

Développement de JGenea Gedcom

Parseurs & générateurs Gedcom

par Templth

Développement de JGenea Gedcom: Parseurs & générateurs Gedcom

par Templth

Ce document, pour le moment, définit la structure et l'organisation du sous-projet JGenea Gedcom.

Il ne donne pour le moment aucune indication concernant la génération de fichiers gedcom à partir de données en base. Cette lacune sera comblée dans une prochaine version du document.

Table des matières

1. Présentation	1
Types de parseur	1
Organisation du sous-projet	1
Organisation des packages java	1
Compilation	3
Exécution et test	5
2. Configuration	6
Fichier de configuration	6
Chargement de la configuration	7
Accès à la configuration	8
3. Traitement d'un fichier	10
Traitement de l'entête	10
Algorithme général	10
4. Les parseurs	12
Les classes génériques et interfaces	12
Parseur DOM-like	13
Parseur SAX-like	14
Handler pour parseur SAX-like	14
5. Les différents éléments gedcom supportés	16
Document Gedcom	16
Entête Gedcom	16
Personne Gedcom	17
Elements constitutifs d'une personne	18
Famille Gedcom	22
Eléments constitutifs d'une famille	24
Source Gedcom	24
Note Gedcom	25
Archive Gedcom	25
Adresse Gedcom	25
Objets multimédia (images...) Gedcom	25
Soumetteur Gedcom	26
6. Modèle objet des éléments Gedcom	27
Entête	27
Soumetteur	27
Personne	27
Elements constitutifs d'une personne	28
Mariage	28
Elements constitutifs d'un mariage	28
Note	28
Source	28
Archive	28
Adresse	28
Objet multimedia	28

Liste des exemples

4.1. Classement des blocs par type	14
5.1. Exemple d'entête gedcom	17
5.2. Exemple de personnes gedcom	17
5.3. Noms d'une personne	19
5.4. Événements d'une personne	20
5.5. Association de personnes	20
5.6. Exemple de citation de sources	21
5.7. Exemple de notes	22
5.8. Exemple de familles gedcom	23

Chapitre 1. Présentation

Types de parseur

Nous allons nous inspirer pour les parseurs des normes de XML. En effet, en XML, il existe deux types de parseur

1. Les parseurs de type SAX: il s'agit d'un parseur événementiel qui appelle un "handler" à chaque fois qu'une balise xml est trouvée.
2. Les parseurs de type DOM: il s'agit de parseur chargeant tout l'arbre xml en mémoire. Il est ensuite possible de le parcourir grâce à des objets normalisés.

Organisation du sous-projet

Voici l'organisation des répertoires pour le projet. Cette organisation est la même que pour les autres sous-projets de JGenea:

```
<JGENEA_GEDCOM_HOME>
+- build/
|   build.xml
+- src/
|   +- bin/
|   |   test.bat
|   +- classes/
|   |   +- org/
|   |   |   +- genealogie/
|   |   |   |   +- gedcom/
|   +- ressources/
|   |   +- xml/
|   |       configuration-gedcom.xml
```

Organisation des packages java

Le sous-projet JGenea Gedcom comprendra les classes comprises dans le package org.genealogie.gedcom et dans les packages fils.

Les différents packages auront les fonctions suivantes:

1. org.genealogie.gedcom.util:

- classe de chargement de la configuration des traitements gedcom
- classes des éléments gedcom de niveau 0

2. org.genealogie.gedcom.encodages:

- interface que doivent implémenter les classes de décodage / encodage
- classes implémentant les décodages / encodages

3. org.genealogie.gedcom.parseurs

- classes génériques des parseurs

4. org.genealogie.gedcom.parseurs.saxlike

- interface que doivent implémenter les classes des parseurs sax-like
- interface que doivent implémenter les handlers
- classes implémentant les parseurs sax-like

5. org.genealogie.gedcom.parseurs.domlike

- interface que doivent implémenter les classes des parseurs dom-like
- classes implémentant les parseurs dom-like

6. org.genealogie.gedcom.traitement

- classes de traitement des gedcom se basant sur les encodeurs / décodeurs et les parseurs

Note

Le nom du package n'est pas très bien choisi!

Compilation

Pour la compilation, le fichier `build.xml` suivant doit être utilisé avec ant:

```

<?xml version="1.0"?>

<!-- ===== -->
<!-- JGenea Gedcom build file -->
<!-- ===== -->

<project name="JGenea" default="main" basedir="..">

    <target name="init">
        <echo message="build.compiler = ${build.compiler}"/>
        <echo message="java.home = ${java.home}"/>
        <echo message="user.home = ${user.home}"/>
        <echo message="java.class.path = ${java.class.path}"/>
        <echo message=""/>
    </target>

    <property name="Name" value="JGenea Gedcom"/>
    <property name="name" value="jgenea gedcom"/>
    <property name="version" value="2.0"/>

    <property name="build.dir" value="${basedir}/build"/>
    <property name="build.classes.dir" value="${basedir}/build/classes"/>
    <property name="build.lib.dir" value="${basedir}/build/lib"/>

    <property name="javadocs.dir" value="${basedir}/javadocs"/>

    <property name="lib.dir" value="${basedir}/lib"/>
    <property name="lib.ext.dir" value="${basedir}/lib/ext"/>
    <property name="src.dir" value="${basedir}/src/classes"/>
    <property name="src.lib.dir" value="${basedir}/src/lib"/>
    <property name="bin.dir" value="${basedir}/bin"/>
    <property name="src.bin.dir" value="${basedir}/src/bin"/>

    <property name="packages" value="org.genealogie.gedcom.*"/>

    <property name="build.compiler" value="classic"/>

    <!-- Build classpath -->
    <path id="classpath">
        <fileset dir="${src.lib.dir}">
            <include name="**/*.jar"/>
        </fileset>
    </path>

    <!-- ===== -->
    <!-- Prepares the build directory -->
    <!-- ===== -->
    <target name="prepare" depends="init">

        <mkdir dir="${lib.dir}"/>

    </target>

    <!-- ===== -->
    <!-- Compiles the source code -->
    <!-- ===== -->
    <target name="compile" depends="prepare">
        <mkdir dir="${build.classes.dir}"/>
        <javac srcdir="${src.dir}"
            destdir="${build.classes.dir}"
            debug="on"
            deprecation="off"
            optimize="on"
        >
        <classpath refid="classpath"/>
    </javac>
    </target>

    <!-- ===== -->
    <!-- Creates the jar archives --> 4
    <!-- ===== -->
    <target name="jar" depends="compile">

        <mkdir dir="${build.lib.dir}"/>
        <mkdir dir="${build.classes.dir}"/>

```


Exécution et test

Les scripts source de test et d'exécution se trouvent dans le répertoire `<JGENEA_GEDCOM_HOME>/src/bin` et sont copiés par ant dans le répertoire `<JGENEA_GEDCOM_HOME>/bin` lors d'un changement effectué dans ceux-ci.

Pour tester, il suffit donc de se placer dans le dernier répertoire et de lancer les scripts adéquates.

Chapitre 2. Configuration

Fichier de configuration

Ce paragraphe définit la configuration des parseurs gedcom. Voici un exemple de fichier de configuration xml:

```
<gedcom>

  <encodages>
    <!-- Encodeur / Décodeur ANSEL -6>
    <encodage type="ANSEL" id="encodage_ansel" libelle="">
      <classe>org.genealogie.gedcom.caracteres.AnselEncoder</classes>
      <param nom="param1"> ... </param>
      <param nom="param2"> ... </param>
      ...
    </encodage>
  </encodages>

  <parseurs>
    <!-- Parseur par défaut -->
    <parseur type="defaut" version="*" methode="DOMLike" libelle="">
      <classe>org.genealogie.gedcom.parseurs.domlike.Parseur</classes>
      <param nom="param1"> ... </param>
      <param nom="param2"> ... </param>
      ...
    </parseur>
    <!-- Parseur spécifique -->
    <parseur type="heredis" version="7" methode="SAXLike" libelle="">
      <classe>org.genealogie.gedcom.parseurs.saxlike.Parseur</classes>
      <param nom="param1"> ... </param>
      <param nom="param2"> ... </param>
      ...
    </parseur>
  </parseurs>

  <balises>
    <!-- Balise FAM -->
    <balise id="FAM">
      <methode> ... </methode>
      <methode-sax> ... </methode-sax>
      <type>
        <classe>org.genealogie.gedcom.utils.FamilleGedcomUtils</classes>
      </type>
    </balise>

    <!-- Balise INDI -->
    <balise id="INDI">
      <methode> ... </methode>
      <methode-sax> ... </methode-sax>
      <type>
        <classe>org.genealogie.gedcom.utils.IndividuGedcomUtils</classes>
      </type>
    </balise>
  </balises>
</gedcom>
```

Note

Dans la configuration des parseurs, des valeurs par défaut devront être précisées comme la valeur par défaut pour parser un chaîne de lieu, la définition des mariages civils et religieux.

Ce fichier xml de doit permettre de configurer les choses suivantes:

1. Ce fichier doit permettre de configurer dynamiquement des classes d'encodage / décodage des caractères des caractères du fichier gedcom,
2. Il doit également permettre de définir la liste des parseurs possibles et de rediriger dynamiquement vers le bon parseur en fonction des données présentes dans l'entête du fichier gedcom,
3. Il doit enfin permettre de configurer les balises gedcom lues dans le fichier. Il s'agit des balises de niveau 0 dans le fichier.

L'élément `methode` de la balise `balise` permet de définir la méthode à appeler sur le parseur pour lire celle-ci.

L'élément `methode-sax` de la balise `balise` permet de définir la méthode à appeler sur l'handler sax lorsque la balise a été lue.

Pour finir, le type définit la classe utilitaire mappant la balise.

Chargement de la configuration

Le chargement de la configuration se fait grâce à une classe de type singleton. Il s'agit en fait du même type de chargement que pour la configuration des propriétés de JGenea (voir le fichier `configuration.xml` [] dans le cvs du sous-projet `jgenea-ihm`, répertoire `src/ressources/xml/`).

La classe de chargement de la configuration gedcom, `org.genealogie.ConfigurationGedcomUtils`, sera similaire à celle citée précédemment (voir la classe `org.genealogie.utils.ConfigurationUtils` [] dans le cvs du sous-projet `jgenea-dao`, répertoire `src/classes/org/genealogie/utils/`) et de la forme suivante:

```
package org.genealogie.gedcom.utils;

public class ConfigurationGedcomUtils {

    //METHODES STATIQUES

    // A reprendre!
    public static String getValeur(Node noeud) {
    }

    // A reprendre!
    public static String getAttribut(Node noeud,String cle) {
    }

    // Chargement de la configuration avec le nom d'un fichier
    public static void chargerConfigurationGedcom(String fichier) {
        ...
    }

    // Chargement de la configuration avec un flux d'entrée
    public static void chargerConfigurationGedcom(InputStream fis) throws Exception {
        ...
    }

    // Chargement de la configuration dans les objets de la classe
    // à partir de l'arbre xml chargé
    public static void chargerConfigurationGedcom(Document document) throws Exception {
        ...
    }

    //METHODES DU SINGLETON
    //Récupération du singleton
    public static ConfigurationGedcomUtils getInstance() {
        if( configurationGedcomUtils==null ) {
            configurationGedcomUtils=new ConfigurationGedcomUtils();
        }
        return configurationGedcomUtils;
    }

    //METHODES DE L'INSTANCE
    // Constructeur d'initialisation
    public ConfigurationGedcomUtils() {
        ...
    }

    // Méthodes de récupération des listes ( encodage, parseurs & balises )
    // Voir le paragraphe suivant
}
```

Accès à la configuration

L'accès à la configuration se fera pas la classe `org.genealogie.gedcom.utils.ConfigurationGedcomUtils` vue dans le paragraphe suivant. Les méthodes suivantes de la classes permettront de récupérer les informations:

```
package org.genealogie.gedcom.utils;

public class ConfigurationGedcomUtils {
    private Hashtable listeEncodagePossibles;
    ...

    //METHODES DE L'INSTANCE
    // Constructeur d'initialisation
    public ConfigurationGedcomUtils() {
        this.listeEncodagePossibles=new Hashtable();
        ...
    }

    // Méthodes de récupération des listes ( encodage, parseurs & balises )
    public void putEncodage(String nomEncodage,DescriptionEncodageUtils encodage) {
        if( listeEncodagePossibles!=null )
            listeEncodagePossibles.put(nomEncodage,encodage);
    }

    public DescriptionBaseUtils getEncodage(String nomEncodage) {
        if( listeEncodagePossibles!=null )
            return (DescriptionEncodageUtils)listeEncodagePossibles.get(nomEncodage);
        else
            return null;
    }

    public Hashtable getListeEncodagesPossibles() {
        return listeEncodagesPossibles;
    }

    // Idem pour les parseurs et les balises
}
```

Lors du traitement du fichier gedcom et après la lecture des informations contenues dans l'entête de celui-ci, la classe de traitement fera appel à ces méthodes pour charger les différents éléments nécessaires.

Chapitre 3. Traitement d'un fichier

Traitement de l'entête

Le fichier gedcom à traiter sera ouvert normalement en unicode car à ce moment précis, on ne possède aucune information concernant l'encodage utilisé par celui-ci.

Les informations suivantes devront être récupérées:

1. L'application (nom et version) qui a générée le fichier. Ceci sera utile pour rediriger les traitements éventuellement vers un parseur spécifique.
2. Le type d'encodage du fichier. Ceci sera également utile pour choisir le décodeur à choisir. Il doit obligatoirement y avoir un encodeur /décodeur ANSEL.
3. Le format des chaînes de lieu du fichier. Ceci est indispensable pour reconnaître les lieux.
4. La langue du fichier.
5. La version de la norme Gedcom utilisée.

Voici l'exemple d'un entête de fichier gedcom contenant ces informations:

```
0 HEAD
1 SOUR HEREDIS 6.1 PC
  2 VERS 6.1
  2 NAME HEREDIS PC
  2 CORP BSD Concept Ã
    3 ADDR www.heredis.com
1 DATE 13 OCT 2002
  2 TIME 23:42:10
1 GEDC
  2 VERS 5.5
  2 FORM LINEAGE-LINKED
1 CHAR ANSEL
1 LANG FRENCH
1 PLAC
  2 FORM Ville , Code Lieu , D  partement , R  gion , Pays, Subdivision
1 SUBM @S0@
0 @S0@ SUBM
1 NAME BONNEVILLE
1 ADDR
1 _EMAIL Jo  l.bonneville@Free.fr
```

Note

Il est n  cessaire apr  s le chargement du d  codeur de d  coder toutes les informations des champs de l'ent  te car il peut y avoir des accents ou des caract  res qui peuvent   tre encod  s diff  rents qu'en Unicode.

Algorithme g  n  ral

Ce paragraphe explique les algorithmes utilis  s pour traiter un fichier gedcom. Deux algorithmes seront utilis  s:

1. Dans le cas de l'utilisation d'un parseur DOM-like
2. dans le cas de l'utilisation d'un parseur SAX-like

Voici la liste des étapes dans le premier cas:

1. Chargement éventuel de la configuration si cela n'est pas déjà fait
2. Lecture de l'entête du fichier gedcom et mémorisation des informations précédemment citées
3. Chargement de la classe de décodage appropriée
4. Chargement du parseur approprié dans la liste des parseurs DOM-like
5. Chargement du fichier à partir du parseur
6. Récupération du parseur d'une instance de l'objet `org.genealogie.gedcom.utils.DocumentGedcomUtils` qui permettra de lire les différents éléments du fichier

Voici la liste des étapes dans le second cas:

1. Chargement éventuel de la configuration si cela n'est pas déjà fait
2. Lecture de l'entête du fichier gedcom et mémorisation des informations précédemment citées
3. Chargement de la classe de décodage appropriée
4. Chargement du parseur approprié dans la liste des parseurs SAX-like
5. positionnement du handler sur le parseur
6. traitement du fichier à partir du parseur qui appellera les différentes méthodes configurées du handler à chaque fois qu'une balise de niveau 0 aura été lue

Chapitre 4. Les parseurs

Les classes génériques et interfaces

Tous les parseurs devront hérités d'une classe générique abstraite `ParseurGedcomGeneric` du package `org.genealogie.gedcom.parseur`.

Cette classe définit les différentes méthodes pour construire les objets éléments gedcom à partir du bloc sous forme de chaîne de caractères.

```
package org.genealogie.gedcom.parseurs;

public abstract class ParseurGedcomGeneric {
    protected PersonneGedcom parserPersonne(String bloc) {
    }

    protected FamilleGedcom parserFamille(String bloc) {
    }

    protected SourceGedcom parserSource(String bloc) {
    }

    protected NoteGedcom parserNote(String bloc) {
    }

    protected ArchiveGedcom parserArchive(String bloc) {
    }

    ...
}
```

Chaque parseur doit implémenter une des deux interfaces suivantes:

1. ParseurDomLike: définissant un parseur de type DOM-like

```
package org.genealogie.gedcom.parseurs;

public interface ParseurDOMLike {
    public DocumentGedcom parser(String nomFichier);
    public DocumentGedcom parser(InputStream fichier);

    public EnteteGedcom parserEntete(String nomFichier);
    public EnteteGedcom parserEntete(InputStream fichier);

    public void setEncodage(Encodage encodage);
}
```


2. ParseurSAXLike: définissant un parseur de type SAX-like

```
package org.genealogie.gedcom.parseurs;

public interface ParseurSAXLike {
    public void parser(String nomFichier);
    public void parser(InputStream fichier);

    public void parserEntete(String nomFichier);
    public void parserEntete(InputStream fichier);

    public void setHandler(Handler handler);
    public void setEncodage(Encodage encodage);
}
```

Parseur DOM-like

D'après le paragraphe précédent, un parseur gedcom de ce type sera de la forme suivante:

```
package org.genealogie.gedcom.parseurs.domlike;

public class ParseurDOMLikeDefautImpl extends ParseurGedcomGeneric implements ParseurDOMLike {
    public DocumentGedcom parser(String nomFichier) {
    }
    public DocumentGedcom parser(InputStream fichier) {
    }

    public EnteteGedcom parserEntete(String nomFichier) {
    }
    public EnteteGedcom parserEntete(InputStream fichier) {
    }

    public void setEncodage(Encodage encodage) {
    }
}
```

La première étape est la lecture de l'entête du fichier qui permettra par la suite d'aiguiller vers le bon parseur et le bon encodeur / décodeur. C'est pourquoi la lecture de cette entête se fait avec le parseur par défaut. Seule les informations concernant l'encodage et l'application qui a générée le fichier.

Note

Ceci implique les méthodes `parserEntete` des deux interfaces `ParseurDOMLike` et `ParseurSAXLike`.

Attention

Il ne faut pas lire dans cette étape le contenu de la chaîne des lieux car il peut y avoir dedans des caractères spéciaux.

A partir de ces informations, le véritable parcours du fichier gedcom peut commencer.

Le parseur va détecter toutes les balises gedcom de niveau 0 et isoler le bloc dans une chaîne de caractère. D'après le fichier de configuration, le parseur pourra invoquer dynamiquement la bonne méthode pour traiter cette balise et l'ajouter dans la bonne table de hachage.

Il s'agit en effet d'une table de hachage ayant pour clé le type de balise traitée et pour clé une autre table de hachage comprenant la liste de ces blocs avec pour entrée leur identifiant.

Exemple 4.1. Classement des blocs par type

```
Table de hachage générale
clé: INDI --> Table des INDI
           clé: id --> instance de PersonneGedcom correspondante
```

Parseur SAX-like

D'après le paragraphe précédent, un parseur gedcom de ce type sera de la forme suivante:

```
package org.genealogie.gedcom.parseurs.saxlike;

public class ParseurSAXLikeDefaultImpl extends ParseurGedcomGeneric implements ParseurSAXLike {
    public DocumentGedcom parser(String nomFichier) {
    }
    public DocumentGedcom parser(InputStream fichier) {
    }

    public EnteteGedcom parserEntete(String nomFichier) {
    }
    public EnteteGedcom parserEntete(InputStream fichier) {
    }

    public void setHandler(Handler handler) {
    }

    public void setEncodage(Encodage encodage) {
    }
}
```

La première étape est la lecture de l'entête du fichier qui permettra par la suite d'aiguiller vers le bon parseur et le bon encodeur / décodeur. C'est pourquoi la lecture de cette entête se fait avec le parseur par défaut. Seule les informations concernant l'encodage et l'application qui a générée le fichier.

A partir de ces informations, le véritable parcours du fichier gedcom peut commencer.

Le parseur va détecter toutes les balises gedcom de niveau 0 et isoler le bloc dans une chaîne de caractère. D'après le fichier de configuration, le parseur pourra invoquer dynamiquement la bonne méthode du handler qui fera lui-même ses traitements sur l'instance de l'objet correspondante à la balise.

Handler pour parseur SAX-like

Chaque handler devra implémenter l'interface `GedcomSAXLikeHandler` défini de la manière suivante pour pouvoir être invoquée sur chaque élément gedcom traité:

```
package org.genealogie.gedcom.parseur;

public interface GedcomSAXLikeHandler {
    public void traiterPersonne(PersonnesGedcom personne);
    public void traiterFamille(FamillesGedcom famille);
    public void traiterSource(SourceGedcom source);
    public void traiterNote(NoteGedcom note);
    public void traiterArchive(ArchiveGedcom archive);
}
```

Note

L'appel aux méthodes de l'interface est configuré par le fichier xml.

Note

Le nombre des méthodes n'est pas figé et peut être étendu si d'autres éléments gedcom doivent être traitées.

Chapitre 5. Les différents éléments gedcom supportés

Document Gedcom

Cet objet doit pouvoir contenir tous les éléments que l'on peut trouver dans un fichier gedcom:

1. L'entête gedcom
2. La liste des personnes avec les caractéristiques et événements associés
3. La liste des mariages avec les événements associés
4. La liste des sources
5. La liste des notes
6. La liste des archives
7. La liste des adresses
8. Les liaisons entre les éléments cités, les citations et les notes
9. Les objets multimédia
10. Les soumetteurs

Donc, cet objet devra donc avoir un attribut entête avec ses accesseurs ainsi que des attributs de type Vector avec également leurs accesseurs.

Entête Gedcom

Toutes les informations contenues dans l'entête doivent être mappées dans un objet. Voici le prototype d'une entête. Ceci est tiré de la norme gedcom et les pages font référence aux pages de ce document.

```
n HEAD {1:1}
+1 SOUR <APPROVED_SYSTEM_ID> {1:1} p.38
+2 VERS <VERSION_NUMBER> {0:1} p.55
+2 NAME <NAME_OF_PRODUCT> {0:1} p.48
+2 CORP <NAME_OF_BUSINESS> {0:1} p.48
+3 <<ADDRESS_STRUCTURE>> {0:1} p.29
+2 DATA <NAME_OF_SOURCE_DATA> {0:1} p.48
+3 DATE <PUBLICATION_DATE> {0:1} p.51
+3 COPR <COPYRIGHT_SOURCE_DATA> {0:1} p.39
+1 DATE <TRANSMISSION_DATE> {0:1} p.55
+2 TIME <TIME_VALUE> {0:1} p.55
+1 SUBM @XREF:SUBM@ {1:1} p.55
+1 FILE <FILE_NAME> {0:1} p.44
+1 COPR <COPYRIGHT_GEDCOM_FILE> {0:1} p.39
+1 GEDC {1:1}
+2 VERS <VERSION_NUMBER> {1:1} p.55
+2 FORM <GEDCOM_FORM> {1:1} p.44
+1 CHAR <CHARACTER_SET> {1:1} p.39
+2 VERS <VERSION_NUMBER> {0:1} p.55
+1 LANG <LANGUAGE_OF_TEXT> {0:1} p.45
+1 NOTE <GEDCOM_CONTENT_DESCRIPTION> {0:1} p.44
+2 [CONT|CONC] <GEDCOM_CONTENT_DESCRIPTION> {0:M}
```

Exemple 5.1. Exemple d'entête gedcom

```
0 HEAD
1 SOUR HEREDIS 6.1 PC
  2 VERS 6.1
  2 NAME HEREDIS PC
  2 CORP BSD Concept Ã
    3 ADDR www.heredis.com
1 DATE 13 OCT 2002
  2 TIME 23:42:10
1 GEDC
  2 VERS 5.5
  2 FORM LINEAGE-LINKED
1 CHAR ANSEL
1 LANG FRENCH
1 PLAC
  2 FORM Ville , Code Lieu , D  partement , R  gion , Pays, Subdivision
1 SUBM @S0@
0 @S0@ SUBM
1 NAME BONNEVILLE
1 ADDR
1 _EMAIL Jo  el.bonneville@Free.fr
```

Personne Gedcom

Les personnes, g  r  es par JGenea, doivent avoir le prototype suivant. Ceci est tir   de la norme gedcom et les pages font r  f  rence aux pages de ce document.

```
n @XREF:INDI@ INDI {1:1}
+1 <<PERSONAL_NAME_STRUCTURE>> {0:M} p.34
+1 SEX <SEX_VALUE> {0:1} p.53
+1 <<INDIVIDUAL_EVENT_STRUCTURE>> {0:M} p.31
+1 <<INDIVIDUAL_ATTRIBUTE_STRUCTURE>> {0:M} p.30
+1 <<ASSOCIATION_STRUCTURE>> {0:M} p.29
+1 <<SOURCE_CITATION>> {0:M} p.34
+1 <<MULTIMEDIA_LINK>> {0:M} p.33,26
+1 <<NOTE_STRUCTURE>> {0:M} p.33
+1 <<CHANGE_DATE>> {0:1} p.29
```

Attention

Je ne sais pas trop s'il faut g  rer les tags ALIA, ANCI et DESI. Je pense que non!

```
+1 ALIA @<XREF:INDI>@ {0:M} p.55
+1 ANCI @<XREF:SUBM>@ {0:M} p.55
+1 DESI @<XREF:SUBM>@ {0:M} p.55
```

Exemple 5.2. Exemple de personnes gedcom

```
0 @246I@ INDI
1 NAME Amâelie Jeanne Ernestine/BAUDET/
2 GIVN Amâelie Jeanne Ernestine
2 SURN BAUDET
1 SEX F
1 BIRT
2 DATE 19 MAR 1916
2 PLAC Bazaiges (Canton d'Eguzon),36270,Indre (Arrondissement de la
Châatre),France,EGUSON CHANTOME
1 DEAT
2 DATE 27 SEP 1986
2 PLAC Argenton sur Creuse,36200,Indre,France,
2 SOUR @253S@
1 BURI
2 NOTE Inhumâee au cimetière communale de Bazaize, son village natal.
1 OCCU Couturière en chemiserie.
1 FAMC @5958U@

0 @247I@ INDI
1 NAME Jean Baptiste/BAUDET/
2 GIVN Jean Baptiste
2 SURN BAUDET
1 SEX M
1 BIRT
2 DATE 11 DEC 1875
3 TIME 5:0
2 PLAC Celon,36200,Indre,France,"Les PASCAUDS"
2 SOUR @263S@
2 ASSO @266I@
3 TYPE INDI
3 RELA Witness
2 ASSO @260I@
3 TYPE INDI
3 RELA Informant
2 ASSO @269I@
3 TYPE INDI
3 RELA Witness
1 DEAT
2 DATE 29 JUN 1929
3 TIME 6:0
2 PLAC Bazaiges (Canton d'Eguzon),36270,Indre (Arrondissement de la
Châatre),France,
2 SOUR @271S@
2 ASSO @274I@
3 TYPE INDI
3 RELA Informant
1 BURI
2 DATE AFT 29 JUN 1999
2 PLAC Bazaiges (Canton d'Eguzon),36270,Indre (Arrondissement de la
Châatre),France,Cimetière communal
1 OCCU Cantonnier & cultivateur
1 FAMS @5958U@
1 FAMC @5963U@
```

Attention

Il est nécessaire de bien gérer les relations entre les personnes (tag ASSO) car elles sont gérées par JGenea.

Elements constitutifs d'une personne

PERSONAL_NAME_STRUCTURE

```
n NAME <NAME_PERSONAL> {1:1} p.48
+1 GIVN <NAME_PIECE_GIVEN> {0:1} p.49
+1 NICK <NAME_PIECE_NICKNAME> {0:1} p.49
+1 SURN <NAME_PIECE_SURNAME> {0:1} p.49
+1 <<SOURCE_CITATION>> {0:M} p.34
+1 <<NOTE_STRUCTURE>> {0:M} p.33E
```

Exemple 5.3. Noms d'une personne

```
1 NAME Jean Baptiste/BAUDET/
2 GIVN Jean Baptiste
2 SURN BAUDET
```

INDIVIDUAL_EVENT_STRUCTURE

```
n [ BIRT | CHR ] [Y|<NULL>] {1:1}
+1 <<EVENT_DETAIL>> {0:1} p.29
+1 FAMC @<XREF:FAM>@ {0:1} p.55
|
n [ DEAT | BURI | CREM ] [Y|<NULL>] {1:1}
+1 <<EVENT_DETAIL>> {0:1} p.29
|
n ADOP [Y|<NULL>] {1:1}
+1 <<EVENT_DETAIL>> {0:1} p.29
+1 FAMC @<XREF:FAM>@ {0:1} p.55
+2 ADOP <ADOPTED_BY_WHICH_PARENT> {0:1} p.37
|
n [ BAPM | BARM | BASM | BLES ] [Y|<NULL>] {1:1}
+1 <<EVENT_DETAIL>> {0:1} p.29
|
n [ CHRA | CONF | FCOM | ORDN ] [Y|<NULL>] {1:1}
+1 <<EVENT_DETAIL>> {0:1} p.29
|
n [ NATU | EMIG | IMMI ] [Y|<NULL>] {1:1}
+1 <<EVENT_DETAIL>> {0:1} p.29
|
n [ CENS | PROB | WILL ] [Y|<NULL>] {1:1}
+1 <<EVENT_DETAIL>> {0:1} p.29
|
n [ GRAD | RETI ] [Y|<NULL>] {1:1}
+1 <<EVENT_DETAIL>> {0:1} p.29
|
n EVEN {1:1}
+1 <<EVENT_DETAIL>> {0:1} p.29
```

EVENT_DETAIL

```
n TYPE <EVENT_DESCRIPTOR> {0:1} p.43
n DATE <DATE_VALUE> {0:1} p.42/41
n <<PLACE_STRUCTURE>> {0:1} p.34
n <<ADDRESS_STRUCTURE>> {0:1} p.29
n AGE <AGE_AT_EVENT> {0:1} p.37
n CAUS <CAUSE_OF_EVENT> {0:1} p.38
n <<SOURCE_CITATION>> {0:M} p.34
n <<MULTIMEDIA_LINK>> {0:M} p.33,26
n <<NOTE_STRUCTURE>> {0:M} p.33
```

PLACE_STRUCTURE

```
n PLAC <PLACE_VALUE> {1:1} p.51
+1 <<SOURCE_CITATION>> {0:M} p.34
+1 <<NOTE_STRUCTURE>> {0:M} p.33
```

Exemple 5.4. Evénements d’une personne

```
1 BIRT
2 DATE 11 DEC 1875
3 TIME 5:0
2 PLAC Celon,36200,Indre,,France,"Les PASCAUDS"
2 SOUR @263S@
2 ASSO @266I@
3 TYPE INDI
3 RELA Witness
2 ASSO @260I@
3 TYPE INDI
3 RELA Informant
2 ASSO @269I@
3 TYPE INDI
3 RELA Witness
1 DEAT
2 DATE 29 JUN 1929
3 TIME 6:0
2 PLAC Bazaiges (Canton d'Eguzon),36270,Indre (Arrondissement de la
Châatre),France,
2 SOUR @271S@
2 ASSO @274I@
3 TYPE INDI
3 RELA Informant
1 BURI
2 DATE AFT 29 JUN 1999
2 PLAC Bazaiges (Canton d'Eguzon),36270,Indre (Arrondissement de la
Châatre),France,Cimetière communal
```

INDIVIDUAL_ATTRIBUTE_STRUCTURE

```
n NATI <NATIONAL_OR_TRIBAL_ORIGIN> {1:1} p.49
+1 <<EVENT_DETAIL>> {0:1} p.29
|
n NCHI <COUNT_OF_CHILDREN> {1:1} p.39
+1 <<EVENT_DETAIL>> {0:1} p.29
|
n NMR <COUNT_OF_MARRIAGES> {1:1} p.39
+1 <<EVENT_DETAIL>> {0:1} p.29
|
n OCCU <OCCUPATION> {1:1} p.50
+1 <<EVENT_DETAIL>> {0:1} p.29
```

ASSOCIATION_STRUCTURE

```
n ASSO @<XREF:INDI>@ {0:M} p.55
+1 TYPE <RECORD_TYPE> {1:1} p.51
+1 RELA <RELATION_IS_DESCRIPTOR> {1:1} p.52
+1 <<NOTE_STRUCTURE>> {0:M} p.33
+1 <<SOURCE_CITATION>> {0:M} p.34
```


Exemple 5.5. Association de personnes

```
1 ASSO @886I@
2 TYPE INDI
2 RELA Friend
2 NOTE Ils se sont connus au Lycêee Technique d'Etat "Cyprien Desgroux"
qu'ils frâequentâerent tous les deux de 1961 âa 1965 et oâu ils
obtienrent tous deux le BEI d'Electromecanicien.
1 ASSO @959I@
2 TYPE INDI
2 RELA Friend
2 NOTE Ils se sont connus au Lycêee Technique d'Etat "Cyprien Desgroux"
qu'ils frâequentâerent tous les deux de 1961 âa 1965 et oâu ils
obtienrent tous deux le BEI d'Electromecanicien.
```

Attention

Il va falloir regarder tous les types d'association et voir s'il est possible de les insérer dans JGenea.

SOURCE_CITATION

```
n SOUR @<XREF:SOUR>@ /* pointer to source record */ {1:1} p.55
+1 PAGE <WHERE_WITHIN_SOURCE> {0:1} p.55
+1 DATA {0:1}
+2 DATE <ENTRY_RECORDING_DATE> {0:1} p.43
+2 TEXT <TEXT_FROM_SOURCE> {0:M} p.54
+3 [ CONC | CONT ] <TEXT_FROM_SOURCE> {0:M}
+1 QUAY <CERTAINTY_ASSESSMENT> {0:1} p.38
+1 <<MULTIMEDIA_LINK>> {0:M} p.33,26
+1 <<NOTE_STRUCTURE>> {0:M} p.33
| /* Systems not using source records */
n SOUR <SOURCE_DESCRIPTION> {1:1} p.53
+1 [ CONC | CONT ] <SOURCE_DESCRIPTION> {0:M}
+1 TEXT <TEXT_FROM_SOURCE> {0:M} p.54
+2 [ CONC | CONT ] <TEXT_FROM_SOURCE> {0:M}
+1 <<NOTE_STRUCTURE>> {0:M} p.33
```

Exemple 5.6. Exemple de citation de sources

```
1 GRAD
2 DATE 28 JUN 1990
2 PLAC Versailles,78000,Yvelines (Arrond. de Versailles),„France,
2 SOUR @989S@
```

Les deux types de citations (directe ou par référence à une source) doivent être gérés et il est prtès important que la balise DATA contenant le contenu de la citation le soit aussi.

Attention

Comment gérer les actes qui appartiennent à une source mais ne sont pas une citation d'un événement ou d'autres choses?

MULTIMEDIA_LINK

```
n OBJE @<XREF:OBJE>@ {1:1} p.55 /* embedded form*/
|
n OBJE {1:1} p.55 /* linked form*/
+1 FORM <MULTIMEDIA_FORMAT> {1:1} p.48
+1 TITL <DESCRIPTIVE_TITLE> {0:1} p.43
+1 FILE <MULTIMEDIA_FILE_REFERENCE> {1:1} p.47
+1 <<NOTE_STRUCTURE>> {0:M} p.33
```

Les deux types de références (directe ou par référence à un objet multimédia) doivent être gérées.

NOTE_STRUCTURE

```
n NOTE @<XREF:NOTE>@ {1:1} p.55
+1 <<SOURCE_CITATION>> {0:M} p.34
|
n NOTE [SUBMITTER_TEXT> | <NULL>] {1:1} p.54
+1 [ CONC | CONT ] <SUBMITTER_TEXT> {0:M}
```

Exemple 5.7. Exemple de notes

```
1 BURI
2 NOTE Inhumâee au cimetière communale de Bazaize, son village natal.

0 @1002I@ INDI
1 NOTE 12 Mars 1976 - Acte N°129 - Etat Civil de Saint OUEN (Seine
Saint Denis)
2 CONT Acte de naissance de BONNEVILLE Loïc
2 CONT Le douze mars mil neuf cent soixante seize à onze heures
quarante cinq minutes, est née 27, avenue Gabriel Pâeri* : Loïc du sexe
masculin, de Joël BONNEVILLE, née à PARIS, 6ème, le dix juin milneuf
cent quarante sept, agent de l'Electric
2 CONC itée de France, et de Catherine Marie Madeleine CHAUVET, née à
PARIS 10ème, le trente Octobre mil neuf cent quarante huit, employée
administrative, son épouse, domiciliée à TRIEL sur SEINE, (Yvelines),
75 rue des Temporaets.
2 CONT Dressée le treize mars mil neuf cent soixante seize, à onze
heures quarante minutes, sur la déclaration du père, qui, lecture faite
et invitée à lire l'acte, a signé avec Nous, Jean JOUVIN, Adjoint au
Maire de Saint OUEN, officier de l'Et
2 CONC at Civil par délégation.
2 CONT Suivent les signature de Jean JOUVIN et Joël BONNEVILLE.
2 CONT * Note: au 27 Avenue Gabriel PERI, à Saint OUEN, (Seine
Saint-DENIS), se trouvait la clinique privée d'accouchement " Clinique des
Rosiers ". Le médecin accoucheur (et patron) de cette clinique était le
Docteur BEREND.
```

Attention

Les deux types de notes (directe ou par référence à une note) doivent être gérés.

Famille Gedcom

Les familles, gérées par JGenea, doivent avoir le prototype suivant. Ceci est tiré de la norme gedcom et les pages font référence aux pages de ce document.

```
n @<XREF:FAM>@ FAM {1:1}
+1 <<FAMILY_EVENT_STRUCTURE>> {0:M} p.30
+2 HUSB {0:1}
+3 AGE <AGE_AT_EVENT> {1:1} p.37
+2 WIFE {0:1}
+3 AGE <AGE_AT_EVENT> {1:1}
+1 HUSB @<XREF:INDI>@ {0:1} p.55
+1 WIFE @<XREF:INDI>@ {0:1} p.55
+1 CHIL @<XREF:INDI>@ {0:M} p.55
+1 NCHI <COUNT_OF_CHILDREN> {0:1} p.39
+1 <<SOURCE_CITATION>> {0:M} p.34
+1 <<MULTIMEDIA_LINK>> {0:M} p.33,26
+1 <<NOTE_STRUCTURE>> {0:M} p.33
+1 <<CHANGE_DATE>> {0:1} p.29
```

Exemple 5.8. Exemple de familles gedcom

```
0 @5933U@ FAM
1 HUSB @106I@
1 WIFE @3965I@
1 NCHI 1
1 MARR
2 TYPE Religious marriage
2 DATE 22 AUG 1998
3 TIME 15:0
2 PLAC La Trinite,06340,Alpes-Maritimes,,France,
2 SOUR @5936S@
1 CHAN
2 DATE 14 MAY 2002

0 @5953U@ FAM
1 HUSB @221I@
1 WIFE @5885I@
1 NCHI 0
1 MARR
2 DATE 30 JUN 1990
2 NOTE Mariage câelâebrâe dans la plus stricte intimitâe (lieu non communiquâe)
3 CONT Au moment du mariage, les parents de la mariâee habitaient: M
& Mme Renâe Villeneuve, 22 rue de Paumule, 36200, le Pâechereau.
3 CONT Les parents du mariâe habitaient: M & Mme Jean Barre,
Chateauguillaume, 36370, Lignac
1 CHAN
2 DATE 14 MAY 2002

0 @5958U@ FAM
1 HUSB @247I@
1 WIFE @248I@
1 CHIL @246I@
1 CHIL @256I@
1 CHIL @276I@
1 CHIL @278I@
1 CHIL @282I@
1 CHIL @287I@
1 NCHI 6
1 MARR
2 DATE 14 APR 1903
2 PLAC Bazaiges (Canton d'Eguzon),36270,Indre (Arrondissement de la
Châatre),France,Mairie de Bazaiges
2 SOUR @5960S@
2 NOTE Tâemoins de mariage : BAUDET Charles, 28ans, cultivateur,
VILLENEUVE Emile, 21ans, LIVERNETTE Auguste, 43ans, cultivateur, tous trois
demeurant âa Bazaiges. LETELLE Sylvain, 40ans cultivateur,demeurant aux
Pascauds, âa Celon.
3 CONT Câelâebrâe par PERNIN Sylvain, adjoint.
1 MARC
2 NOTE Non
1 CHAN
2 DATE 14 MAY 2002
```

Attention

Il faut faire attention de bien gérer le type d'un mariage. Ceci est indispensable car JGenea gère les mariages religieux et civils.

Éléments constitutifs d'une famille

Pour ces éléments, se reporter au paragraphe Éléments constitutifs d'une personne.

Source Gedcom

Les sources, gérées par JGenea, doivent avoir le prototype suivant. Ceci est tiré de la norme gedcom et les pages font référence aux pages de ce document.

```
n @<XREF:SOUR>@ SOUR {1:1}
+1 DATA {0:1}
+2 EVEN <EVENTS_RECORDED> {0:M} p.44
+3 DATE <DATE_PERIOD> {0:1} p.41
+3 PLAC <SOURCE_JURISDICTION_PLACE> {0:1} p.54
+2 AGNC <RESPONSIBLE_AGENCY> {0:1} p.52
+2 <<NOTE_STRUCTURE>> {0:M} p.33
+1 AUTH <SOURCE_ORIGINATOR> {0:1} p.54
+2 [CONT|CONC] <SOURCE_ORIGINATOR> {0:M} p.54
+1 TITL <SOURCE_DESCRIPTIVE_TITLE> {0:1} p.53
+2 [CONT|CONC] <SOURCE_DESCRIPTIVE_TITLE> {0:M} p.53
+1 ABBR <SOURCE_FILED_BY_ENTRY> {0:1} p.53
+1 PUBL <SOURCE_PUBLICATION_FACTS> {0:1} p.54
+2 [CONT|CONC] <SOURCE_PUBLICATION_FACTS> {0:M} p.54
+1 TEXT <TEXT_FROM_SOURCE> {0:1} p.54
+2 [CONT|CONC] <TEXT_FROM_SOURCE> {0:M} p.54
+1 <<SOURCE_REPOSITORY_CITATION>> {0:1} p.35
+1 <<MULTIMEDIA_LINK>> {0:M} p.33,26
+1 <<NOTE_STRUCTURE>> {0:M} p.33
+1 <<CHANGE_DATE>> {0:1} p.29
```

Note Gedcom

Les notes, gérées par JGenea, doivent avoir le prototype suivant. Ceci est tiré de la norme gedcom et les pages font référence aux pages de ce document.

```
n @<XREF:NOTE>@ NOTE <SUBMITTER_TEXT> {1:1} p.54
+1 [ CONC | CONT] <SUBMITTER_TEXT> {0:M}
+1 <<SOURCE_CITATION>> {0:M} p.34
+1 <<CHANGE_DATE>> {0:1} p.29
```

Archive Gedcom

Les archives, gérées par JGenea, doivent avoir le prototype suivant. Ceci est tiré de la norme gedcom et les pages font référence aux pages de ce document.

```
n @<XREF:REPO>@ REPO {1:1}
+1 NAME <NAME_OF_REPOSITORY> {0 dsdscsd:1} p.48
+1 <<ADDRESS_STRUCTURE>> {0:1} p.29
+1 <<NOTE_STRUCTURE>> {0:M} p.33
+1 <<CHANGE_DATE>> {0:1} p.29
```

Adresse Gedcom

Les adresses, gérées par JGenea, doivent avoir le prototype suivant. Ceci est tiré de la norme gedcom et les pages font référence aux pages de ce document.

```
n ADDR <ADDRESS_LINE> {0:1} p.37
+1 CONT <ADDRESS_LINE> {0:M} p.37
+1 ADR1 <ADDRESS_LINE1> {0:1} p.37
+1 ADR2 <ADDRESS_LINE2> {0:1} p.37
+1 CITY <ADDRESS_CITY> {0:1} p.37
+1 STAE <ADDRESS_STATE> {0:1} p.37
+1 POST <ADDRESS_POSTAL_CODE> {0:1} p.37
+1 CTRY <ADDRESS_COUNTRY> {0:1} p.37
```

Objets multimédia (images...) Gedcom

Les objets multimédia, gérés par JGenea, doivent avoir le prototype suivant. Ceci est tiré de la norme gedcom et les pages font référence aux pages de ce document.

```
n @XREF:OBJE@ OBJE {1:1}
+1 FORM <MULTIMEDIA_FORMAT> {1:1} p.48
+1 TITL <DESCRIPTIVE_TITLE> {0:1} p.43
+1 <<NOTE_STRUCTURE>> {0:M} p.33
+1 BLOB {1:1}
+2 CONT <ENCODED_MULTIMEDIA_LINE> {1:M} p.43
+1 OBJE @<XREF:OBJE>@ /* chain to continued object */ {0:1} p.26
+1 <<CHANGE_DATE>> {0:1} p.29
```

Soumetteur Gedcom

Les adresses, gérées par JGenea, doivent avoir le prototype suivant. Ceci est tiré de la norme gedcom et les pages font référence aux pages de ce document.

```
n @<XREF:SUBM>@ SUBM {1:1}
+1 NAME <SUBMITTER_NAME> {1:1} p.54
+1 <<ADDRESS_STRUCTURE>> {0:1} p.29
+1 LANG <LANGUAGE_PREFERENCE> {0:3} p.45
+1 <<CHANGE_DATE>> {0:1} p.29
```

Chapitre 6. Modèle objet des éléments Gedcom

Ce paragraphe décrit en détail le modèle objet des éléments gedcom décrits dans le paragraphe précédent.

Entête

Cette entête sera stocké dans l'objet `org.genealogie.gedcom.utils.EnteteGedcomUtils`.

```
public class EnteteGedcomUtils {
    private String gedcomGénérateur;
    private String gedcomGénérateurVersion;
    private String gedcomGénérateurName;
    private String gedcomGénérateurDéveloppement;
    private String gedcomDateGénération;

    private String gedcomVersion;
    private String gedcomForm;

    private String gedcomEncodage;
    private String gedcomLangue;

    private String gedcomFormatLieu;
    private SoumetteurGedcomUtils gedcomAuteur;

    ...
    // Liste des accesseurs en lecture et écriture
    ...
}
```

Soumetteur

Cet élément sera stocké dans l'objet `org.genealogie.gedcom.utils.SoumetteurGedcomUtils`.

```
public class SoumetteurGedcomUtils {
    private String nom;
    private AdresseGedcomUtils adresse;
    private String langue;
    private String java.util.Date dateModification;

    ...
    // Liste des accesseurs en lecture et écriture
    ...
}
```

Personne

Cet élément sera stocké dans l'objet `org.genealogie.gedcom.utils.PersonneGedcomUtils`.

```
public class PersonneGedcomUtils {  
    private Vector noms; // Vecteur de PersonneNomGedcomUtils  
    private Vector caracteristiques; // Vecteur de PersonneCaracteristiqueGedcomUtils  
    private Vector evenements; // Vecteur de PersonneEvenementGedcomUtils  
    private Vector notes; // Vecteur de NoteGedcomUtils  
    private Vector citations; // Vecteur de CitationGedcomUtils  
    private String sexe;  
    private Vector elementsMultimedia; // Vecteur de ObjetsMultimediaGedcomUtils  
    private Vector associations; // Vecteur de AssociationGedcomUtils  
  
    private String java.util.Date dateModification;  
}
```

Elements constitutifs d'une personne

Mariage

Elements constitutifs d'un mariage

Note

Source

Archive

Adresse

Objet multimedia