

Niveau de Distribution		
Limité		2
Interne		3
Externe	X	4

Annuaire Interne des Personnels v 1.0 Manuel Technique

(V1.4)

Rédaction		Compléments / Vérification		Approbation		Diffusion	
Nom :	Olivier CHATOR	Nom :		Nom :		Nom :	
Date :	04/01/2010	Date :		Date :		Date :	
Visa :	OC	Visa :		Visa :		Visa :	

1. Historique et documents de référence

1.1. Historique

Date	Version	Nom	Description
27/02/09	V1	Olivier CHATOR	Création du document
04/03/09	V1.1	Laurent CHELET	Relecture
09/03/09	V1.2	Olivier CHATOR	Ajout d'un paragraphe : Cas particulier de la balise « <i>mailMappingChampsModifies</i> »
06/05/09	V 1.3	Olivier CHATOR	Mise à jour suite à : <ul style="list-style-type: none">• Changement de JDK (1.4.2 vers 1.5)• Evolution des copies d'écran• Démarrage sans les photos agent en Juin 2009
04/01/10	V 1.4	Olivier CHATOR	Mise à jour suite à de nombreuses évolutions et modifications

1.2. Documents de référence

Document	Nom
Manuel d'Installation du logiciel A.I.P v 1.x	AIP v 1.0 - Manuel Installation (version).doc
Manuel d'administration du logiciel A.I.P v 1.x	AIP v 1.0 - Manuel Administration (version).doc
Manuel de l'utilisateur du logiciel A.I.P v 1.x	AIP v 1.0 - Manuel Utilisateur (version).doc
Manuel d'exploitation du logiciel A.I.P v 1.x	AIP v 1.0 - Manuel Exploitation (version).doc

Table de matières

1.	HISTORIQUE ET DOCUMENTS DE REFERENCE	2
1.1.	Historique	2
1.2.	Documents de référence	2
2.	INTRODUCTION	5
2.1.	Objectifs du document	5
2.2.	Limite de la v 1.0	5
3.	ARCHITECTURE LOGICIELLE DE L'APPLICATION	6
3.1.	Pré-requis	6
3.1.1.	Les produits logiciels indispensables	6
3.1.2.	Le montage réseau d'accès aux photos agent	6
3.1.3.	Paramétrage Tomcat spécifique au CG33	6
3.1.3.1.	Fichier \${CATALINA_HOME}/conf/web.xml	6
3.1.3.2.	Fichier \${CATALINA_HOME}/conf/context.xml	6
3.1.4.	Paramétrage standard OpenLDAP	7
3.1.5.	Spécification du niveau de log OpenLDAP	8
3.2.	L'arborescence générale du logiciel AIP v 1.0	10
3.3.	Contenu de chaque répertoire de l'application	11
4.	COMPOSANTS HORS « TRAITEMENT CŒUR »	12
4.1.	Fichier params.xml	12
4.1.1.	Structure	12
4.1.2.	Le paramétrage global du logiciel	13
4.1.3.	Cas particulier de la balise « mailMappingChampsModifies »	17
4.2.	Fichiers .properties	18
4.2.1.	Objectifs	18
4.2.2.	Multilinguisme	18
4.2.3.	Structure d'un fichier .properties	18
4.3.	Fichiers .log & .log-yyyy-mm-dd	19
4.3.1.	Objectifs	19
4.3.2.	Paramétrage	19
4.3.2.1.	WEB-INF/classes/log4j.properties	19
4.3.2.2.	conf/params.xml	19
4.3.3.	Structure et contenu du fichier de log	20
4.3.3.1.	Fichier AIPLog.log	20
4.3.4.	Mécanisme d'historisation automatique	21
4.3.5.	Les erreurs	21
4.3.5.1.	Fichier AIPLog.log	21
4.3.5.2.	Fichier catalina.out	21
4.3.5.3.	Cas particulier	22

4.4.	Fichier AIP.css	22
4.5.	Fichier AIP.js	22
4.6.	Fichier params.xml.template	22
4.7.	Fichier log4j.properties. template	22
4.8.	Fichier inetorgperson.schema	22
4.9.	Fichier slapd.conf	23
4.10.	Fichiers ASupprimer.txt	23
4.11.	Fichier MANIFEST.MF	23
4.12.	Fichier web.xml	23
5.	COMPOSANTS « TRAITEMENT CŒUR »	24
5.1.	Introduction	24
5.2.	Servlets	24
5.2.1.1.	Déclaration des Servlets	24
5.2.1.2.	Structure des Servlets	24
5.3.	Librairie AIP.jar	25
6.	COMMENT MODIFIER DES COMPOSANTS CŒUR ?	26
6.1.	Introduction	26
6.2.	Etape 0 : Pré-requis	26
6.3.	Etape 1: Création du projet Eclipse	27
6.4.	Etape 2 : Modification & test des composants	28
6.5.	Etape 3 : Création du fichier AIP.jar	28
6.6.	Etape 4 : Génération du fichier AIP.war	28
6.7.	Etape 5 : Mise à disposition du fichier AIP.war	28
7.	ANNEXE : INTERVENTION DIRECTE SUR LES DONNEES DE L'ANNUAIRE	29
7.1.	Introduction	29
7.2.	L'outil JExplorer	29
7.3.	Connexion au Ldap	29
7.4.	Modification de données	30
7.5.	Déconnexion	30

2. Introduction

2.1. Objectifs du document

L'Annuaire Interne des Personnels (désigné par l'acronyme A.I.P.) s'inscrit dans le cadre du projet global « Annuaire du CG33 » qui regroupe également les annuaires applicatifs internes et externes (SSO).

Dans ce contexte, le présent document vise à décrire tous les aspects techniques relatifs au logiciel A.I.P. en v 1.0 pour ce qui concerne les points suivants :

- Architecture interne du logiciel
- Description des composants applicatifs

2.2. Limite de la v 1.0

Dans sa forme actuelle, la v 1.0 ne dispose pas d'un IHM web d'administration. La modification des fichiers de paramétrage doit donc se faire de manière manuelle.

3. Architecture logicielle de l'application

3.1. Pré-requis

3.1.1. Les produits logiciels indispensables

Pour pouvoir être exécuté de manière satisfaisante, le logiciel A.I.P. v 1.0 nécessite de disposer, sur le serveur (**machine virtuelle Linux** par exemple avec **4Go** de RAM si possible) qui l'héberge :

- 1) D'un serveur d'application web de type Tomcat en version 6.0.20 (racine d'installation : **`\${CATALINA_HOME}`**)
- 2) D'un **JDK 1.6** (les composants logiciels ayant été compilés avec la version 1.6.0-16)
- 3) D'un accès à l'annuaire LDAP qui héberge les données de l'annuaire lui-même. On parle bien ici d'un accès, car il n'est pas obligatoire que le serveur LDAP soit lui-même installé sur le même serveur à partir du moment où le LDAP est accessible via une url de la forme **ldap://serveur :port**. A titre d'illustration, nous avons réalisé les développements sur un serveur **OpenLDAP 2.0.27**.

3.1.2. Le montage réseau d'accès aux photos agent

Au CG33, l'AIP s'appuie sur le référentiel des photos (fichiers jpg) géré par le logiciel EDIGRAPH. Il conviendra donc de réaliser un montage réseau CIFS du répertoire **photo** dans la distribution AIP (voir plus loin dans ce document) vers le répertoire [\\srvphoto\DRH\EDIGRAPH\CG33\images](#) en mode **lecture seule**.

3.1.3. Paramétrage Tomcat spécifique au CG33

3.1.3.1. Fichier **`\${CATALINA_HOME}/conf/web.xml**

Uniquement pour le CG33 qui utilise une interface avec l'application Abyla de la Direction de Patrimoine, il est nécessaire d'ajouter les éléments suivants en fin du **web.xml**, avant le dernier tag **</web-app>** :

```
<!-- Datasource ABYLA -->
<resource-ref>
  <description>ABYLA Oracle Datasource</description>
  <res-ref-name>jdbc/abyla</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

3.1.3.2. Fichier **`\${CATALINA_HOME}/conf/context.xml**

Uniquement pour le CG33 qui utilise une interface avec l'application Abyla de la Direction de Patrimoine, il est nécessaire d'ajouter les éléments suivants en fin du **context.xml**, avant le dernier tag **</Context>** :

```
<!-- Datasource ABYLA -->
<Resource name="jdbc/abyla" auth="Container"
  type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver"
  url="jdbc:oracle:thin:@serveur:0000:SID"
  username="user" password="password" maxActive="20" maxIdle="10"
  maxWait="-1"/>
```

3.1.4. Paramétrage standard OpenLDAP

On se référera au manuel d'installation mais on rappelle pour mémoire

- 1) Qu'il convient de rechercher le fichier **LDAP Schema/inetorgperson.schema** dans la distribution AIP et de le détacher dans le répertoire **`\${LDAP_HOME}/etc/openldap/schema** en remplacement du fichier original du même nom .
- 2) Qu'il faut ensuite adapter le contenu du fichier **`\${LDAP_HOME}/etc/openldap/slapd.conf** au serveur et à l'environnement.
- 3) Qu'il convient enfin d'ajouter les 2 lignes d'index (***index employeeNumber eq*** et ***ou eq***) en fin de fichier.
Exemple :

```
# $OpenLDAP: pkg/ldap/servers/slapd/slapd.conf,v 1.8.8.7 2001/09/27 20:00:31 kurt
Exp $
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include      /opt/openldap/etc/openldap/schema/core.schema
include      /opt/openldap/etc/openldap/schema/cosine.schema
include      /opt/openldap/etc/openldap/schema/inetorgperson.schema

# Define global ACLs to disable default read access.

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral ldap://root.openldap.org

pidfile      /opt/openldap/var/slapd.pid
argsfile     /opt/openldap/var/slapd.args

# Load dynamic backend modules:
# modulepath /home/oc/openLdap2027/libexec/openldap
# moduleload back_ldap.la
# moduleload back_ldbm.la
# moduleload back_passwd.la
# moduleload back_shell.la

#
# Sample Access Control
#   Allow read access of root DSE
#   Allow self write access
#   Allow authenticated users read access
#   Allow anonymous users to authenticate
#
#access to dn="" by * read
#access to *
#   by self write
#   by users read
#   by anonymous auth
#
# if no access controls are present, the default is:
#   Allow read by all
#
# rootdn can always write!

#####
# ldbm database definitions
#####

database ldbm
suffix      "dc=cg33,dc=cg33"
rootdn      "cn=cg33,dc=cg33,dc=cg33"

#rootdn      "cn=Manager,o=My Organization Name,c=US"
```

```
# Cleartext passwords, especially for the rootdn, should
# be avoid. See slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw cg33
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /opt/openldap/var/openldap-ldb

# Indices to maintain
index default pres,eq
index uid,cn,sn,givenName pres,eq,sub
index ou,serviceLong pres,eq,sub
index employeeNumber eq
index manager eq
index destinationIndicator pres,eq,sub
index objectClass eq

# cache
cachesize 10000
dbcachesize 2000000

# LogLevel
LogLevel 256
```

ATTENTION à bien positionner les index pour éviter les problèmes de performance dans la restitution des informations et l'affichage des écrans !!! Les index positionnés ci-dessus (paragraphe **Indices to maintain**) semblent impératifs pour garantir de bonnes performances

3.1.5. Spécification du niveau de log OpenLDAP

Il est possible d'activer les logs OpenLDAP qui sont désactivés par défaut.

1) Editer le fichier `/opt/openldap/etc/openldap/slapd.conf` et ajouter à la fin :

```
# Niveau de Log openldap  
loglevel 256
```

2) Editer le fichier `/etc/syslog.conf` et ajouter à la fin :

```
# Log slapd openldap  
local4.* -/var/log/slapd.log
```

3) Relancer syslog :

```
/etc/init.d/syslog restart
```

4) Relancer openldap :

```
/etc/init.d/syslog restart
```


Pour info :

1) Niveaux de log OpenLDAP :

number	hex-value	log-name	Logging description
-1	0xFFFF		enable all logging
0	0x0000	-	logging inhibited - no logging occurs including critical errors. Not recommended.
1	0x1	acl	trace function calls
2	0x2	packets	debug packet handling
4	0x4	args	heavy trace debugging
8	0x8	conns	connection management
16	0x10	BER	print out packets sent and received
32	0x20	filter	search filter processing
64	0x40	config	configuration file processing
128	0x80	ACL	access control list processing
256	0x100	stats	stats log connections/operations/results
512	0x200	stats2	stats log entries sent
1024	0x400	shell	print communication with shell backends
2048	0x800	parse	print entry parsing debugging
4096	0x1000	cache	caching (unused)g
8192	0x2000	index	indexing (unused)
16384	0x4000	sync	print syncrepl (replica) logging
32768	0x8000	none	A misnomer - it will log message that are not categorized including curial messages

On les positionne dans le slapd.conf avec **loglevel v** où **v** peut être la somme de différents niveaux.
Exemple :

```
loglevel 255
# sets 1, 2, 4, 8, 16, 32, 64 and 128
# adds all the numbers
```

2) Par exemple, avec un loglevel **256**, le fichier **/var/log/slapd.log** contient après redémarrage et une recherche :

```
Jan 12 12:05:48 serveur slapd[21601]: slapd shutdown: waiting for 0 threads to terminate
Jan 12 12:05:49 serveur slapd[21601]: slapd stopped.
Jan 12 12:05:50 serveur slapd[13639]: slapd starting
Jan 12 12:08:02 serveur slapd[13639]: daemon: conn=0 fd=7 connection from IP=0.0.0.0:40135
(IP=0.0.0.0:0000) accepted.
Jan 12 12:08:02 serveur slapd[13639]: conn=0 op=0 BIND dn="CN=CG33,DC=CG33,DC=CG33"
method=128
Jan 12 12:08:02 serveur slapd[13639]: conn=0 op=0 RESULT tag=97 err=0 text=
Jan 12 12:08:03 serveur slapd[13639]: conn=0 op=1 SRCH base="dc=cg33,dc=cg33" scope=2
filter="(ou=*)"
Jan 12 12:08:07 serveur slapd[13639]: conn=0 op=1 SEARCH RESULT tag=101 err=0 text=
Jan 12 12:08:15 serveur slapd[13639]: conn=0 op=2 SRCH base="dc=cg33,dc=cg33" scope=2
filter="(sn=chator*)"
Jan 12 12:08:15 serveur slapd[13639]: conn=0 op=2 SEARCH RESULT tag=101 err=0 text=
Jan 12 12:08:18 serveur slapd[13639]: conn=0 op=3 SRCH
base="ou=DGAF_DSI_SPSI,ou=DGAF_DSI,ou=DGAF,dc=cg33,dc=cg33" scope=2
filter="(employeeNumber=9603158)"
Jan 12 12:08:18 serveur slapd[13639]: conn=0 op=3 SEARCH RESULT tag=101 err=0 text=
Jan 12 12:08:18 serveur slapd[13639]: conn=0 op=5 SRCH base="dc=cg33,dc=cg33" scope=2
filter="(manager=employeeNumber=9603158)"
```

Jan 12 12:08:19 serveur slapd[13639]: conn=0 op=6 SRCH base="dc=cg33,dc=cg33" scope=2 filter="(employeeNumber=311522)"

Jan 12 12:08:19 serveur slapd[13639]: conn=0 op=5 SEARCH RESULT tag=101 err=0 text=

Jan 12 12:08:19 serveur slapd[13639]: conn=0 op=6 SEARCH RESULT tag=101 err=0 text=

Ce loglevel ne semble pas pénaliser les perfs, mais attention au volume du fichier de log compte tenu du nombre de requêtes pas jour !

3.2. L'arborescence générale du logiciel AIP v 1.0

En se référant au manuel d'installation, nous avons vu que tous les composants sont installés sous le répertoire racine du logiciel A.I.P. v 1.0, i.e. sous le répertoire **`\${CATALINA_HOME}/webapps/AIP_CG33`**. On admet que ce répertoire racine de l'application sera désigné par la variable **`\${AIP_HOME}`** dans tout ce qui suit.

Ainsi, à partir de cette racine d'installation principale, on trouvera l'arborescence suivante :

..	30/10/2009 14:...	rwxr-xr-x	aip
conf	14/12/2009 17:...	rwxr-xr-x	aip
css	14/12/2009 16:...	rwxr-xr-x	aip
images	10/12/2009 17:...	rwxr-xr-x	aip
js	14/12/2009 12:...	rwxr-xr-x	aip
jsp	23/11/2009 09:...	rwxr-xr-x	aip
lang	17/11/2009 16:...	rwxr-xr-x	aip
LDAP Schema	14/09/2009 15:...	rwxr-xr-x	aip
logs	16/12/2009 08:...	rwxrwxr-x	aip
META-INF	21/08/2009 12:...	rwxr-xr-x	aip
photos	16/09/2009 15:...	rwxr-xr-x	aip
src	25/09/2009 10:...	rwxr-xr-x	aip
temp	15/12/2009 17:...	rwxrwxr-x	aip
WEB-INF	21/08/2009 12:...	rwxr-xr-x	aip
index.jsp	61 26/02/2009 09:...	rwxr-xr-x	aip

3.3. Contenu de chaque répertoire de l'application

Dans les répertoires listés ci-dessus, on trouve tous les fichiers de l'application

Répertoire ou fichier	Fichiers	Descriptions / Commentaires
conf	params.xml params.xml.template log4j.properties.template	Fichier global de paramétrage du logiciel Fichier modèle de paramétrage du logiciel (en cas de pb) Fichier modèle du fichier <i>WEB-INF/classes/log4j.properties</i>
css	AIP.css	Feuille de style globale du logiciel (cadres, fontes, etc.)
images	*.gif, *.jpg, *.png	Images, icônes, etc.
js	AIP.js	Contient toutes les fonctions Javascript du logiciel
jsp	AnalyseDonnees.jsp	Fichier pouvant être supprimé et qui ne sert qu'à fournir un état des lieux de complétude de certaines données de l'annuaire. Il n'est jamais utilisé par les autres composants mais peut aider à fournir une analyse des données pour les administrateurs
lang	aip_fr.properties aip_en.properties	Fichiers de langue contenant tous les libellés de tous les écrans (1 fichier par langue). Paramétré dans la balise <i>ficLang</i> dans le fichier global de paramétrage <i>params.xml</i>
LDAP Schema	inetorgperson.schema slapd.conf	Conservation (Utilisé uniquement pour l'installation initiale) : <ul style="list-style-type: none"> • Du schéma modifié d'annuaire Open LDAP de l'AIP. • Du fichier de paramétrage OpenLDAP
logs	AIPLog.log AIPLog.log.yyyy-mm-dd	Fichier de log principal du logiciel Fichiers de log historisés (1 par jour)
META-INF	MANIFEST.MF	Fichier standard d'une application Web décrivant sommairement l'application
WEB-INF	web.xml	Fichier standard d'une application web avec déclaration des servlets applicatives via les tags standards <i><servlet></i> et <i><servlet-mapping></i>
WEB-INF/classes	log4j.properties	Fichier de paramétrage des mécanismes de traçage de l'application, basés sur le standard log4j
WEB-INF/classes/AIP		Répertoire vide mais qui recueille, lors du développement, les classes du package AIP, issues de la compilation des sources dans le répertoire `\${AIP_HOME}/src , et à partir desquelles on crée le fichier `\${AIP_HOME}/lib/AIP.jar
WEB-INF/lib	AIP.jar activation.jar axis.jar log4j-1.2.14.jar mail.jar	Librairie du logiciel (code java + classes compilées) Librairie standard utilisée pour le dev. Des Web Services AIP Librairie standard utilisée pour le dev. Des Web Services AIP Librairie standard de trace log4j Librairie standard d'envoi de mail
photos	NOM_PRENOM_employ eeNumber.jpg	Le répertoire photos est un montage réseau vers http://srvphoto\DRH\EDIGRAPH\CG33\images pour accéder aux photographies des agents (au format spécifié dans l'exemple ci-contre à gauche et sous forme de fichiers jpg).
src	AIP_WS_*.java AIP/*.java	Code source des Web Services de consultation de l'AIP Code source des servlets & objets (IHM de consultation)
src/javadoc	*.html	Javadoc de tous les fichiers source du package AIP
temp		Répertoire de travail utilisé par le logiciel.
	Index.jsp	Fichier global de démarrage du logiciel de consultation AIP

☞ Remarque :

L'AIP est une application Full Open Source :

- Toutes les bibliothèques utilisées sont complètement Open Source, aucune n'est propriétaire
- Le serveur d'application Tomcat l'est également
- Même constat pour Open LDAP

4. Composants hors « traitement cœur »

4.1. Fichier params.xml

4.1.1. Structure

C'est le fichier principal de paramétrage global du logiciel A.I.P. v 1.0. , où tout le paramétrage est centralisé, accessible dans le répertoire **conf**.

On notera que :

1. La liste des paramètres est fournie entre les balises **<Parametres>** et **</Parametres>**
2. Chaque paramètre est décrit entre des balises **<Param>** et **</Param>** et représenté par un nom (balise **<nom>**) avec une valeur associée (balise **<valeur>**).

De manière synthétique, le fichier **params.xml** est formaté comme suit :

```
<?xml version="1.0"?>

<Parametres>

  <Param>
    <nom>paramètre N°1</nom>
    <valeur>valeur      du      paramètre
N°1</valeur>
  </Param>
  .
  .
  .
  <Param>
    <nom>paramètre N°n</nom>
    <valeur>valeur      du      paramètre
N°n</valeur>
  </Param>

</Parametres>
```



Lors de l'enregistrement du fichier **params.xml** modifié, il faut impérativement redémarrer tomcat pour que les changements soient pris en compte. En effet, ce fichier est chargé par les servlets uniquement au moment du démarrage et une fois pour toute.

4.1.2. Le paramétrage global du logiciel

Paramètre	Exemple de valeur	Commentaire
Paramétrage des tailles des pools de connexion à l'annuaire LDAP		
xxxxxConnectionPoolSizexxxx	5	Il y a 8 servlets dans AIP v 1.0 et donc 8 paramétrages de pool, soit 1 pour chaque écran : ldapConnectionPoolSizeFormulaireRechercher ldapConnectionPoolSizeListeResultats ldapConnectionPoolSizeListeResultatsXLS ldapConnectionPoolSizeOrganigramme ldapConnectionPoolSizeExplorateur ldapConnectionPoolSizeFicheAgent ldapConnectionPoolSizeFicheElu ldapConnectionPoolSizeFicheAgentModifiable ldapConnectionPoolSizeEnvoyerFicheParMail
Données de connexions à l'annuaire LDAP		
initialContextFactory	com.sun.jndi.ldap.LdapCtxFactory	Contexte d'accès au serveur LDAP (mode JNDI)
providerURL	ldap://serveur :port	URL LDAP d'accès au serveur d'annuaire
securityAuthentication	simple	Type d'authentification sur le serveur le LDAP
baseRecherche	dc=cg33,dc=cg33	Racine de base de recherche dans le LDAP
securityPrincipal	cn=cg33,dc=cg33,dc=cg33	Utilisateur de consultation de l'annuaire LDAP
securityCredentials		Mot de passe de l'utilisateur
topLevel	cg33	Libellé de la racine de haut niveau dans le LDAP
topLevelLong	CONSEIL GENERAL DE LA GIRONDE	Libellé long de la racine de haut niveau dans le LDAP
defaultSearchScope	SUBTREE_SCOPE	Scope de recherche par défaut dans l'annuaire
Paramétrage des charsets		
xxxxxxCharsetEncoding	ISO-8859-1 ou UTF-8	4 paramétrages de charset sont possibles selon que l'élément à paramétrer se situe dans un environnement windows (ISO-8859-1) ou Linux (UTF-8) : ldapServerCharsetEncoding : celui de l'annuaire LDAP serveurCharsetEncoding : celui du serveur qui fait tourner l'AIP exportCharsetEncoding : celui du fichier XLS exporté (ISO-8859-1 puisque vers le monde Windows) mailCharsetEncoding : celui du fichier joint au mail de demande de modification de la fiche agent qui est envoyé au support (généralement ISO-8859-1 puisque vers le monde Windows)
Répertoires de l'application, relativement à la racine \${AIP_HOME}		
repImages	images	Répertoire des images
repJsp	Jsp	Répertoire de la JSP de contrôle des données
repCss	css	Répertoire du fichier css
repJs	js	Répertoire du fichier javascript
repLib	lib	Non utilisé dans la version industrielle (servlets)
repLang	lang	Répertoire des fichiers de langue (.properties)
repPhotos	photos	Répertoire des photos (montage réseau)
repTemp	temp	Répertoire de travail (création du fichier joint au mail de demande de modification de fiche agent)
repPourAbyla	temp	Spécifique CG33 : Répertoire de travail (création du fichier de mise à jour des localisations dans l'application Abyla de la Direction du Patrimoine)

Paramètre	Exemple de valeur	Commentaire
Paramétrage du système de trace (log4j)		
log4JRepLogs	logs	Répertoire du fichier de log courant et des logs historisés (nomLog.yyyy-mm-dd)
log4JLogLevel	INFO	Niveau de log (nomenclature log4j)
log4JLogFileFileName	AIPLog.log	Nom du fichier de log courant
log4JPatternLayout	%d{yyyy/MM/dd HH:mm:ss,SSS} %-5p %m %n	Format des lignes du fichier de log (nomenclature log4j)
log4JStackTraceDepth	4	Non utilisé par l'IHM de consultation AIP qui est l'objet de ce document, mais par le batch global d'intégration de données.
Paramétrage divers		
tagEstAgent	Agent	Texte contenu dans le champs « description » du schéma d'annuaire LDAP qui permet de faire la distinction entre agents et élus
tagToutesFonctions	Toutes	Texte représentant « toutes les fonctions » en tête de la liste des fonctions si elle est affichée dans les critères du formulaire de recherche (débrillé en v 1.0 au CG33 via le paramètre « filtrerFonction » positionné à « N »)
tagSplitSecretaires	;	Tag permettant de séparer, dans le champs « secretary » du schéma d'annuaire LDAP les secrétaires qui sont notées sous la forme « EmployeeNumber=xxx ;EmployeeNumber=yy y » (débrillé en v 1.0 au CG33 via le paramètre « afficherSecretaires » positionné à « N »)
tagSplitOU	/	Tag permettant de séparer les services sous forme hiérarchiosée dans le champs « ou » du schéma d'annuaire LDAP. Exemple : CG33/DGAF/DSI
tagPhoto	photo	Tag technique représentant la photo d'un agent ou d'un élu lors de sa demande de modification de fiche. Ce tag est un paramètre passé entre les servlets qui gèrent les demandes de modification de fiche annuaire
tagCommentaires	commentaires	Tag technique représentant le commentaire que peut laisser un agent ou un élu lors de sa demande de modification de fiche. Ce tag est un paramètre passé entre les servlets qui gèrent les demandes de modification de fiche annuaire
Paramétrage des fichiers de ressource de l'application		
ficLang	aip_fr.properties	Nom du fichier de langue utilisé, localisé dans le répertoire défini par le paramètre <i>repLang</i>
ficCss	AIP.css	Nom du fichier css, localisé dans le répertoire défini par le paramètre <i>repCss</i>
ficJs	AIP.js	Nom du fichier de fonctions javascript utilisé, localisé dans le répertoire <i>repJs</i>
Mapping entre le nom interne à l'application et les champs de données de l'annuaire LDAP		
dn	dn	dn A.I.P. = dn LDAP
matricule	employeeNumber	Matricule A.I.P. = employeeNumber LDAP
.....
numCanton	departmentNumber	numCanton A.I.P. = departmentNumber LDAP

Paramètre	Exemple de valeur	Commentaire
Activation / inhibition de certains critères de recherche sur le formulaire principal		
filtrerFonction	N	Inhibé en v 1.0 au CG33 (demande DRH)
filtrerService	O	Activé en v 1.0 au CG33
filtrerServiceLong	O	Activé en v 1.0 au CG33
filtrerMotCle	O	Activé en v 1.0 au CG33
Activation / inhibition d'affichage de certaines informations dans les écrans ou exports de données		
afficherFonction	N	Inhibé en v 1.0 au CG33 (demande DRH)
afficherMailFonctionnel	N	Inhibé en v 1.0 au CG33
afficherBtnExporter	O	Active / inhibe la possibilité d'export XLS
afficherSecretaires	N	Inhibé en v 1.0 au CG33 (demande DRH)
afficherPhotos	O	Active / inhibe l'affichage des photos
afficherListePersonnesTopLevelDans Organigramme	N	Inhibe l'affichage des listes de personne sous les service pour le niveau le plus haut de l'organisation (problème de performance sinon)
activerAppelTelAutomatique	O	Active / inhibe l'appel tel. Automatique sur clic
autoriserModificationsFicheAgent	O	Active / inhibe la possibilité de demander la modification de la fiche agent
autoriserDemandeModifPhotos	N	Active / inhibe la possibilité de demander la modification de sa photo
autoriserCommentaires	N	Active / inhibe la possibilité de laisser un commentaire dans la fiche agent modifiable
experimentation	N	Active / inhibe l'affichage d'un cartouche en tête du formulaire de recherche en cas de version d'expérimentation de l'AIP
exportDetaille	N	Active / inhibe l'export des colonnes « numéros de matricule des agents » (EmployeeNumber) , bâtiment, étage & pièce
debug	N	Active / inhibe le mode de débogage (affichage en direct des stack trace java si problème)
activerControleMatriculePourModif	O	Active / inhibe la demande de saisie du numéro de matricule (EmployeeNumber) pour donner l'accès à la fiche agent modifiable. Utilisé tant que pas de SSO disponible, contrôle minimaliste.
accederOrganigramme	O	Active / inhibe l'accès à l'organigramme
accederExplorateur	O	Active / inhibe l'accès à l'explorateur
accederEquipe	O	Active / inhibe l'accès à la liste des agents rattachés à un manager
accederPlan	O	Active / inhibe l'accès à l'application (Abyla au CG33), via appel d'URL, d'affichage du plan du bureau de l'agent
accederCarte	O	Active / inhibe l'accès à l'application (Google au CG33), via appel d'URL, d'affichage du plan de quartier pour se rendre au bâtiment de l'agent
accederFicheCanton	O	Active / inhibe l'accès à l'application (portail institutionnel au CG33), via appel d'URL, d'affichage de la fiche Canton d'un élu
afficherMessageAlternatifPlan	N	Si positionné à O, permet d'inhiber temporairement l'appel URL à l'application de restitution des plans de bureau et affiche un message alternatif paramétré dans le fichier de langue aip_xxx.properties => Relance Tomcat
afficherMessageAlternatifCarte	N	Si positionné à O, permet d'inhiber temporairement l'appel URL à l'application de restitution des plans de quartier et affiche un message alternatif paramétré dans le fichier de langue aip_xxx.properties => relance Tomcat

Paramètre	Exemple de valeur	Commentaire
afficherMessageAlternatifFicheCanton	N	Si positionné à O, permet d'inhiber temporairement l'appel URL à l'application de restitution des fiches canton d'un élu et affiche un message alternatif paramétré dans le fichier de langue aip_XXX.properties => Nécessite un redémarrage Tomcat pour prise en compte
Paramétrage de la cryptographie (pour cryptage des matricules agent dans les pages HTML)		
algorithmDeCrypto	PBEWithMD5AndTripleDES	Utilisé pour le cryptage des numéros de matricule (EmployeeNumber) dans les différents écrans. Les valeurs possibles sont : Blowfish, DES, DESede, PBEWithMD5AndDES, PBEWithMD5AndTripleDES ou TripleDES
cleDeCrypto		Clé de cryptage / décryptage
Spécifique au CG33. Laisser les valeurs à « N » ou vides en standard. Utilisation de l'application métier Abyla		
utiliserAbylaPourLocalisations	O	Mise en œuvre des mécanismes de validation des demandes de modification de localisation
envoyerMailSiLocalisationInconnue	O	Spécifie si l'on envoie un mail à la Direction du Patrimoine si un agent ne trouve pas sa localisation dans les listes proposées
mailDestinataireLocalisationInconnue	xxxxx@cg33.fr	Mail à la Direction du Patrimoine si un agent ne trouve pas sa localisation dans les listes proposées
mailCopieLocalisationInconnue	xxxxx@cg33.fr	Mail en copie si un agent ne trouve pas sa localisation dans les listes proposées
ABYLAAdapterType	jdbc:oracle:thin	Driver JDBC d'accès au SGBD d'Abyla
ABYLAHost	Serveur	Hostname de l'application Abyla
ABYLAHostPort	10000	Port de l'application Abyla
ABYLASid	SID	Sid Oracle
ABYLALogin	user	Login sur la base Abyla
ABYLAPassword		Mot de passe sur la base Abyla
ABYLAFetchSize	200	Taille (nb. D'enregistrements) ramenés du serveur
ABYLAFileModifs	ABYLAModifs.csv	Fichier dans le répertoire défini par le paramètre <i>repPourAbyla</i> qui contient les données pour modification de localisation des agents
ABYLANbEltsDansListes	4	Nombre d'éléments affichés dans les listes de sites, bâtiments, étages et pièce
ABYLAValidationSecretariat	O	Envoi ou non d'une demande de validation à la BAL fonctionnelle de la direction de rattachement de l'agent
URLs des applications externes (SIG, affichage de plans, etc.) appelées par l'AIP		
urlSig	http://.....	Dans ces URLs, au sein du fichier params.xml : <ul style="list-style-type: none"> • Les caractères @ sont remplacés dynamiquement par le code au moment de l'exécution. • Les séquences & ; remplacent les caractères & rencontrés dans les URLs
urlAbyla	http://.....	
urlAnnuaireElus	http://.....	
urlOmnivista	http://.....	
Paramétrage des envois des fiches agents modifiées (à la demande de l'utilisateur)		
mailSupportDestinataire		Adresse mail du support à qui faire l'envoi lors de modifications de fiche agent. N'est plus utilisé, cf. mécanismes du paramètre « mailMappingChampsModifies » explicités ci-dessous

Paramètre	Exemple de valeur	Commentaire
mailMappingChampsModifies		Structuration des envois de mail des fiches agent modifiées (cf. plus loin)
mailEmetteur	AIP@cg33.fr	Adresse mail émetteur (générique)
mailServeur	Serveur	Serveur de mail à utiliser
mailServeurUsername	AIP-CG33	Utilisateur de connexion au mailServeur
mailServeurPassword		Mot de passe de connexion au mailServeur
mailDebugMode	N	Laisser à N
SupprimerFichierTrvApresEnvoiMail	O	Supprimer ou non les fichiers générés dans <i>repTemp</i> après envoi du mail au support
Paramétrage descriptif de la structure du fichier params.xml		
XML_XXXXXXXXXXXXXXXXXX		Décrit chacune des 9 balises de la structure du fichier params.xml. NE PAS MODIFIER SVP

4.1.3. Cas particulier de la balise « mailMappingChampsModifies »

Cette balise sert à décrire, pour chaque champs modifié de la fiche agent si on le souhaite, vers qui (un ou plusieurs) ont doit envoyer un mail (en destinataire principal et/ou en copie) pour actions éventuelles de mise à jour dans les applications métier qui sont les source de données de l'AIP.

Sur une seule ligne, entre les balises **<valeur>** et **</valeur>**, on définit toutes les règles de routage des mails. Chaque règle est séparée de la suivante par un point-virgule.

Chaque règle se compose elle-même de 3 éléments :

- La liste des champs (1 ou plusieurs séparés par des virgules) pour lesquels s'applique la règle
- La liste de diffusion principale (adresses séparées par des virgules), derrière le texte « to : »
- La liste de diffusion en copie (adresses séparées par des virgules), derrière le texte « cc : »

Ainsi, si on établit 3 règles de routage (par exemple) :

- Règle **R1**
- Règle **R2**
- Règle **R3**

Elles peuvent se traduire ainsi :

photo,departmentNumber,commentaires,sn,givenName,ou,organizationalUnitName,title,manager,secretary,to: o.chator@cg33.fr,cc:p1@cg33.fr,telephoneNumber,mobilePhone,destinationIndicator,fax,to:p1@cg33.fr,cc:p2@cg33.fr;mail,functionalMail,to:o.chator@cg33.fr,cc:o.chator@cg33.fr;street,postalCode,postOfficeBox,physicalDeliveryOfficeName,etage,piece,roomNumber,to:o.chator@cg33.fr,cc:o.chator@cg33.fr

4.2. Fichiers .properties

4.2.1. Objectifs

Ces fichiers sont accessibles dans le répertoire « **lang** ». Ils permettent :

- De ne jamais avoir à éditer de code (servlets) pour modifier un libellé de zone d'écran ou le texte d'un bouton
- De gérer le multilinguisme

4.2.2. Multilinguisme

Le nom du fichier de langue choisi est référencé dans le fichier global de configuration (**params.xml**) au travers de la balise **lang**. Dans l'illustration ci-dessous, on utilise le fichier **aip_fr.properties**

```
<Param>
  <nom>lang</nom>
  <valeur>aip_fr.properties</valeur>
</Param>
```

Pour passer en langue anglaise, il suffit de changer le nom du fichier de langue, donc de remplacer dans la balise ci-dessus **aip_fr.properties** par **aip_en.properties**

4.2.3. Structure d'un fichier .properties

Chacune des lignes « utiles » du fichier est de la forme **attribut=valeur** où :

- **attribut** est le nom interne (manipulé par les servlets) du libellé à afficher dans un écran
- **valeur** est la valeur réelle du libellé affiché sur l'écran

On aura, par exemple pour le formulaire de recherche principal :

```
frmRecherche.titre=Rechercher
frmRecherche.critere.1.label=Nom contient
frmRecherche.critere.2.label=Pr&eacute;nom contient
frmRecherche.critere.3.label=T&eacute;l&eacute;phone (num&eacute;ro de poste)
frmRecherche.critere.4.label=DGA ou Direction ou Service
frmRecherche.critere.5.label=Libell&eacute; long de Service contient
frmRecherche.critere.6.label=Fonction
frmRecherche.critere.7.label=Mot cl&eacute;
frmRecherche.explorateurEtorganigramme.label=Organigramme et explorateur des
services
```



Pour éviter tout problème de transcodage (ISO8859-1 vers UTF-8 par exemple), les valeurs des libellés sont **exprimées au format HTML** ! Ainsi, par exemple, le « é » sera représenté sous sa forme HTML **é** ;

Comme pour toute modification dans le fichier **params.xml** modifié, il faut impérativement redémarrer tomcat pour que les changements intégrés dans les fichiers .properties soient pris en compte. En effet, ils sont également chargés au moment du démarrage et une fois pour toute.

4.3. Fichiers .log & .log-yyyy-mm-dd

4.3.1. Objectifs

Ces fichiers sont accessibles dans le répertoire « **logs** ». Ils permettent de conserver la trace des appels de chacune des servlets qui sont autant d'écran du logiciel. Les mécanismes de traçage s'appuie sur le standard **log4j**.

4.3.2. Paramétrage

4.3.2.1. WEB-INF/classes/ log4j.properties

Ce fichier, qu'il n'est pas nécessaire de modifier normalement, contient les éléments suivants, normés log4j :

```
log4j.rootLogger = ${log.level}, AIPLog
log4j.appender.AIPLog = org.apache.log4j.DailyRollingFileAppender
log4j.appender.AIPLog.DatePattern='.'yyyy-MM-dd
log4j.appender.AIPLog.layout=org.apache.log4j.PatternLayout
log4j.appender.AIPLog.layout.ConversionPattern = ${log.pattern}
log4j.appender.AIPLog.file = ${log.chemin}/${log.fichier}
```

Les variables de la forme **\${.....}** sont des variables issues du fichier **params.xml** (cf. paragraphe suivant).

4.3.2.2. conf/params.xml

Pour assurer une prise en compte souple des modifications sans se disperser et par soucis de centralisation des opérations, l'utilisation des variables **\${.....}** permet de déporter le paramétrage du système de log dans le seul fichier **params.XML** => un seul fichier de paramètres pour toute l'application.

Variable dans Log4j.properties	Variable dans Params.xml	Valeur initiale	Commentaire
<code>\${log.level}</code>	log4JLogLevel	INFO	Deux niveau de trace principaux : <ul style="list-style-type: none">• INFO : standard• TRACE : pour debug, très verbeux
<code>\${log.pattern}</code>	log4JPatternLayout	<code>%d{yyyy/MM/dd HH:mm:ss,SSS}</code> <code>%-5p %m %n</code>	Au format log4j standard
<code>\${log.chemin}</code>	log4JRepLogs	logs	Répertoire de stockage du log
<code>\${log.fichier}</code>	log4JLogFileName	AIPLog.log	Nom du fichier de log

4.3.3. Structure et contenu du fichier de log

L'AIP utilise deux types de fichier de log :

- De manière globale le fichier `$(AIP_HOME)/logs/AIPLog.log`
- De manière exceptionnelle le fichier de log Tomcat standard `$(CATALINA_HOME)/logs/catalina.out`

4.3.3.1. Fichier AIPLog.log

En mode de traçage **INFO** (variable `log4JLogLevel` dans `params.xml`) et en mode de fonctionnement nominal, tout appel à un écran de l'A.I.P. génère à minima 2 lignes de traces qui sont horodatées :

- Une trace de début (appel)
- Une trace de fin (fin d'affichage d'une page HTML à l'écran)

Illustration :

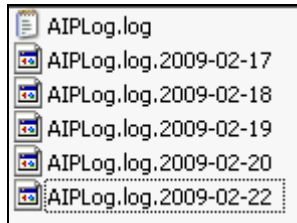
```
2009/02/26 16:36:30,463 INFO 1 127.0.0.1 AfficherFormulaireRecherche - service () - Debut
2009/02/26 16:36:32,448 INFO 1 127.0.0.1 AfficherFormulaireRecherche - service () - Fin
2009/02/26 16:36:39,495 INFO 1 127.0.0.1 AfficherListeResultats - service () - Debut
2009/02/26 16:36:40,136 INFO 1 127.0.0.1 AfficherListeResultats - service () - Fin
```

Chaque ligne du fichier de log (variable `log4JLogFileName` dans `params.xml`) est structuré selon la norme log4j et le format décrit via la variable `log4JPatternLayout` (dans `params.xml` également). Ainsi, si l'on détaille la structure de chaque ligne, on obtient l'illustration concrète avec la première fournie dans l'illustration ci-dessus) :

Valeur	Description
2009/02/26	Date au format yyyy/mm/dd
16:36:30,463	Heure avec les centièmes de seconde
INFO	Niveau de trace
1	Numéro d'appel séquentiel de la page. Il est incrémenté de 1 à chaque appel, ce qui permet de retrouver rapidement le nombre d'appel de chaque page et ainsi d'ajuster éventuellement les pools de connexion
127.0.0.1	Adresse IP ayant effectué l'appel. Ceci permettra, à terme de suivre (à des fins statistique, le profil des transactions effectuées). IL NE S'AGIT EN AUCUN CAS DE SURVEILLER CE QUE FONT LES AGENTS VIA L'AIP. D'ailleurs, les paramètres passés (ce qu'ils recherchent), ne sont pas restitués. Le seul intérêt est de savoir par où passe « une transaction » pour optimiser les pools de connexion mais également mettre l'accent sur tel ou tel amélioration dans le futur.
AfficherFormulaireRecherche	Nom de la servlet effectuant la trace
service ()	Nom de la fonction dans la servlet effectuant la trace
Debut	Contenu du message de trace

4.3.4. Mécanisme d'historisation automatique

La ligne `log4j.appender.AIPLog = org.apache.log4j.DailyRollingFileAppender` du fichier `log4j.properties` permet de définir que l'on souhaite, tous les jours et automatiquement, que le fichier `log.fichier` soit historisé sous la forme `log.fichier.yyyy-mm-dd`. Illustration concrète :



Ainsi :

- Chaque **AIPLog.log** est historisé sous la forme **AIPLog.log.yyyy-mm-dd**, l'historisation étant basée sur la date de création initiale du **AIPLog.log** courant.
- Il devient possible d'archiver (ou de supprimer) les anciens **AIPLog.log.yyyy-mm-dd** selon des modalités à définir avec les équipes de production

☞ Remarque :

Plus l'on peut conserver de **AIPLog.log.yyyy-mm-dd**, plus l'on pourra faire des statistiques sur une période étendue (cf. structure & contenu du fichier de log plus loin)

4.3.5. Les erreurs

☞ Rappel :

L'AIP utilise deux types de fichier de log :

- De manière globale le fichier `$(AIP_HOME)/logs/AIPLog.log`
- De manière exceptionnelle le fichier de log Tomcat standard `$(CATALINA_HOME)/logs/catalina.out`

4.3.5.1. Fichier AIPLog.log

☞ A noter :

Toute erreur laisse 2 types de traces :

- Celle de « haut niveau » comme par exemple une Exception, généralement résumée sur la première ligne
- La stack trace java (si le paramètre **debug** dans le fichier `$(AIP_HOME)/conf/params.xml` est à **O**)

La structure des lignes de trace est quasiment identique (hormis l'ajout du type d'erreur ou de **StackTrace** derrière le nom de la fonction de la servlet), seul le niveau de trace (**INFO** si tout va bien) passe à **ERROR**. Par exemple :

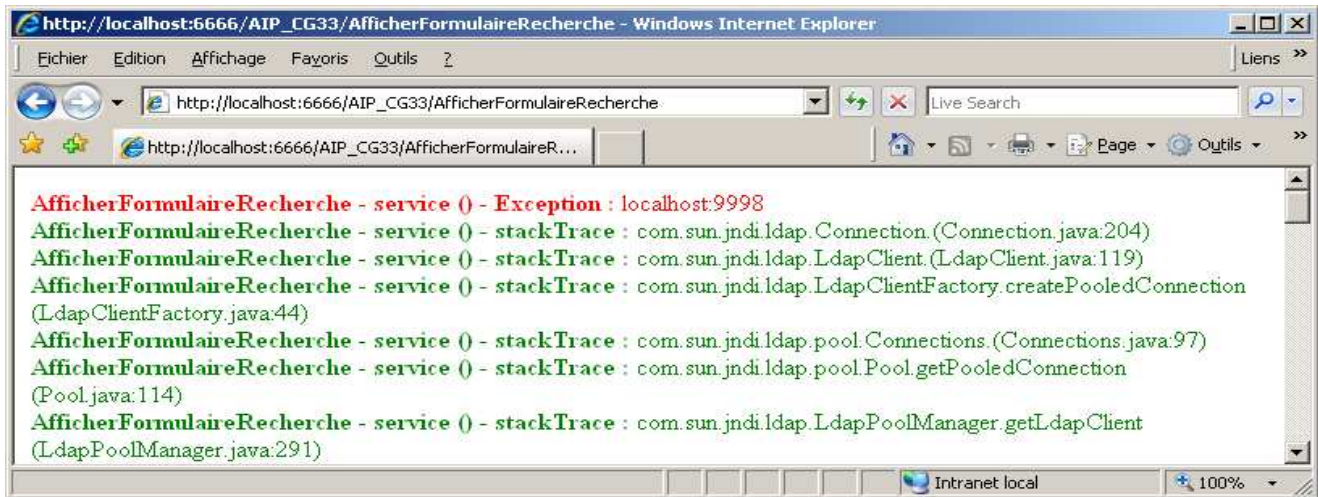
```
2009/02/27 10:43:30,007 ERROR 1 127.0.0.1 InterfaceAnnuaire - setSearchScope (String) - NamingException : [LDAP:
error code 32 - No Such Object]
2009/02/27 10:43:30,007 ERROR 1 127.0.0.1 InterfaceAnnuaire - setSearchScope (String) - NamingException - stackTrace
: com.sun.jndi.ldap.LdapCtx.mapErrorCode(LdapCtx.java:3013)
2009/02/27 10:43:30,007 ERROR 1 127.0.0.1 InterfaceAnnuaire - setSearchScope (String) - NamingException - stackTrace
: com.sun.jndi.ldap.LdapCtx.processReturnCode(LdapCtx.java:2934)
2009/02/27 10:43:30,007 ERROR 1 127.0.0.1 InterfaceAnnuaire - setSearchScope (String) - NamingException - stackTrace
: com.sun.jndi.ldap.LdapCtx.processReturnCode(LdapCtx.java:2740)
```

4.3.5.2. Fichier catalina.out

Même chose que pour le fichier de log principal **AIPLog.log**, même formalisme des lignes de trace.

4.3.5.3. Cas particulier

Si l'erreur se produit au moment de l'initialisation des servlets (comme par exemple l'impossibilité de trouver le serveur LDAP au démarrage), l'erreur est restituée directement à l'écran lors de l'appel au premier écran **mais pas dans un des fichiers** :



```
AfficherFormulaireRecherche - service () - Exception : localhost:9998
AfficherFormulaireRecherche - service () - stackTrace : com.sun.jndi.ldap.Connection (Connection.java:204)
AfficherFormulaireRecherche - service () - stackTrace : com.sun.jndi.ldap.LdapClient (LdapClient.java:119)
AfficherFormulaireRecherche - service () - stackTrace : com.sun.jndi.ldap.LdapClientFactory.createPooledConnection
(LdapClientFactory.java:44)
AfficherFormulaireRecherche - service () - stackTrace : com.sun.jndi.ldap.pool.Connections (Connections.java:97)
AfficherFormulaireRecherche - service () - stackTrace : com.sun.jndi.ldap.pool.Pool.getPooledConnection
(Pool.java:114)
AfficherFormulaireRecherche - service () - stackTrace : com.sun.jndi.ldap.LdapPoolManager.getLdapClient
(LdapPoolManager.java:291)
```

4.4. Fichier AIP.css

Ce fichier se situe dans le répertoire **`\${AIP_HOME}/css`**.

Il est au format standard pour une application web et ne contient aucune particularité.

4.5. Fichier AIP.js

Ce fichier se situe dans le répertoire **`\${AIP_HOME}/js`**.

Il est au format standard pour une application web et ne contient aucune particularité.

4.6. Fichier params.xml.template

Ce fichier se situe dans le répertoire **`\${AIP_HOME}/conf`**.

Il est une copie initiale du fichier **params.xml** et peut servir de base pour la création d'un nouveau fichier de paramétrage.

4.7. Fichier log4j.properties.template

Ce fichier se situe dans le répertoire **`\${AIP_HOME}/conf`**.

Il est une copie initiale du fichier **log4j.properties** qui doit se trouver sous **`\${AIP_HOME}/WEB-INF/classes** et peut servir de base pour la création d'un nouveau fichier de paramétrage.

4.8. Fichier inetorgperson.schema

Ce fichier se situe dans le répertoire **`\${AIP_HOME}/LDAP Schema`**.

Il n'est d'aucune utilité pour le logiciel AIP v 1.0 lui-même mais intègre une copie du schéma d'annuaire modifié au format LDAP. Il est à considérer en quelque sorte comme une sauvegarde liée à la structure de l'annuaire lui-même mais pas à l'interface de consultation objet de ce document.

4.9. Fichier slapd.conf

Ce fichier se situe dans le répertoire `${AIP_HOME}/LDAP Schema`.

Il n'est d'aucune utilité pour le logiciel AIP v 1.0 lui-même mais intègre une copie du fichier de paramétrage OpenLDAP. Il est à considérer en quelque sorte comme une sauvegarde liée à la structure de l'annuaire lui-même mais pas à l'interface de consultation objet de ce document.

4.10. Fichiers ASupprimer.txt

Ils sont au nombre de 2 et localisés dans les répertoires :

- `${AIP_HOME}/logs`
- `${AIP_HOME}/temp`

Question : Pourquoi laisser un fichier vide (**ASupprimer.txt**) dans ces répertoires ?

Réponse : Certains outils de compression n'embarquent pas les répertoires vides dans les archives compressées.

Nous verrons plus loin, au sein du paragraphe « [Etape 5 : Génération du fichier .war](#) » toute l'utilité de ces fichiers.

4.11. Fichier MANIFEST.MF

Ce fichier se situe dans le répertoire `${AIP_HOME}/META-INF`.

Il est au format standard pour une application web et ne contient aucune particularité déclarative :

```
Manifest-Version: 1.0
AIP_CG33: 1.0
Created-By:    O.  CHATOR    (CG33
2009)
```

4.12. Fichier web.xml

Ce fichier se situe dans le répertoire `${AIP_HOME}/WEB-INF`.

Il est au format standard d'une application web et contient la déclaration des 8 servlets utilisées dans le logiciel.

Exemple pour une des Servlets :

```
<servlet>
  <servlet-name> AfficherFormulaireRecherche</servlet-name>
  <servlet-class>AIP. AfficherFormulaireRecherche</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>AfficherFormulaireRecherche</servlet-name>
  <url-pattern>/AfficherFormulaireRecherche</url-pattern>
</servlet-mapping>
```

Le caractère / devant leur nom dans la balise url-pattern traduit le fait que leur URL d'appel est toujours de la forme http://serveur:port/AIP_CG33/servlet

5. Composants « traitement cœur »

5.1. Introduction

Le logiciel A.I.P. v 1.0 utilise les composants « cœur » intégrés au package java `${AIP_HOME}/WEB-INF/lib/ AIP.jar`
Les servlets de l'IHM de consultation sont au nombre de 13 dont 4 spécifiques au CG33 de part l'utilisation de l'application Abyla.

5.2. Servlets

La « bascule » entre mode spécifique CG33 et mode standard de traitements'effectue via valorisation (O ou N) du paramètre `utiliserAbylaPourLocalisations` dans le fichier `${AIP_HOME}/conf/params.xml` :

Servlet	Utilisé spécifiquement au CG33	Utilisée si le paramètre <code>utiliserAbylaPourLocalisations</code> vaut :
<code>ABYLAAfficherFicheAgentModifiable.class</code>	Oui	O
<code>ABYLAEnvoyerFicheParMail.class</code>	Oui	O
<code>ABYLAInterface.class</code>	Oui	O
<code>ABYLAModifierLocalisation.class</code>	Oui	O
<code>AfficherCarte.class</code>	Non	
<code>AfficherExplorateur.class</code>	Non	
<code>AfficherFicheAgent.class</code>	Non	
<code>AfficherFicheAgentModifiable.class</code>	Non	
<code>AfficherFicheElu.class</code>	Non	
<code>AfficherFormulaireRecherche.class</code>	Non	
<code>AfficherListeResultats.class</code>	Non	
<code>AfficherOrganigramme.class</code>	Non	
<code>AfficherPlan.class</code>	Non	

5.2.1.1. Déclaration des Servlets

Comme pour toute application web, les Servlets sont déclarés dans le fichier `web.xml` (localisé sous `${AIP_HOME}/WEB-INF`) au travers des balises `servlet-name` et `servlet-mapping`. On pourra se référer au paragraphe sur ce fichier ([Fichier web.xml](#)) plus haut dans ce document































5.2.1.2. Structure des Servlets

Elles sont toutes bâties sur le même modèle et suivent le standard Java en terme d'élaboration. 3 méthodes principales :

- `init ()` : Initialisation des pools de connexion, initialisations globales
- `service ()` : génération de la page HTML suite à appel par un navigateur
- `destroy ()` : vide

5.3. Librairie AIP.jar

Cette librairie, localisée sous **\$(AIP_HOME)/WEB-INF/lib**, contient tous les objets compilés (.class) et leur code source (.java) utiles aux Servlets vues ci-dessus. Son contenu :

Nom	Type	Date de modification	Taille
 ABYLAAfficherFicheAgentModifiable.class	Fichier CLASS	22/12/2009 17:21:16	24,304
 ABYLAEnvoyerFicheParMail.class	Fichier CLASS	22/12/2009 17:21:16	27,303
 ABYLAInterface.class	Fichier CLASS	22/12/2009 17:21:16	10,300
 ABYLAModifierLocalisation.class	Fichier CLASS	22/12/2009 17:21:16	14,187
 AfficherCarte.class	Fichier CLASS	22/12/2009 17:21:16	6,434
 AfficherExplorateur.class	Fichier CLASS	22/12/2009 17:21:16	9,013
 AfficherFicheAgent.class	Fichier CLASS	22/12/2009 17:21:16	22,682
 AfficherFicheAgentModifiable.class	Fichier CLASS	22/12/2009 17:21:16	19,598
 AfficherFicheElu.class	Fichier CLASS	22/12/2009 17:21:16	20,584
 AfficherFormulaireRecherche.class	Fichier CLASS	22/12/2009 17:21:16	16,006
 AfficherListeResultats.class	Fichier CLASS	22/12/2009 17:21:16	23,554
 AfficherOrganigramme.class	Fichier CLASS	22/12/2009 17:21:16	11,105
 AfficherPlan.class	Fichier CLASS	22/12/2009 17:21:16	6,490
 EnvoyerFicheParMail.class	Fichier CLASS	22/12/2009 17:21:16	24,551
 Explorateur.class	Fichier CLASS	22/12/2009 17:21:16	6,975
 ExporterListeResultats.class	Fichier CLASS	22/12/2009 17:21:16	11,910
 Fonction.class	Fichier CLASS	22/12/2009 17:21:16	1,947
 InterfaceAnnuaire.class	Fichier CLASS	22/12/2009 17:21:16	59,140
 ListeDeNoeuds.class	Fichier CLASS	22/12/2009 17:21:16	1,138
 ListeParametres.class	Fichier CLASS	22/12/2009 17:21:16	69,871
 MyLogger.class	Fichier CLASS	22/12/2009 17:21:16	3,254
 Noeud.class	Fichier CLASS	22/12/2009 17:21:16	1,283
 Organigramme.class	Fichier CLASS	22/12/2009 17:21:16	10,361
 Outils.class	Fichier CLASS	22/12/2009 17:21:16	24,872
 Parametre.class	Fichier CLASS	22/12/2009 17:21:16	2,788
 Personne.class	Fichier CLASS	22/12/2009 17:21:16	14,067
 Photo.class	Fichier CLASS	22/12/2009 17:21:16	3,248
 Service.class	Fichier CLASS	22/12/2009 17:21:16	3,353
 ServiceNomCourtComparator.class	Fichier CLASS	22/12/2009 17:21:16	788
 ServiceNomLongComparator.class	Fichier CLASS	22/12/2009 17:21:16	784

Pour accéder aux fichiers source, décompressez **AIP.jar** dans un répertoire de travail.

6. Comment modifier des composants cœur ?

6.1. Introduction

S'il est possible de modifier les fichiers **params.xml**, **AIP.css**, **AIP.js**, **web.xml**, **aip_xxx.properties** et **log4j.properties** de manière simple sans passer par un environnement de développement, il en va autrement des 13 servlets principales et des fichiers sources contenus dans le répertoire **`\${AIP_HOME}/src** puisque ce sont des composants java et qu'ils nécessitent une recompilation. Ce chapitre décrit le mode opératoire de modification de ces composants.

6.2. Etape 0 : Pré-requis

Pour pouvoir modifier les composants cœur (*.java), vous devez disposer de l'environnement de développement Eclipse installé sur votre poste de développement avec un JDK (1.6 dans la version actuelle de l'AIP).

Pour tester vos modifications, vous devez disposer :

- D'un serveur Tomcat (6.0.20)
- D'un JDK (1.6)
- D'un accès à un annuaire LDAP (Open LDAP dans la version actuelle)

On se référera au manuel d'installation cité en référence qui décrit complètement l'installation des pré-requis en environnement Linux. Sous Windows, on utilisera les installeurs adaptés à cet OS.

On suppose dans tout ce qui suit que vous disposez des pré-requis cités ci-dessus et nous partirons de l'hypothèse que vous disposez d'un tomcat en local dans lequel vous avez déployé (cf. également manuel d'installation) l'application (**AIP.war**).

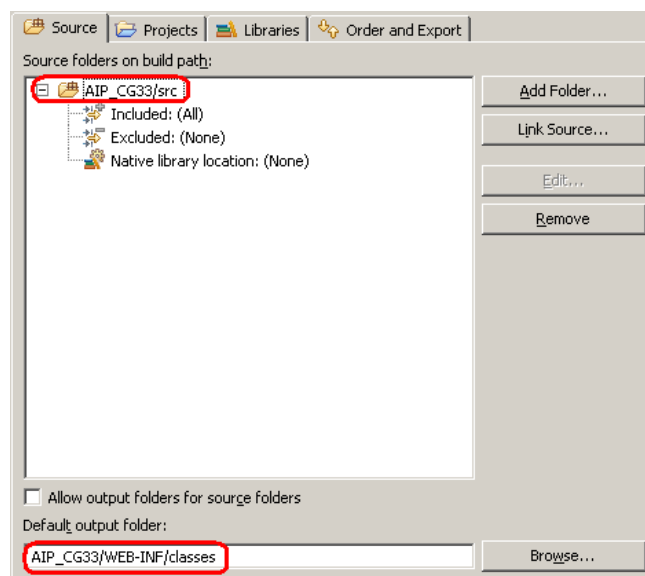
Les sources (.java) des objets utilisés par les 13 servlets principales sont regroupés dans le répertoire **`\${AIP_HOME}/src**. Si vous devez en modifier au moins un, enchaenez les actions suivantes :

6.3. Etape 1: Création du projet Eclipse

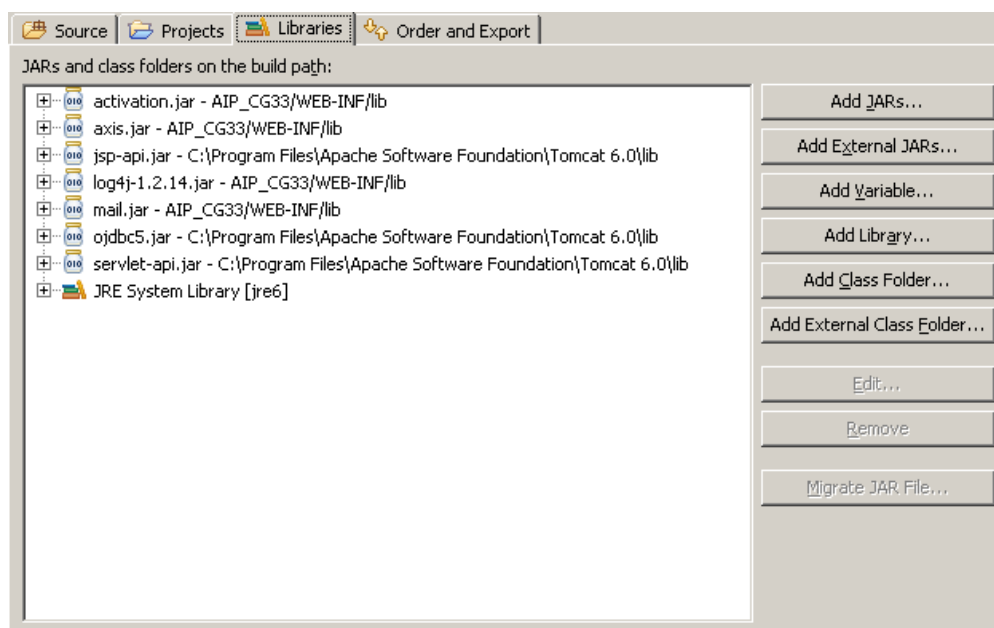
Enchaînez les opérations suivantes :

- 1) Lancez Eclipse
- 2) Créer un nouveau projet java (**file -> New -> Java Project**) en allant :
 - Chercher le répertoire **#{AIP_HOME}** comme répertoire initial
 - Nommer le projet **AIP_CG33**
 - Sélectionner le **JDK 1.6**
- 3) Vérifier (ou mettez à jour) dans **Project -> Properties**, sous le **chemin de compilation java**, que vous avez bien les éléments suivants de positionnés dans les onglets « **Source** » et « **Librairies** »

🔗 Onglet source :



🔗 Onglet librairies :



- 4) Sélectionnez **Project -> build all**, pour vérifier que tout est correct à la compilation

6.4. Etape 2 : Modification & test des composants

Modifiez les composants, recompilez (génération des .class par sélection du menu **Project -> build all**).

6.5. Etape 3 : Création du fichier AIP.jar

Une fois toutes vos modifications effectuées sur les objets génériques :

- 1) Fermer Eclipse
- 2) Créez un répertoire **AIP** sous un répertoire de travail (exemple : `${AIP_HOME}/temp`)
- 3) Déplacez (**couper / coller**) tous les fichiers compilés (.class) listés au paragraphe [Librairie AIP.jar](#) (cf. plus haut dans ce document) de `${AIP_HOME}/WEB-INF/classes/AIP` vers `${AIP_HOME}/temp/AIP`.
- 4) Créez un fichier **AIP.jar** en sélectionnant le répertoire `${AIP_HOME}/temp/AIP`
- 5) Remplacez l'ancienne version de **AIP.jar** sous `${AIP_HOME}/WEB-INF/lib` par le nouveau fichier que vous venez de créer
- 6) Supprimer `${AIP_HOME}/temp/AIP` et tout ce qu'il contient

Pensez également à « faire suivre » les fichiers source sous `${AIP_HOME}/src/AIP/*.java`

6.6. Etape 4 : Génération du fichier AIP.war

Une fois vos composants modifiés et validés, si vous souhaitez les déployer ailleurs (VA et/ou production), enchaînez les opérations suivantes :

- 1) Sélectionnez tous les fichiers sous `${AIP_HOME}` **ET PAS LE SEUL REPERTOIRE `${AIP_HOME}`** lui-même
- 2) Avec votre outil de compression, créez le fichier **AIP.war**

ATTENTION : si vous avez testé avec le Tomcat local à votre poste de travail, pensez à :

- 1) Purger le répertoire `${AIP_HOME}/logs` MAIS laissez le fichier **ASupprimer.txt**
- 2) Purger le répertoire `${AIP_HOME}/temp` MAIS laissez le fichier **ASupprimer.txt**

☛ Remarque :

Question : Pourquoi laisser un fichier vide (**ASupprimer.txt**) dans ces répertoires ?

Réponse : Certains outils de compression n'embarquent pas les répertoires vides dans les archives compressées.

6.7. Etape 5 : Mise à disposition du fichier AIP.war

Par analogie avec ce qui est décrit dans le manuel d'installation cité en référence, pensez à mettre le nouveau **AIP.war** à disposition (via winSCP ou autre mode de transfert de votre choix) sur le serveur de VA ou de production dans le répertoire de dépôt de packages d'installation (par exemple `/home/aip/installers`)

Concernant l'installation, suivre le mode opératoire fourni dans le document cité en référence qui concerne cette phase.

7. Annexe : Intervention directe sur les données de l'annuaire

7.1. Introduction

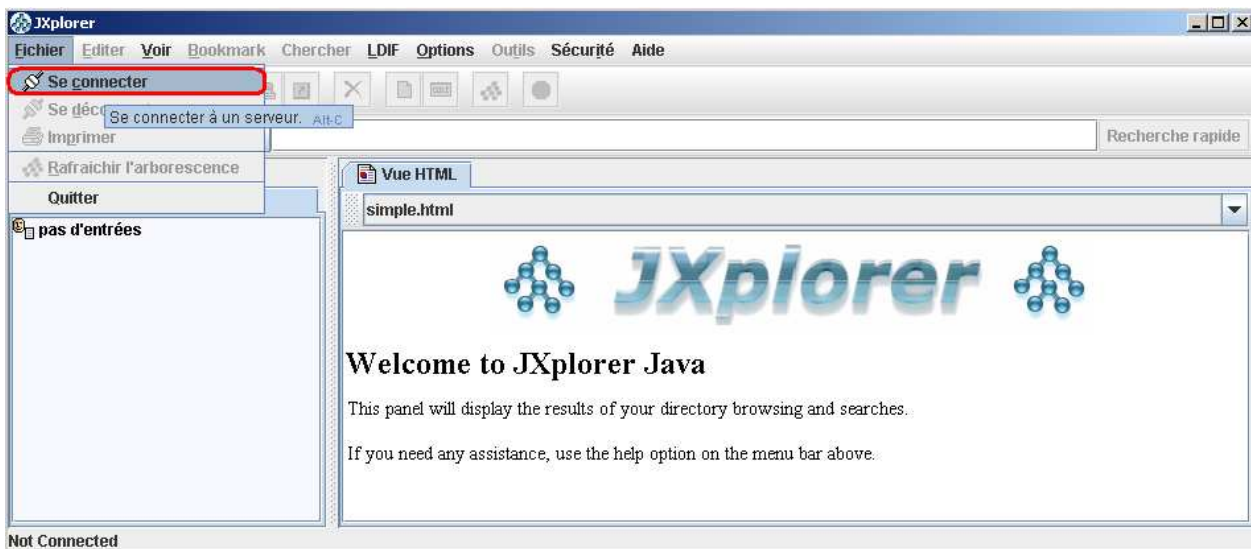
Ce chapitre **n'est à mettre en œuvre qu'en cas d'urgence élevée** où une modification en direct sur les données s'avère nécessaire. Il ne doit être utilisé dans aucun autre cas puisque l'alimentation en données est issue d'interface automatisées à partir des applications métier amont

7.2. L'outil JExplorer

C'est un outil Open Source qui peut être déployé sur un client Linux ou Windows puisqu'il est entièrement écrit en langage Java. Il est directement téléchargeable sur le site : <http://jxplorer.org/>

7.3. Connexion au Ldap

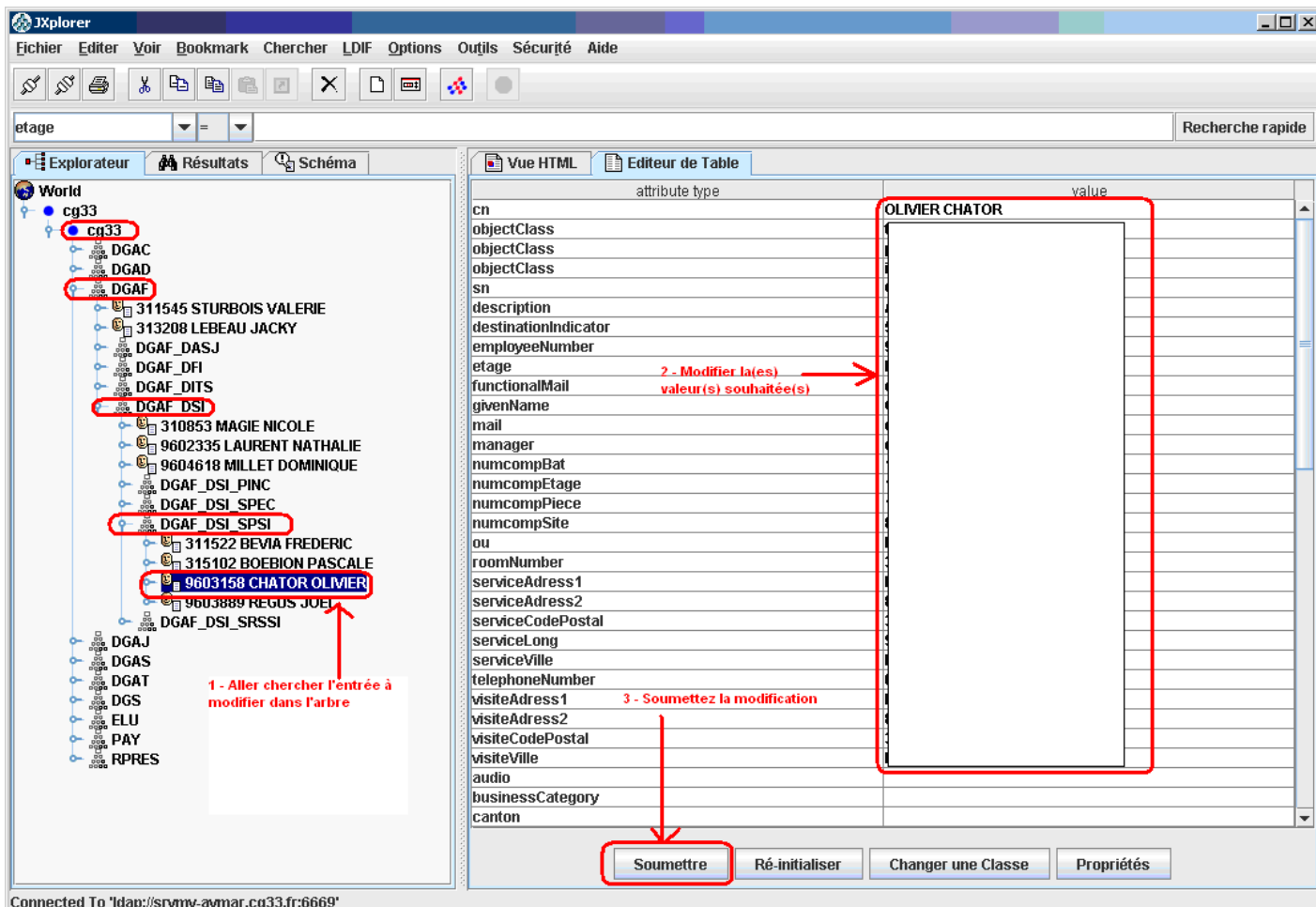
Etape 1 : Sur l'interface graphique de l'outil, sélectionner **Fichier -> Se connecter** :



Etape 2 : Remplir les données de connexion à l'annuaire Open LDAP puis cliquer sur le bouton **OK** :

7.4. Modification de données

Utilisez l'arbre de gauche pour aller chercher l'entrée à modifier puis l'onglet « *Editeur de table* » sur celui de droite pour visualiser toutes les données de l'entrée choisie.



Le fait de cliquer sur le bouton « Soumettre » implique une modification immédiate des données qui est répercutée aussitôt sur l'IHM de consultation de l'annuaire.

7.5. Déconnexion

Sélectionnez *Fichier* -> *Se déconnecter*

