xSNOWTAM – TEC

Mapping Document From AMDB Files Into AIXM 5.0 Database

TECHNICAL ANALYSIS

Version 1.0

DOCUMENT DISTRIBUTION LIST

NAME	COMPANY/SERVICE	ACRONYM	ROLE
Eduard POROSNICU	EUROCONTROL	EPR	
Tudor IVANOV	Pulsar/Consultancy	TIV	Project Manager
Ludovic LAMBRECHTS	Pulsar/Consultancy	LLS	Technical Leader
Jeremy GRUMBACH	Pulsar/Consultancy	JGR	Analyst

DOCUMENT CHANGE RECORD

Date	Author(s)	Review	Description	Sections
24 Mar 2009	JGR		V0.1: first version	All
21 Apr 2009	JGR		V0.9: complete review (including information coming from emails with SWI)	All
27 Apr 2009	JGR		V1.0: updated after SWI review	3.4.1, 3.5, 3.6

CONTENTS

1 Purpose

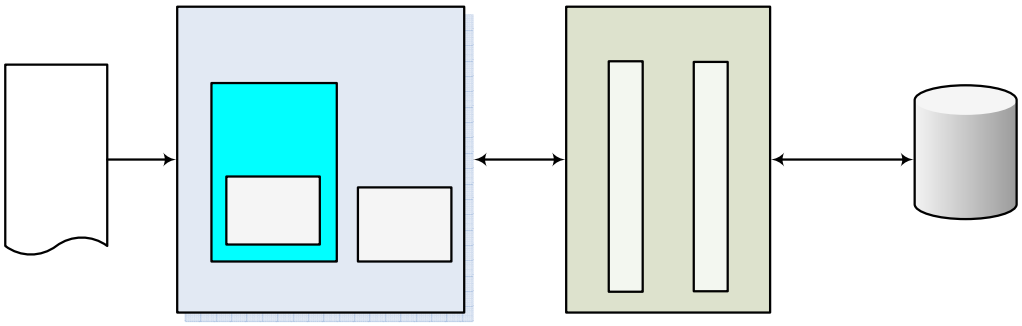
The purpose of this document is to describe technical details of the mapping from AMDB features to the digital SNOWTAM Database. The scope of the mapping is limited to the features necessary for Digital SNOWTAM Trial (xSNOWTAM). Therefore, it is not an exhaustive mapping of all possible data from AMDB but a subset containing the following features:

- *RunwayElement;*
- *TaxiwayElement;*
- *ApronElement;*
- *GuidanceLine.*

(This includes associated objects such as ElevatedSurface, SurfaceCharacteristics, etc.)

2 Architecture

The mapping will be implemented in Java. It will be based on a generic architecture also used for other mappings (SNOWTAM to AIXM5, AIXM45 to AIXM5, etc…):



2.1 Architecture Modules

- The ***XSD AIXM 5.0 MODEL*** is a set of Java Classes representing AIXM 5.0. This set of classes is directly generated from the AIXM 5.0 xsd files;
- The ***AMDB INPUT MODULE*** will read an AMDB XML file and transform it into the ***XSD AIXM 5.0 MODEL***;
- The ***AIXM 5.0 MODEL*** a set of Java Classes representing AIXM 5.0. This set of classes is directly generated from the xSNOWTAM Database (and consequently is simpler than the ***XSD AIXM 5.0 MODEL*** which has deeper levels);
- The ***Converter*** will transform XSD AIXM 5.0 features to AIXM 5.0 features using, if necessary, the SNOWTAM Kernel to retrieve associated data in database (see chapter **Error! Reference source not found.**). The final step is to save the feature (and/or timeslice) in the SNOWTAM database also via the SNOWTAM Kernel.

AMDB To Digital SNOWTA
CONVERTER

2.2 AMDB XML Files

AMDB Files use the AIXM Basic Message format:

```
<AIXMBasicMessage>
    <message:hasMember>
        <aixm:feature-name>
            […]
        </ aixm:feature-name>
    </message:hasMember>
    <message:hasMember>
        <aixm:feature-name>
            […]
        </ aixm:feature-name>
```

AMDB
XML FILE

AMDB
INPUT MODULE
(load method)

XSD
AIXM 5.0
MODEL

```
        </message:hasMember>
</AIXMBasicMessage>
```

It is not possible to detect if an AMDB feature already exists in the database (AMDB features do not have a natural key to identify them). Therefore, each AMDB feature will be considered as a new feature.

Special Rule:

- ⇨ To be able to manage updates/deletes, a solution would be to completely remove the previous AMDB content of the database, and to reload it with features coming from the new AMDB file. This assumes of course that each AMDB file contains the whole AMDB data.

Comment: Agreed. I'll make this a rule.

3 Fields Mapping

3.1 Technical Fields

Followings fields are found in most of the features, timeslices or objects and have default values. They are described only once, in the table below:

Feature	
Column	Default value
SEQID	Technical key (Sequence)
CREATED_BY	'MAPPING AMDB'
CREATED_ON	sysdate
LAST_MODIFIED_BY	'MAPPING AMDB'
LAST_MODIFIED_ON	sysdate

3.2 Temporality Fields

Some fields concerning temporality are common to all timeslices, they are described only once:

For each AMDB feature, we will:

- Create a new AIXM 5.0 feature;
- Create a PERMDELTA and a BASELINE associated to this new feature, with the following field values:

Timeslice	
Column	Default value
INTERPRETATION	BASELINE or PERMDELTA
SEQUENCE_NUMBER	1
CORRECTION_NUMBER	0
START_DATE	Start date defined in the <*gml:validTime*>
START_OF_LIFE	Start date defined in the <*aixm:featureLifetime*>

3.3 GML Identifier

We will use the gml:identifier of the AMDB file. It will be saved in the "UUID" field of the feature table.

Ex: <*gml:identifier codeSpace="http://www.eurocontrol.int/amdb*">
2439fad1-8061-4c71-96f6-22dcd92d0982
</*gml:identifier*>

3.4 Parent Search

3.4.1 General case

All AMDB features are linked to a parent feature which should already exist in the xSNOWTAM database. We will search for it, based on its Natural Key:

Feature	*Parent Feature*	*Parent Natural Key*
RunwayElement	Runway	Airport Heliport Designator / Runway Designator
TaxiwayElement	Taxiway	Airport Heliport Designator / Taxiway Designator
ApronElement	Apron	Airport Heliport Designator / Apron Name
GuidanceLine	GateStand	To be determined
	RunwayCentrelinePoint	To be determined
	Apron	Airport Heliport Designator / Apron Name
	Taxiway	Airport Heliport Designator / Taxiway Designator

Example:

In case of the Guidance Line, it can be linked to a taxiway. The short-hand link looks like:

```
<aixm:connectedTaxiway
xlink:href="http://www.eurocontrol.int/amdb#xpointer(/aixm:Airport[ESSA]/aixm:Taxiway[UF]"/>
```

It must be parsed to retrieve the taxiway Natural Key, that is to say “ESSA/UF”.

3.4.2 Error Case

If the associated feature is not found in the SNOWTAM Database, the feature is not treated and an error is logged in either a log file.

Enhancement:

- ⇨ After having made some tests, it appears that lots of parent features cannot be retrieved in the xSNOWTAM database (especially Aprons and Taxiways). A solution could be to create empty Aprons or Taxiways *on the fly*. They will of course only have a designator.
  By the way, these new Aprons and Taxiways might be involved in contaminations, therefore it will not be possible to remove them afterwards from the database.

3.5 Geometry

In the AMDB subset provided, there are two types of Geometries: elevated surface and elevated curve.

3.5.1 Elevated Surface

An elevated surface is simply a polygon in two dimensions (the last point coordinates are equal the first point coordinates). It will be saved as a polygon in Oracle.

```
<aixm:extent>
  <aixm:ElevatedSurface srsName="urn:ogc:def:crs:OGC:1.3:CRS84" srsDimension="2" gml:id="afed323b">
    <gml:polygonPatches>
      <gml:PolygonPatch>
        <gml:exterior>
          <gml:LinearRing>
            <gml:posList>
              16.23669102 58.59026703 16.23656106 58.59012501 16.23643092 58.58998299
```

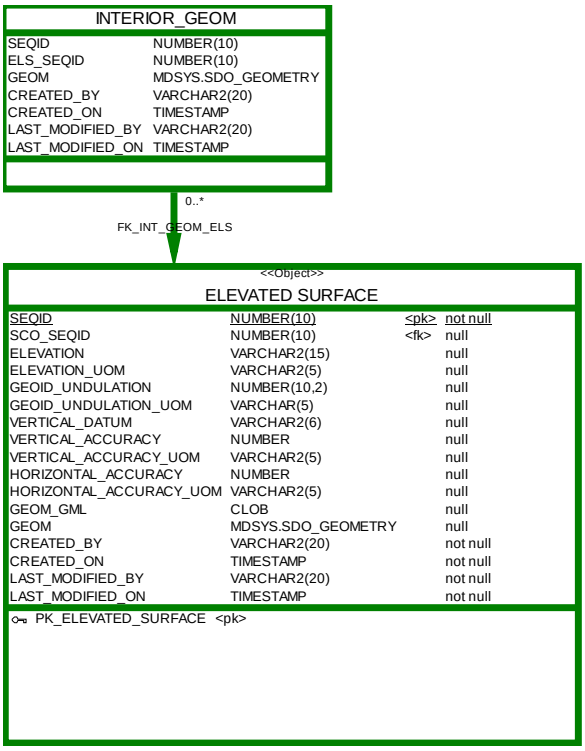
```
                16.24566402 58.58768304 16.24579398 58.58782497 16.24592394 58.58796699
                16.23669102 58.59026703
              </gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:PolygonPatch>
      </gml:polygonPatches>
    </aixm:ElevatedSurface>
</aixm:extent>
```

3.5.1.1 Taxiway specific case

For Taxiways, the geometry is more complex. It is composed of an exterior geometry and interior ones. The exterior geometry can be a line or a multiline. Each interior geometry is also a line or a multiline. The final taxiway geometry is the result of the subtraction of the interior geometries to the exterior one:

```
<aixm:ElevatedSurface srsName="urn:ogc:def:crs:OGC:1.3:CRS84" srsDimension="2" gml:id="aad1a709">
            <gml:polygonPatches>
              <gml:PolygonPatch>
                <gml:exterior>
                  <gml:Ring>
                    <gml:curveMember>
                      <gml:LineString gml:id="ef83aa75">
                        <gml:posList>17.920413 59.643415  17.92166 59.646558 17.92129 59.645541
17.921115 59.64513501 17.921072 59.645048 17.920718 59.64422799 17.920636 59.644038 17.920625
59.644013 17.920499 59.64372001 17.920413 59.643415</gml:posList>
                      </gml:LineString>
                    </gml:curveMember>
                    <gml:curveMember>
                      <gml:LineString gml:id="c0fd49d7">
                        <gml:posList>17.932661 59.65483701 17.932717 59.654852 17.932752 59.654874
17.932762 59.65489801 17.932744 59.654921 17.932717 59.65493601 17.932672 59.65495201 17.932616
59.65495201 17.93256 59.654944 17.932498 59.654921 17.932483 59.65488399 17.932509 59.654856
17.932557 59.65484401 17.932624 59.654836 17.932661 59.65483701</gml:posList>
                      </gml:LineString>
                    </gml:curveMember>
                  </gml:Ring>
                </gml:exterior>
                <gml:interior>
                  <gml:Ring>
                    <gml:curveMember>
                      <gml:LineString gml:id="ae547aa8">
                        <gml:posList>17.923547 59.649319 17.92352 59.64931 17.923476 59.64931
17.923444 59.64932001 17.923432 59.649335 17.923432 59.64935099 17.923438 59.649362 17.923459
59.649371 17.92349 59.649375 17.923526 59.649371 17.923551 59.649359 17.92356 59.64934301 17.923557
59.64933099 17.923547 59.649319</gml:posList>
                      </gml:LineString>
                    </gml:curveMember>
                  </gml:Ring>
                </gml:interior>
              </gml:PolygonPatch>
            </gml:polygonPatches>
          </aixm:ElevatedSurface>
```

From a technical point of view, the exterior geometry is saved in the GEOM field of the ELEVATED_SURFACE table. Each interior geometry is saved in the GEOM field of the INTERIOR_GEOM. A trigger will then substract each inserted interior geometry to the GEOM of the ELEVATED_SURFACE.



3.5.2 Elevated Curve

An elevated curve is composed of one or **more** gml:LineStringSegment, in two dimensions. It will be saved as a line or multiline in Oracle.

```
   <aixm:extent>
     <aixm:ElevatedCurve srsName="urn:ogc:def:crs:OGC:1.3:CRS84" srsDimension="2"
gml:id="d5936664">
        <gml:segments>
          <gml:LineStringSegment>
            <gml:posList>
                17.919225 59.643849 17.919293 59.64404 17.919303 59.64406799 17.919411 59.644367
                17.919437 59.64444001 17.919519 59.64466501 17.919715 59.64522899
                </gml:posList>
            </gml:LineStringSegment>
          <gml:LineStringSegment>
            <gml:posList>
                17.918747 59.64253001 17.918909 59.642978 17.919118 59.643557 17.919213 59.64381701
            </gml:posList>
          </gml:LineStringSegment>
        </gml:segments>
      </aixm:ElevatedCurve>
```

```
</aixm:extent>
```

3.6 Other fields

The AMDB files are AIXM 5.0 compliant and the xSNOWTAM database is also directly inspired by AIXM 5.0. Consequently, the mapping is quite straightforward (nothing to transform, just put the right thing at the right place).

Example:

For Surface Characteristics, the AMDB xml looks like this:

```
<aixm:surfaceProperties>
  <aixm:SurfaceCharacteristics>
    <aixm:composition>ASPH</aixm:composition>
    <aixm:classPCN>55</aixm:classPCN>
    <aixm:pavementTypePCN>FLEXIBLE</aixm:pavementTypePCN>
    <aixm:pavementSubgradePCN>B</aixm:pavementSubgradePCN>
    <aixm:maxTyrePressurePCN>X</aixm:maxTyrePressurePCN>
    <aixm:evaluationMethodPCN>TECH</aixm:evaluationMethodPCN>
    <aixm:classLCN>35</aixm:classLCN>
    <aixm:weightSIWL uom="KG">20</aixm:weightSIWL>
    <aixm:tyrePressureSIWL uom="BAR">1.1</aixm:tyrePressureSIWL>
    <aixm:weightAUW uom="LB">10</aixm:weightAUW>
  </aixm:SurfaceCharacteristics>
</aixm:surfaceProperties>
```

The corresponding table in database is:

<<Object>>
SURFACE_CHARACTERISTICS

SEQID	NUMBER(10)	<pk>
COMPOSITION	VARCHAR2(15)	
PREPARATION	VARCHAR2(8)	
SURFACE_CONDITION	VARCHAR2(8)	
CLASS_PCN	VARCHAR2(5)	
PAVEMENT_TYPE_PCN	VARCHAR2(8)	
PAVEMENT_SUBGRADE_PCN	VARCHAR2(5)	
MAX_TYRE_PRESSURE_PCN	VARCHAR2(1)	
EVALUATION_METHOD_PCN	VARCHAR2(4)	
CLASS_LCN	NUMBER(8,2)	
WEIGHT_SIWL	NUMBER(12,4)	
WEIGHT_SIWL_UOM	VARCHAR2(5)	
TYRE_PRESSURE_SIWL	NUMBER(12,4)	
TYRE_PRESSURE_SIWL_UOM	VARCHAR2(5)	
WEIGHT_AUW	NUMBER(12,4)	
WEIGHT_AUW_UOM	VARCHAR2(5)	
CREATED_BY	VARCHAR2(20)	
CREATED_ON	TIMESTAMP	
LAST_MODIFIED_BY	VARCHAR2(20)	
LAST_MODIFIED_ON	TIMESTAMP	

Therefore, the mapping will be:

AMDB XML fields	*Database fields*
<aixm:composition>	COMPOSITION
<aixm:classPCN>	CLASS_PCN

<aixm:pavementTypePCN>	PAVEMENT_TYPE_PCN
<aixm:pavementSubgradePCN>	PAVEMENT_SUBGRADE_PCN
<aixm:maxTyrePressurePCN>	MAX_TYRE_PRESSURE_PCN
<aixm:evaluationMethodPCN>	EVALUATION_METHOD_PCN
<aixm:classLCN>	CLASS_LCN
<aixm:weightSIWL uom="KG">	WEIGHT_SIWL and WEIGHT_SIWL_UOM
<aixm:tyrePressureSIWL uom="BAR">	TYRE_PRESSURE_SIWL and TYRE_PRESSURE_UOM
<aixm:weightAUW uom="LB">	WEIGHT_AUW and WEIGHT_AUW_UOM