

# **EarlyWarning**

## Guide de Configuration

Thomas Kowalski

Juillet 2018

# Table des matières

<b>1</b>	<b>Fonctionnement général de l'alarme précoce</b>	<b>4</b>
<b>2</b>	<b>Introduction et remarques générales</b>	<b>6</b>
2.1	Arguments de la ligne de commande pour EarlyWarning . . . . .	6
2.2	Commandes de redémarrage . . . . .	6
<b>3</b>	<b>Installations des pré-requis</b>	<b>7</b>
3.1	Installation de Git . . . . .	7
3.2	Installation de Java . . . . .	7
3.3	Installation d'Asterisk . . . . .	7
3.4	Installation de Maven . . . . .	7
3.5	Récupération des sources de l'application . . . . .	7
3.6	Réinitialisation du dépôt Maven local . . . . .	7
<b>4</b>	<b>Compilation d'EarlyWarning</b>	<b>8</b>
4.1	Lecture et installation des dépendances . . . . .	8
4.2	Tests unitaires . . . . .	8
4.3	Compilation . . . . .	8
4.4	<i>Packaging</i> (en utilisant Maven) . . . . .	9
4.5	<i>Packaging</i> (avec copie automatique des ressources) . . . . .	9
4.6	Génération de la <i>Javadoc</i> . . . . .	9
4.7	Makefile . . . . .	9
<b>5</b>	<b>Configuration d'Asterisk</b>	<b>10</b>
5.1	Configuration de la connexion Asterisk $\longleftrightarrow$ passerelle AudioGuides . . . . .	10
5.2	Configuration de l'AMI (Asterisk Manager Interface) . . . . .	10
5.3	Configuration du <i>dialplan</i> . . . . .	10
<b>6</b>	<b>Structure du fichier de configuration</b>	<b>11</b>
<b>7</b>	<b>Configuration de la connexion réseau (<i>network</i>)</b>	<b>11</b>
<b>8</b>	<b>Configuration des déclencheurs (<i>triggers</i>)</b>	<b>12</b>
<b>9</b>	<b>Configuration des passerelles téléphoniques (<i>gateway</i>)</b>	<b>13</b>
9.1	Configuration de la passerelle Asterisk . . . . .	13
9.2	Configuration de la passerelle Charon . . . . .	13
<b>10</b>	<b>Configuration de la gestion des contacts (<i>contacts</i>)</b>	<b>15</b>
10.1	Configuration des listes de contacts . . . . .	15
10.2	Aperçu d'une configuration complète des contacts . . . . .	15
10.3	Modification des paramètres liés au serveur Web . . . . .	15
<b>11</b>	<b>Sons utilisables par les <i>triggers</i> (<i>sounds</i>)</b>	<b>16</b>
11.1	Principe général . . . . .	16
11.2	Conversion automatique des sons dans un dossier . . . . .	16
11.3	Configuration des correspondances entre noms de sons et noms de fichiers dans EarlyWarning . . . . .	16
<b>12</b>	<b>Messagerie électronique (<i>mail</i>)</b>	<b>18</b>
<b>13</b>	<b>Audio et port série (<i>audioserial</i>)</b>	<b>19</b>
<b>14</b>	<b>Redondance et système haute-disponibilité (<i>failover</i>)</b>	<b>20</b>
<b>15</b>	<b>Démarrage de l'application</b>	<b>21</b>
<b>16</b>	<b>Test de fonctionnement de l'application</b>	<b>22</b>

<b>17 Modification des listes d'appel via l'interface Web</b>	<b>23</b>
17.1 Introduction . . . . .	23
17.2 Ajout d'un contact . . . . .	23
17.3 Désactivation d'un contact . . . . .	24
17.4 Suppression d'un contact . . . . .	24
17.5 Réglage du contact prioritaire . . . . .	24

# 1 Fonctionnement général de l'alarme précoce

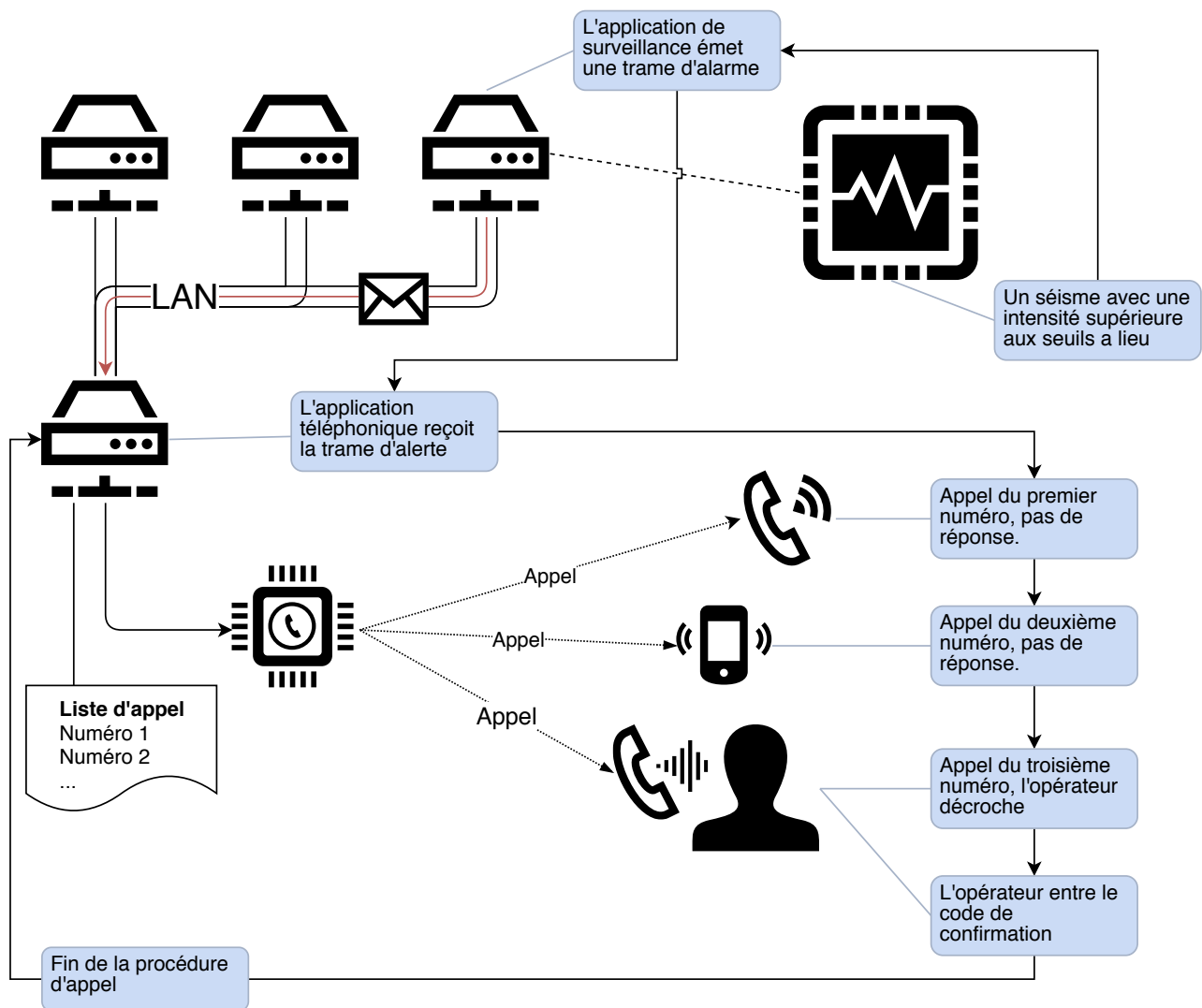


FIGURE 1 – Schéma du fonctionnement général de l'Alarme Précoce

Le principe général du système d'alarme précoce est de déclencher des appels téléphoniques vocaux interactifs et de diffuser des messages d'alertes par différents modes de communication (SMS, e-mail) en fonction d'un certain nombre de déclencheurs (*triggers*).

Un déclencheur peut être une sismicité importante, le dépassement de valeurs de seuil (RSAM1 par exemple), une panne matérielle, logicielle ou électrique, *etc.*

Les déclencheurs sont créés par des logiciels spécifiques, hébergés sur des serveurs informatiques. Les déclencheurs transitent via le réseau informatique.

Lorsqu'un déclencheur est émis, le système d'alarme précoce l'analyse. Il en tire un certain nombre d'informations :

- L'annuaire à utiliser pour les appels ;
- La séquence de confirmation ;
- La priorité du message ;
- Le message en lui-même ;
- *etc.*

Commence ensuite une phase d'appel. Celle-ci se base sur la liste d'appel contenant un nombre illimité de numéros de téléphone. Le système d'alarme commence par le premier contact de la liste et initie l'appel.

L'appel émis est vocal et interactif. Le message pré-enregistré est énoncé et la personne appelée doit saisir une séquence de confirmation (suite de numéros à saisir sur le clavier de son téléphone) pour valider la réception du message.

Tant que la séquence de validation n'a pas été enregistrée ou l'appel décroché, l'application va continuer d'appeler les numéros présents dans son annuaire. La phase d'appel laisse sonner un nombre maximum de fois avant d'abandonner un appel. Elle détermine lorsqu'un répondeur prend l'appel et abandonne l'appel. Elle raccroche tout appel décroché au bout d'un certain délai, que l'utilisateur ait ou non entré la séquence de confirmation (pour libérer la ligne et passer à l'appel suivant).

De plus, si la passerelle téléphonique par défaut (Asterisk) est inaccessible, l'application peut également utiliser une alarme anti-intrusion (module *Charon I*) comme passerelle téléphonique pour émettre une alerte d'urgence.

Enfin, l'application intègre différents outils, notamment une interface Web d'édition d'annuaires, un logiciel d'envoi de *triggers* permettant la vérification du bon fonctionnement du système.

## 2 Introduction et remarques générales

### 2.1 Arguments de la ligne de commande pour EarlyWarning

- testcall** Créé et ajoute à la pile un *trigger* afin de vérifier le bon fonctionnement de la passerelle téléphonique ;
- testcalls** Créé et ajoute à la pile deux *triggers* afin de vérifier le fonctionnement de la passerelle téléphonique ;
- searchresources** Cherche automatiquement dans les sous dossiers du dossier actuel (*working directory*) le dossier où se trouve la configuration. Utile avant tout lors du débogage, pour tester une nouvelle version sans avoir à la déplacer dans le dossier où se trouvent les ressources.

### 2.2 Commandes de redémarrage

Toute modification de la configuration d'Asterisk ou de la configuration d'EarlyWarning demande un redémarrage des applications concernées.

**Asterisk** Utiliser la commande `service asterisk restart`, le redémarrage est immédiat ;

**EarlyWarning** Utiliser la commande `systemctl restart earlywarning.service`.

**Attention** Le téléchargement des dépendances peut prendre beaucoup de temps après la réinitialisation.

## 3 Installations des pré-requis

### 3.1 Installation de Git

Git s'installe facilement avec Aptitude :

```
apt-get install git
```

### 3.2 Installation de Java

Sous Debian, on utilisera l'OpenJDK dans sa dernière version disponible.

#### Exemple avec la version 8

```
apt-get install openjdk-8-jdk
```

L'installation du **JDK** est nécessaire pour pouvoir compiler l'application. Le **JRE** ne suffit pas.

**Remarque** La nouvelle version de l'alarme précoce est conçue pour fonctionner avec Java à partir de sa version 7.

### 3.3 Installation d'Asterisk

Asterisk s'installe facilement avec Aptitude :

```
sudo apt-get install asterisk
```

### 3.4 Installation de Maven

Le projet utilise la technologie Maven, que l'on peut installer avec Aptitude :

```
apt-get install maven
```

### 3.5 Récupération des sources de l'application

Les sources se récupèrent depuis le dépôt Git avec la commande `clone`

```
git clone https://github.com/kowalskithomas/alarmeprecoceovpf.git
```

### 3.6 Réinitialisation du dépôt Maven local

Dans de rares cas, des problèmes de compilation peuvent survenir avec Maven. La réinitialisation du dépôt local peut résoudre ces problèmes.

```
rm -rf ~/.m2
```

## 4 Compilation d'EarlyWarning

EarlyWarning utilise Maven pour la gestion des dépendances, les tests unitaires, la compilation et le *packaging*.

Cette section couvre ces différents aspects.

**Remarque** Les commandes Maven doivent être exécutées depuis le dossier `java` qui contient le dossier `src` et le fichier `pom.xml`.

### 4.1 Lecture et installation des dépendances

Grâce au fichier `pom.xml` présent dans le code source, Maven peut déterminer l'arbre des dépendances de l'artefact EarlyWarning.

EarlyWarning utilise deux types de dépendances :

- Des dépendances libres (*Apache Commons*, *LF4J*) ;
- Des dépendances propriétaires ou internes.

Les premières peuvent directement être récupérées et installées par Maven depuis le Maven Repository dans le *local Maven repository*.

Les secondes doivent cependant être installées par Maven depuis des archives JAR fournies avec le code source. Cette installation se fait grâce au *plugin maven-install*.

Pour installer les dépendances (libres et propriétaires), utiliser la commande

```
mvn validate
```

Le déroulement de cette commande peut prendre un certain temps, notamment si le dépôt Maven local est vierge. En effet, toutes les dépendances doivent alors être téléchargées.

### 4.2 Tests unitaires

Maven permet de lancer directement les tests unitaires. Pour cela, utiliser la commande suivante :

```
mvn clean test
```

**Remarque** Lors de la première exécution des tests, le téléchargement de dépendances additionnelles sera nécessaire.

**Remarque** `mvn clean` permet de supprimer les binaires existants et de forcer une recompilation de l'ensemble des sources.

La sortie finale est censée ressembler à

```
[INFO] Results:
[INFO]
[INFO] Tests run: 25, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.501s
[INFO] Finished at: Tue Jul 17 06:09:57 UTC 2018
[INFO] Final Memory: 14M/36M
[INFO] -----
```

### 4.3 Compilation

Pour compiler l'ensemble des classes, utiliser

```
mvn clean compile
```



## 4.4 *Packaging* (en utilisant Maven)

Enfin, pour créer une archive Java (JAR) du projet EarlyWarning, utiliser

```
mvn package
```

Le fichier `pom.xml` est configuré pour que l'étape de **packaging** utilise le *plugin* Maven *shadow*. Celui-ci permet d'inclure dans le JAR final toutes les dépendances du projet.

Ainsi, lors du déploiement, seul le fichier généré `AlarmePrecoce-x.x.x.jar` est nécessaire à l'exécution de l'application.

Les ressources ne sont, quant à elles, pas incluses dans le JAR et doivent donc être copiées dans le même dossier où le JAR est déployé.

```
-rw-r--r--  1 root root 5382533 Jul 17 06:13 AlarmePrecoce-1.0.1.jar
-rw-r--r--  1 root root    5508 Jul 17 06:13 dependency-reduced-pom.xml
-rw-r--r--  1 root root  445621 Jul 17 06:13 original-AlarmePrecoce-1.0.1.jar
...
```

Le fichier `original-AlarmePrecoce-x.x.x.jar` correspond à l'archive ne contenant que les sources propres au projet. Le fichier `AlarmePrecoce-x.x.x.jar` contient, lui, toutes les dépendances (et est donc beaucoup plus lourd).

## 4.5 *Packaging* (avec copie automatique des ressources)

Utiliser le Makefile fourni :

```
make package
```

Il appellera lui-même Maven, nettoiera le dossier de sortie `target` puis y copiera les ressources de `src/main`.

## 4.6 Génération de la *Javadoc*

Pour générer la Javadoc, on peut à nouveau utiliser le Makefile

```
make javadoc
```

## 4.7 Makefile

Le Makefile a plusieurs options simples qui appellent directement Maven :

`default` : voir `package` ;

`package` : crée le package ;

`javadoc` : génère la Javadoc dans `target` ;

`build, compile` : compile les sources Java ;

`clean` : nettoie le dossier des fichiers compilés.

## 5 Configuration d'Asterisk

### 5.1 Configuration de la connexion Asterisk $\longleftrightarrow$ passerelle AudioGuides

Ouvrir le fichier `/etc/asterisk/sip.conf` et remplacer son contenu par :

```
[general]
context=default
disallow=all
allow=ulaw
allow=alaw

[pstn-out]          ; Utilisé pour les appels sortants
type=peer
allow=ulaw
context=outbound
dtmfmode=inband
host=<IP DU MP-114> ; IP de la passerelle
nat=no
qualify=no

[pstn-in]           ; Utilisé pour les appels entrants
canreinvite=no
context=inbound     ; Le contexte à utiliser pour les appels entrants dans le dialplan
dtmfmode=inband
host=<IP DU MP-114> ; IP de la passerelle
nat=never
type=user
```

### 5.2 Configuration de l'AMI (Asterisk Manager Interface)

Ouvrir le fichier `/etc/asterisk/manager.conf` et le remplir avec les informations suivantes :

```
[general]
enabled = yes
port = <PORT AMI> (par défaut 5038)
bindaddr = 127.0.0.1

[<NOM D'UTILISATEUR AMI>]
enabled = yes
port = 5038
allow = 0.0.0.0/0.0.0.0
bindaddr = <IP DU SERVEUR ASTERISK ICI>
secret = <MOT DE PASSE AMI>
read = all
write = all
```

### 5.3 Configuration du *dialplan*

EarlyWarning n'a pas besoin de configuration du *dialplan* (fichier `extensions.conf`).

## 6 Structure du fichier de configuration

Le fichier de configuration `earlywarning.xml` comprend un certain nombre de sections principales que nous allons détailler par la suite.

Chaque section commence par une balise ouvrante (`<network>` par exemple) et se termine par une balise fermante (`</network>` par exemple).

Une section peut contenir d'autres sections (des sous-sections) ou des paramètres.

Les paramètres commencent par une balise ouvrante (`<port>` par exemple) suivie de la valeur du paramètre (4445 par exemple) et se terminent par une balise fermante (`</port>` par exemple).

## 7 Configuration de la connexion réseau (*network*)

La section `network` ne comprend qu'un paramètre : `port`. Il correspond au port réseau qu'utilise le système d'alarme précoce.

Sa valeur par défaut est 4445.

Listing 1 – Aperçu de la section `network`

```
<network>  
  <port>4445</port>  
</network>
```

## 8 Configuration des déclencheurs (*triggers*)

Cette section définit plusieurs paramètres pour les *triggers*.

Elle comprend une sous-section **defaults** qui contient les paramètres par défaut des *triggers*.

**create\_trigger\_on\_error** Indique au système s'il doit générer des appels téléphoniques en cas d'erreurs ou de mauvaise réception de déclencheurs. La valeur de ce paramètre peut être **true** (vrai : on génère des appels en cas d'erreur) ou **false** (faux : on ne génère pas d'appel en cas d'erreur). La valeur par défaut est **false**.

**priority** Indique la priorité par défaut des *triggers* qui n'ont pas de priorité défini. La valeur de ce paramètre est un entier de 1 à 9.

1 correspond à la priorité la plus élevée. La valeur par défaut est 2.

**confirm\_code** Correspond au code de confirmation par défaut des *triggers* qui n'ont pas de code de confirmation défini.

La valeur de ce paramètre est une suite d'entiers. La valeur par défaut est 11.

**repeat** Indique si une liste d'appel doit être répétée en cas de non confirmation. La valeur de ce paramètre peut être **true** (le message sera répété) ou **false** (le message ne sera pas répété). La valeur par défaut est **true**.

*Attention, cette fonctionnalité n'est pas disponible dans la version 1.1 de l'application.*

Listing 2 – Aperçu de la section triggers

```
<triggers>
  <create_trigger_on_errors>>false</create_trigger_on_errors>
  <defaults>
    <priority>2</priority>
    <confirm_code>11</confirm_code>
    <repeat>true</repeat>
  </defaults>
</triggers>
```

## 9 Configuration des passerelles téléphoniques (*gateway*)

Cette section contient deux paramètres et plusieurs sous-sections, correspondant aux différentes passerelles.

**active** Indique quelle passerelle l'application doit utiliser. Ce paramètre peut prendre les valeurs suivantes :

- **asterisk**;
- **charon**;

**failover\_enabled** (Utilisée uniquement si la passerelle active n'est pas **charon**) Indique s'il faut utiliser l'alarme anti-intrusion en cas de défaillance de la passerelle par défaut (par exemple, Asterisk).

Listing 3 – Configuration des paramètres généraux de passerelle

```
<gateway>
  <active>
    asterisk
  </active>
  <failover_enabled>
    true
  </failover_enabled>
</gateway>
```

### 9.1 Configuration de la passerelle Asterisk

Cette section contient trois paramètres et une sous-section dédiée aux identifiants Asterisk.

**retries** Correspond au nombre d'essais auquel a le droit l'opérateur lorsqu'il entre le code ;

**ring\_timeout** Correspond au temps (en millisecondes) pendant lequel le téléphone doit sonner avant de passer à l'opérateur suivant ;

**agi\_server\_host** Correspond à l'adresse du serveur AGI local. Dans le cas de l'implémentation par défaut, c'est l'adresse IP de la machine exécutant l'application Alarme Précoce (par défaut : **localhost**).

La sous-section **settings** permet de configurer la connexion à l'interface AMI. Elle contient quatre paramètres :

**ami\_host** L'adresse du serveur Asterisk ;

**ami\_port** Le port qu'utilise l'interface AMI (par défaut 5038) ;

**ami\_user** Le nom d'utilisateur (du manager) à utiliser sur l'AMI ;

**ami\_password** Le mot de passe du manager à utiliser sur l'AMI.

Listing 4 – Configuration de l'AMI

```
<gateway>
  <asterisk>
    <retries>3</retries>
    <ring_timeout>15000</ring_timeout>
    <agi_server_host>localhost</agi_server_host>
    <settings>
      <ami_host>IP DU SERVEUR ASTERISK</ami_host>
      <ami_port>PORT AMI</ami_port>
      <ami_user>NOM UTILISATEUR AMI</ami_user>
      <ami_password>MOT DE PASSE AMI</ami_password>
    </settings>
  </asterisk>
</gateway>
```

### 9.2 Configuration de la passerelle Charon

Cette sous-section contient trois paramètres.

**host** L'adresse du module Charon ;

**port** Le port TCP à utiliser pour communiquer avec le module ;

**timeout** Le *timeout* à utiliser lors de la communication avec le module.

Listing 5 – Configuration de l'accès au module Charon I

```
<charon>  
  <host>195.83.188.220</host>  
  <port>23</port>  
  <timeout>60000</timeout>  
</charon>
```

## 10 Configuration de la gestion des contacts (*contacts*)

La section *contacts* comporte une sous-section et deux paramètres.

**home** Le port de connexion HTTP pour l'interface Web ;

**www-home** Le dossier où se trouvent les objets à servir (pages HTML, scripts JavaScript, *etc.*).

Listing 6 – Configuration du serveur Web

```
<contacts>
  <home>resources/www/</home>
  <port>6001</port>
</contacts>
```

### 10.1 Configuration des listes de contacts

La sous section **lists** permet de gérer les listes de contacts à utiliser. Elle contient autant de sous-sections qu'il y a des listes de contacts disponibles.

Une liste de contacts se définit de la façon suivante, elle contient deux paramètres.

**id** L'identifiant de la liste de contacts (qui sera utilisé par les *triggers*) ;

**path** Le chemin du fichier correspondant à la liste de contacts.

Listing 7 – Définition d'une liste de contacts

```
<list>
  <id>nom_de_la_liste</id>
  <path>chemin/du/fichier/de/liste.json
</list>
```

### 10.2 Aperçu d'une configuration complète des contacts

Listing 8 – Exemple de configuration des listes

```
<contacts>
  <home>resources/www/</home>
  <port>6001</port>
  <lists>
    <list>
      <id>default</id>
      <path>resources/contacts/default.json
    </list>
    <list>
      <id>tech</id>
      <path>resources/contacts/techniciens.json
    </list>
  </lists>
</contacts>
```

Les identifiants (ici, **default** et **custom\_list**) sont ensuite utilisable dans les datagrammes des *triggers*.

**Remarque** il doit toujours y avoir une liste **default** à utiliser lorsque la liste demandée par le *trigger* n'est pas disponible. Si aucune liste par défaut n'est donnée, EarlyWarning ne démarrera pas.

**Remarque 2** à l'heure actuelle, seules les listes d'appel au format **json** sont supportées par EarlyWarning.

### 10.3 Modification des paramètres liés au serveur Web

L'interface de gestion des contacts a deux paramètres modifiables :

Ces éléments peuvent être modifiés dans le fichier de configuration :

**Remarque** Le dossier des objets à servir n'a, *a priori*, pas de raison d'être modifié.

## 11 Sons utilisables par les *triggers* (*sounds*)

### 11.1 Principe général

La passerelle Asterisk permet d'ajouter des sons correspondant à des messages personnalisés. Ces sons doivent être au format **gsm** pour pouvoir être utilisés par Asterisk.

Une fois les sons convertis au format **gsm** (le plus simple étant de les enregistrer au format Wave puis de les convertir grâce à l'utilitaire **sox** installé par défaut sur Debian), copier les fichiers dans le dossier `/usr/share/asterisk/sounds` :

```
root@textor:/usr/share/asterisk/sounds# ls -la
total 472
drwxr-xr-x  3 root    root      4096 Jul  3 07:24 .
drwxr-xr-x 10 root    root      4096 Jun 19 11:27 ..
-rw-r--r--  1 root    root     12969 Jul  3 07:18 accueil.gsm
-rw-r--r--  1 asterisk asterisk 35640 Jun 25 12:30 accueilvopf.gsm
lrwxrwxrwx  1 root    root        36 Dec 30 2017 custom -> ../../local/share/asterisk/sounds
-rw-r--r--  1 root    root      8184 Jul  3 07:18 demandecode.gsm
lrwxrwxrwx  1 root    root        36 Jun 19 11:27 en -> /etc/alternatives/asterisk-prompt-en
lrwxrwxrwx  1 root    root        39 Jun 19 11:27 en_US -> /etc/alternatives/asterisk-prompt-en-us
drwxr-xr-x  8 root    root     20480 Jul  3 07:21 en_US_f_Allison
-rw-r--r--  1 root    root     10362 Jul  3 07:18 incorrect.gsm
-rw-r--r--  1 root    root      8448 Jul  3 07:18 incorrect2.gsm
-rw-r--r--  1 root    root      8349 Jul  3 07:18 merci.gsm
lrwxrwxrwx  1 root    root        31 Dec 30 2017 recordings -> /var/lib/asterisk/sounds/custom
-rw-r--r--  1 root    root      8118 Jul  3 07:18 sismicite.gsm
-rw-r--r--  1 asterisk asterisk 345526 Jun 25 06:49 test.sln
```

**Attention** Après ajout des sons, un redémarrage d'Asterisk est nécessaire afin de les réindexer.

### 11.2 Conversion automatique des sons dans un dossier

Afin de convertir plus rapidement un grand nombre de sons, on peut utiliser le script Shell suivant :

Listing 9 – Script de conversion de fichiers WAV en GSM

```
for i in *.wav ; do
    sox "$i" -r 8000 -c1 "$(basename "${i%.wav}").gsm"
done
```

Les fichiers d'entrée doivent être des fichiers audio Wave (**.wav**), sans restriction sur le *bitrate*.

**Remarque** Il est **impératif** de vérifier le fonctionnement des sons après installation ! En effet, si un son inconnu est demandé à Asterisk lors de l'exécution d'un script AGI (c'est-à-dire au moment de l'émission d'un appel), aucun son ne sera joué (pas un son par défaut) et l'appelé ne pourra être mis au courant du risque.

### 11.3 Configuration des correspondances entre noms de sons et noms de fichiers dans EarlyWarning

Chaque *trigger* peut intégrer un son personnalisé (par exemple, un *trigger* pour un disque plein demandera un son **disque\_plein** et un *trigger* pour une sismicité importante demandera un son **sismicite**).

Cependant, ces noms de sons ne sont que des identifiants, et le son à jouer lors d'un appel téléphonique dépend de l'environnement : dans notre cas, il dépend principalement de la passerelle téléphonique utilisée.

Il faut donc faire un lien entre les identifiants de son (par exemple, **sismicite**) et les sons correspondants (par exemple, **sismicite\_importante.gsm**).

Ce lien est effectué au moment de l'appel grâce au fichier de configuration **earlywarning.xml**.

Afin de pouvoir utiliser autant de sons que possible, il faut relier chaque identifiant à un son existant :

Listing 10 – Aperçu de la section sounds

```
<sounds>
  <sound>
    <id>default</id>
    <asterisk>warning</asterisk>
```



```

        <charon>5</charon>
    </sound>
    <sound>
        <id>welcome</id>
        <asterisk>accueil</asterisk>
        <charon>-1</charon>
    </sound>
    <sound>
        <id>login</id>
        <asterisk>demandecode</asterisk>
        <charon>-1</charon>
    </sound>
    <sound>
    </sounds>

```

**Attention** Toutes les passerelles doivent avoir un son d'avertissement par défaut. Il sera joué si un son particulier demandé par un *trigger* (par exemple, `disque_plein`) est indisponible. Un son est dit *indisponible* qu'à la condition qu'aucun lien ne soit fait entre son identifiant et un fichier dans le fichier de configuration :

Listing 11 – Exemple de configuration avec son indisponible

```

<sounds>
    <sound>
        <id>default</id>
        <asterisk>warning_default</asterisk>
        <voicent>warning.wav</voicent>
    </sound>
    <sound>
        <id>disque</id>
        <asterisk>disque_plein</asterisk>
        <voicent>disque_plein.wav</voicent>
    </sound>
    <sound>
        <id>sismicite</id>
        <asterisk>sismicite</asterisk>
    </sound>
</sounds>

```

Ici, le son `sismicite` est indisponible pour la passerelle `voicent`. S'il est demandé par un *trigger*, ce sera le son par défaut (`default`) qui sera joué.

Si le fichier `disque_plein` n'existe pas sur le serveur Asterisk, **il n'est pas pour autant marqué indisponible, car il n'existe aucun moyen de le vérifier à l'exécution. Cela risque d'entraîner un problème à l'exécution (la personne n'aura pas de message d'avertissement à valider).**

**Remarque** Lors du démarrage, l'application `EarlyWarning` vérifie que toutes les passerelles existantes (c'est-à-dire l'ensemble des passerelles qui ont au moins une clé définie dans `sounds`) ont un son par défaut. Si ce n'est pas le cas, l'application refuse de démarrer.

De plus, elle émettra un avertissement (*Warning*) pour tous les sons configurés pour au moins une passerelle, mais pas pour toutes (dans l'exemple, un avertissement aurait été émis pour `sismicite` pour la passerelle `voicent`).

## 12 Messagerie électronique (*mail*)

Cette section définit la fonctionnalité de messagerie électronique (ou e-mail). Elle comprend une sous-section **smtp** et une sous-section **mailinglist**. Cette dernière comprend une à plusieurs sous-sections **contact**.

**use\_mail** permet d'activer ou de désactiver la fonctionnalité mail de l'alarme précoce. La valeur peut être **true** (utilisation de la fonctionnalité mail) ou **false**; (fonctionnalité mail désactivée);

**host** correspond au serveur d'envoi de mail. La valeur doit être le nom ou l'adresse IP du serveur de mail;

**port** correspond au port réseau du serveur d'envoi de mail. La valeur doit être un entier. Par défaut la valeur 25 est utilisée (port SMTP par défaut);

**username** correspond au nom d'utilisateur du serveur SMTP. La valeur est une chaîne de caractère.

**password** correspond au mot de passe de l'utilisateur du serveur SMTP. La valeur est une chaîne de caractère.

**from** correspond à l'adresse mail qui envoie le message. La valeur du paramètre est une chaîne de caractère.

La sous-section **mailinglist** peut contenir autant d'éléments **contact** que l'on veut. Chaque élément **contact** correspond à un destinataire pour l'envoi de messages.

Pour chaque noeud **contact**, le seul paramètre, **email**, correspond à l'adresse du destinataire.

Listing 12 – Aperçu de la section mail

```
<mail>
  <use_mail>false</use_mail>
  <smtp>
    <host>smtp.mail.com</host>
    <port>25</port>
    <username>username</username>
    <password>password</password>
    <from>observatoire@mail.com</from>
  </smtp>
  <mailinglist>
    <contact>
      <email>premier.contact@mail.com</email>
    </contact>
    <contact>
      <email>deuxieme.contact@ipgp.fr</email>
    </contact>
    <contact>
      <email>troisieme.contact@mail.fr</email>
    </contact>
  </mailinglist>
</mail>
```

## 13 Audio et port série (*audioserial*)

La section `audioserial` comprend une sous-section : `serial` et cinq paramètres.

**use\_audioserial** permet d'activer ou désactiver la fonction Audio et port série. La valeur par défaut est `true` (activé) ;

**delay** correspond au délai (en secondes) entre l'envoi du texte sur le port série et le début de la lecture du fichier audio. La valeur par défaut est `10` ;

**message** correspond au préfixe du message qui sera envoyé sur le port série. La valeur par défaut est `Message d'alerte...` ;

**begin\_commands** Chemin du fichier binaire qui permet de configurer la radio connectée éventuellement au port série. Le contenu du fichier sera envoyé à l'ouverture du port série. La valeur par défaut est `resources/beginCMD` ;

**end\_commands** Chemin du fichier binaire qui permet de configurer la radio connectée éventuellement au port série. Le contenu du fichier sera envoyé après l'envoi du message texte sur le port série. La valeur par défaut est `resources/endCMD`.

La sous-section `serial` permet de configurer le port série.

**port** correspond au nom du port série. La valeur par défaut est `COM1` ;

**speed** correspond à la vitesse (en bauds) du port série. La valeur par défaut est `9600`.

Listing 13 – Aperçu de la section `audioserial`

```
<audioserial>
  <use_audioserial>true</use_audioserial>
  <serial>
    <port>COM1</port>
    <speed>9600</speed>
  </serial>
  <delay>10</delay>
  <message>Message d'alerte de l'Observatoire...</message>
  <begin_commands>resources/beginCMD</begin_commands>
  <end_commands>resources/endCMD</end_commands>
</audioserial>
```

## 14 Redondance et système haute-disponibilité (*failover*)

Dans le cas de l'installation d'un système redondant (plusieurs instances de l'application fonctionnant en parallèle sur le réseau), la configuration doit être mise à jour pour utiliser cette fonctionnalité.

La section `failover` comprend trois paramètres.

`is_failover` définit si l'instance configurée est une instance principale (`true`) ou non (`false`);

`main` définit l'adresse de l'instance principale. *Cette valeur n'est pas lue lors du démarrage de l'instance principale, seulement lors du démarrage d'une instance secondaire*;

`heartbeat_port` définit le port TCP à utiliser pour la communication entre une instance principale et une instance secondaire. Sa valeur par défaut est 6002.

Listing 14 – Aperçu de la section `failover`

```
<failover>
  <is_failover>true</is_failover>
  <main>195.83.188.10</main>
  <heartbeat_port>6002</heartbeat_port>
</failover>
```

**Remarque** Lors du démarrage, une instance secondaire vérifie la présence de l'instance principale. En cas de refus de connexion ou de tout type d'erreur, elle émettra un avertissement :

```
WARN [main] 2018-07-30 11:41:44,220 -
  Current instance is configured to be a failover but the main
  instance that should be running at 195.83.188.10 IS NOT RESPONDING.
  Is the configuration wrong or is there an error on the other side?
```

## 15 Démarrage de l'application

Une fois l'application installée dans un dossier (à savoir l'archive Java, `AlarmePrecoce.jar`, le dossier `resources` ainsi que le fichier de configuration dûment rempli), on peut l'exécuter comme un service (*daemon*) grâce au script de gestion de l'application.

Celui-ci peut être trouvé dans le dossier `resources` et doit être configuré puis copié dans `/etc/init.d` afin qu'il soit automatiquement exécuté au démarrage.

Pour configurer le fichier, vous aurez besoin

- Du nom d'utilisateur qui exécutera l'application (par défaut `sysop`), dénoté par `<UTILISATEUR>` dans la suite;
- De l'emplacement d'installation du script de démarrage de l'application (qui peut être obtenu avec `pwd`), dénoté par `<DOSSIER D'INSTALLATION>` dans la suite.

```
[Unit]
Description=AlarmePrecoce
After=network.target

[Service]
Type=simple
User=<UTILISATEUR>
WorkingDirectory=<DOSSIER D'INSTALLATION>
ExecStart=/bin/bash <DOSSIER D'INSTALLATION>/EarlyWarning.sh
Restart=on-failure
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=/var/log/earlywarning.log

[Install]
WantedBy=multi-user.target
```

Enfin, installez l'Alarme Précocce en tant que service à l'aide de ces commandes, à exécuter dans le dossier contenant le JAR et le dossier `resources` :

```
cp resources/EarlyWarning.sh .
cp resources/earlywarning.service /etc/systemd/system
systemctl daemon-reload
systemctl enable earlywarning.service
```

Le service sera démarré automatiquement au prochain démarrage.  
Pour démarrer manuellement le service, on peut utiliser

```
systemctl start earlywarning.service
```

De même avec `stop`, `restart` ou `status`.

**Accès aux journaux** Pour accéder aux journaux de l'application, utiliser la commande

```
journalctl -u earlywarning.service
```

## 16 Test de fonctionnement de l'application

Pour vérifier le bon fonctionnement de la partie téléphonique de l'application Alarme Précoce, on peut utiliser son argument `-testcall`, qui émettra un *trigger* et un donc un appel quelques secondes après le démarrage de l'application.

## 17 Modification des listes d'appel via l'interface Web

Les listes d'appel ne sont pas censées être modifiées directement via les fichiers correspondants (pour des risques évidents de risque d'erreur de syntaxe, par exemple).

La modification doit se faire à travers l'interface Web. Pour ce faire :

- Lancer l'application EarlyWarning ;
- Se rendre avec un navigateur Web à l'adresse `http://<adresse du serveur EarlyWarning>:<port>` (le port par défaut est 6001).

FIGURE 2 – Capture d'écran de l'interface d'édition de listes d'appel

### 17.1 Introduction

L'interface Web se décompose en deux colonnes.

La colonne de gauche correspond à la liste d'appel, à savoir quelles sont les personnes à appeler et dans quel ordre. Les éléments en couleur plus foncée correspondent au contact prioritaire. Ainsi, le premier élément est toujours de couleur foncée.

La colonne de droite se décompose en deux parties :

**L'annuaire** contient toutes les personnes connues du fichier ; on peut les y supprimer ou y définir le contact prioritaire ;

**Le formulaire d'ajout de contact** permet d'ajouter un nouveau contact au fichier.

Pour ajouter un contact à la liste d'appel, il suffit de faire glisser le contact correspondant de la liste de droite à celle de gauche, à la position souhaitée.

Pour modifier l'ordre d'appel, on peut directement réordonner (en les faisant glisser) les éléments de la liste de gauche.

### 17.2 Ajout d'un contact

L'ajout d'un contact à la liste d'appels se fait de la façon suivante :

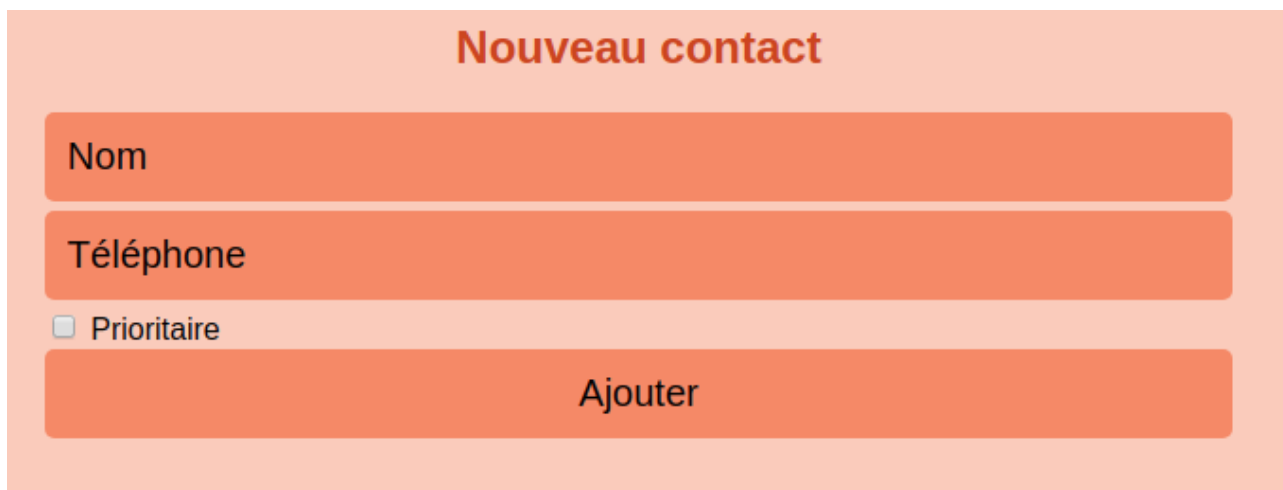
- Remplir les champs *Nom* et *Téléphone* ;
- Cocher au besoin la case *Prioritaire* ;
- Valider avec le bouton *Valider*.

#### Case *Prioritaire*

La case *Prioritaire* permet de définir un contact comme étant *prioritaire*. Cela signifie que le premier de la liste d'appel sera *toujours* celui-ci.

Un contact prioritaire peut se trouver à plusieurs endroits dans la liste d'appels, mais le premier élément sera toujours un contact prioritaire (si la liste d'appel contient un contact prioritaire).

Un seul contact peut être prioritaire à la fois.

Le formulaire est intitulé "Nouveau contact" en rouge. Il contient deux champs de saisie : "Nom" et "Téléphone". En dessous de "Téléphone", il y a une case à cocher non cochée suivie du mot "Prioritaire". Tout cela est regroupé dans une section orange. En dessous de cette section, il y a un grand bouton orange "Ajouter".

**Nouveau contact**

Nom

Téléphone

☐ Prioritaire

Ajouter

FIGURE 3 – Formulaire de création de contact

### 17.3 Désactivation d'un contact

Pour désactiver un contact (ou, s'il est présent plusieurs fois dans la liste d'appel, supprimer une de ses occurrences), utiliser la croix qui apparaît à gauche de l'élément lorsque l'on passe le pointeur de la souris sur l'élément dans la liste d'appel (à droite).

*Vous ne pouvez pas supprimer tous les contacts de la liste d'appel.*

### 17.4 Suppression d'un contact

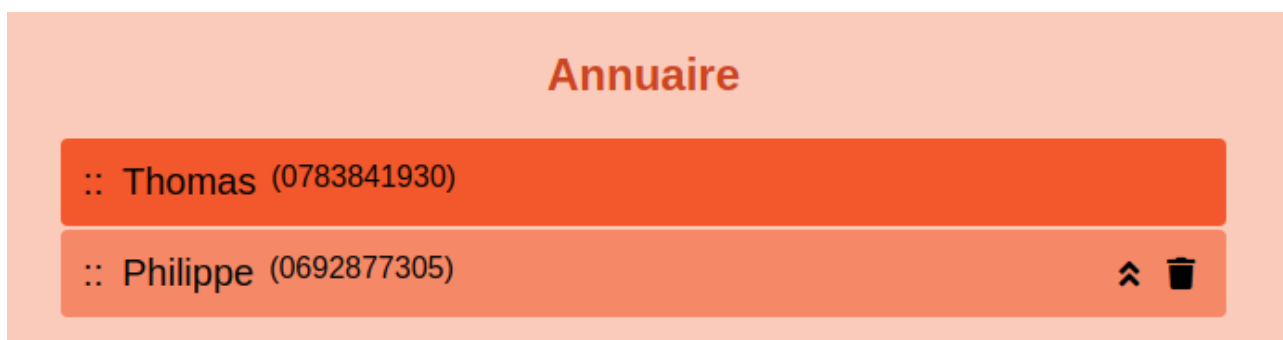
Pour supprimer un contact de la liste d'appel, utiliser la croix qui apparaît à gauche de l'élément correspondant dans la liste des contacts disponibles (à gauche).

*Supprimer un contact supprime également toutes ses occurrences dans la liste d'appel.*

### 17.5 Réglage du contact prioritaire

Pour définir le contact prioritaire, utiliser le bouton avec les deux flèches vers le haut qui apparaît au passage de la souris dans l'annuaire.

Une nouvelle entrée sera alors automatiquement ajoutée à la liste d'appel pour que le nouveau contact prioritaire soit présent en haut.

L'annuaire est intitulé "Annuaire" en rouge. Il contient deux entrées de contact. La première entrée est "Thomas (0783841930)". La deuxième entrée est "Philippe (0692877305)". À droite de l'entrée "Philippe", il y a deux icônes : une flèche vers le haut et une poubelle.

**Annuaire**

:: Thomas (0783841930)

:: Philippe (0692877305) ⬆️ 🗑️

FIGURE 4 – Définition du contact prioritaire