

---

# **INTERFACE D'IMPORTATION PROJET GIE / TF1**

---

***Documentation Technique  
Active X PubImport V4.0***

**FEVRIER 2021**

---

## I. DESCRIPTION DE L'INTERFACE D'IMPORTATION

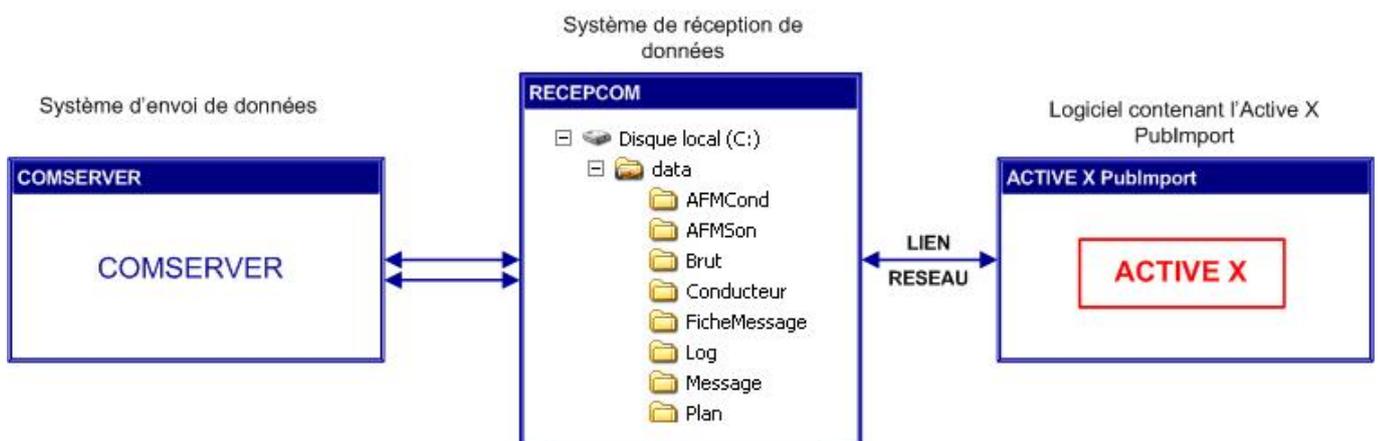
### 1. Définition

Il s'agit d'une interface d'importation entre le routeur et les différents systèmes de diffusion disponibles sur le marché et agréés par le G.I.E. Les Indépendants

Elle permet, de façon simple, de récupérer les informations émanant de la régie publicitaire via le routeur.

### 2. Vue générale

#### Architecture « RouteurGIE »



Toutes les données émanant du ComServer sont stockées dans le répertoire "c:\data\brut".  
Toutes les données stockées dans le répertoire "c:\data\log" seront renvoyées vers le ComServer.  
Le répertoire "c:\data" sera accessible sous la forme d'un partage Windows.

#### Architecture « Web API »



### 3. Fonctionnalités

Les fonctionnalités de l'interface d'importation sont les suivantes :

- ✓ Listage des campagnes.
- ✓ Listage des campagnes ajoutées, modifiées ou supprimées.
- ✓ [Listage des campagnes pour une journée donnée](#)
- ✓ Listage des descriptions de messages.
- ✓ Listage des descriptions de messages ajoutées, modifiées ou supprimées.
- ✓ [Listage des descriptions de messages pour une journée donnée](#)
- ✓ [Remise dans brut des fichiers publicitaires \(ou remise à télécharger selon l'architecture cible\)](#)
- ✓ Récupération des messages sonores (avec copie réseau ou internet).
- ✓ Récupération d'informations des messages sonores (sans copie).
- ✓ Pointage des spots (un spot étant un message dans un écran donné). Le pointage des spots permet de générer un rapport de diffusion.
- ✓ Emission d'un historique d'utilisation permettant d'indiquer les fonctions de L'interface d'importation ayant été utilisées.
- ✓ Listage des conducteurs.
- ✓ Listage des conducteurs ayant été ajoutés ou modifiés.
- ✓ Traçage de la modification du contenu sonore d'un message national.

### 4. Description Fonctionnelle

Pour récupérer les publicités nationales, différentes étapes sont à réaliser telles que :

- ✓ La première étape consiste à initialiser le répertoire de stockage des données.  
Pour cela, on utilise la commande "**InitRepertoire**" suivi du nom de chemin réseau sur lequel se trouvent les données ou de la chaîne de connexion à l'API (selon l'architecture cible).  
Si vous adressez une Web API, vous pouvez initialiser l'accès aux données en utilisant la fonction **InitWebApi**(url, user, password).
- ✓ La seconde étape consiste à réceptionner les nouvelles campagnes.  
Pour cela, on utilise la commande "**ListeCampagnesModifiees**". Cette dernière retourne un fichier XML de descriptions des campagnes pour celles qui sont ajoutées, modifiées ou supprimées.  
Attention, lorsque la fonction a été appelée, l'active X considère que les campagnes concernées ne doivent plus apparaître lors du prochain appel de cette fonction.
- ✓ La troisième étape consiste à réceptionner la description des nouveaux messages.  
Pour cela, on utilise la commande "**ListeMessagesModifies**". Cette dernière retourne un fichier XML de descriptions de messages pour ceux qui sont ajoutés, modifiés ou supprimés.  
Attention, lorsque cette fonction a été appelée, l'active X considère que les descriptions de messages concernées ne doivent plus apparaître lors du prochain appel de cette fonction.
- ✓ La quatrième étape consiste à récupérer le contenu sonore des messages.  
Pour cela, on utilise la commande "**RecupererMessage**". Cette dernière copie le contenu sonore du message dans un fichier spécifié dans la commande.
- ✓ La dernière étape consiste à pointer la diffusion du spot.

Pour cela, 3 commandes sont à utiliser : "**DemarrerPointageDeSpot**", "**PointageDeSpot**" ou "**PointageDeSpotCC**" et "**ArreterPointageDeSpot**". Ces dernières permettent d'indiquer que les messages ont bien été diffusés dans la tranche horaire spécifiée.

#### **4.1 Description Fonctionnelle Autoroute FM**

- ✓ La première étape consiste à initialiser le répertoire de stockage des données.  
Pour cela, on utilise la commande "**InitRepertoire**" suivi du nom de chemin réseau sur lequel se trouvent les données.
- ✓ La seconde étape consiste à réceptionner les conducteurs et les sons  
Pour les conducteurs, on utilise la commande "**AfmRecupererConducteur**" en indiquant le nom de fichier source ainsi que celui de destination.  
Pour les sons, on utilise la commande "**AfmRecupererSons**" en indiquant le nom de fichier source ainsi que celui de destination.
- ✓ La dernière étape consiste à pointer la diffusion du spot.  
Pour cela, 3 commandes sont à utiliser : "**DemarrerPointageDeSpot**", "**PointageDeSpot**" ou "**PointageDeSpotCC**" et "**ArreterPointageDeSpot**". Ces dernières permettent d'indiquer que les messages ont bien été diffusés dans la tranche horaire spécifiée.

## 4.2 Description Fonctionnelle : Fonctions Génériques

### 4.2.1 : Fonctions génériques normales

Les fichiers publicitaires (Plans, Sons et Fiches Messages) sont envoyés aux routeurs avec un préfixe décrivant le produit auquel ils appartiennent (sans préfixe pour le National GIE, IDF pour Puissance Capital...).

A partir de la version 2.3 de l'ActiveX PubImport, des fonctions permettant d'importer les fichiers de type générique ont été ajoutées. Toutes les fonctionnalités présentes pour le traitement de la publicité Nationale ou Ile-DeFrance sont utilisables pour tout autre type de produit.

Les fonctions ajoutées sont de la forme :

- ✓ **GeneNomDeLaFontionInitiale**(ListeParamFonctionInitial, string Prefixe)

Exemple:

- ✓ long **ListeCampagnes** (string FichierDestination)

devient

- ✓ long **GeneListeCampagnes** (string FichierDestination, string Prefixe)

Utilisation :

Pour lister les campagnes de la publicité nationale GIE :

**ListeCampagnes** ("C:\CpgNatio.xml")

ou

**GeneListeCampagnes** ("C:\ CpgNatio.xml", "")

Pour lister les campagnes de la publicité Ile-De-France:

**IDFListeCampagnes** ("C:\CpgIDF.xml")

ou

**GeneListeCampagnes** ("C:\ CpgIDF.xml", "IDF")

Pour lister les campagnes de la publicité Autoroute Info (Préfixe=AFMINFO)

**GeneListeCampagnes** ("C:\CpgAFMInfo.xml", "AFMINFO")

### 4.2.2 : Fonctions génériques radios

Dans la version 3.3 de l'ActiveX PubImport, des fonctions génériques permettant de n'importer que des fichiers adressés à une radio donnée ont été intégrées. Ce sont des déclinaisons des fonctions génériques existantes prenant un paramètre supplémentaire indiquant la radio concernée. Elles sont de la forme :

- ✓ **GeneRadioNomDeLaFontionInitiale**(ListeParamFonctionInitial, string Radio)

**Radio** est une chaîne de caractères correspondant à l'identifiant régie de la radio.

Attention, en raison d'une limitation du nombre de caractères dans le nom d'une fonction, certains noms ont été réduits. C'est le cas pour :

**GeneRadioListeCpgModifiees** au lieu de **GeneRadioListeCampagnesModifiees**  
**GeneRadioListeCondModifies** au lieu de **GeneRadioListeConducteursModifiees**  
**GeneRadioVisuCondArchives** au lieu de **GeneRadioVisuConducteursArchives**

Ces fonctions peuvent également être utilisées pour importer des fichiers non adressés à une radio donnée (par exemple le produit National) en passant comme identifiant régie de radio la valeur vide "".

Pour lister les campagnes de la publicité nationale GIE :

**ListeCampagnes** ("C:\CpgNatio.xml")

ou

**GeneListeCampagnes** ("C:\ CpgNatio.xml", "")

ou

**GeneRadioListeCampagnes** ("C:\ CpgNatio.xml", "", "")

Pour lister les campagnes de la publicité Ile-De-France:

**IDFListeCampagnes** ("C:\CpgIDF.xml")

ou

**GeneListeCampagnes** ("C:\ CpgIDF.xml", "IDF")

ou

**GeneRadioListeCampagnes** ("C:\ CpgIDF.xml", "IDF", "")

Pour lister les campagnes Multiville pour la radio ALO:

**GeneRadioListeCampagnes** ("C:\CpgMultiVilleALO.xml", "MV", "ALO")

#### 4.2.3 : Fonctions datées

Depuis la version 4.0, l'ActiveX PubImport peut récupérer les données depuis une API Web. Contrairement à l'architecture adressant un répertoire partagé (typiquement [\\routeurgie\data\\$](#)), l'API Web est capable de déterminer l'utilité d'un fichier pour une date. De fait, des fonctions ont été ajoutées permettant de récupérer tous les fichiers « utile » pour une date donnée. Elles se basent sur les fonctions génériques et n'existe pas sous la forme des fonctions historiques permettant d'adresser directement le produit sans préfixe ou IDF. Un paramètre supplémentaire permet d'indiquer la date choisie (au format AAAAMMDD)

- Récupérer toutes les campagnes GIE utiles pour le 01/01/2020 :

**GeneListeCampagnesDate** ("C:\CpgNatio.xml", "", "20200101")

- Récupérer toutes les campagnes IDF utiles pour le 01/02/2020 :

**GeneListeCampagnesDate** ("C:\CpgNatio.xml", "IDF", "20200201")

- Récupérer toutes les campagnes Multiville de la radio 'CON' utiles pour le 01/02/2020 :

**GeneRadioListeCampagnesDate** ("C:\CpgNatio.xml", "MV", "CON", "20200201")

- Récupérer toutes les fiches messages GIE utiles pour le 01/02/2020 :

**GeneListeMessagesDate** ("C:\Fiches.xml", "", "20200201")

#### 4.3 Description Fonctionnelle : Fichier de type MultiVille

Dans sa version 3.3, l'ActiveX PubImport a été modifié afin de permettre l'importation de fichiers dont l'export est groupé par radio et adressant une ou plusieurs zones de cette radio. Ces modifications permettent de gérer informatiquement le produit Multiville.

En effet, une campagne Multiville peut-être envoyée pour chacune des radios alimentées par une machine de réception publicitaire (RouteurGIE). De plus, chaque campagne Multiville d'une radio peut adresser une ou plusieurs zones de diffusion de cette radio.

Le fonctionnement interne de l'ActiveX PubImport a donc été modifié afin de générer, à destination de la pige du GIE « Les Indes Radios », la description et l'horodatage des importations avec les informations de radios et de zones.

Le préfixe correspondant au produit Multiville est « MV ».

L'importation des fichiers de type Multiville peut se faire de 2 manières différentes :

- soit en utilisant les fonctions génériques important les fichiers de toutes les radios
- soit en utilisant les fonctions génériques « radios » important les fichiers d'une radio donnée

#### 4.3.1 : Utilisation des fonctions génériques

Le premier mode de fonctionnement utilise les fonctions génériques existantes (cf. 4.2.1 Fonctions Génériques) et permet d'importer en un seul appel, les fichiers Multiville correspondant à la requête (Liste des campagnes, liste des conducteurs modifiés ou non) pour TOUTES les radios alimentées par le RouteurGIE.

Pour l'ensemble des résultats retournés, il convient de récupérer le champ « IdentifiantRadio » de l'en-tête et de ne traiter le résultat que pour la radio correspondante.

Par exemple, l'appel de la fonction **GeneListeCampagnesModifiees**("C:\result.xml", "MV") retournera l'ensemble des campagnes Multiville nouvellement reçues (présentes dans Brut) quelque soit la radio concernée.

Dans ce mode de fonctionnement, il est préférable de prévoir la récupération d'un résultat pour une radio non configurée dans le système.

#### 4.3.2 : Utilisation des fonctions génériques « Radios »

Le second mode de fonctionnement utilise les déclinaisons « radio » des fonctions génériques (cf. 4.2.2 Fonctions Génériques Radios). Ces fonctions prennent toutes un paramètre supplémentaire (une chaîne de caractères) indiquant l'identifiant régie de la radio pour laquelle les résultats doivent être retournés.

Par exemple, pour récupérer la liste des plans Multiville nouvellement arrivés (présentes dans Brut) pour la radio identifiée par ALO, on appelle la fonction **GeneRadioListeCpgModifiees**("C:\result.xml", "MV", "ALO").

Il est à noter que, dans ce mode de fonctionnement, une radio alimentée en Multiville par le RouteurGIE, mais non renseignée dans le système (donc d'identifiant radio inconnu) ne pourra pas être récupérée.

#### 4.3.3 : Exemple de résultats retournés par l'appel de différentes fonctions

Soit une machine de réception publicitaire alimentant 3 radios AAA, AAB et AAC. Si le contenu de cette machine est le suivant :

Répertoire BRUT	Répertoire de Sauvegarde (PLAN ou Conducteur)
MVPLAN00400000.AAA.000 (1)	MVPLAN00400000.AAA.000 (5)
MVPLAN00400000.AAB.000 (2)	MVPLAN00400000.AAC.000 (6)
MVCOND20120601.AAA.XML (3)	MVCOND20120601.AAA.XML (7)
MVCOND20120601.AAB.XML (4)	MVCOND20120601.AAC.XML (8)
	MVPLAN00400001.AAA.000 (9)

Fonction appelée	Résultat obtenu
<b>GeneListeCampagnesModifiees</b> ("C:\CpgMV.xml", "MV")	MVPLAN00400000.AAA.000 (1) MVPLAN00400000.AAB.000 (2)
<b>GeneListeCampagnes</b> ("C:\CpgMV.xml", "MV")	MVPLAN00400000.AAA.000 (5) MVPLAN00400000.AAC.000 (6)
<b>GeneListeConducteursModifies</b> (("C:\CondMV.xml", "MV")	MVCOND20120601.AAA.XML (3) MVCOND20120601.AAB.XML (4)
<b>GeneListeConducteurs</b> (("C:\CondMV.xml", "MV")	MVCOND20120601.AAA.XML (7) MVCOND20120601.AAC.XML (8)
<b>GeneImporterCampagne</b> (400000, "C:\Cpg400000.xml", "MV")	MVPLAN00400000.AAA.000 (1) MVPLAN00400000.AAB.000 (2) MVPLAN00400000.AAC.000 (6)
<b>GeneImporterConducteur</b> (20120601, "C:\CondMV20120601.xml", "MV")	MVCOND20120601.AAA.XML (3) MVCOND20120601.AAB.XML (4) MVCOND20120601.AAC.XML (8)
<b>GeneRadioListeCpgModifiees</b> ("C:\CpgMV.xml", "MV", "AAA")	MVPLAN00400000.AAA.000 (1)
<b>GeneRadioListeCampagnes</b> ("C:\CpgMV.xml", "MV", "AAA")	MVPLAN00400000.AAA.000 (5) MVPLAN00400001.AAA.000 (9)
<b>GeneRadioListeCpgModifiees</b> ("C:\CpgMV.xml", "MV", "AAC")	Rien
<b>GeneRadioListeCampagnes</b> ("C:\CpgMV.xml", "MV", "AAC")	MVPLAN00400000.AAC.000 (6)
<b>GeneRadioListeConducteurs</b> (("C:\CondMV.xml", "MV", "AAA")	MVCOND20120601.AAA.XML (7)
<b>GeneRadioListeCondModifies</b> ("C:\CondMV.xml", "MV", "AAA")	MVCOND20120601.AAA.XML (3)

## 5. Adresser l'architecture Web

En partant d'un code existant adressant une architecture basée sur un « RouteurGIE », vous pouvez adresser une architecture Web en passant une chaîne de connexion à la place du répertoire de données du RouteurGIE.

Adressage d'un RouteurGIE :

InitRepertoires(« [\\routeurgie\data\\$](#) »)

Adressage d'une architecture Web :

InitRepertoires(« [url | user | password](#) »)

Toutes les autres fonctions appelées se comportent de la même manière et retournent les mêmes valeurs.

## II. DOCUMENTATION TECHNIQUE

Afin de répondre à la description de l'interface d'importation, cette dernière se présentera sous la forme d'un Active X. Cet Active X fonctionne sous les systèmes d'environnement Windows 98 et supérieur et, s'importe aisément dans tout autre logiciel.

[A partir de la version 4.0, l'ActiveX PubImport ne fonctionne que sous les systèmes Windows 7 et ultérieur.](#)

### 1. Liste des fonctionnalités

✓ long **InitRepertoire** (string Repertoire)

Repertoire est une chaîne de caractères de longueur illimitée.

Cette fonction permet d'indiquer à l'interface d'importation où se trouve le répertoire de données du routeur. Elle renvoie la valeur 0 en cas d'échec de la fonction sinon toute autre valeur.

Par exemple : InitRepertoire ( \\routeur\data\$)

Cette fonction peut également être utilisée pour adresser une WebApi, en passant paramètre la chaîne de connexion au lieu du chemin d'accès. La chaîne de connexion est composée de l'url, du compte et du mot de passe séparé par des |. Il s'agit d'une surcouverte à l'appel de la fonction InitWebApi.

Par exemple : InitRepertoire([www.pub.com](http://www.pub.com) | moncompte | monpassword)

✓ long **InitWebApi** (string url, string user, string password)

Url est une chaîne de caractères de longueur illimitée.

User est une chaîne de caractères de longueur illimitée.

Password est une chaîne de caractères de longueur illimitée.

Cette fonction permet d'indiquer à l'interface d'importation les données de connexion à la WebAPI. Elle renvoie la valeur 0 en cas d'échec de la fonction sinon toute autre valeur.

Par exemple : InitWebApi ( [www.pub.com](http://www.pub.com), moncompte, monpassword)

✓ long **Conducteur** (string FichierDestination, string Date, string HeureDebut, string HeureFin)

**(Cette fonction n'existe plus)**

✓ long **ListeCampagnes** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des campagnes. Le contenu du fichier XML sera plus amplement détaillé dans le point 3 du présent dossier. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette liste est établie à partir des plans de roulement contenus dans le répertoire "c:\data\plan" du routeur.

✓ long **ListeCampagnesModifiees** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des campagnes ayant subi une modification (ajout, modification, suppression). Le contenu du fichier XML est identique au fichier XML de la fonction ListeCampagne. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette liste est établie à partir des plans de roulement contenus dans le répertoire "c:\data\brut" du routeur. Ces plans sont ensuite copiés dans le répertoire "c:\data\plan".

✓ long **ListeMessages** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des descriptions de messages. Le contenu du fichier XML sera plus amplement détaillé dans le point 3 du présent dossier. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette liste est établie à partir des fiches messages contenues dans le répertoire "c:\data\FicheMessage" du routeur.

✓ long **ListeMessagesModifies** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des messages ayant subi une modification (ajout, modification, suppression). Le contenu du fichier XML est identique au fichier XML de la fonction ListeMessages. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette liste est établie à partir des fiches messages contenues dans le répertoire "c:\data\brut" du routeur. Ces fiches messages sont ensuite copiées dans le répertoire "c:\data\FicheMessage".

✓ long **DemarrerPointageDeSpot** (string NomAntenne)

NomAntenne est une chaîne correspondant à l'Antenne qui a diffusé les spots.

Cette fonction est à utiliser avec les 2 autres fonctions de pointage. Elle initialise le pointage des spots sur l'antenne sélectionnée. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Le nom de l'antenne doit être défini comme suit

- En cas de Radio Mono Zone de diffusion publicitaire  
Nomdelaradio / Nomdelaradio
- En cas de Radio Multi Zone de diffusion publicitaire  
Nomdelaradio / Nomdelazone

**ATTENTION : Il doit y avoir un espace avant et après le /**

✓ long **ArreterPointageDeSpot** ()

Cette fonction termine OBLIGATOIREMENT le pointage de spot. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

✓ long **PointageDeSpot** (String DateDeDiffusion, String HeureDeDiffusion, long IdSpot, long IdMessage, long EtatDeDiffusion, long CodeAnomalieDeDiffusion, long DureeDuSpotDiffuse, String CodeEcran,

String HeureDeDiffusionEcran, long DureeEcran, long NbSpotsEcran, long PlaceDuSpotDansEcran, String Commentaire, long TypeSpot)

DateDeDiffusion est une chaîne de 8 caractères définie comme suit "aaaammjj" correspondant à la date de diffusion du spot pointé.

HeureDeDiffusion est une chaîne de 6 caractères définie comme suit "hhmmss" correspondant à l'heure de diffusion du spot pointé.

IdSpot est un entier long correspondant au code spot du spot pointé

IdMessage est un entier long correspondant au code message du spot pointé

EtatDeDiffusion est un entier long correspondant à l'état de diffusion du spot défini comme suit :

0 : Non diffusé

1 : Diffusé

2 : Diffusé avec anomalie mineure (Obsolète : considéré comme une erreur)

3 : Diffusé avec anomalie grave (Obsolète : considéré comme une erreur)

4 : non prévu (Obsolète : considéré comme une erreur)

CodeAnomalieDeDiffusion est un entier long correspondant au code anomalie de diffusion dans le cas où EtatDeDiffusion égale à 0, 2 ou 3. La valeur par défaut est 0. (Obsolète)

DureeDuSpotDiffuse est un entier long correspondant à la durée du spot diffusé en millisecondes

CodeEcran est une chaîne de caractère correspondant au libellé du code de l'écran

HeureDeDiffusionEcran est une chaîne de 8 caractères définie comme suit "hhmmss" correspondant à l'heure de diffusion de l'écran.

DureeEcran est un entier long correspondant à la durée de l'écran diffusé en millisecondes

NbSpotsEcran est un entier long correspondant au nombre de spot de l'écran.

PlaceDuSpotDansEcran est un entier long correspondant à la place du spot dans l'écran.

Commentaire est une chaîne de caractères correspondant au commentaire (facultatif)

TypeSpot est un entier long correspondant au type de spot, à savoir

- 0 : Publicité Nationale
- 1 : Publicité Locale
- 2 : Elément sonore avec citation annonceur
- 3 : Elément sonore sans citation annonceur
- 4 : Editorial (Rubriques, météo) (obsolète : considéré comme un élément sonore sans citation annonceur)
- 5 : Multiville
- 6 : Virgule

La description de la valeur de chaque élément lorsque des positions sont imposées est décrite en annexe.

Cette fonction permet le pointage des spots émanant de la régie publicitaire pour chacun des sites d'une radio. Cette information sera transmise au site central sous la forme d'un fichier XML. Ce fichier est stocké dans le répertoire "c:\data\log" du routeur. Ce fichier est ensuite traduit par le système de surveillance associé au serveur de communication. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Ces 3 dernières fonctions doivent être utilisées de la manière suivante obligatoirement :

```
DemarrerPointageDeSpot(« Mon site de diffusion »)
PointageDeSpot( ... paramètres du spot 1 ...)
PointageDeSpot( ... paramètres du spot 2 ...)
...
PointageDeSpot( ... paramètres du spot n ...)
ArreterPointageDeSpot()
```

Le détail des éléments requis et leurs valeurs lors du pointage des spots émanant de la régie publicitaire est présenté en annexe. Il est à noter que depuis la version 3.1 de l'ActiveX PublImport, cette fonction ajoute par défaut un code concurrence égal à 0.

- ✓ long **PointageDeSpotCC** (String DateDeDiffusion, String HeureDeDiffusion, long IdSpot, long IdMessage, long EtatDeDiffusion, long CodeAnomalieDeDiffusion, long DureeDuSpotDiffuse, String CodeEcran, String HeureDeDiffusionEcran, long DureeEcran, long NbSpotsEcran, long PlaceDuSpotDansEcran, String Commentaire, long TypeSpot, long CodeConcurrence)

Cette fonction est identique à **PointageDeSpot** mais demande un paramètre supplémentaire **CodeConcurrence**, de type entier, indiquant le code concurrence (code secteur) du spot pointé.

Ce dernier paramètre ne sera utilisé que pour les spots de type 1 (Local), 2 (Elément sonore avec citation d'annonceur) et 5 (Multiville). Il n'est pas interdit d'indiquer un code concurrence pour les autres types de spot mais celui-ci sera ignoré lors du traitement du rapport de diffusion. En effet, les codes concurrences des spots nationaux sont connus et ceux des spots sans citation d'annonceur ou des virgules n'ont pas de raison d'exister. Les codes concurrence indiqués doivent être présents dans le fichier envoyé par la régie qui récupérable via la fonction **ImporterListeCodeConcurrence**, sinon les spots seront considérés en erreur.

- ✓ long **NombreDeSpotsPointes()**

Cette fonction renvoie le nombre de spots pointés depuis le dernier appel à la fonction DemarrerPointageDeSpots.

Attention, la valeur retournée n'est pas réinitialisée après l'appel de la fonction ArreterPointageDeSpot.

- ✓ long **RecupererMessage** (long CodeMessage, string CheminDestination, string FichierDestination)

CodeMessage est un entier long correspondant à l'identifiant du message. Le CodeMessage doit être récupéré via la description des spots d'une campagne.

CheminDestination est une chaîne de caractères de longueur illimitée.

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de copier le message correspondant à CodeMessage dans le CheminDestination spécifié avec le nom FichierDestination. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Les messages sont stockés dans le répertoire "c:\data\brut" du routeur. Ils sont ensuite copiés dans le répertoire "c:\data\message" du routeur.

- ✓ long **DetailMessage** (long CodeMessage, string FichierDestination)

CodeMessage est un entier long correspondant à l'identifiant du message. Le CodeMessage doit être récupéré via la description des spots d'une campagne.

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la description du message indiqué par le CodeMessage. Le contenu du fichier XML sera plus amplement détaillé dans le point 3 du présent.

La fonction renvoie la valeur -1 si le code message n'a pas été trouvé

La fonction renvoie la valeur 2 pour indiquer que le message est nouvellement arrivé

La fonction renvoie la valeur 1 dans les autres cas

Cette fonction n'affecte pas les fichiers dans les différents répertoires du routeur

✓ long **AfmRecupererConducteur** (string FichierSource, string FichierDestination)

FichierSource est le nom du fichier conducteur.

FichierDestination est le nom de fichier destination

La fonction renvoie la valeur du code de retour de la fonction Win32 « CopyFile »

✓ long **AfmRecupererSons** (string FichierSource, string FichierDestination)

FichierSource est le nom du fichier sons.

FichierDestination est le nom de fichier destination

La fonction renvoie la valeur du code de retour de la fonction Win32 « CopyFile »

✓ long **ImporterCampagne** (long CodeCampagne, string FichierDestination)

CodeCampagne est un entier long correspondant à l'identifiant de la campagne.

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la description de la campagne passée en paramètre. Le contenu du fichier XML est de la même forme que pour la fonction listeCampagnesModifiees. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette description est établie à partir de la campagne contenue dans le répertoire "c:\data\brut" du routeur ou, si elle est déjà importée, du répertoire "c:\data\Plan". Cette campagne est ensuite copiée dans le répertoire "c:\data\ Plan".

✓ long **ImporterFicheMessage**(long CodeMessage, string FichierDestination)

CodeMessage est un entier long correspondant à l'identifiant du message.

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la description de la fiche message passée en paramètre. Le contenu du fichier XML est de la même forme que pour la fonction listeMessagesModifiees. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette description est établie à partir de la fiche message contenue dans le répertoire "c:\data\brut" du routeur ou, si elle est déjà importée, du répertoire "c:\data\FicheMessage". Cette fiche message est ensuite copiée dans le répertoire "c:\data\FicheMessage".

✓ long **VisualisationCampagnesBrut** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste **simplifiée** des campagnes à importer. Le contenu du fichier XML est de la même forme que pour la fonction listeCampagnes, **sauf que les spots des campagnes ne sont pas énumérés**. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette description est établie à partir des campagnes contenues dans le répertoire "c:\data\brut" du routeur. Ces campagnes **ne sont pas copiées** dans le répertoire "c:\data\Plan".

✓ long **VisualisationCampagnesArchives** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste **simplifiée** des campagnes déjà importées. Le contenu du fichier XML est de la même forme que pour la fonction listeCampagnes, **sauf que les spots des campagnes ne sont pas énumérés**. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette description est établie à partir des campagnes contenues dans le répertoire "c:\data\plan" du routeur.

✓ long **VisualisationFichesMessagesBrut** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des fiches messages à importer. Le contenu du fichier XML est de la même forme que pour la fonction listeMessages. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette description est établie à partir des fiches messages contenues dans le répertoire "c:\data\brut" du routeur. Ces fiches messages **ne sont pas copiées** dans le répertoire "c:\data\FicheMessage".

✓ long **VisualisationFichesMessagesArch**(string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des fiches messages déjà importés. Le contenu du fichier XML est de la même forme que pour la fonction listeMessages. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette description est établie à partir des fiches messages contenues dans le répertoire "c:\data\FicheMessage" du routeur.

✓ long **VisualisationMessagesBrut** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des messages (en mp2) à importer. Le contenu du fichier XML est de la même forme que pour la fonction DetailMessage. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette description est établie à partir des messages contenus dans le répertoire "c:\data\brut" du routeur. Ces fiches messages **ne sont pas copiées** dans le répertoire "c:\data\Message".

✓ long **VisualisationMessagesArchives**(string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des messages (en mp2) déjà importés. Le contenu du fichier XML est de la même forme que pour la fonction DetailMessage. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette description est établie à partir des messages contenus dans le répertoire "c:\data\ Message " du routeur.

✓ long **ListeConducteurs** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des conducteurs. Le contenu du fichier XML sera plus amplement détaillé dans le point 3 du présent dossier. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette liste est établie à partir des conducteurs contenus dans le répertoire "c:\data\conducteur" du routeur.

✓ long **ListeConducteursModifies** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant la liste des conducteurs ayant subi une modification (ajout, modification, suppression). Le contenu du fichier XML est identique au fichier XML de la fonction ListeConducteur. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette liste est établie à partir des conducteurs contenus dans le répertoire "c:\data\brut" du routeur. Ces conducteurs sont ensuite copiés dans le répertoire "c:\data\conducteur".

✓ long **ImporterConducteur** (string date, string FichierDestination)

date est une chaîne de 8 caractères contenant la date du conducteur à récupérer au format "AAAAMMJJ".

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de créer un fichier XML nommé "FichierDestination" contenant le conducteur de la date passée en paramètre. Le contenu du fichier XML est de la même forme que pour la fonction listeConducteursModifies. La fonction renvoie la valeur 0 en cas d'échec sinon toute autre valeur.

Cette description est établie à partir du conducteur contenu dans le répertoire "c:\data\brut" du routeur ou, si il a déjà été importé, du répertoire "c:\data\Conducteur". Cette campagne est ensuite copiée dans le répertoire "c:\data\Conducteur".

✓ Fonctions spéciales Ile-De-France

long **IDFListeCampagnes** (string FichierDestination)

long **IDFListeCampagnesModifiees** (string FichierDestination)

long **IDFListeMessages** (string FichierDestination)

long **IDFListeMessagesModifies** (string FichierDestination)

long **IDFRecupererMessage** (long CodeMessage, string CheminDestination, string FichierDestination)

long **IDFDetailMessage** (long CodeMessage, string CheminXML)

long **IDFImporterFicheMessage** (long CodeMessage, string FichierDestination)

long **IDFImporterCampagne** (long CodeCampagne, string FichierDestination)

long **IDFVisuCampagnesBrut** (string FichierDestination)

long **IDFVisuCampagnesArchives** (string FichierDestination)  
long **IDFVisuFichesMessagesBrut** (string FichierDestination)  
long **IDFVisuFichesMessagesArch** (string FichierDestination)  
long **IDFVisuMessagesBrut** (string FichierDestination)  
long **IDFVisuMessagesArchives** (string FichierDestination)  
long **IDFListeConducteurs** (string FichierDestination)  
long **IDFListeConducteursModifies** (string FichierDestination)  
long **IDFImporterConducteur** (string date, string FichierDestination)

Toutes les fonctions commençant par IDF pour Ile-De-France, se comportent exactement de la même façon que les fonctions initiales. Seules les campagnes, les fiches messages et les messages utilisés pour le produit Ile-de-France sont importés.

✓ Fonctions Génériques

long **GeneListeCampagnes** (string FichierDestination, string Prefixe)  
long **GeneListeCampagnesModifiees** (string FichierDestination, string Prefixe)  
long **GeneListeMessages** (string FichierDestination, string Prefixe)  
long **GeneListeMessagesModifies** (string FichierDestination, string Prefixe)  
long **GeneRecupererMessage** (long CodeMessage, string CheminDestination, string FichierDestination, string Prefixe)  
long **GeneDetailMessage** (long CodeMessage, string CheminXML, string Prefixe)  
long **GeneImporterFicheMessage** (long CodeMessage, string FichierDestination, string Prefixe)  
long **GeneImporterCampagne** (long CodeCampagne, string FichierDestination, string Prefixe)  
long **GeneVisuCampagnesBrut** (string FichierDestination, string Prefixe)  
long **GeneVisuCampagnesArchives** (string FichierDestination, string Prefixe)  
long **GeneVisuFichesMessagesBrut** (string FichierDestination, string Prefixe)  
long **GeneVisuFichesMessagesArch** (string FichierDestination, string Prefixe)  
long **GeneVisuMessagesBrut** (string FichierDestination, string Prefixe)  
long **GeneVisuMessagesArchives** (string FichierDestination, string Prefixe)  
long **GeneListeConducteurs** (string FichierDestination, string Prefixe)  
long **GeneListeConducteursModifies** (string FichierDestination, string Prefixe)  
long **GeneImporterConducteur** (string date, string FichierDestination, string Prefixe)  
long **GeneVisuConducteursBrut** (string FichierDestination, string Prefixe)  
long **GeneVisuConducteursArchives** (string FichierDestination, string Prefixe)

Toutes les fonctions génériques se comportent de la même façon que les fonctions spécifiques de la publicité Nationale ou Ile-De-France. Seul un paramètre décrivant le préfixe des fichiers à traiter est à ajouter lors de l'appel de la fonction.

✓ Fonctions Génériques « Radios »

long **GeneRadioListeCampagnes** (string FichierDestination, string Prefixe, string radio)  
long **GeneRadioListeCpgModifiees** (string FichierDestination, string Prefixe, string radio)  
long **GeneRadioVisuCampagnesBrut** (string FichierDestination, string Prefixe, string radio)  
long **GeneRadioVisuCampagnesArchives** (string FichierDestination, string Prefixe, string radio)  
long **GeneRadioImporterCampagne** (long CodeCampagne, string FichierDestination, string Prefixe, string radio)  
long **GeneRadioListeConducteurs** (string FichierDestination, string Prefixe, string radio)  
long **GeneRadioListeCondModifies** (string FichierDestination, string Prefixe, string radio)  
long **GeneRadioVisuConducteursBrut** (string FichierDestination, string Prefixe, string radio)  
long **GeneRadioVisuCondArchives** (string FichierDestination, string Prefixe, string radio)  
long **GeneRadioImporterConducteur** (string date, string FichierDestination, string Prefixe, string radio)

Ces fonctions sont équivalentes aux fonctions de gestion de campagnes et de conducteurs génériques mais prennent un paramètre supplémentaire indiquant la radio concernée par l'appel. L'identifiant radio est une chaîne de caractères.

Les fonctions génériques « radios » se comportent de la même manière que leurs homologues « normales » dans le cas où l'identifiant radio passé en paramètre est vide (""). Si ce n'est pas le cas, les résultats retournés se limiteront à ceux concernant la radio passée en paramètre.

✓ Fonctions Génériques datées

Les fonctions datées fonctionnent de la même manière que les fonctions classiques mais ne retournent que les fichiers nécessaires à la diffusion d'une date passée en paramètre. La date est au format « AAAAMMDD ».

long **GeneListeCampagnesDate**(string FichierDestination, string Prefixe, string date)

long **GeneRadioListeCampagnesDate**(string FichierDestination, string Prefixe, string radio, string date)

long **GeneListeMessagesDate**(string FichierDestination, string Prefixe, string date)

✓ Remise de fichiers publicitaires dans Brut / à télécharger

Bien qu'il soit possible de récupérer nommément une campagne, une fiche message ou un conducteur déjà importé (via les fonctions du style **ImporterFichier**(id\_du\_fichier)), il peut être intéressant de pouvoir remettre des fichiers dans Brut / à télécharger pour les récupérer avec un processus classique à base de **ListeFichierModifiees**.

Pour cela, vous pouvez utiliser les fonctions :

long **GeneRemiseCampagneBrut**(long idcampagne, string prefixe)

idcampagne est l'identifiant de la campagne

prefixe est le préfixe du produit (vide pour Les Indés Radios, IDF pour Les Indés Capitale...)

Retourne 1 si le fichier a été trouvé et remis dans Brut / à télécharger. 0 sinon.

Si vous appelez cette fonction sur un produit adressé (par exemple Multiville), cela remettra dans Brut / à télécharger, l'ensemble des campagnes ayant pour identifiant « idcampagne », quelle que soit la radio associée.

long **GeneRadioRemiseCampagneBrut**(long idcampagne, string prefixe, string radio)

idcampagne est l'identifiant de la campagne

prefixe est le préfixe du produit (vide pour Les Indés Radios, IDF pour Les Indés Capitale...)

radio est l'identifiant de la radio recherchée (pour les produits adressés comme le Multiville)

Si ce paramètre est vide, cela revient à appeler la fonction

**GeneRemiseCampagneBrut**(idcampagne, prefixe).

Retourne 1 si le fichier a été trouvé et remis dans Brut / à télécharger. 0 sinon

long **GeneRemiseFicheBrut**(long idMessage, string prefixe)

idMessage est l'identifiant du message

prefixe est le préfixe du produit (vide pour Les Indés Radios, IDF pour Les Indés Capitale...)

Retourne 1 si le fichier a été trouvé et remis dans Brut / à télécharger. 0 sinon.

long **GeneRemiseConducteurBrut**(string date, string prefixe)

date est la date désirée au format AAAAMMJJ

prefixe est le préfixe du produit (vide pour Les Indés Radios, IDF pour Les Indés Capitale...)

Retourne 1 si le fichier a été trouvé et remis dans Brut / à télécharger. 0 sinon.

Si vous appelez cette fonction sur un produit adressé (par exemple Multiville), cela remettra dans Brut / à télécharger, l'ensemble des conducteurs ayant pour identifiant « date », quelle que soit la radio associée.

long **GeneRadioRemiseConducteurBrut** (string date, string prefixe, string radio)

date est la date désirée au format AAAAMMJJ

prefixe est le préfixe du produit (vide pour Les Indés Radios, IDF pour Les Indés Capitale...)

radio est l'identifiant de la radio recherchée (pour les produits adressés comme le Multiville)

Si ce paramètre est vide, cela revient à appeler la fonction **GeneRemiseConducteurBrut** (date, prefixe).

Retourne 1 si le fichier a été trouvé et remis dans Brut / à télécharger. 0 sinon

✓ Fonction de récupération des fichiers de configuration

long **NouvelleListeCodeConcurrence**()

Cette fonction renvoie 1 si une nouvelle liste de code concurrence a été reçue, 0 sinon.

long **ImporterListeCodeConcurrence** (string FichierDestination)

FichierDestination est une chaîne de caractères de longueur illimitée.

Cette fonction permet de récupérer la liste des codes concurrence dans « FichierDestination ». Le format de ce fichier est décrit dans la partie 3 de ce document. Elle renvoi 0 en cas d'échec, sinon toute autre valeur.

Le fichier retourné correspond toujours à la dernière version de la liste des codes concurrence envoyée par la régie. C'est-à-dire que cette fonction renvoi le fichier présent dans le répertoire de réception (Brut) s'il existe (si **NouvelleListeCodeConcurrence** renvoi 1), sinon elle renvoi le fichier sauvegardé présent dans le répertoire Config.

✓ **Fonction de traçage des remplacements de contenus sonores.**

Ces fonctions permettent de tracer le remplacement du contenu sonore d'un message national (tout produit confondu). Elles seront notamment utiles pour la pîge de la diffusion publicitaire dans le cadre d'une alerte de la régie. Elles doivent être utilisées par le SDA lorsque l'utilisateur remplace le contenu sonore d'un message national.

Vous devez avoir utilisé la méthode **InitRepertoire** pour initialiser le chemin d'accès au RouteurGIE par l'OCX avant l'appel d'une de ces fonctions.

long **RemplaceMessageParLocal**( long CodeMessageInitial,  
long CodeSecteurMsgRemplacement)

- [CodeMessageInitial](#) est l'identifiant du message national remplacé
- [CodeSecteurMsgRemplacement](#) est le code secteur du spot local de remplacement

Indique le remplacement du message national dont l'identifiant (idMessage) est [CodeMessageInitial](#), par un spot local dont le secteur d'activité est [CodeSecteurMsgRemplacement](#). Le secteur d'activité doit appartenir à la liste des secteurs autorisé par la régie.

Dans le cas où [CodeSecteurMsgRemplacement](#) est égal à 0, le message national sera considéré comme étant remplacé par un élément sonore sans citation d'annonceur (comblage...).

long **RemplaceMessage**(long [CodeMessageInitial](#), long [CodeMessageRemplacement](#))

- [CodeMessageInitial](#) est l'identifiant du message national remplacé
- [CodeMessageRemplacement](#) est l'identifiant du spot national de remplacement

Indique le remplacement du message national dont l'identifiant (idMessage) est [CodeMessageInitial](#), par un autre message national dont l'identifiant est [CodeMessageRemplacement](#).

Dans le cas où [CodeMessageRemplacement](#) est égal à 0, le message national sera considéré comme étant remplacé par un élément sonore sans citation d'annonceur (comblage...).

Ces 2 fonctions renvoient 1 si le traçage s'est bien passé, 0 sinon (pas d'accès en écriture au répertoire du RouteurGIE). Attention, si la fonction InitRepertoire n'a pas été appelée sur l'instance de l'ActiveX PublImport utilisée, la fonction renverra tout de même 1 mais générera les logs localement.

L'appel de ces fonctions génère un fichier de log dans le répertoire [\\routeurgie\data\\$\log](#). Il contient, en outre, une ligne de texte indiquant la modification effectuée.

### Adressage d'une architecture WebApi : Optimisation du comportement des fonctions

Lorsque vous adressez une architecture basée sur une WebApi, le comportement des fonctions permettant de récupérer toutes les campagnes ou toutes les fiches message a été optimisé pour ne retourner que les fichiers « Utiles », c'est-à-dire, les fichiers nécessaires à la diffusion de la journée en cours et des journées futures.

Cela concerne les fonctions :

ListeCampagnes

IDListeCampagnes

GeneListeCampagnes

GeneRadioListeCampagnes

ListeMessages

IDListeMessages

GeneListeMessages

### ✓ Historique d'utilisation

Tous les appels vers les fonctions de l'interface d'importation sont logués sous la forme d'un fichier construit de la manière suivante : "aaaammjjhhmss.log". Ce dernier indique l'année en 4 caractères, le mois en 2 caractères, le jour en 2 caractères, l'heure en 2 caractères, les minutes en 2 caractères et enfin, les secondes en 2 caractères.

Cette précaution permet de connaître les résultats renvoyés par l'active X. On peut ainsi tracer toutes les opérations effectuées par l'active X. En cas de litige, cela permettra de connaître exactement le déroulement des opérations effectuées.

Ce fichier est stocké dans le répertoire "c:\data\log\" du routeur.

## 2. Description des répertoires

Ces informations ne concernent que l'architecture basée sur un répertoire partagé.

Les fichiers de données sont représentés suivant l'architecture suivante :

- ✓ Le répertoire "C:\data\brut" contient les fichiers émanant du site central.
- ✓ Le répertoire "C:\data\log" contient les fichiers de l'historique d'utilisation et des informations de pointage de spots.
- ✓ Le répertoire "C:\data\plan" contient les informations relatives aux campagnes.
- ✓ Le répertoire "C:\data\FicheMessage" contient les descriptions de messages.
- ✓ Le répertoire "C:\data\Message" contient le contenu sonore des messages.
- ✓ Le répertoire "C:\data\Conducteurs" contient les conducteurs importés.
- ✓ Le répertoire "C:\data\Config" contient les fichiers de configuration (par exemple le fichier contenant la liste des codes concurrence).

### 3. Description des fichiers XML

#### ✓ Le fichier XML de campagne

Ce fichier contient la liste des campagnes. L'élément suivant a été défini :

- Un champ "NbCampagnes" correspondant au nombre de campagnes listées.

#### **ATTENTION**

A partir de la version 2.5 de l'ActiveX, l'élément « campagne » n'est plus défini par la balise <campagne> mais par la balise <Campagne>. Il est également conseillé de traiter les 2 types de balises (Campagne et campagne) afin d'assurer la compatibilité avec les fichiers au format « texte ».

Pour chaque campagne, sont définis les éléments suivants :

- Un champ "Nom" correspondant au nom de la campagne.
- Un champ "CodeSecteur" correspondant au code du secteur d'activité de la campagne.
- Un champ "Secteur" correspondant au libellé du secteur d'activité de la campagne.
- Un champ "Identifiant" correspondant au numéro d'identification de la campagne.
- Un champ "Supprimee" indiquant si la campagne a été supprimée. Pour indiquer qu'une campagne a été supprimée, le champ contient la valeur "1", dans le cas contraire, le champ contient la valeur "0".
- Un champ "DateDebut" correspondant à la date de début de la campagne sous la forme "aaaammjj".
- Un champ "DateFin" correspondant à la date de fin de la campagne sous la forme "aaaammjj".
- Un champ "NomAnnonceur" correspondant au nom de l'annonceur de la campagne.
- Un champ "DateDebutSpot" correspondant à la date du premier spot de la campagne sous la forme "aaaammjj".
- Un champ "DateFinSpot" correspondant à la date du dernier spot de la campagne sous la forme "aaaammjj".
- Un élément "ListeJumelage" contenant une liste de champs "IdCampJumelee" correspondant à l'identifiant d'une campagne jumelée.
- Un champ "NbSpots" correspondant au nombre de spots définis pour cette campagne.

#### **ATTENTION**

A partir de la version 2.5 de l'ActiveX, l'élément « spot » n'est plus défini par la balise <spot> mais par la balise <Spot>. Il est également conseillé de traiter les 2 types de balises (Spot et spot) afin d'assurer la compatibilité avec les fichiers au format « texte ».

En plus des différents éléments précités, à chaque campagne est liée une liste des spots à diffuser. Cette dernière comprend les éléments suivants :

- Un champ "Date" correspondant à la date de diffusion du spot sous la forme "aaaammjj"
- Un champ "HeureDebut" correspondant à l'heure de début de la tranche horaire de diffusion du spot sous la forme "hh:mm"
- Un champ "HeureFin" correspondant à l'heure de fin de la tranche horaire de diffusion du spot sous la forme "hh:mm"
- Un champ "IdentifiantMessage" correspondant à l'identification du message à diffuser. Dans le cas où cette information n'est pas disponible, le champ contient la valeur 0.
- Un champ "DureeCommerciale" correspondant à la durée commerciale du spot exprimée en secondes.
- Un champ "Doublon" permettant de détecter si un spot est présent deux fois dans la même tranche horaire. Dans ce cas, le champ contient le numéro de passage du spot.
- Un champ "DoublonMemeEcran" indiquant comment les spots en doublon doivent être programmés. 0 = indifférent, 1 : même écrans, 2 : écrans distincts.
- Un champ "Rigueur" permettant d'indiquer la position du message dans l'écran. Ce champ contient la valeur :
  - 'T' si le spot doit être placé en tête d'écran
  - '2' si le spot doit être en deuxième position dans l'écran
  - '3' si le spot doit être en troisième position dans l'écran
  - 'F' si le spot doit être placé en fin d'écran
  - 'A' si le spot doit être placé soit en début, soit en fin d'écran.
  - ' ' (espace) si aucune rigueur n'est retenue
- Un champ "PositionPreferentielle" indiquant si la position de rigueur est obligatoire (Valeur "O") ou souhaitée par le client (Valeur "S"). Aucune valeur si aucune position n'est demandée.
- Un champ "IdentifiantSpot" correspondant à l'identifiant du spot.
- Un champ "CodeConcurrence1" correspondant au code du premier secteur d'activité associé au message.
- Un champ "CodeConcurrence2" correspondant au code du deuxième secteur d'activité associé au message.
- Un champ "~~LibelleConcurrence1~~" correspondant au libellé du premier secteur d'activité associé au message.
- Un champ "~~LibelleConcurrence2~~" correspondant au libellé du deuxième secteur d'activité associé au message.
- Un champ "DemiHeure" indiquant le numéro de l'écran de la tranche horaire dans laquelle le spot a été vendu. Ce champs peut prendre comme valeur : "P" pour le premier écran, "D" pour le dernier écran, "E" pour l'écran écran (ou écran star) et "" sinon.
- Un champ "OrdreDiff" indiquant l'ordre de diffusion pour les spots ou 0 si aucun ordre n'est spécifié.
- Un champ "RegleDiff" correspondant à la référence de la règle de diffusion utilisée. Ce champs peut prendre comme valeur : 1 pour "Même écran - séparés", 2 pour "Même écran - collés", 3 pour "Ecrans distincts" et 0 ou vide si aucune règle de diffusion n'est applicable.
- Un champ "~~RegleDiffLib~~" correspondant au libellé de la règle de diffusion à appliquer ("" si non applicable)
- Un champ "~~Jumele~~" indiquant si l'ordre de diffusion est précisé et que la campagne est jumelée. Ce champs prend comme valeur 1 si c'est le cas, 0 sinon.

### ✓ Le fichier XML de message

Ce fichier contient la liste des messages. L'élément suivant a été défini :

- Un champ "NbMessages" correspondant au nombre de messages listés.

#### **ATTENTION**

A partir de la version 2.5 de l'ActiveX, l'élément « message » n'est plus défini par la balise <message> mais par la balise <Message>. Il est également conseillé de traiter les 2 types de balises (Message et message) afin d'assurer la compatibilité avec les fichiers au format « texte ».

Pour chaque message, sont définis les éléments suivants :

- Un champ "Nom" correspondant au nom du produit.
- Un champ "Commentaire" correspondant au commentaire associé au message.
- Un champ "DureeCommerciale" correspondant à la durée commerciale du message en secondes.
- Un champ "IdentifiantMessage" correspondant à l'identifiant Radio Pilot du message.
- Un champ "NumeroRegie" correspondant au numéro de régie du message associé. (*Obsolète*)
- Un champ "Valide" permettant d'indiquer la validité du message. Dans le cas où le message est actif, le champ contient la valeur "1". Dans le cas contraire, le champ contient la valeur "0".

### ✓ Le fichier XML pour le détail des messages

Ce fichier contient la liste des messages. Pour chaque message, sont définis les éléments suivants :

- Un champ "CodeMessage" correspondant au code du message.
- Un champ "Date" correspondant à la date du message dans le routeur, ce champ est au format "YYYYMMDDHHMMSS00".
- Un champ "Taille" correspondant à la taille en octet du message.

### ✓ Le fichier XML conducteur

Ce fichier contient la liste des conducteurs. Pour chaque conducteur, sont définis les éléments suivants :

- Un champ "Date" correspondant à la date concernée par le conducteur sous la forme "aaaammjj"
- Une liste d'éléments "Antenne" décrivant l'antenne concernée par le conducteur. Dans le cadre des transferts automatiques des pré conducteurs, une seule antenne sera présente par fichier Conducteur.

Pour chaque élément Antenne sont définis les champs suivants :

- Un champ "Nom" correspondant au nom de l'antenne concernée. Ce champ devrait toujours prendre comme valeur : "LES INDEPENDANTS" pour les conducteur du produit National et "PUISSANCE CAPITAL" pour les pré conducteurs du produit IDF.
- Une liste d'éléments "Tranche" correspondant aux tranches horaires de la journée décrite.

Pour chaque élément Tranche, sont définis les champs suivants :

- Un champ "Nom" correspondant au nom de la tranche horaire concernée.
- Un élément "SansContrainteDemiHeure" correspondant à la description de la tranche horaire dont les spots n'ont pas de contrainte d'écran. Cet élément peut ne pas être présent dans le fichier XML, si aucun spot ne correspond.
- Un élément "PremierEcran" correspondant à la description de la tranche horaire dont les spots doivent être placés dans le premier écran. Cet élément peut ne pas être présent dans le fichier XML, si aucun spot ne correspond.
- Un élément "DernierEcran" correspondant à la description de la tranche horaire dont les spots doivent être placés dans le dernier écran. Cet élément peut ne pas être présent dans le fichier XML, si aucun spot ne correspond.

Les éléments "SansContrainteDemiHeure", "PremierEcran" et "DernierEcran" sont définis par :

- Une liste d'éléments "Spot" correspondant à la description des spots à diffuser.

Pour chaque élément Spot sont définis les champs suivants :

- Un champ "Date" correspondant à la date de diffusion du spot sous la forme "aaaammjj".
- Un champ "HeureDebut" correspondant à l'heure de début de la tranche horaire de diffusion du spot sous la forme "hh:mm"
- Un champ "HeureFin" correspondant à l'heure de fin de la tranche horaire de diffusion du spot sous la forme "hh:mm"
- Un champ "Rigueur" permettant d'indiquer la position du message dans l'écran. Ce champs contient la valeur :
  - 'T' si le spot doit être placé en tête d'écran
  - '2' si le spot doit être en deuxième position dans l'écran
  - '3' si le spot doit être en deuxième position dans l'écran
  - 'F' si le spot doit être placé en fin d'écran
  - 'A' si le spot doit être placé soit en début, soit en fin d'écran.
  - ' ' (espace) si aucune rigueur n'est retenue
- Un champ "Doublon" permettant de détecter si un spot est présent deux fois dans la même tranche horaire. Dans ce cas, le champ contient le numéro de passage du spot.
- Un champ "DureeCommerciale" correspondant à la durée commerciale du spot exprimée en secondes.
- Un champ "~~LibelleConcurrence1~~" correspondant au libellé du premier secteur d'activité associé au message.
- Un champ "~~LibelleConcurrence2~~" correspondant au libellé du deuxième secteur d'activité associé au message.

- Un champ "CodeConcurrence1" correspondant au code du premier secteur d'activité associé au message.
- Un champ "CodeConcurrence2" correspondant au code du deuxième secteur d'activité associé au message.
- Un champ "IdentifiantSpot" correspondant à l'identifiant du spot.
- Un champ "PositionPreferentielle" indiquant si la position de rigueur est obligatoire (Valeur "O") ou souhaitée par le client (Valeur "S"). Aucune valeur si aucune position n'est demandée.
- Un champ "DoublonMemeEcran" indiquant si le spot est flagué Doublon, dans ce cas, ce champs vaut 1.
- Un champ "DemiHeure" indiquant la demi-heure de la tranche horaire dans laquelle le spot a été vendu. Ce champs peut prendre comme valeur : "P" pour la première demi-heure, "D" pour la deuxième demi-heure et "" sinon.
- Un champ "OrdreDiff" indiquant l'ordre de diffusion pour les spots ou 0 si aucun ordre n'est spécifié.
- Un champ "RegleDiff" correspondant à la référence de la règle de diffusion utilisée. Ce champs peut prendre comme valeur : 1 pour "Même écran - séparés", 2 pour "Même écran - collés", 3 pour "Ecrans distincts" et 0 ou vide si aucune règle de diffusion n'est applicable.
- Un champ ~~"RegleDiffLib" correspondant au libellé de la règle de diffusion à appliquer ("" si non applicable).~~
- Un champ ~~"Jumelo" indiquant si l'ordre de diffusion est précisé et que la campagne est jumelée. Ce champs prend comme valeur 1 si c'est le cas, 0 sinon.~~
- Un élément "Message" correspondant à la description du message à diffuser.
- Un élément "Campagne" correspondant à la description de la campagne de ce spot.
- Un élément "ListeJumelage" à la liste des jumelages de ce spot.

Pour l'élément Message, sont définis les champs suivants :

- Un champ "Nom" correspondant au nom du produit.
- Un champ "Commentaire" correspondant au commentaire associé au message.
- Un champ "DureeCommerciale" correspondant à la durée commerciale du message en secondes.
- Un champ "IdentifiantRadioPilot" correspondant à l'identifiant du message. Ce champ est identique au champ IdentifiantMessage.
- Un champ "NumeroRegie" correspondant au numéro de régie du message associé.
- Un champ "Valide" permettant d'indiquer la validité du message. Dans le cas où le message est actif, le champ contient la valeur "V". Dans le cas contraire, le champ contient la valeur "I".

Pour l'élément Campagne sont définis les champs suivants :

- Un champ "Nom" correspondant au nom de la campagne.
- Un champ "CodeSecteur" correspondant au code du secteur d'activité de la campagne.
- Un champ "Secteur" correspondant au libellé du secteur d'activité de la campagne.
- Un champ "Identifiant" correspondant au numéro d'identification de la campagne.

- Un champ "DateDebut" correspondant à la date de début de la campagne sous la forme "aaaammjj".
- Un champ "DateFin" correspondant à la date de fin de la campagne sous la forme "aaaammjj".
- Un champ "NomAnnonceur" correspondant au nom de l'annonceur de la campagne.
- Un champ "DateDebutSpot" correspondant à la date du premier spot de la campagne sous la forme "aaaammjj".
- Un champ "DateFinSpot" correspondant à la date du dernier spot de la campagne sous la forme "aaaammjj".

L'élément ListeJumelage est défini par une liste de champs :

- "IdCampJumelee" correspondant l'identifiant d'une campagne jumelée.

Vous trouverez en annexe un exemple de pré conducteur.

#### ✓ Le fichier XML CodeConcurrence

Ce fichier contient la liste des codes concurrences valides. L'élément de niveau supérieur est "ListeSecteurs". Pour chaque secteur d'activité "Secteur", sont définis les éléments suivants :

- Un champ "Code" correspondant au code concurrence. Cette valeur est au même format que pour les champs "CodeSecteur" des éléments "Campagne", "CodeConcurrence1" et "CodeConcurrence2" des éléments "Spot".
- Un champ "Libellé" correspondant au libellé du code concurrence.

Exemple :

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
- <ListeSecteurs>
  - <Secteur>
    <Code>00000101</Code>
    <Libelle>Boissons non alcoolisées</Libelle>
  </Secteur>
  - <Secteur>
    <Code>00000102</Code>
    <Libelle>Boissons alcoolisées</Libelle>
  </Secteur>
  ...
</ListeSecteurs>
```

### III. ANNEXE

#### Description d'un pré conducteur XML

```

<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
- <Conducteur>
<Date>20070429</Date>
- <Antenne>
<Nom>LES INDEPENDANTS</Nom>
- <Tranche>
  <Nom>05h00 - 06h00</Nom>
  - <SansContrainteDemiHeure>
    - <Spot>
      <Date>20070429</Date>
      <HeureDebut>05:00</HeureDebut>
      <HeureFin>06:00</HeureFin>
      <Rigueur>T</Rigueur>
      <Doublon />
      <DureeCommerciale>040</DureeCommerciale>
      <IdentifiantSpot>13607429</IdentifiantSpot>
      <CodeConcurrence1>18019999</CodeConcurrence1>
      <CodeConcurrence2>18019999</CodeConcurrence2>
      <PositionPreferentielle>0</PositionPreferentielle>
      <DoublonMemeEcran>0</DoublonMemeEcran>
      <DemiHeure />
      <OrdreDiff>000</OrdreDiff>
      <RegleDiff>0</RegleDiff>
      - <Message>
        <Nom>MAPA - E1</Nom>
        <Commentaire>Mme Niepceron B.427</Commentaire>
        <DureeCommerciale>040</DureeCommerciale>
        <IdentifiantRadioPilot>630298</IdentifiantRadioPilot>
        <NumeroRegie>157962</NumeroRegie>
        <Valide>V</Valide>
      </Message>
    - <Campagne>
      <Identifiant>165369</Identifiant>
      <Nom>MAPA FIL ROUGE IND-185819</Nom>
      <Secteur>Assurances professionnelles (risques
      industriels...)</Secteur>
      <DateDebut>20070301</DateDebut>
      <DateFin>20071230</DateFin>
      <CodeSecteur>18010308</CodeSecteur>
      <NomAnnonceur>MUTUELLE ASSURANCE PROF
      ALIMENT</NomAnnonceur>
      <DateDebutSpot>20070301</DateDebutSpot>
      <DateFinSpot>20071230</DateFinSpot>
    </Campagne>
  </ListeJumelage />
</Spot>
</SansContrainteDemiHeure>
</Tranche>
- <Tranche>
  <Nom>11h00 - 12h00</Nom>

```

```

- <SansContrainteDemiHeure>
  - <Spot>
    <Date>20070429</Date>
    <HeureDebut>11:00</HeureDebut>
    <HeureFin>12:00</HeureFin>
    <Rigueur />
    <Doublon />
    <DureeCommerciale>030</DureeCommerciale>
    <IdentifiantSpot>13721551</IdentifiantSpot>
    <CodeConcurrence1>00000001</CodeConcurrence1>
    <CodeConcurrence2>00000001</CodeConcurrence2>
    <PositionPreferentielle />
    <DoublonMemeEcran>0</DoublonMemeEcran>
    <DemiHeure />
    <OrdreDiff>000</OrdreDiff>
    <RegleDiff>0</RegleDiff>
  - <Message>
    <Nom>KRONENBOURG</Nom>
    <Commentaire>1/Concours rugissement
    (16/04/07)</Commentaire>
    <DureeCommerciale>030</DureeCommerciale>
    <IdentifiantRadioPilot>629842</IdentifiantRadioPilot>
    <NumeroRegie>157537</NumeroRegie>
    <Valide>V</Valide>
  </Message>
  - <Campagne>
    <Identifiant>165630</Identifiant>
    <Nom>KRONENBOURG IND-186194</Nom>
    <Secteur>Boissons alcoolisées</Secteur>
    <DateDebut>20070416</DateDebut>
    <DateFin>20070429</DateFin>
    <CodeSecteur>01020101</CodeSecteur>
    <NomAnnonceur>KRONENBOURG
    BRASSERI</NomAnnonceur>
    <DateDebutSpot>20070416</DateDebutSpot>
    <DateFinSpot>20070429</DateFinSpot>
  </Campagne>
  <ListeJumelage />
</Spot>
</SansContrainteDemiHeure>
</Tranche>
- <Tranche>
  <Nom>23h00 - 24h00</Nom>
  - <SansContrainteDemiHeure>
    - <Spot>
      <Date>20070429</Date>
      <HeureDebut>23:00</HeureDebut>
      <HeureFin>23:59</HeureFin>
      <Rigueur />
      <Doublon />
      <DureeCommerciale>030</DureeCommerciale>
      <IdentifiantSpot>13779784</IdentifiantSpot>
      <CodeConcurrence1>00001805</CodeConcurrence1>
      <CodeConcurrence2>00001805</CodeConcurrence2>
      <PositionPreferentielle />

```

```
<DoublonMemeEcran>0</DoublonMemeEcran>
<DemiHeure />
<OrdreDiff>000</OrdreDiff>
<RegleDiff>0</RegleDiff>
- <Message>
    <Nom>MINISTERE DE L'EQUIPEMENT</Nom>
    <Commentaire>2/Vitesse (15/03/07)</Commentaire>
    <DureeCommerciale>030</DureeCommerciale>
    <IdentifiantRadioPilot>627577</IdentifiantRadioPilot>
    <NumeroRegie>155411</NumeroRegie>
    <Valide>V</Valide>
</Message>
- <Campagne>
    <Identifiant>166839</Identifiant>
    <Nom>MINISTERE DE L'EQUIPEMENT IND-187815</Nom>
    <Secteur>Informations gouvernementales</Secteur>
    <DateDebut>20070315</DateDebut>
    <DateFin>20070507</DateFin>
    <CodeSecteur>18050301</CodeSecteur>
    <NomAnnonceur>MINISTERE DE
    L'EQUIPEMENT</NomAnnonceur>
    <DateDebutSpot>20070315</DateDebutSpot>
    <DateFinSpot>20070507</DateFinSpot>
</Campagne>
<ListeJumelage />
</Spot>
</SansContrainteDemiHeure>
</Tranche>
</Antenne>
</Conducteur>
```

**Description de la valeur de chaque type d'élément composant un écran publicitaire pour la vérification des positions imposées**

Les positions préférentielles vendues par la régie publicitaire doivent être respectées. Seuls les éléments de type Virgule(6) ne comptent pas lors de la vérification du placement des spots. Il est cependant autorisé, dans le cas d'une fin d'écran, de faire suivre le spot imposé(FE) par un élément sonore sans citation annonceur (3).

Le tableau ci-dessous décrit les différents types de spots valides et leur utilisation lors de la vérification du placement par le GIE

TypeSpot	Libellé	Ancienne valeur	Description
0	Publicité nationale		
1	Publicité locale		
2	Elément sonore avec citation annonceur	Autos-promos	Compte comme une publicité
3	Elément sonore sans citation annonceur	Comblage	Compte pour 0 dans le calcul du respect des rigueurs de positionnement si l'élément est en fin d'écran. Un élément de ce type peut être placé derrière une fin d'écran mais pas devant une tête d'écran
5	Publicité Multiville		
6	Virgule		Compte pour 0 dans le calcul du respect des rigueurs de positionnement. Un élément de ce type peut être placé devant une tête d'écran, derrière une fin d'écran...

### Détail des éléments imposés lors du pointage des spots

Afin de contrôler au maximum la diffusion des publicités nationales par les radios, le GIE « Les Indépendants » a décidé d'imposer la remontée d'informations cohérentes lors du pointage des spots pour les champs suivants des fonctions **PointageDeSpot** et **PointageDeSpotCC** décrites plus haut :

Nom du paramètre	Description	
DateDeDiffusion	Date de diffusion du spot	AAAAMMJJ
HeureDeDiffusion	Heure de diffusion du spot	HHMMSS
IdSpot	Identifiant du spot	
IdMessage	Identifiant du message diffusé	
EtatDeDiffusion	Etat de diffusion du spot	0 : Non diffusé 1 : Diffusé <b>2-3-4 : Jamais utilisé (anomalie mineure, majeure, spot non prévu...) =&gt; Obsolète</b>
DureeDuSpotDiffuse	Durée réelle du spot diffusé	
CodeEcran	Code de l'écran dans lequel a été diffusé le spot	Il doit s'agir d'un code alphanumérique identifiant de manière unique l'écran au minimum dans l'heure et au mieux dans la journée (exemple N° de l'écran dans l'heure ou dans la journée)
HeureDeDiffusionEcran	Heure de diffusion de l'écran dans lequel a été diffusé le spot	HHMMSS
NbSpotsEcran	Nombre de spots de l'écran dans lequel a été diffusé le spot	>0
PlaceDuSpotDansEcran	Position du spot dans l'écran dans lequel il a été diffusé	0<Position<=NbSpotsEcran
DureeEcran	Durée totale de l'écran publicitaire	En millisecondes
TypeSpot	Type du spot diffusé	0 : Publicité Nationale 1 : Publicité Locale 2 : Elément sonore avec citation annonceur 3 : Elément sonore sans citation annonceur 4 : Editorial (Rubriques, météo) ( <b>obsolète : considéré comme un élément sonore sans citation annonceur</b> ) 5 : Multiville 6 : Virgule
CodeConcurrence	Code concurrence du spot	Ce code doit faire partie de la liste des codes concurrences définis par la régie (présents dans la liste des codes concurrence présente sur le routeur). La valeur de ce code peut être indifféremment du type complété à droite ou complété à gauche par des 0. 101 = 1010000

## **Pige publicitaire des codes concurrence**

La fonction PointageDeSpotCC prend en dernier paramètre le code concurrence du spot pointé.

La liste des codes concurrences valides est disponible sur chaque routeur GIE (machine de réception publicitaire) dans le répertoire Brut (s'il n'a pas été importé) ou dans le répertoire Config (après son importation). Il peut être récupéré en utilisant les fonctions NouvelleListeCodeConcurrence et ImporterListeCodeConcurrence.

Il est à noter que le code concurrence est un numéro composé de 8 chiffres incluant le code famille et éventuellement les codes classe, secteur et variété. Dans la mesure où le numéro proposé ne correspond pas à 8 chiffres, le code informatique utilisé dans le flux de diffusion de la régie et pour la pige publicitaire est complété de chiffres « 0 » se positionnant devant les chiffres des codes familles et éventuellement les codes classe, secteur et variété. Concernant le pointage des spots, la pige acceptera indifféremment les codes concurrence complétés à droite ou à gauche par des '0'. Par exemple, pour un spot de secteur Alimentation, les codes concurrence suivants sont considérés comme équivalents dans la pige: 2 = 2000000 (=00000002 = 02000000 si on affiche les 0 non significatifs)

Toute remontée de logs informatiques comportant un code concurrence ne figurant pas dans la table de codes concurrence secteur radio mise à jour, sera traitée en incident de diffusion « erreur code concurrence ».

Afin de permettre une vérification du respect des codes concurrence, il est nécessaire que le champ suivant soit indiqué pour tout élément composant un écran publicitaire de TypeSpot 0 (facultatif car connu depuis les fichiers exportés par la régie), 1, 2 et 5.

Les éléments de TypeSpot 1, 2 et 5 pour lesquels le champ CodeConcurrence n'est pas renseigné seront traités en incident de diffusion « erreur code concurrence ». Il est à noter que dans la mesure où le paramètre CodeConcurrence serait renseigné pour des éléments de TypeSpot 3 et 6, ce paramètre ne serait pas pris en compte dans la pige publicitaire.

## **Règles afférentes aux codes concurrence et principes de la pige publicitaire**

Les incidents de diffusion codifiés « erreur code concurrence » correspondent aux cas suivants :

- Deux spots nationaux collés avec même CodeConcurrence ou séparés (i) d'un élément publicitaire avec un code concurrence distinct ou (ii) d'un élément non publicitaire (TypeSpot 3 ou 6) inférieur à 3/5 secondes.
- Un spot national et un spot local collés avec même CodeConcurrence ou séparés (i) d'un élément publicitaire avec un code concurrence distinct ou (ii) d'un élément non publicitaire (TypeSpot 3 ou 6) inférieur à 3 secondes.
- Un spot national et un spot extra-local collés avec même CodeConcurrence ou séparés (i) d'un élément publicitaire avec un code concurrence distinct ou (ii) d'un élément non publicitaire (TypeSpot 3 ou 6) inférieur à 3 secondes.
- Un spot national et un spot d'Auto-Promotion AVEC citation annonceur collés avec même CodeConcurrence ou séparés (i) d'un élément publicitaire avec un code concurrence distinct ou (ii) d'un élément non publicitaire (TypeSpot 3 ou 6) inférieur à 3/5 secondes.
- CodeConcurrence pour les spots nationaux, locaux, Auto-promotion, extra-locaux ne figurant pas dans la table des codes concurrence secteur radio ou non renseigné.

- Deux spots extra-locaux collés avec même CodeConcurrence ou séparés (i) d'un élément publicitaire avec un code concurrence distinct ou (ii) d'un élément non publicitaire (TypeSpot 3 ou 6) inférieur à 3/5 secondes (\*).
- Un spot extra-local et un spot local collés avec même CodeConcurrence ou séparés (i) d'un élément publicitaire avec un code concurrence distinct ou (ii) d'un élément non publicitaire (TypeSpot 3 ou 6) inférieur à 3/5 secondes (\*).
- Un spot extra-local et un spot d'Auto-Promotion AVEC citation annonceur collés avec même CodeConcurrence ou séparés (i) d'un élément publicitaire avec un code concurrence distinct ou (ii) d'un élément non publicitaire (TypeSpot 3 ou 6) inférieur à 3/5 secondes (\*).

(\*) Pige publicitaire à prévoir dans l'automatisation de la diffusion Multiville

Par dérogation aux principes énoncés, ne sont pas traités en erreur de diffusion :

- Les écrans collés (MEC) codifiés dans la diffusion nationale transmis par la régie ne doivent pas être considérés comme concurrent, s'agissant d'un mode de diffusion souhaité par l'annonceur.
- Les spots nationaux collés avec même CodeConcurrence et emplacements préférentiels TE1 et TE2 codifiés dans la diffusion nationale transmis par la régie.

Exemple :

TE1 – spot national – Code Secteur 00000002 – Alimentation

TE2 – spot national – Code Secteur 00000002 – Alimentation

Dans le cas où, les spots nationaux auraient été codifiés TE2 et TE3, le spot codifié TE2 aurait dû être « surclassé » en TE1.

# Bibliographie

➤ <http://www.chez.com/xml/initiation/index.htm>

## Version :

- 3.4.0 : Ajout de fonctions permettant le traçage du remplacement du contenu sonore d'un message national.
- 3.3.0 : Correction des fonctions d'importation des conducteurs nationaux.
- 3.2.0 : Modification interne de l'ActiveX PubImport afin de permettre le traitement de fichier type Multiville.  
  
Suppression de champs inutiles dans la description des spots des fichiers Campagne et Conducteur (LibelleConcurrence1, LibelleConcurrence2, RegleDiffLib, Jumele)
- 3.1.0 : Ajout de la fonction PointageDeSpotCC  
  
Ajout des fonctions NouvelleListeCodeConcurrence et ImporterListeCodeConcurrence permettant la récupération de la liste des codes concurrences exportées par la régie.
- 3.0.1 : Ajout du détail des éléments nécessaires imposés lors du pointage des spots. Ajout de la valeur des types d'élément lors du pointage des spots. Modification des valeurs des types de spots et leur description dans la fonction PointageDeSpot. Passage en « Obsolète » des anomalies de diffusion (anomalie mineure et majeure). Ajout de la notion d'écran écran dans le champ DemiHeure de l'élément Spot.
- 3.0 : Ajout de la liste des campagnes jumelées à la description de l'élément Campagne. Ajout de champs DemiHeure, OrdreDiff, RegleDiff, RegleDiffLib et Jumele à la description des éléments Spot. Ajout des fonctions permettant l'importation des Pré Conducteurs envoyés par LAP.
- 2.5 : Modification interne du fonctionnement de l'ActiveX permettant l'importation des fichiers campagnes et fiche message au format XML.
- 2.3 : Ajout des fonctions génériques permettant le traitement de tous les fichiers de type CSE (National GIE, IDF, Autoroute Info...)
- 2.2 : Ajout des fonctions et champs permettant de connaître le contenu quantitatif du XML obtenu (nombre de messages, de campagnes, de spots par campagne et de spots pointés)
- 2.1 : Ajout des fonctions AutorouteFM.
- 2.0 : Ajout des fonctions Ile-De-France. Ajout des fonctions de Visualisation. Ajout des fonctions ImporterCampagne et ImporterFicheMessage.

- 1.1 : Modification de la description du champ doublonmêmedecran.
- 1.0 : Constitution du document.