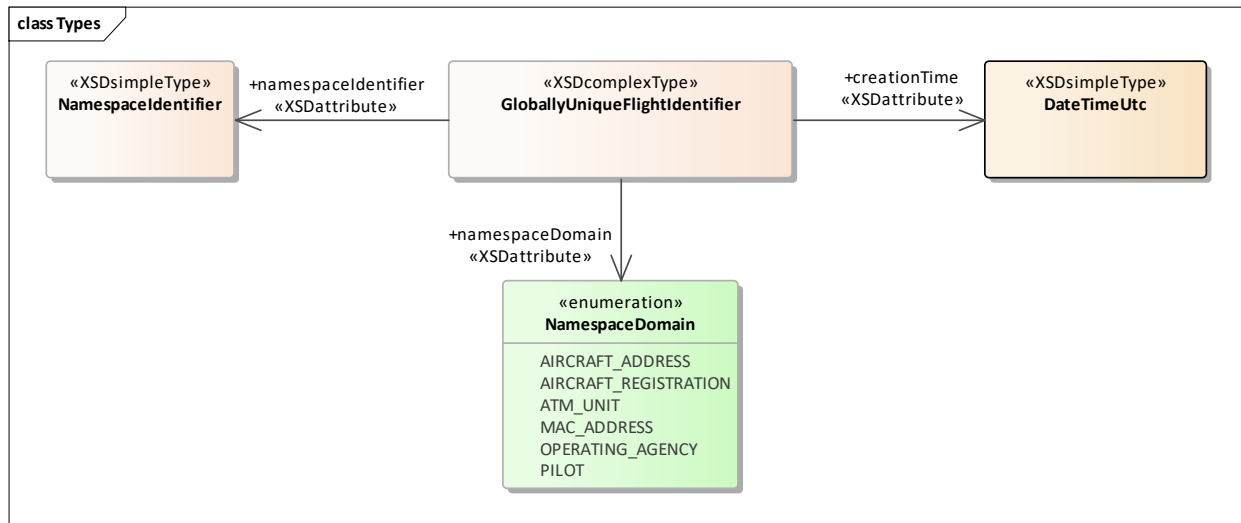
- Name: OPERATING\_AGENCY
- Definition: An operating agency is to be identified by its three letter Operating Agency Designator (ICAO Doc. 8585). A flight planning service provider's namespace content is composed of their Doc. 8585 designator. Example: ABC.  
If the provider files from different locations, a four letter Location Indicator (LOCID as determined in ICAO Doc. 7910) can be used to indicate from which station the flight plan is filed. For example, operator ABC filing from Austin, Texas, U.S.A. (LOCID KAUS) might use: ABC-KAUS. If an operating agency generates GUFIs from more than one station, a numeric station identifier can be added as well. Example: ABC-12 or ABC-KAUS-3.
- Name: PILOT
- Definition: A pilot is identified by a pilot license number. For example: ABCD1234.

Create an association named "namespaceDomain" between the GloballyUniqueFlightIdentifier class and the NamespaceDomain enumeration with the following definition: "Identifies the particular type of namespace used by the originator of a GUFIs". Apply the <<XSDattribute>> stereotype to this association and add a tag with the name "use" and the value "required".

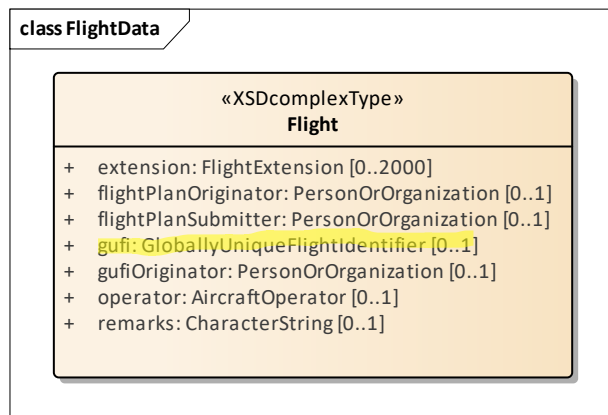
Create an <<XSDSimpleType>> class named NamespaceIdentifier that inherits from CharacterString and has the following definition: "The namespace chosen by the GUFIs originator. Used to reduce the likelihood of GUFIs collisions and provide traceability as to which entity generated a particular GUFIs". Set the maxLength tag for this class to 12 to accommodate the longest allowable namespace value.

Create an association named "namespaceIdentifier" between the GloballyUniqueFlightIdentifier class and the newly created NamespaceIdentifier class with the following definition: "The namespace chosen by the GUFIs originator. Used to reduce the likelihood of GUFIs collisions and provide traceability as to which entity generated a particular GUFIs". Apply the <<XSDattribute>> stereotype to this attribute and add a tag with the name "use" and the value "required".

Create an association named “creationTime” between the GloballyUniqueFlightIdentifier class and the DateTimeUtc class with the following definition: “The time at which the GUFU was created. Used to reduce the likelihood of GUFU collisions and ensure GUFUs remain perpetually unique.”. Apply the <<XSDattribute>> stereotype to this association and add a tag with the name “use” and the value “required”.



In the FlightData package, change the type of the “gufi” attribute of the Flight class to use the newly created GloballyUniqueFlightIdentifier class.



This new GUFU format will result in XML structured like the following example:

```

<gufi codeSpace="urn:uuid" creationTime="2021-03-24T09:30:10Z" namespaceDomain="ATM_UNIT"
namespaceIdentifier="KZDC">2b1bf9a9-c516-46be-bdc9-4926d9b84c8e</gufi>

```

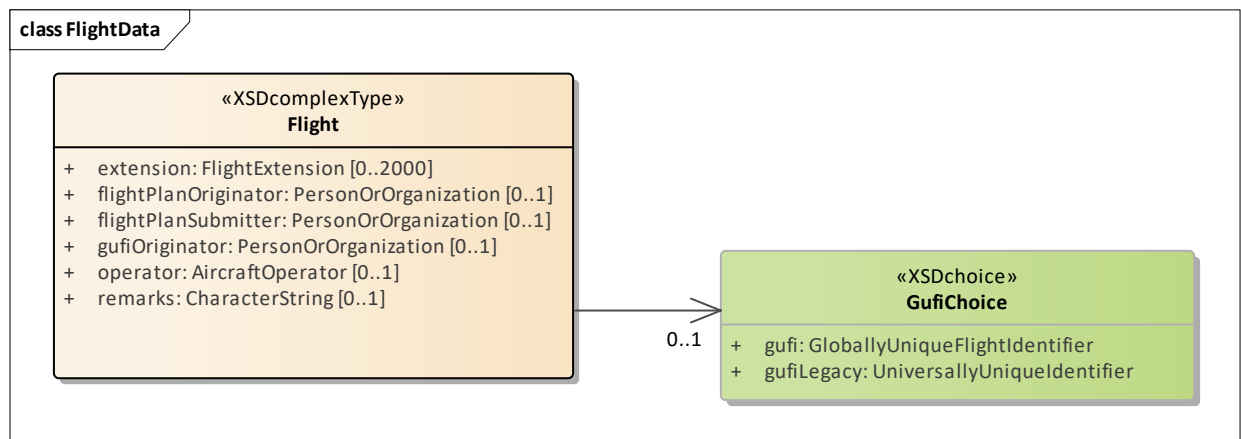
### Update based on CR TIMs and CCB review.

It was noted that the inclusion of these new required attributes in Core 4.3.0 would create a backwards compatibility issue with this minor release. GUFIs created in Core 4.2.0 would not include these fields. As such, there would be no way basis for what to put in them if information about a flight with a 4.2.0 GUFIs were to be published in 4.3.0 format. To address this, one possible way forward is to create a choice structure within FIXM allowing for representation of legacy 4.2.0 GUFIs in 4.3.0.

To accomplish this, create a new class with the <<XSDchoice>> stereotype named GufiChoice with the following definition: “Helper class to create an embedded choice structure.”.

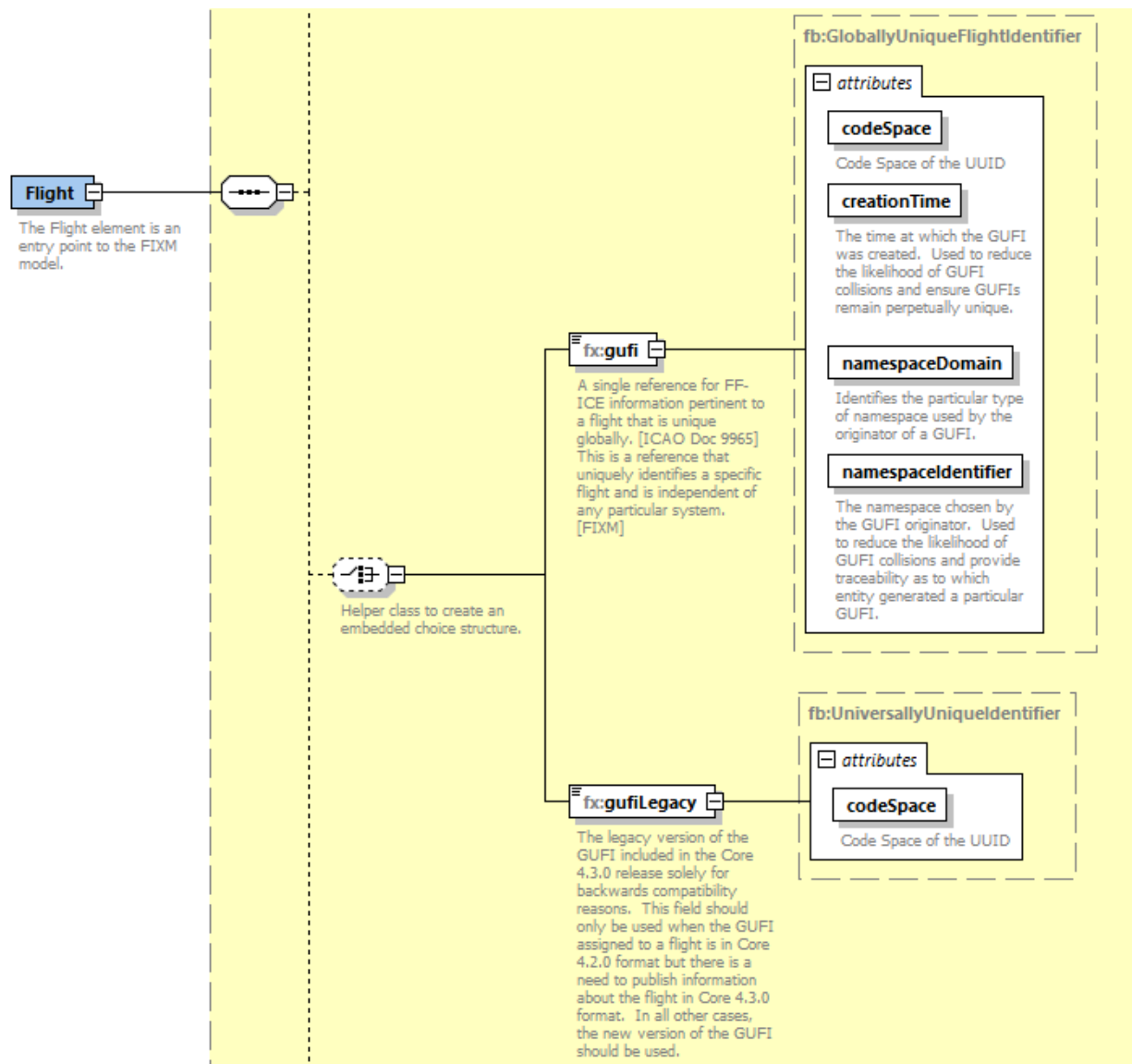
Move the existing “gufi” attribute (changed as noted above to be of type GloballyUniqueFlightIdentifier) from the Flight class to this new GufiChoice class and change its multiplicity to make it required. Then create a new required attribute under GufiChoice named “gufiLegacy” of type UniversallyUniquelIdentifier with the following definition: “The legacy version of the GUFIs included in the Core 4.3.0 release solely for backwards compatibility reasons. This field should only be used when the GUFIs assigned to a flight is in Core 4.2.0 format but there is a need to publish information about the flight in Core 4.3.0 format. In all other cases, the new version of the GUFIs should be used.”.

Finally, create an unnamed association between the Flight class and this new GufiChoice class with a multiplicity of 0..1.



This approach offers a clear distinction between the new FF-ICE compliant GUFIs and the legacy 4.2.0 GUFIs which should help FIXM users understand the nature of this change and the limited circumstances under which the legacy version should be used.

In the physical model, this approach will create an embedded choice structure in the flight element’s sequence as shown below.



Because of this modeling, when it comes time to remove this choice structure and only retain the new FF-ICE compliant GUFU format, the change should have no impact to the resulting XML other than the “gufuLegacy” element no longer being available.

CCB SECRETARIAT				
External Standard Consistency Checked	<input type="checkbox"/> AIDX <input type="checkbox"/> AIRM			
CR Status	<input checked="" type="checkbox"/> Proposed <input type="checkbox"/> Implemented <input type="checkbox"/> Withdrawn <input type="checkbox"/> Rejected			
Decision Date	Click or tap to enter a date			



Implemented In	X.X.X
Comments	Click or tap here to add any additional information or comments.

#### IMPLEMENTATION NOTES (CCB SECRETARIAT TO COMPLETE)

CCB Secretariat notes on any deviations that were required during implementation.