MESSAGES FORMATS USED BY THE IMPORT AND CONTROL SERVICES

26/08/2015 VERSION 1.2

SOMMAIRE

1. INTRODUCTION

The application lets the establishments to manage their registration cards via the web site and via web services:

- On the web site the establishments can manually navigate and fill forms to create, update and read their cards. They can also use the transfer page to create and update several cards automatically.

- With web services the establishment can connect their internal system to create, update and read their cards automatically.

This documentation focuses on the messages formats used by the transfer page and the web services. For more information about managing registration cards manually, please read the website user manual.

The messages exchanged between the establishments and the application are encoded in XML files. Their formats are strictly defined with XSD files.

2. XML MESSAGES

2.1. XML SCHEMA AKA XSD

XSD files are XML documents standardized by the World Wide Web Consortium (W3C – website: http://www.w3.org). XSD stands for "XML Schema Document": it is used to express a set of rules to which an XML document must conform in order to be considered 'valid' according to that schema (see http://en.wikipedia.org/wiki/XML_Schema_(W3C) for more information).

Although these files are all formatted in XML, they are not meant to be read with a simple text editor but with tools like Altova XML Spy (http://www.altova.com/xml-editor) or Oxygen XML Editor (http://www.oxygenxml.com).

> ### Note:
>
> *The XSD files located in the annex folder of this document contain a lot of documentation about the different fields: it is not the goal of this document to copy them here, but to explain how to read them.*

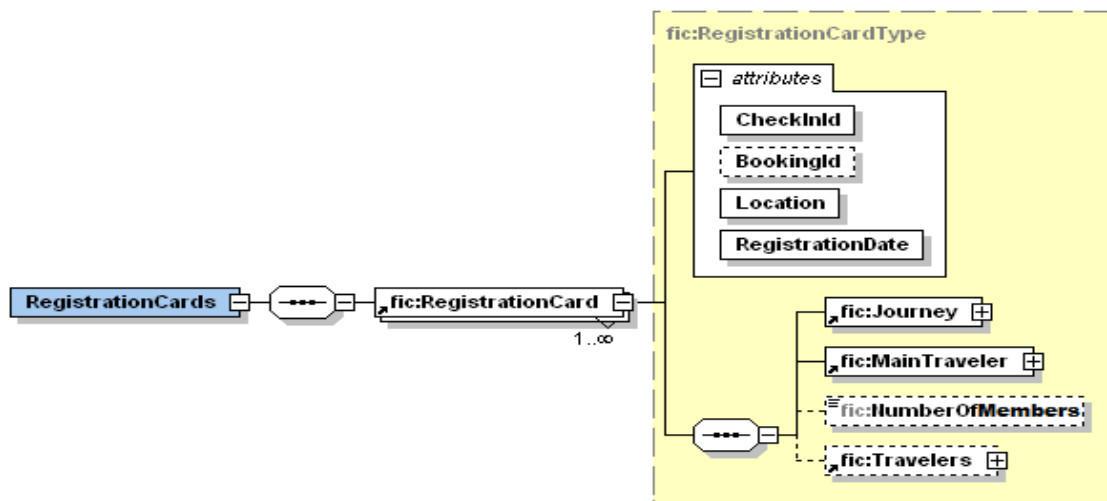## 2.2. REGISTRATION CARDS FORMAT

2.2.1. Understanding XSDs

The transfer function on the website and the web services use the same format to describe registration cards; thus this section explains how to use XSDs by using registration cards as a sample.

The file **data.xsd** describes how to format the registration cards in XML. Open it with a XML editor (here Oxygen XML Editor):

|	
element	Transmission
complexType	DetailAddressType
complexType	DetailTravelerType
complexType	EstablismentCodeType
complexType	JourneyType
complexType	MainTravelerType
complexType	RegistrationCardsType
complexType	RegistrationCardType

You will get a list of elements and types, browse on the element **RegistrationCards**:


As you can see this view explains clearly how to format the XML documents; for example with only the information of the previous picture we can already create the following document:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<fic:RegistrationCards
        xmlns:fic="http://xml.public.lu/schema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://xml.public.lu/schema file:///./data.xsd">
        <!-- Registration card without optional element -->
        <fic:RegistrationCard RegistrationDate="2008-11-21T22:03:31.062+02:00" Location="312" CheckInId="123">
                <fic:Journey .../>
                <fic:MainTraveler ...>…</fic:MainTraveler>
        </fic:RegistrationCard>
        <!-- Registration card with some optional elements -->
        <fic:RegistrationCard RegistrationDate="2009-11-21T22:03:31.062+02:00" Location="'22'" CheckInId="32" bookingId="12">
                <fic:Journey .../>
                <fic:MainTraveler ...>…</fic:MainTraveler>
        </fic:RegistrationCard>
</fic:RegistrationCards>
```

XSD files are mostly used to validate XML documents which are written by hand or generated. Tools like Altova XML Spy can generate empty XML documents based on XSDs:
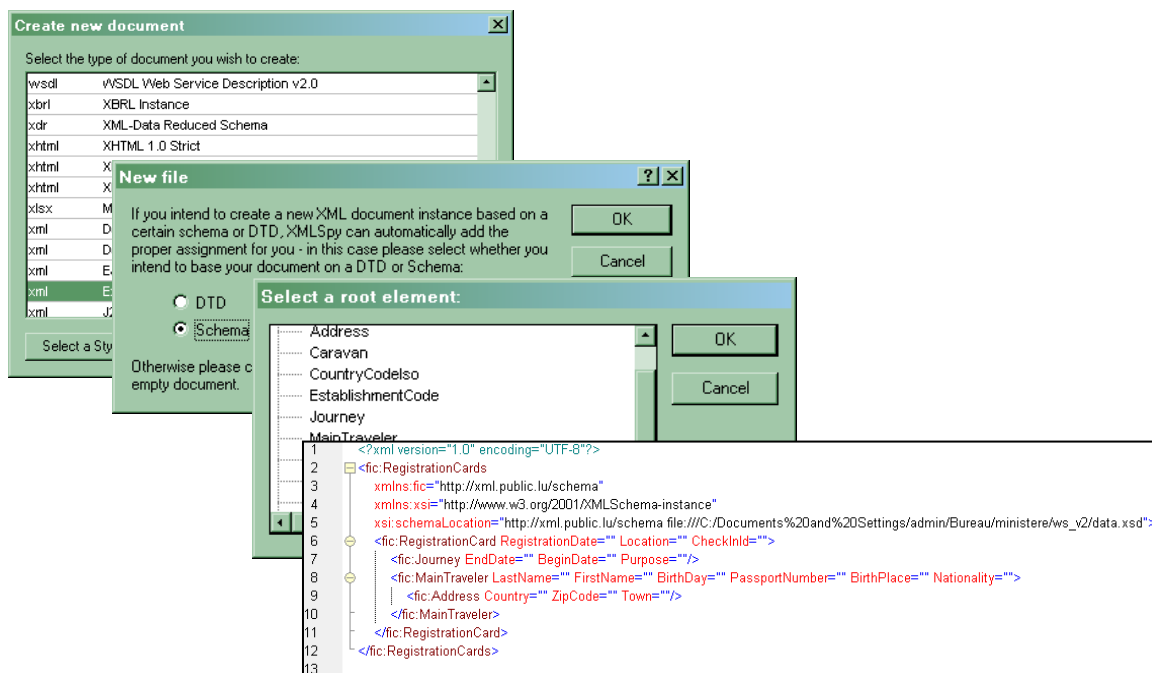
They can also indicate errors precisely (in this sample the zip code has been forgotten):
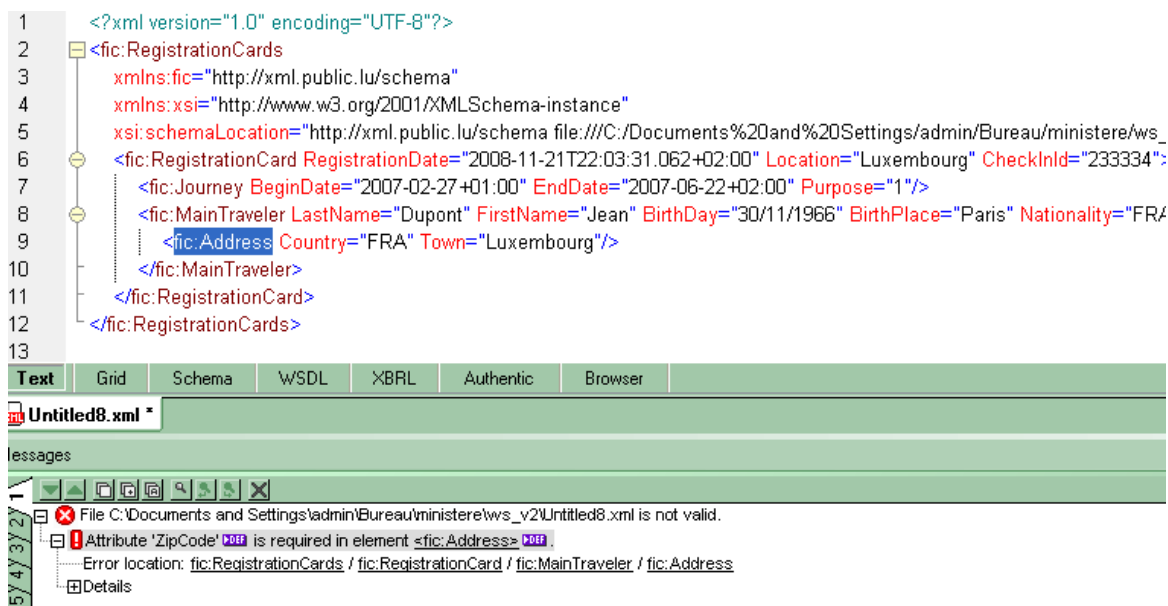
Validating a XML document before sending it is a must. However "XSD valid" doesn't mean the document is 100% right: everything cannot be checked with a schema, for example the uniqueness of an ID in the database.

2.2.2. Registration cards

The registration card is composed of:

- A journey
    - Purpose - allowed values:
        - "1" : Leisure/Other
        - "2" : Business/Congress

    - *CampJourneyType* (**Only for camping**) – allowed values:
        - "1" : Residential
        - "2" : Other

    - BeginDate (e.g. 2011-01-11)
    - EndDate (e.g. 2011-01-12)

- A main traveler – composed of:
    - PassportNumber (e.g. 07AB178386)
    - Nationality (e.g. FRA)
    - BirthPlace (e.g. Metz)
    - BirthDay (e.g. 25/11/1985)
    - FirstName (e.g. John)
    - LastName (e.g. Doe)
    - Address – composed of:
        - Country (e.g. FRA)
        - Town (e.g. Paris)
        - ZipCode (e.g. 75000)
        - StreetNumber (e.g. 11)
        - StreetAddress (e.g. rue des Lilas)

2.2.3. Changes compared to the previous version

- The group card cannot be used anymore. The only remaining format is the individual card.

- Concerning the "Purpose", previously, there were 4 values which have to match to the 2 new values:
    - "1" (Leisure) => "1" Leisure / Other
    - "2" (Business) => "2" Business / Congress
    - "3" (Congress) => "2" Business / Congress
    - "4" (Other) => "1" Leisure / Other

- Concerning the "CampJourneyType", there is only 2 values remaining:
    - "1" (Residential) => "1" Residential

- "2" (Seasonal) => "2" Other
- "3" (In passing) => "2" Other

- The "AccompanyingPerson" element cannot be specified anymore.

- The "Travelers" and "Traveler" elements cannot be specified anymore.

- The "Vehicle" element cannot be specified anymore.

- The "Caravan" element cannot be specified anymore.

- The "NumberOfChildren" element cannot be specified anymore.

- The sum of the number of the "Travelers", "AccompanyingPerson" and "NumberOfChildren" is specified in the new element "NumberOfMembers". This element does not include the main traveler in the sum of members.

The following table illustrates each case with a sample (for a hotel):

Case	Sample

1 adult	```xml
	<RegistrationCard
	        RegistrationDate="2011-01-11T00:00:00Z"
	        Location="217"
	        CheckInId="1"
	        BookingId="1">
	        <Journey
	                Purpose="1"
	                BeginDate="2011-01-11"
	                EndDate="2011-01-12"/>
	        <MainTraveler
	                PassportNumber="243021491"
	                Nationality="FRA"
	                BirthPlace="Paris"
	                BirthDay="01/01/1959"
	                FirstName="Jean"
	                LastName="Dupont">
	                <Address
	                        Country="FRA"
	                        Town="Paris"
	                        ZipCode="75000"
	                        StreetNumber="1"
	                        StreetAddress="rue des Lilas" />
	        </MainTraveler>
	</RegistrationCard>
	``

n adults	```xml
	<RegistrationCard
	        RegistrationDate="2011-01-11T00:00:00Z"
	        Location="217"
	        CheckInId="1"
	        BookingId="1">
	        <Journey
	                Purpose="1"
	                BeginDate="2011-01-11"
	                EndDate="2011-01-12"/>
	        <MainTraveler
	                PassportNumber="243021491"
	                Nationality="FRA"
	                BirthPlace="Paris"
	                BirthDay="01/01/1959"
	                FirstName="Jean"
	                LastName="Dupont">
	                <Address
	                        Country="FRA"
	                        Town="Paris"
	                        ZipCode="75000"
	                        StreetNumber="1"
	                        StreetAddress="rue des Lilas" />
	        </MainTraveler>
	        <NumberOfMembers>n-1</NumberOfMembers>
	</RegistrationCard>
	``

The following table illustrates each case with a sample (for a camping):

Case	Sample

1 adult	```xml
	<RegistrationCard
	        RegistrationDate="2011-01-11T00:00:00Z"
	        Location="217"
	        CheckInId="1"
	        BookingId="1">
	        <Journey
	                Purpose="1"
	                CampJourneyType="1"
	                BeginDate="2011-01-11"
	                EndDate="2011-01-12"/>
	        <MainTraveler
	                PassportNumber="243021491"
	                Nationality="FRA"
	                BirthPlace="Paris"
	                BirthDay="01/01/1959"
	                FirstName="Jean"
	                LastName="Dupont">
	                <Address
	                        Country="FRA"
	                        Town="Paris"
	                        ZipCode="75000"
	                        StreetNumber="1"
	                        StreetAddress="rue des Lilas" />
	        </MainTraveler>
	</RegistrationCard>
	``

n adults	```xml
	<RegistrationCard
	        RegistrationDate="2011-01-11T00:00:00Z"
	        Location="217"
	        CheckInId="1"
	        BookingId="1">
	        <Journey
	                Purpose="1"
	                CampJourneyType="1"
	                BeginDate="2011-01-11"
	                EndDate="2011-01-12"/>
	        <MainTraveler
	                PassportNumber="243021491"
	                Nationality="FRA"
	                BirthPlace="Paris"
	                BirthDay="01/01/1959"
	                FirstName="Jean"
	                LastName="Dupont">
	                <Address
	                        Country="FRA"
	                        Town="Paris"
	                        ZipCode="75000"
	                        StreetNumber="1"
	                        StreetAddress="rue des Lilas" />
	        </MainTraveler>
	        <NumberOfMembers>n-1</NumberOfMembers>
	</RegistrationCard>
	``

Note:

CampJourneyType only for camping

3. IMPORTING CARDS

This service let the establishments to create and update several registration cards automatically.

Connect to the website and go to the **TRANSFER** page:

The file to upload is described by the **data.xsd** file; the root element is **Transmission**:
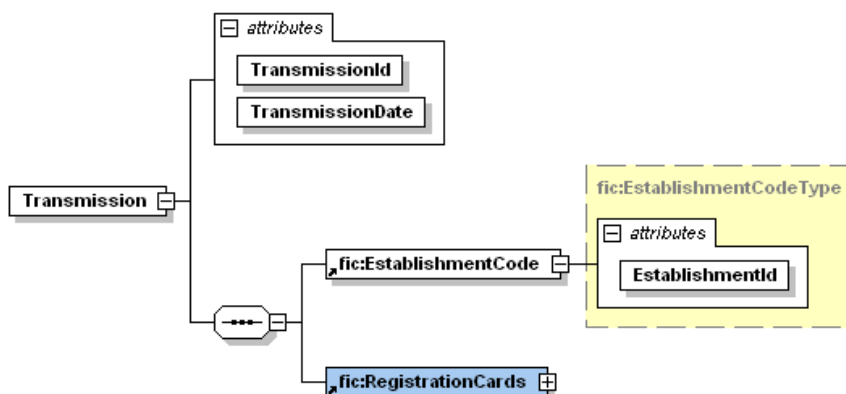
The **TransmissionId** is set by the client and must be unique; it is used to identify his request in order to help him to match it with the response and call the help desk in case of a problem.

The **EstablishmentId** is a code which respects the STATEC nomenclature.

The **RegistrationCards** element has already been described in the previous section.

Here are sample XML documents:

- Personal form

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Transmission TransmissionDate="2010-07-09T21:40:36Z"
              TransmissionId="86147" >
          <EstablishmentCode EstablishmentId="0000106"/>
          <RegistrationCards>
                    <RegistrationCard RegistrationDate="2011-02-08T00:00:00Z"
                               Location="306"
                               CheckInId="889"
                               BookingId="1">
                              <Journey Purpose="1"
                                       BeginDate="2011-02-08"
                                       EndDate="2011-02-10"/>
                              <MainTraveler PassportNumber="243021491"
                                       Nationality="FRA"
                                       BirthPlace="Paris"
                                       BirthDay="6/10/1959"
                                       FirstName="Molitor"
                                       LastName="Mario">
                                    <Address Country="ITA"
                                             Town="Jarville"
                                             ZipCode="54500"
                                             StreetNumber="10"
                                             StreetAddress="avenue andre malraux"/>
                              </MainTraveler>
                              <NumberOfMembers>1</NumberOfMembers>
                    </RegistrationCard>
          </RegistrationCards>
</Transmission>
```

After sending a XML document the following page is displayed:

It displays the success and errors detected during the importation. More details are sent to the application administrators so please call them in case of a problem.

4. WEB SERVICES

Web services let the establishments to connect their internal system to create, update and read the registration cards automatically. These services are designed for big establishments who must handle a lot of cards; moreover the development of the bridge between the internal

system and the web services is more complex than just generating a XML file for the website.

If you are interested to use the web services, please contact the Helpdesk (address: helpdesk@cie.etat.lu), they will provide the documentation.

5. ANNEXES

Because of the file sizes, the annex files are located in the folder "annexes" provided with this document.