
Projet « Lot de messages » [LDM]

Table des matières

1	CONTEXTE	2
2	OBJECTIFS	2
3	CONCEPTION	2
4	RÉALISATION	3
A.	LE MÉCANISME DE TRANSFERT DE FICHIERS.....	3
B.	STRUCTURE DU MESSAGE « FICHIER DESCRIPTIF / VOUCHER »	5
5	SPÉCIFICATIONS	13
A.	STRUCTURE DU FICHIER DE DONNÉES	13
B.	CONVENTION DE NOM DU VOUCHER.....	13
C.	CONVENTION DE NOM POUR LES FICHIERS DE DONNÉES	14
6	ANNEXE : EXEMPLE DE CONTENU DE FICHIERS «DESCRIPTIF / VOUCHER»	15
A.	PRISONERSTATUS	15
B.	DIMONA	16

1 Contexte

La migration des applications du mainframe, l'adoption des standards XML (W3C) pour l'échange des données ont poussé les responsables informatiques de la BCSS à repenser la procédure de l'échange des fichiers.

D'une part, il y a des fichiers qui reprennent une suite de messages à traiter individuellement et d'autres dont chacun des messages font partie d'un tout. D'autre part, leur format et leur codage peuvent variés : longueur variable / longueur fixe (mainframe) 'carriage return-line feed' / 'line feed' (windows-linux) ebcdic/ascii/utf8.

En complément de la problématique de l'échange physique de fichiers, il y a le besoin de communiquer une synthèse de leur contenu.

2 Objectifs

Nous avons dès lors énoncé la liste des fonctionnalités à couvrir.

- Définir une procédure commune quel que soit le type de fichiers, pour échanger des « **Lots de Messages** ¹ » ; ces lots devant être traités par la BCSS et ses partenaires.
- Etre à même d'orienter les '**Lots de Messages** vers le processus de traitement idoine à partir d'éléments explicites.
- Disposer d'un mode de communication afin d'en expliciter leur contenu.
- Garantir l'intégrité du LDM (fichier) transmis ou disposer d'un moyen pour vérifier l'intégrité du LDM (fichier) reçu.
- Identifier l'expéditeur.
- Identifier univoquement chacun des LDM
- LDM doit pouvoir s'appliquer à tous les types de fichier ou de contenu :
 - fichiers de records de taille fixe
 - fichiers spécifiques à des applications non génériques : ponctuelles, ...
 - fichiers reprenant un seul message XML reprenant plusieurs occurrences d'informations (exemple : les mutations du cadastre des pensions, la liste des personnes protégées dans le cadre du tarif social, ...),
 - avec des messages SOAP (conforme à des WSDL) (WS-addressing),

3 Conception

Nous avons d'une part le ou les fichiers à échanger que nous pouvons nommer « fichiers des données » et d'autre part, un fichier descriptif qui permet à chacun des interlocuteurs de « savoir » l'objet de ces fichiers de données.

¹ Les termes **Lot de Messages** ont été choisis pour s'adapter à toutes les situations.

Ce fichier descriptif reprend les informations relatives à :

- **son identification :**
 - son nom
 - l'organisation qui l'a émis,
 - son espace temporel
 - un identifiant univoque

- **son traitement**
 - sa structure
 - contenu/ format
 - codage
 - destinataire

- **son intégrité**
 - la taille
 - le moyen de contrôle
 - l'empreinte : Message Digest algorithme de hashage (MD5)

4 Réalisation

La démarche suivie a été de définir un mécanisme pour l'échange d'un ou de plusieurs fichiers de données puis de détailler le contenu du fichier descriptif pour répondre aux fonctionnalités.

Dès lors, nous avons défini un message XML spécifique pour décrire le contenu de ce fichier descriptif. L'élément «*lotPackageVoucher* » encapsule tous les paramètres permettant de répondre aux fonctionnalités décrites précédemment. [Cfr schéma *LotPackageVoucher_20090716.xsd*].

Nous avons également arrêté une convention de nom pour distinguer ces fichiers descriptifs des fichiers de données.

A. LE MÉCANISME DE TRANSFERT DE FICHIERS

La BCSS comme le partenaire peuvent décider de mettre à disposition un ou plusieurs fichiers de données à traiter. D'autre part, chaque type de données (chaque fichier) est lié à une application spécifique (générique ou pas). Pour éviter une multitude de convention de noms pour chacune d'elle, la BCSS a mis au point un système basé sur un fichier descriptif général qui accompagne le ou les fichiers en relation.

La convention de nom pour reconnaître les fichiers descriptifs est primordiale.

Transfert du partenaire vers la BCSS

Le principe de fonctionnement est le suivant :

- Le partenaire place, en premier lieu, tous les fichiers de données sur le serveur FTP.

- Il met en dernier lieu le fichier descriptif (voucher) sur le serveur ftp.
- Le destinataire (*la BCSS*) recherche à intervalle régulier la présence de fichiers répondant au critère « *voucher.xml ».
- La BCSS télécharge le voucher.
- L'exploite :
 - Valide le voucher:
 - validation par rapport au schéma,
 - vérifie la valeur de l'élément *environment*,
 - vérifie que le voucher avec cet ID et de cet expéditeur n'a pas encore été traité,
 - vérifie le contenu du bloc *addressee*.
 - ouvre le voucher et dispose d'info :
 - un ID de voucher,
 - l'application concernée (**applicationCode** et **operationCode** (pour *xml* = le nom de l'élément root)),
 - une liste de fichiers (ou un seul fichier) de données pour l'application concernée,
 - le moyen de le(s) valider.
 - télécharge chacun des fichiers et vérifie la valeur de l'empreinte MD5 (il s'agit de l'**empreinte** du **fichier non-comprimé**)
 - notifie le résultat du déroulement (par email)
 - fournit ces fichiers aux applications concernées (si intégrité positive)
 - supprime du serveur FTP en cas de vérification positive : le « voucher et les fichiers de données ». Le principe est de laisser sur le serveur, les fichiers en suspens.

Si la BCSS constate que le fichier n'est pas valide, alors l'équipe exploitation de la BCSS contactera le partenaire.

Transfert de la BCSS vers le partenaire

Le principe de fonctionnement est le suivant :

- *La BCSS* place, en premier lieu, tous les fichiers de données sur le serveur FTP.
- Il met en dernier lieu le fichier descriptif (**voucher**) sur le serveur ftp.
- Le destinataire (*le partenaire*) recherche à intervalle régulier la présence de fichiers répondant au critère « *voucher.xml ».
- Le destinataire télécharge le voucher.
- L'exploite :
 - Valide le voucher:
 - validation par rapport au schéma
 - vérifie la valeur de l'élément *environment*
 - vérifie que le voucher avec cet ID et de cet expéditeur n'a pas encore été traité
 - vérifie le contenu du bloc *addressee*
 - ouvre le voucher et dispose d'info :
 - un ID de voucher
 - l'application concernée (**applicationCode** et **operationCode** (pour *xml* = le nom de l'élément root))
 - une liste de fichiers de données (ou un seul) pour l'application concernée,

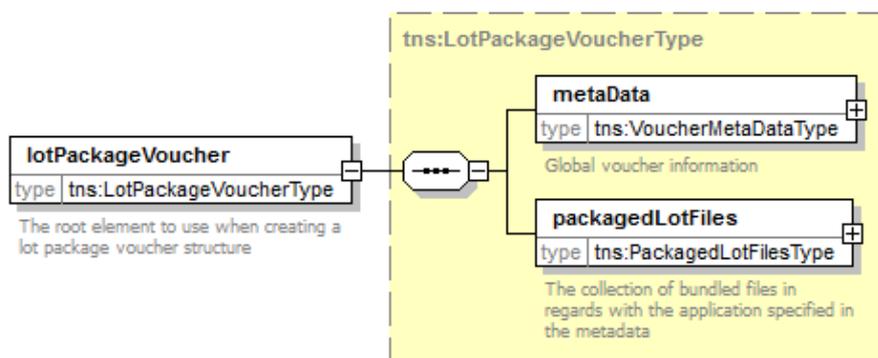
- le moyen de le(s) valider
 - décharge chacun des fichiers et vérifie la valeur de l’empreinte MD5 (il s’agit de l’empreinte du fichier non-comprimé)
 - fournit ces fichiers aux applications concernées (si intégrité positive)
 - supprime du serveur FTP en cas de vérification positive : le « voucher et les fichiers de données ». Le principe est de laisser sur le serveur, les fichiers en suspens.

Si le partenaire constate que le fichier n’est pas valide, alors l’équipe exploitation du partenaire contactera la BCSS.

B. STRUCTURE DU MESSAGE « FICHIER DESCRIPTIF / VOUCHER »

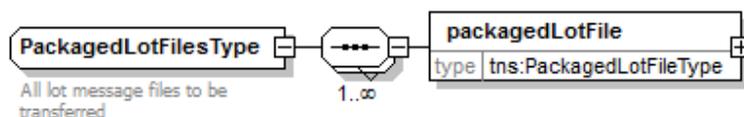
Nous nous limiterons à décrire brièvement le contenu de ce message. Le schéma reprend en commentaire, une description beaucoup plus détaillée.

Le fichier descriptif se présente comme suit :

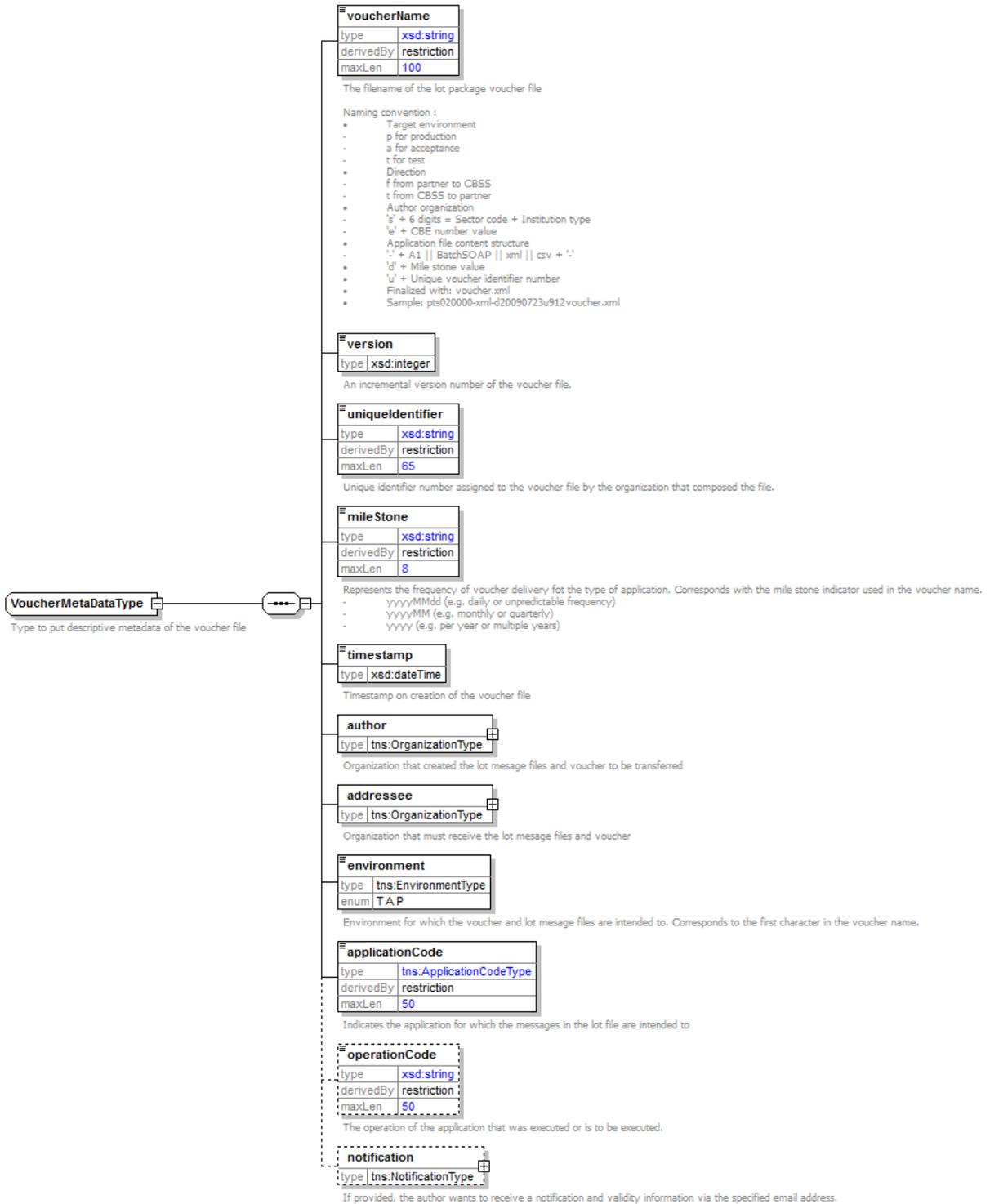


Nom de l’élément	Description
metadata	contient l’information pour identifier le Lot de Messages et le traitement attendu
packagedLotFiles	liste les fichiers de données, leurs spécifications (structure et contenu) et le moyen de vérifier leur intégrité

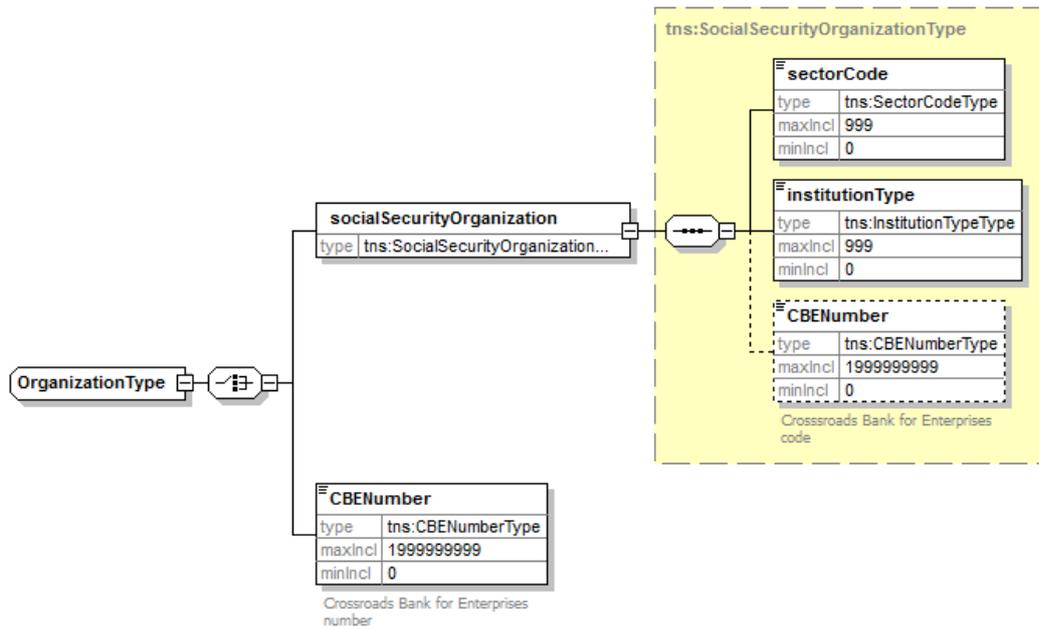
Un Lot de Messages peut être constitué d’un ou de plusieurs fichiers de données.



La partie méta data :

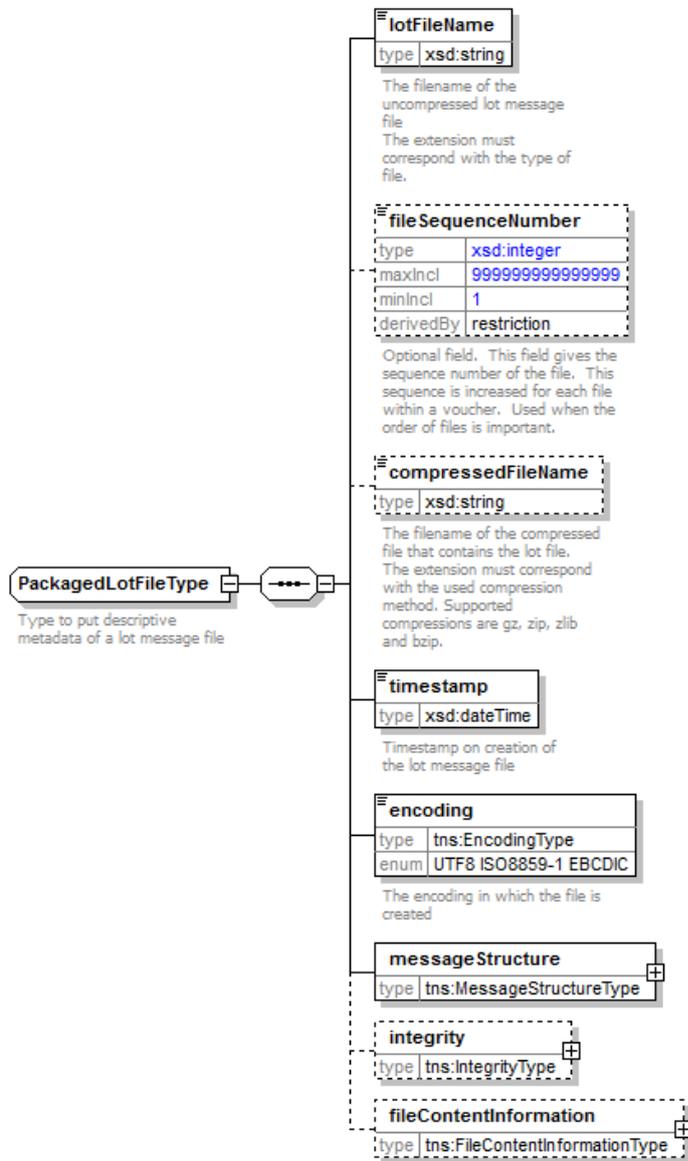


Nom de l'élément	Description
voucherName	Reprend le nom du fichier descriptif.
version	Version prévue en cas de "rerun" dû à une erreur. Si le voucher est envoyé pour la première fois, cet élément contient la valeur 1. Remarque : Si le voucher doit être envoyé à nouveau, avec le même <i>uniqueIdentifier</i> , il faut prendre contact avec le service exploitation de la BCSS.
uniqueIdentifier	Identifie le lot de messages. La valeur correspond à celle du champ "identification unique" du nom du voucher. Pour chaque application, cette valeur est déterminée en consultation avec la BCSS. L'identifiant à utiliser sera définie dans la documentation TSS (<i>Technical Service Specifications</i>) de l'application.
mileStone	Permet de relier à une fréquence d'exécution. Sera souvent la date de création. Exemple: 20180201.
timestamp	Heure à laquelle le fichier descriptif a été créé. Exemple: 2010-01-05T03:03:16.809
author	Expéditeur
addressee	Destinataire (permet de détecter incohérence)
environment	Environnement d'exécution : 'T' pour test, 'A' pour acceptation ou 'P' pour production.
applicationCode	Nom ou acronyme de l'application. Exemples: DismissalCompensation, DIMONA
operationCode	Nom de l'élément root du fichier de données en cas d'XML
notification	Une adresse e-mail peut être fournie pour recevoir une notification du téléchargement et de sa validité



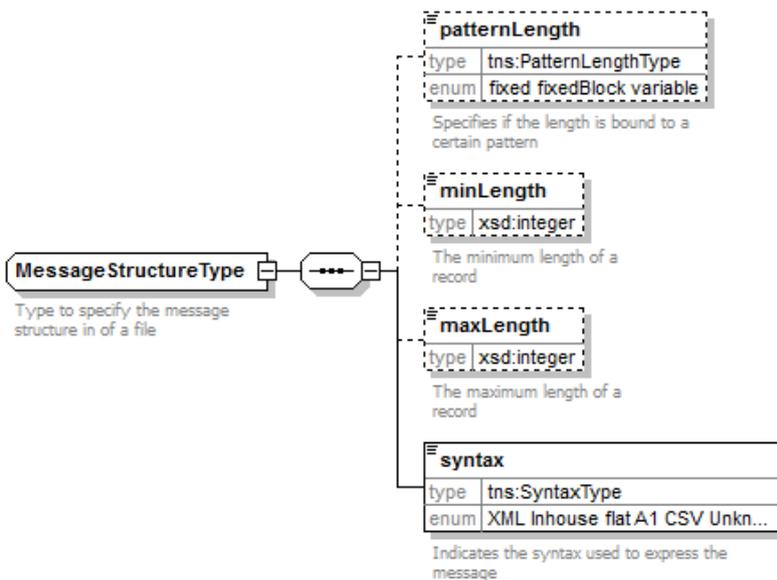
Les organisations peuvent être identifiées par un numéro BCE ou par un code secteur et type d'institution.

Pour chaque fichier repris dans un lot, nous avons les informations suivantes :

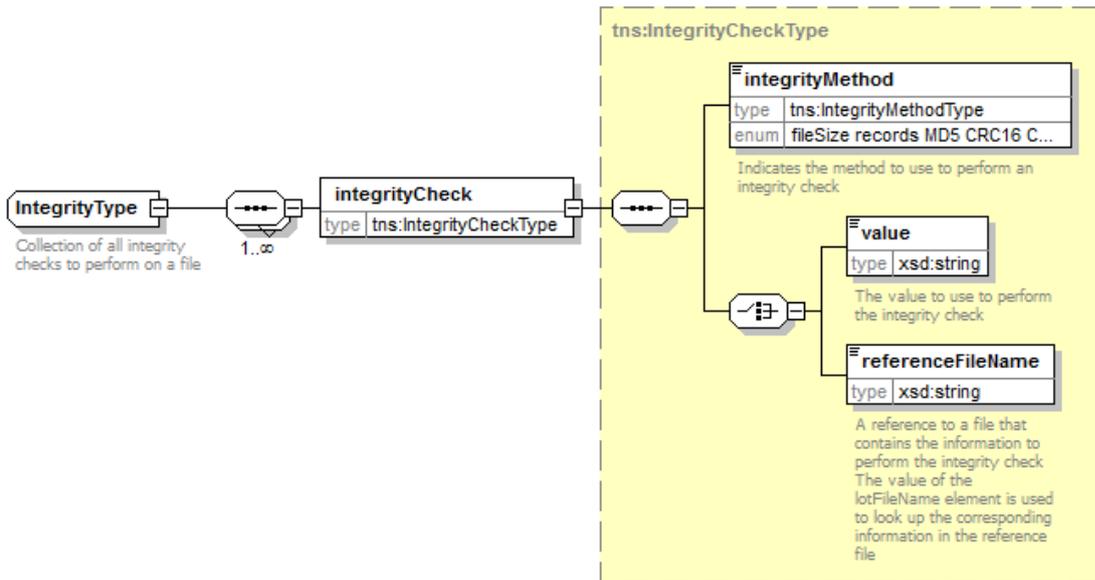


Nom de l'élément	Description
lotFileName	Donne le nom du fichier physique original à traiter. La valeur doit correspondre au nom dans le fichier zip si le fichier se trouve dans un fichier zip.
fileSequenceNumber	Ce champ n'est plus supporté et ne devrait pas être présent. Il était prévu pour permettre la numérotation des fichiers dans un voucher. Mais si l'ordre des fichiers est important l'ordre sera garanti par le champs

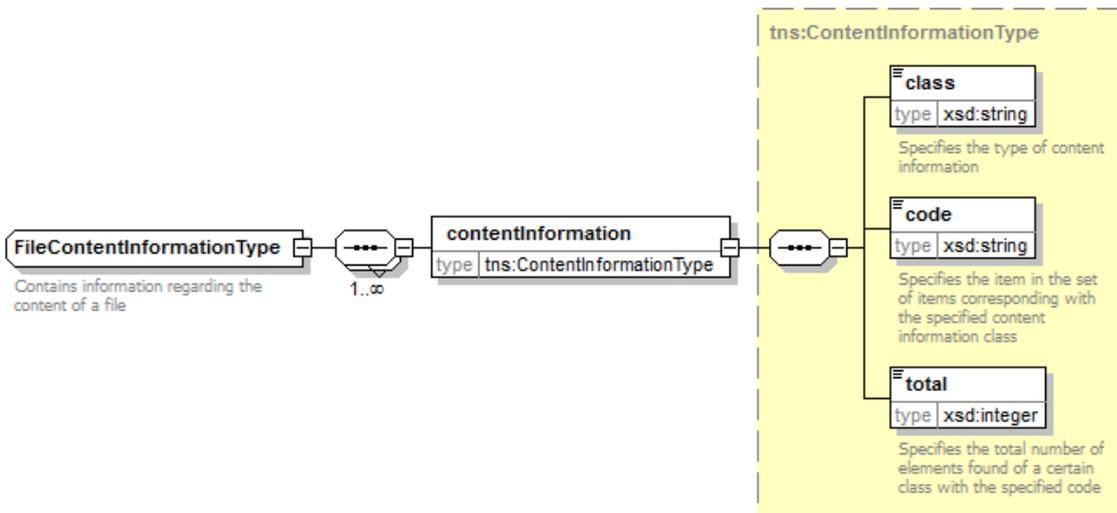
	sequenceNumber dans les fichiers de données. Si le champ est toujours présent, la BCSS ne fera rien avec la valeur de fileSequenceNumber.
compressedFileName	Eventuellement le nom du fichier à télécharger s'il est zippé. Les algorithmes de compression supportés sont: zip et gzip.
timestamp	Heure à laquelle le fichier de données correspondant a été créé
encoding	Précise le codage des caractères. Valeurs possibles: UTF8, ISO8859-1.
messageStructure	Décrit la structure des messages du fichier de données
integrity	Méthode et valeur d'intégrité
fileContentInformation	Fournit les métadonnées et les statistiques du contenu fonctionnel du fichier de données



Nom de l'élément	Description
patternLength	Spécifie si la longueur d'un message est fixe ou pas. Absent en cas de XML. Valeurs possibles : fixed, fixedBlock, variable.
minLength	La longueur minimale d'un message. Absent en cas de XML.
maxLength	La longueur maximale d'un message. Absent en cas de XML.
syntax	Syntaxe du message. Valeurs possibles : XML, flat, CSV.



Nom de l'élément	Description
integrityMethod	Indique la méthode à utiliser pour vérifier l'intégrité du fichier. Valeurs possibles : MD5. Remarque: FileSize est temporairement pris en charge pendant un certain temps, mais sera supprimé du schéma dans une version ultérieure. La bcss préfère utiliser une somme de contrôle MD5 pour le contrôle d'intégrité.
value	La valeur à utiliser pour vérifier l'intégrité.
referenceFileName	Cette fonctionnalité n'est plus supportée par la BCSS. Ce champ sera supprimé du schéma dans une version ultérieure.



Nom de l'élément	Description
class	Le type d'information. Exemples : Node, LegalContext.
code	Le code qui correspond à ce type d'information. Par exemple, le nom d'un élément ou le nom du contexte légal.
total	Le nombre total d'éléments pour cette classe et ce code.

Exemple :

```

<fileContentInformation>
  <contentInformation>
    <class>LegalContext</class>
    <code>NEO:CAREER</code>
    <total>1</total>
  </contentInformation>
  <contentInformation>
    <class>Node</class>
    <code>updateNotification</code>
    <total>1214</total>
  </contentInformation>
</fileContentInformation>
    
```

5 Spécifications

A. Structure du fichier de données

Dans le cadre des projets de ce type², il a été convenu d'utiliser un format XML pour organiser les informations.

Pour que ce fichier soit valide,

- il doit être bien formé,
- Il doit être conforme au schéma.

Les fichiers XML contiennent un certain nombre de blocs standard: sender, receiver, legalContext, sequenceNumber et dataFilters. La documentation de ces blocs peut être trouvée dans le document sur les définitions de messages des services BCSS. Ce document peut être consulté sur le site internet de la BCSS sous la rubrique Services et support / Méthode de travail / Architecture orientée service.

B. Convention de nom du voucher

La convention de nom pour le **voucher** répond à la convention suivante :

<env>	<direction>	<organisation>	<catégorie application>	<milestone>	<identification unique>	suffixe voucher.xml
t / a / p : définit l'environnement						
from/ to : spécifie l'envoi ou la réception de la part du partenaire de la BCSS						
snnnnnn : le secteur et le type d'institution <ul style="list-style-type: none"> • s012000 = ONSS, • s013002 = ORPSS <i>ou</i> en{10} le numéro d'entreprise <ul style="list-style-type: none"> • e0308357753 = SPF Justice 						
-xml- le contenu répond à la norme xml						
dyyy dyyyMM						

² Il s'agit d'applications comme les mutations du cadastre de pension, celles de DIMONA, des interruptions de carrière, Handichild, etc ...

dyyyyMMdd donne la fréquence et le repère temporel
u identifiant qui sera défini dans la documentation TSS (Technical Service Specifications) de l'application

Par exemple :

tts012000-xml-d20180401uDIMONA.notifyResponse.000000023.voucher.xml

tts013002-xml-d20171001uDIMONA.notifyResponse.000000010.voucher.xml

afe0308357753-xml-d20171210uPrisonerStatus.notifyPrisonerMovementRequest.000000010.voucher.xml

C. Convention de nom pour les fichiers de données

Le respect de cette convention n'est pas obligatoire, mais la BCSS l'utilisera progressivement pour ses échanges. Il est d'application pour les messages XML.

<env>	<direction>	<organisation>	<catégorie application>	<milestone>	<identification unique>	suffixe .xml
u/a/p : définit l'environnement						
from/to : spécifie l'envoi ou la réception de la part du partenaire de la BCSS						
snnnnnn : le secteur et le type d'institution <ul style="list-style-type: none"> s012000 = ONSS, s013002 = ORPSS ou en{10} le numéro d'entreprise <ul style="list-style-type: none"> e0308357753 = SPF Justice 						
-xml- le contenu répond à la norme xml						
dyyyy dyyyyMM dyyyyMMdd donne la fréquence et le repère temporel						
u identifiant qui sera défini dans la documentation TSS (Technical Service Specifications) de l'application						

Par exemple : pour les partenaires repris dans les exemples.

tts012000-xml-d20180401uDIMONA.notifyResponse.000000123.xml

tts013002-xml-d20171001uDIMONA.notifyResponse.000000011.xml

afe0308357753-xml-d20171210uPrisonerStatus.notifyPrisonerMovementRequest.000000010.xml

6 Annexe : exemple de contenu de fichiers «descriptif / voucher»

A. PrisonerStatus

tfe0308357753-xml-d20180105uPrisonerStatus.notifyPrisonerMovementRequest.000000006voucher.xml

- **t**= test
- **f**= from
- **e**= enterprise (SPF JUSTICE)

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:lotPackageVoucher
xmlns:tns=http://kszbcss.fgov.be/types/Batch/External/LotPackageVoucherObjects/20090716>
  <metaData>
    <voucherName>tfe0308357753-xml-
d20180105uPrisonerStatus.notifyPrisonerMovementRequest.000000006voucher.xml</voucherName>

    <version>1</version>
    <uniqueIdentifier>PrisonerStatus.notifyPrisonerMovementRequest.000000006</uniqueIdentifier>
    <mileStone>20180105</mileStone>
    <timestamp>2018-01-05T17:29:03.708</timestamp>
    <author>
      <CBENumber>0308357753</CBENumber>
    </author>
    <addressee>
      <socialSecurityOrganization>
        <sectorCode>25</sectorCode>
        <institutionType>0</institutionType>
      </socialSecurityOrganization>
    </addressee>
    <environment>T</environment>
    <applicationCode>PrisonerStatus</applicationCode>
    <operationCode>notifyPrisonerMovementRequest</operationCode>
  </metaData>
  <packagedLotFiles>
    <packagedLotFile>
```

```
<lotFileName>tfe0308357753-xml-  
d20180316uPrisonerStatus.notifyPrisonerMovementRequest.0000000007.xml</lotFileName>  
<timestamp>2018-01-05T17:30:08.463</timestamp>  
<encoding>UTF8</encoding>  
<messageStructure>  
  < syntax>XML</syntax>  
</messageStructure>  
<integrity>  
  <integrityCheck>  
    <integrityMethod>MD5</integrityMethod>  
    <value>39fe7909bd76826e2437b238d15cda76</value>  
  </integrityCheck>  
</integrity>  
<fileContentInformation>  
  <contentInformation>  
    <class>Node</class>  
    <code>prisonerMovementNotification</code>  
    <total>38</total>  
  </contentInformation>  
</fileContentInformation>  
</packagedLotFile>  
</packagedLotFiles>  
</tns:lotPackageVoucher>
```

B. DIMONA

tts012000-xml-d20180112uDIMONA.notifyResponse.0000000037voucher.xml

- **t**= test
- **t**= to
- **s**= secteur/institution (012000)

```
<?xml version="1.0" encoding="UTF-8"?>  
<tns:lotPackageVoucher  
  xmlns:tns="http://kszbcss.fgov.be/types/Batch/External/LotPackageVoucherObjects/20090716">  
  <metaData>  
    <voucherName>tts012000-xml-d20180112uDIMONA.notifyResponse.0000000037voucher.xml</voucherName>  
    <version>1</version>  
    <uniqueIdentifier>DIMONA.notifyResponse.0000000037</uniqueIdentifier>  
    <mileStone>20180112</mileStone>  
    <timestamp>2018-01-12T12:20:51.051</timestamp>  
    <author>  
      <socialSecurityOrganization>  
        <sectorCode>25</sectorCode>  
        <institutionType>0</institutionType>  
        <CBENumber>244640631</CBENumber>  
      </socialSecurityOrganization>  
    </author>  
    <addressee>  
      <socialSecurityOrganization>  
        <sectorCode>12</sectorCode>  
        <institutionType>0</institutionType>
```

```
        </socialSecurityOrganization>
    </addressee>
    <environment>T</environment>
    <applicationCode>DIMONA</applicationCode>
    <operationCode>notifyDimonaDeclarationResponse</operationCode>
</metaData>
<packagedLotFiles>
    <packagedLotFile>
        <lotFileName>tts012000-xml-d20180112uDIMONA.notifyResponse.0000000085.xml</lotFileName>
        <compressedFileName>tts012000-xml-
d20180112uDIMONA.notifyResponse.0000000085.xml.zip</compressedFileName>
        <timestamp>2010-01-12T12:21:52.297</timestamp>
        <encoding>UTF8</encoding>
        <messageStructure>
            <patternLength>variable</patternLength>
            <syntax>XML</syntax>
        </messageStructure>
        <integrity>
            <integrityCheck>
                <integrityMethod>MD5</integrityMethod>
                <value>e7d873ae0372326a5e0f894b77bc4a9c</value>
            </integrityCheck>
        </integrity>
        <fileContentInformation>
            <contentInformation>
                <class>Node</class>
                <code>dimonaDeclarationResponse </code>
                <total>10</total>
            </contentInformation>
            <contentInformation>
                <class>StatusValue</class>
                <code>OK</code>
                <total>10</total>
            </contentInformation>
            <contentInformation>
                <class>StatusCode</class>
                <code>MSG00000</code>
                <total>10</total>
            </contentInformation>
        </fileContentInformation>
    </packagedLotFile>
</packagedLotFiles>
</tns:lotPackageVoucher>
```