Manuel Configuration AFS

Version AFS :	6.5
Version de documentation :	6.5-0.12
Date :	24/08/2007
Référence :	AFS/DOC/MANCONF

Table des matières

Table des matières

1 Introduction....3
1.1.Choix des fonctionnalités AFS....4
2 Indexation d'une source de données....5
2.1 Configuration technique de l'environnement d'indexation....5
2.1.1 Section Base....6
2.1.2 Section Servers....6
2.1.3 Section Crawling....7
2.1.4 Section Indexation....8
2.1.5 Section Reverse....15
2.1.6 Section Repositories....15
2.2 Configuration d'une indexation Web....16
2.2.1 Récupération des données à indexer : le fichier perimeter.xml....16
2.2.1.1 Configurer la fréquence de crawl....22
2.2.2 Prétraitement des données....23
2.2.2.1 Définir des sites comme valeur de filtre....23
2.2.3 Analyse fine des pages HTML : Les Fichiers aihtml....24
2.2.3.1 Structure des fichiers....24
2.2.3.2 Les règles de traitement aihtml....25
2.3 Configuration de l'indexation d'une base de données....27
2.3.1 Récupération des données : Le fichier aidbml....27
2.4 Configurer l'indexation d'un flux structuré XML....30
2.4.1 Prétraitement des données : Le fichier aixml....31
2.4.1.1 Les sections d'un fichier aixml....32
2.4.2 Indexation des données structurées et déclaration de filtres paramétriques : le fichier shmxml....35
3 Créer un environnement de réponse....39
3.1 Configuration....39
3.1.1 Fichier afs.xml....39
3.1.1.1 Les sections pour les paramètres génériques....40
3.1.1.3 Les sections spécifiques au front-end AFS....45
1.1.1.1.Les section spécifiques au back-office AFS....63
3.1.2 Fichier Services.conf....64
4 Intégration des fonctionnalités AFS....67
4.1.1 Application de méthodes linguistiques lors de l'indexation....67
4.1.1.1 Indexation avec gestion des flexions....67
4.1.1.2 Indexation avec thésaurus....68
4.1.1.3 Indexation avec métaphone....70
4.1.2 Generation d'une base d'expressions RTE :....71
4.1.2.1 Configuration de l'indexation....71
4.1.2.2 Configuration du service de réponse....71
4.1.3 Implantation d'agents spécifiques....72
4.1.3.1 Agent de suggestion orthographique (agent Hint) :....72
4.1.3.1.1 Configuration du service de réponse....72
4.1.4 Recherche transversale : ACC (Automatic Across Content)....72
4.2 Mise à jour des index en temps réél....72

1 Introduction

Ce manuel décrit les différentes façons de configurer AFS pour indexer des sources de données non structurées de type sites web (Internet ou Intranet) ou des sources de données structurées telles que des flux XML et des bases de données.

Il explique également comment configurer un service de recherche sur les sources indexées, comment créer une feuille de style XSL permettant de transformer un flux de réponse XML AFS en un autre format XML ou en une page HTML. Il explique enfin comment installer une boîte de recherche sur un site Web puis décrit une méthodologie permettant d'évaluer et de comparer différents réglages de pertinence.

L'enchaînement de ces différentes étapes permet la mise en œuvre effective du moteur de recherche. Il est cependant nécessaire qu'AFS ait préalablement été installé et configuré tel que décrit dans le Manuel Administrateur AFS.

Par ailleurs, il est vivement conseillé au lecteur de lire les premiers chapitres du Manuel Administrateur qui présentent les fondements d'AFS (architecture conceptuelle, logique et physique), les principaux concepts et certains aspects de la configuration.

L'ensemble de ces aspects de la configuration AFS se fait à travers des fichiers XML qui sont éditables soit directement avec un éditeur de texte, soit à travers les interfaces du back-office. L'objectif de ce document est de décrire le format des différents types de fichiers de configuration AFS.

1.1.Choix des fonctionnalités AFS

Avant même de commencer la phase de configuration de l'indexation, il est important de réunir certaines informations, pour la plupart dépendantes des **fonctionnalités à mettre en œuvre** :

- application de méthodes linguistiques lors de l'indexation : indexation exacte, application d'un dictionnaire de langue, indexation phonétique, recherche floue, ...
- catégorisation des réponses, recherche multicritères, recherche par facette ou paramétrique ;
- suggestion orthographique, extension automatique ;
- extraction des concepts liés (Related Topic Engine) ;
- calcul de la similarité entre les informations pour offrir des fonctions de navigation transversale (Automatic Cross Content).

Pour une bonne compréhension des fonctionnalités, il est recommandé de se reporter au document de présentation des "Fonctionnalités AFS" (***référence AFS/WP/FONCS v6.5***).

Les aspects de **catégorisation et** de **recherche multicritères ou paramétrique** sont particulièrement importants dans la mesure où il convient préalablement à la phase de configuration de **lister les différents filtres qui seront utilisés**. En effet, certains de ces filtres (comme le type de document) sont natifs et sont donc activables dans la configuration, alors que d'autres sont spécifiques à chaque corpus et ils doivent donc être définis.

2 Indexation d'une source de données

Pré-requis : Afin de bien comprendre les différentes étapes de ce traitement, il est nécessaire au préalable d'avoir lu le ***chapitre 1.2.1 du manuel Administrateur*** qui présente les différents composants fonctionnels de l'indexation.

On peut avoir plusieurs formats de source de données :

- Des données semi structurées issues du Web (http)
- Des données structurées issues d'une base de données (SQL)
- Des données structurées au format XML

Pour chacune de ces sources, l'indexation se fait en quatre étapes :

– la récupération des données à indexer

– le prétraitement des données

– l'indexation

– la génération des bases de réponse.

2.1 Configuration technique de l'environnement d'indexation

La configuration technique de l'indexation est définie au sein du fichier afs.xml qui se trouve dans le répertoire conf de l'environnement d'indexation.

Ce fichier `afs.xml` a pour noeud racine <AFS> et les sections principales sont les suivantes :

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no" ?>
<AFS>

   <Base>
          <!-- environnement de base d'AFS  -->
   </Base>

   <Servers>
          <!-- définition précise de chacun des services de recherche -->
   </Servers>


   <Crawling>
          <!-- paramètres techniques pour les ressources de type Crawlers et
Indexers -->
   </Crawling>

   <Indexation>
          <!-- paramètres techniques des différents modules du back-end -->
   </Indexation>

   <Reverse>
          <!-- paramètres propres à la génération des bases d'index de réponse -->
   </Reverse>

   <Repositories>
          <!-- liste des tables SQL stockant les informations de filtrage spécifique
-->
   </Repositories>

</AFS>
```

NB : Il n'est pas nécessaire de renseigner chacune des options si les valeurs par défaut sont utilisées.

2.1.1 Section Base

La section Base peut contenir les définitions suivantes :

```
<Base>
  <!-- Faux si les modules back-end sont tous situés sur le même serveur -->
  <Clustered_Service>false</Clustered_Service>

  <!-- Pour définir la langue. Valeur par défaut : fr_FR -->
  <Locale>fr_FR@euro</Locale>

  <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
  <!-- permet de fixer quelques limites et les droits au niveau du système -->
  <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
  <OS>
     <!-- la charge maximale (load) autorisée. En cas de dépassement, AFS arrête
          de lancer des agents -->
     <Max_Load>50</Max_Load>

     <Privileges>
        <!-- Permet de forcer le user et le group des process AFS à partir
          de leurs noms (setuid, setgid) -->
        <Ownership>
              <!-- Utiliser la valeur false permet de ne plus activer les droits sans
                       supprimer cette section de configuration -->
           <Force_Ownership>true</Force_Ownership>
           <User_Name>antidot</User_Name>
           <Group_Name>antidot</Group_Name>
        </Ownership>
     </Privileges>

     <!-- nombre maximum de process AFS qui peuvent être lancés. La valeur par
              défaut est 980. -->
     <Max_Processes>4096</Max_Processes>
  </OS>

  <!-- Temps d'attente maximum (en secondes) pour un ping applicatif. Défaut=3 -->
  <Ping_Timeout_Seconds>3</Ping_Timeout_Seconds>

  <!-- Temps maximum (en secondes) pour envoyer des données. Défaut=30 -->
  <Default_Timeout_Seconds>20</Default_Timeout_Seconds>

  <!-- Temps d'attente (en secondes) entre deux cycles successifs de vérification
       que tous les serveurs ont assez de composants actifs.
      A ajouter à Fork_Loop_Delay_Milliseconds. Défaut=15 -->
  <Fork_Loop_Delay_Seconds></Fork_Loop_Delay_Seconds>

  <!-- Temps d'attente (en millisecondes) entre deux cycles successifs
          de vérification que tous les serveurs ont assez de composants actifs.
          A ajouter à Fork_Loop_Delay_Seconds. Défaut=0 -->
  <Fork_Loop_Delay_Milliseconds></Fork_Loop_Delay_Milliseconds>

</Base>
```

2.1.2 Section Servers

La section Servers contient le champ **<Base>** qui permet de définir le numéro de port à partir duquel les différents agents se lanceront.

```
  <Servers>
     <Base>2000</Base>
  </Servers>
```

2.1.3 Section Crawling

La section Crawling permet de définir les options de configuration pour les crawleurs ainsi que le nombre de ressources de crawleurs et d'indexeurs par hôte. Si le back-end ne nécessite pas de crawl, il est nécessaire de préciser dans la sous-section Resources que '0' crawleur est impliqué, et d'indiquer le nombre d'indexeurs désiré pour l'indexation.

```
<Crawling>

   <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
   <!-- Les options de configuration des crawlers -->
   <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
   <Crawlers>
      <!-- Nombre maximum de documents à crawler avant sortie du crawleur.
       Défaut=1000 -->
      <Max_Iterations>5000</Max_Iterations>

      <!-- Temps d'attente maximal pour obtenir un document. Défaut=60 -->
      <Abort_After_Seconds>60</Abort_After_Seconds>

      <!-- Nom alternatif de l'agent crawleur pour cacher 'antibot'.
           Défaut=Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR
1.1.4322) -->
      <Alternate_User_Agent></Alternate_User_Agent>

      <!-- Plugin de transformation avant sauvegarde -->
      <Transformation_Plugin>
         <!-- Nom de la bibliothèque dynamique de transformation à charger -->
         <Dynamic_Library_Name>libclient_html2xml.so</Dynamic_Library_Name>

         <!-- Nom de la fonction dans la bibliothèque à utiliser.
              Défaut=user_transform -->
         <Function_Name>html_to_xml</Function_Name>
      </Transformation_Plugin>
   </Crawlers>


   <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
   <!-- Définir le nombre de ressources par hôte  -->
   <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
   <Resources>

      <!-- Nombre de crawleurs sur un hôte, format nb@host -->
      <Crawlers>4@localhost</Crawlers>

      <!-- Nombre d'indexeurs sur un hôte, format=nb@host -->
      <Indexers>6@localhost</Indexers>

      <!-- Désactivation du crawl dans cet intervalle de temps,
           format=Mon:10H00-Fri:20H00-->
      <No_Crawling_Between></No_Crawling_Between>

      <!-- Désactivation de l'indexation dans cet intervalle de temps,
           format=Mon:10H00-Fri:20H00-->
      <No_Indexation_Between></No_Indexation_Between>
   </Resources>
```

```
</Crawling>
```

2.1.4 Section Indexation

La section Indexation possède une partie d'options relatives à cette étape, et des sous-sections dédiées à la configuration des managers (Search Data Manager, URLs Manager, Document Manager, Word Manager, Index Manager, Language Checker) et des indexeurs.

```
  <Indexation>
    <!-- - - - - - - - - - -->
    <!-- Options communes  -->
    <!-- - - - - - - - - - -->

    <!-- Taille de hachage utilisée pour optimiser l'écriture de fichiers
         sur disque. Défaut=11
         Il est conseillé d'avoir une taille de hachage impaire,
         petite (1 ou 3) pour les très petits volumes indexés
         plus grande (31) pour les volumes conséquents
         et de conserver la valeur par défaut (11) pour les autres cas -->
    <Hash_Size>11</Hash_Size>

    <!-- Taille de hachage pour la base de données maître (freshredirect).
         Défaut=63 -->
    <Master_Hash_Size>63</Master_Hash_Size>

    <!-- Taille du cache en Mbits pour les base Berkeley DB. Défaut=64 -->
    <Bdb_Cache_Size_Mb>64</Bdb_Cache_Size_Mb>

    <!-- Choix de normalisation des mots indexés. Défaut=stem.
         Choix possibles :
         stem_add : gestion des flexions (utilise stem.db si disponible)
         stem_replace : lemmatisation des formes
         fuzzy_european : algorithme s'appuyant sur la consonnance des mots
         skos-thesaurus : extensions sémantiques
         reference : tolerance sur les références  -->
    <Normalization_Chain>
       <Normalizer>skos-thesaurus</Normalizer>
       <Normalizer>stem_add</Normalizer>
    </Normalization_Chain>

    <!-- Détecte et supprime l'information de spam sur une page Web. Défaut=false --
  >
    <Remove_Spam>false</Remove_Spam>


    <!-- Type de filtre pour le contenu Web. Défaut=country.
         Choix possibles : country, doctype, user (pour désactiver) -->
    <Engine_Flag_Type>country</Engine_Flag_Type>

    <!-- Informations supplémentaires à stocker en SHM. Nécessite que l'option SHM
   soit activée dans l'environnement de réponse -->
   <Extra_Engine_Flags>
     <!-- Filtre par site -->
      <Enable_Site>true</ Enable_Site>
     <!-- Filtre par doctype -->
      <Enable_Doctype>true</ Enable_Doctype>
      <!-- Filtre par langue -->
      <Enable_Lang>true</ Enable_Lang>
```

```
  </Extra_Engine_Flags>

    <!-- Données supplémentaires stockées par les plugins sont gérées
         lorsque l'option est activée. Défaut=false -->
    <Engine_Store_Extra_Data>false</Engine_Store_Extra_Data>


    <!-- Nombre maximal de positions par mot à mémoriser. Defaut=7
         Pour être efficace (n+1) devrait être multiple d'une puissance de 2
         (n=7, n=15, n=31...)-->
    <Max_Positions_Per_Word>7</Max_Positions_Per_Word>


    <!-- Nombre de bits utilisés pour stocker les positions des mots. Défaut=12.
         Les valeurs possibles sont 12 (4096 positions) à 16 (65536 positions).
         Les mots après ces positions sont indexés mais non disponibles pour
         une recherche exacte -->
    <Number_Of_Bits_Per_Position>12</Number_Of_Bits_Per_Position>


    <!-- Longueur maximale d'une entrée dans l'index. Défaut=4000
         Si elle est dépassée, une nouvelle entrée pour le mot est créée.
         A utiliser avec prudence !
         Suggestion : choisir une longueur proche de la taille
         d'un bloc de disque (4096), et laisser de la place à la dernière occurrence
         pour déborder -->
    <Max_Record_Length>4000</Max_Record_Length>


    <!-- Régler la gestion de poids lorsqu'un mot a plusieurs occurrences
         dans un même document -->
    <Advanced_Multiple_Occurences_Bonus>

       <!-- Active la gestion de poids de mots (recommandé). Défaut=false -->
       <Enabled>true</Enabled>

       <!-- Si 2 occurrences ont le même poids,
            alors l'augmenter de ce pourcentage. Défaut=3
            Effectif si Enabled=true. -->
       <Same_Weight_Bonus_Percent>3</Same_Weight_Bonus_Percent>

       <!-- Si 2 occurrences ont des poids différents, alors garder le meilleur
  poids
            augmenté de ce pourcentage du plus petit poids. Défaut=1
            Effectif si Enabled=true. -->
       <Different_Weight_Bonus_Percent>1</Different_Weight_Bonus_Percent>

    </Advanced_Multiple_Occurences_Bonus>


    <!-- - - - - - - - - - - - - - - - - - - -->
    <!-- Options du Search Data Manager  -->
    <!-- - - - - - - - - - - - - - - - - - - -->
   <!-- Ordonnanceur des crawls et des indexations. Lance les services. -->
   <Search_Data_Manager>
      <!-- Durée maximum pendant laquelle un serveur est bloqué en attente de
       réponse à son robots.txt -->
      <Max_Server_Lock_Duration_Seconds/> <!-- entier : 600 -->

      <!-- Fréquence des mises à jour des status des mots. -->
      <Status_Word_Refresh_Seconds/> <!-- entier : 30 -->

      <!-- Réparer automatiquement le cache après Nb secondes sans rien à indexer.
       0 pour désactiver. -->
```

```
      <Fix_After_Nb_Idle_Seconds/> <!-- entier : 400 -->

      <!--  -->
      <Exit_After_Nb_Idle_Seconds/> <!-- Termine les sessions après Nb secondes
       consécutives. Devrait être au moins deux fois Fix_After_Nb_Idle_Seconds. -->

      <!-- Ignorer les fichiers robots.txt pour accéder à des pages interdites. À
       UTILISER AVEC PRÉCAUTION. -->
      <Ignore_Robots_Txt/> <!-- booléen : false -->
    </Search_Data_Manager>

    <!-- - - - - - - - - - - - - - -->
    <!-- Options du URLs Manager -->
    <!-- - - - - - - - - - - - - - -->
    <!-- Gestionnaire d'URLs -->
    <URL_Manager>
      <Server>
        <!-- Driver (MySQL seulement pour le moment) -->
        <Driver/> <!-- string : 'mysql' -->

        <!-- Hôte SGDB -->
        <Db_Host/> <!-- string : localhost -->

        <!-- Utilisateur SGBD -->
        <Login/> <!-- string -->

        <!-- Password SGDB -->
        <Password/> <!-- string-->
      </Server>

      <Tables>
        <!-- Table des pages -->
        <Pages/> <!-- string ~ 'MY_DATABASE.MY_TABLE' -->

        <!-- Table des paths -->
        <Paths/> <!-- string -->

        <!-- Table des servers -->
        <Servers/> <!-- string -->

        <!-- Table de crawling -->
        <Crawl/> <!-- string -->

        <!-- Table d'indexation -->
        <Index/> <!-- string -->

        <!-- Table des chaines -->
        <Strings/> <!-- string -->

        <!-- Table DMOZ (optionnel) -->
        <DMOZ/> <!-- string : '' -->
      </Tables>

      <Broken_Links>
        <!-- Nombre maximum d'événements stockés pour chaque URL cassée -->
        <History_Window_Size/> <!-- entier : 2 -->

        <Analysis_Tables>
          <!-- Cibles des liens cassés -->
          <To/> <!-- string -->

          <!-- Sources des liens cassés -->
          <From/> <!-- string -->
```

```
            <!-- Paire (from,to) -->
            <Links/> <!-- string -->

            <!-- Historique des liens cassés -->
            <History/> <!-- string -->

            <!-- Statistiques des liens cassés -->
            <Stats/> <!-- string -->
          <Analysis_Tables>
          <Result_Database>
            <!-- Nom de la base -->
            <Name/> <!-- string : 'BROKENLINKS' -->

            <!-- Hôte -->
            <Host/> <!--  -->

            <!-- Utilisateur -->
            <Login/> <!-- string : antisearch -->

            <!-- Mot de passe -->
            <Password/> <!-- string  -->
          </Result_Database>
          <Cache>
            <!-- Durée, en secondes, entre deux requêtes -->
            <Sleep_Time_Seconds/> <!-- entier : 30 -->

            <!-- Nombre d'URLs à demander au cache à chaque requête -->
            <Fetch_Window_Size/> <!-- entier : 1000 -->
          </Cache>
          <Links>
            <!-- Analyse des liens -->
            <Read_Window_Size_Mb/> <!-- entier : 32 -->
          </Links>
          <Tuning>
            <!-- Récupérer les urls à crawler dans des exports 'batch', pas dans la
             table SQL. Recommandé pour les gros volumes. -->
            <Batch_Crawl/> <!-- booléen : false -->

            <!-- Ajouter automatiquement les URLs trouvées lors de l'indexation.
             Déconseillé pour de gros volumes. -->
            <Auto_Add_URLs/> <!-- booléen : true -->

            <!-- Ajoute les nouvelles URLs après N cycles. Si 0, les URLs sont
             ajoutées seulement si il n'y a plus d'URL à indexer. -->
            <Add_URLs_Every_Nb_Cycles/> <!-- entier : 1 -->

            <!-- Si les pages de la base de données sont vides, lancer avec -E pour
             exporter les bases de données des pages/hosts/paths (nécessaire pour
             l'analyse des liens textuels. -->
            <Auto_Export_Pages/> <!-- booléen : true -->

            <!-- Fréquence de mise à jour des statistiques : une donnée tous les Nb
             cycles. -->
            <Update_Stats_Every_Nb_Cycles/> <!-- entier : 1 -->

            <!-- Si vrai, le processus d'inversion peut être complété sans links ni
             hostaliases. -->
            <Allow_Engine_Without_Links/> <!-- booléen : false -->

            <!-- Dans le mode plugin base de donnée, nombre d'URLs à ajouter à une
             seule fenêtre (après lesquelles les caches sont nettoyés. -->
            <Database_Plugin_Add_Window_Size/> <!-- entier : 500000 -->
          </Tuning>
          <Plugins>
```

```
            <!-- Plugin utilisateur pour la connexion à la base de données. Doit se
             trouver dans $AFS/plugins/$SYS_ID -->
            <Db_Connector/> <!-- string : libdatabase_plugin.so -->
          </Plugins>
       </Broken_Links>
    </URL_Manager>

    <!-- - - - - - - - - - - - - - - - -->
    <!-- Options du Document Manager -->
    <!-- - - - - - - - - - - - - - - - -->
    <Document_Manager>
       <!-- Taille maximum d'un document en Mbytes, 0 pour illimité. -->
       <Max_Document_Size_KBytes/> <!-- entier : 0 -->

       <!-- Temps maximum d'attente d'une requête. -->
       <Timeout_Seconds/> <!-- entier : 10 -->

       <!-- Paramétrer le nombre de connexion SQL durant l'indexation-->
       <Nb_Tasks/> <!-- entier : 10 -->

       <!-- Associe un suffixe de fichier (avec le '.') au type mime. Par exemple,
        .pdf pour application/pdf -->
       <Mime_Type_Mapping/> <!-- string_map -->

       <!-- Si non vide, utilise ce chemin pour le chemin du cache. -->
       <Cache_Absolute_Path/> <!-- string -->

       <!-- Si vrai, le cache est localisé sur le système de fichiers local. Le
        positionner à faux, si au moins un des systèmes de fichiers n'est pas local.
        -->
       <Local_Cache/> <!-- booléen : true -->

       <!-- (Tuning) Nombre maximum de documents crawlés qui peuvent attendre leur
        ACK. -->
       <Max_Crawl_Ack_Queue_Size/> <!-- entier : 10000 -->

       <!-- Si vrai, utiliser le lock du système de fichier avant de les modifier.
        Utile dans un mode multi-services/processus. -->
       <Lock_Files/> <!-- booléen : false -->

       <!-- Si positionné à une valeur non nulle, sortira après cette durée quoi
        qu'il arrive. Peut être utilisé comme un palliatif aux locks intempestifs. --
        >
       <Exit_After_Seconds/> <!-- entier : 0 -->

    </Document_Manager>

    <!-- - - - - - - - - - - - - - - - -->
    <!-- Options du Word Manager -->
    <!-- - - - - - - - - - - - - - - - -->
    <Word_Manager>
       <!-- Nombre de tâches de réponses -->
       <Nb_Tasks/> <!-- entier : 10 -->

       <!-- Taille du cache en Mo -->
       <Cache_Size_Mb/> <!-- entier : 32 -->

       <!-- Désactive les nettoyages synchronisés après chaque mise à jour des
        systèmes (Dangereux ! À n'utiliser que pour des systèmes de fichiers fiables.
        -->
       <Disable_Synchronous_Flush/> <!-- booléen : false -->
    </Word_Manager>

    <!-- - - - - - - - - - - - - - - - -->
```

```
    <!-- Options de l'Index Manager  -->
    <!-- - - - - - - - - - - - - - - - -->
    <Index_Manager>
      <!-- Nombre de tâches de réponse. -->
      <Nb_Tasks/> <!-- entier : 3 -->

      <!-- Nombre d'uploads avant nettoyage -->
      <Max_Uploads/> <!-- entier : 10 000 -->

      <!-- Taille maximum autorisée pour un buffer en cours d'utilisation dans un
       indexer. Utiliser 0 pour illimté. -->
      <Max_Work_Buffer_Size_MBytes/> <!-- entier : 512 -->

      <!-- Durée au bout de laquelle on vide si on a rien reçu. Doit être inférieur
       à la moitié du Fix_After_Idle_Seconds du Search Data Manager. -->
      <Flush_After_Idle_Seconds/> <!-- entier : 60 -->

      <!-- Après ce nombre de nettoyage, le manager sort afin de libérer de la
       mémoire et d'être proprement redémarré par le Search Data Manager -->
      <Max_Flushes/> <!-- entier : 1 -->

      <!-- À utiliser avec précaution ! Cette option supprime le stockage de l'index
       et devrait être utilisé pour la découverte d'Url (mode web). -->
      <Discard_Index_Data/> <!-- booléen : false -->
    </Index_Manager>

    <!-- - - - - - - - - - - - - - - - -->
    <!-- Options du Language Checker -->
    <!-- - - - - - - - - - - - - - - - -->
    <Language_Checker>
      <!-- Nombre de tâches de vérification. -->
      <Nb_Tasks/> <!-- entier : 10 -->
    </Language_Checker>

    <!-- - - - - - - - - - - - - - -->
    <!-- Options des indexeurs -->
    <!-- - - - - - - - - - - - - - -->
    <Indexers>
      <!-- Nom du logiciel d'indexation -->
      <Binary_Name/> <!-- as_webindex -->

      <!-- Les indexeurs doivent-ils filtrer individuellement par rapport au
       périmètre les urls trouvées (sinon l'URL Manager le fera plus tard). -->
      <Check_Perimeter/> <!-- booléen : true -->

      <!-- Nombre maximum de documents indexés par un indexer -->
      <Max_Iterations/> <!-- entier : 1000 -->

      <!-- Arrête l'indexation après cette durée. -->
      <Abort_After_Seconds/> <!--  entier : 120 -->

      <!-- Taille pré-allouée pour les strings pour optimiser les allocations. -->
      <Pre_Alloc_Size_Bytes/> <!-- entier : 65536 -->

      <Storage>
        <!-- Longueur maximum d'un titre de document stocké dans l'index (tronqué
         si nécessaire). -->
        <Max_Title_Length/> <!-- entier : 128 -->

        <!-- Longueur maximum du contenu d'un document stocké dans l'index (tronqué
         si nécessaire). -->
        <Max_Contents_Length/> <!-- entier : 8192 -->
      </Storage>
```

```
        <Plugin_Specific>
          <HTML>
            <!-- Indexer les textes des liens ? -->
            <Index_Link_Text/> <!-- booléen : true -->
            <!-- Poids des mots d'un lien -->
            <Link_Text_Score/> <!-- entier : 75 -->

            <!-- Indexer les textes des liens ? -->
            <Index_URL_Text/> <!-- booléen : false -->

            <!-- Poids des mots d'une URL -->
            <URL_Text_Score/> <!-- entier : 75 -->

            <!-- URLs contenues incluses dans la somme de contrôle ? (Utiliser vrai
             sauf si cela affiche des publicités) -->
            <Include_Embed_In_Checksum/> <!-- booléen : true -->
          </HTML>

          <OFFICE>
            <!-- Chemin du binaire utilisé pour lancer office -->
            <Office_Binary/> <!-- string : 'OO/program/soffice' -->

            <!-- Paramètres de connexion utilisés par le server office -->
            <Server_Connection_URL/> <!-- string: '
             -accept=socket,host=localhost,port=8100,tcpNoDelay=1;urp;
             StarOffice.ServiceManager' -->

            <!-- URL utilisé par un client pour se connecter à office -->
            <Client_Connection_URL/> <!-- string : '
             uno:socket,host=localhost,port=8100,tcpNoDelay=1;urp;Star
            Office.ServiceManager' -->

            <!-- Paramètres utilisés pour lancer un serveur office -->
            <Server_Parameters/> <!-- string : '-headless' -->

            <!-- Fichier de registre utilisé pour déclarer les composants d'office et
             les enregistrer. -->
            <Registry_File/> <!-- string : 'office_common.db' -->
          </OFFICE>
          <PDF>
          <!-- Utiliser un indexeur pdf avancé (true) ou indexer à plat (false)-->
           <Use_Advanced_Plugin>true</Use_Advanced_Plugin>
        </Plugin_Specific>
      </Indexers>

      <!-- - - - - - - - - -->
      <!-- Classification  -->
      <!-- - - - - - - - - -->
      <Classification>
        <!-- ACC : Activer ce service ? -->
        <Enabled/> <!-- booléen : false -->

        <!-- Hash size utilisée pour les collocations. Si nul, la hash size de
         l'indexation sera utilisée (valeur suggérée : 3 fois celle de l'indexation)
         -->
        <Hash_Size/> <!-- entier : 0 -->

        <N_grams>
          <!-- RTE : Activer ce service ? -->
          <Enabled/> <!-- booléen : false -->

          <!-- Longueur minimum d'un n-grams -->
          <Min_Length/> <!-- entier : 1 -->
```

```
         <!-- Longueur maximum d'un n-grams -->
         <Max_Length/> <!-- entier : 4 -->

         <!-- Nombre minimum d'occurences d'un n-grams dans le corpus à considérer
          comme une collocation. Peut valoir 0. -->
         <Min_Occurences/> <!-- entier : 1 -->

         <!-- Nombre minimum de documents contenant un n-gram dans le corpus devant
          être considéré comme une collocation.-->
         <Min_Documents/> <!-- entier : 1 -->
      </N_grams>
      <!--  -->
      </> <!--  -->

      <!--  -->
      </> <!--  -->
   </Classification>
 </Indexation>
```

2.1.5 Section Reverse

La section Reverse permet de définir la taille en Mb de la fenêtre de lecture de liens.

```
<Reverse>
         <Scores>
                  <Links_Window_Size_Mb/> <!-- par défaut 64 -->
         </Scores>
</Reverse>
```

2.1.6 Section Repositories

La section Repositories autorise la définition de dépôts.

```
<Repositories>
         <Login/> <!-- antisearch -->
         <Password/> <!-- -->
         <Db_Host/> <!-- localhost -->
         <Table/>
         <Id_type/> <!-- uint32 -->
         <Tag_name/>
</Repositories>
```

2.2 Configuration d'une indexation Web

En fonction du nombre de pages à indexer, la configuration doit être modifiée, ainsi pour un important volume de données réparties sur différents sites, le nombre de crawlers pourra par exemple être augmenté (Section `<Crawler>` du fichier *afs.xml cf §2.1.3)*

2.2.1 Récupération des données à indexer : le fichier perimeter.xml

On appelle **périmètre** un ensemble de pages Web à indexer, que ces pages appartiennent à un seul ou à plusieurs sites. La spécificité d'un périmètre est que **les pages sont accédées par crawl**, c'est-à-dire par un accès HTTP simulant l'action d'un utilisateur.

Lors du crawl, le fichier périmètre sera analysé pour vérifier si les liens suivis font partis des pages à indexer ou si elles doivent être ignorées.

La définition d'un fichier périmètre consiste donc à définir l'ensemble des sites à parcourir avec les paramètres associés, ainsi que les catégories dans lesquelles les différentes pages seront ventilées.

Le fichier périmètre est défini dans le fichier de configuration *afs.xml*, il est généralement séparé dans un fichier *perimeter.xml (entité appartenant au fichier afs.xml qui doit donc le référencer)*.

Ce fichier *perimeter.xml* est destiné à être inclus dans la section `<Crawling>` du fichier de configuration *afs.xml* car il nécessite d'être compilé.

Sa structure globale est la suivante :

```
<Categories>
  <!-- Définition des catégories pour la clusterisation des réponses
           ou le filtrage -->
</Categories>

<Perimeter>
  <!-- Définition du périmètre, qui est constitué d'un ensemble de paramètres par
défaut qui s'appliqueront à tous les sites, puis de la liste des sites à indexer.
-->
  <Defaults>
    <!-- les paramètres par défaut -->
  </Defaults>
  <Hosts>
    <!-- la liste des sites à indexer -->
  </Hosts>
</Perimeter>
```

On note les éléments suivants :

- La section `<Categories>` qui permet de définir les catégories utilisées pour catégoriser les pages.
- La section `<Perimeter>` qui permet de lister le chemin des Urls à parcourir et d' associer une catégorie définie dans la section `<Categories>` aux Urls.

Afin d'illustrer le format de ce fichier XML, nous allons définir un périmètre simple consistant à indexer trois sites d'une entreprise : le site grand public, l'extranet destiné aux partenaires et aux revendeurs, et un mini-site thématique conçu pour le lancement d'un produit.

- Section Categories

La section `<Categories>` permet de lister les différentes catégories dans lesquelles les pages seront ventilées.

Le nombre de catégories que l'on peut définir n'est pas limité.

Pour chaque catégorie, on définit son nom, son identifiant, son type, sa cardinalité et les différentes valeurs possibles :

le **nom de la catégorie** (attribut `name`) est le nom en clair qui peut être utilisé pour l'affichage ;

l'**identifiant** de la catégorie (attribut `id`) est un 'nom raccourci' compatible avec le format des attributs XML. Ce raccourci sera utilisé dans le reste de la configuration du périmètre pour référencer cette catégorie ;

le **type** (attribut `type`) peut prendre comme valeur "`fixed`" (lorsque toutes les valeurs possibles sont connues à l'avance et définies dans la configuration) ou alors "`dynamic`" quand les valeurs sont extraites dynamiquement lors de l'indexation.

la **cardinalité** (attribut `card`) permet de définir si une page peut prendre une seule ou plusieurs valeurs parmi la liste des valeurs possibles pour cette catégorie. La cardinalité peut donc être simple (`card="single"`) ou multiple (`card="multi"`).

Pour illustrer le principe des catégories, toujours dans le cadre de notre moteur exemple qui indexe trois sites, nous allons créer deux catégories :

- la première catégorie permet de définir pour chaque page indexée le type d'utilisateur auquel elle est destinée. On va appeler cette catégorie "`Profil`" (identifiant "`profil`"), les différentes valeurs disponibles sont '`Anonyme`', '`Partenaire`' et '`Client`', et la cardinalité de cette catégorie est multiple puisqu'une même page peut être vue à la fois par un visiteur anonyme, un client ou un partenaire.

- la deuxième catégorie est définie pour catégoriser les pages réponses indépendamment du site qui les contient. Cette catégorisation permettra d'affiner la présentation des réponses et d'offrir une fonctionnalité de filtrage des contenus. On décide d'appeler cette catégorie "`Rubrique`" (identifiant "`rubric`"). Les différentes valeurs possibles sont '`Société`', '`Produits`', '`Technologie`', '`Services`', '`Support`' et '`Divers`'. La cardinalité de cette catégorie est simple puisque dans notre cas, une même page ne traite que d'un seul sujet à la fois.

La déclaration XML correspondant à ces deux catégories est la suivante (on note que deux formes de déclaration sont possibles) :

```
<Categories>

  <!-- première catégorie : les différents profils utilisateur -->
  <Category id='profil' name='Profil' card='multi' type='fixed'>
    <Label id='anon' name='Anonyme'    />
    <Label id='part' name='Partenaire' />
    <Label id='cli'  name='Client'     />
  </Category>

  <!-- deuxième catégorie permettant de définir le rubriquage des pages -->
  <Category id='rubric' name='Rubrique' card='single' type='fixed'>
    <Label name='Société'    >soc</Label>
    <Label name='Produits'   >prod</Label>
    <Label name='Technologie'>tech</Label>
    <Label name='Services'   >serv</Label>
    <Label name='Support'    >support</Label>
    <Label name='Divers'     >div</Label>
  </Category>
</Categories>
```

La définition d'un périmètre est tout d'abord constituée de l'ensemble des paramètres par défaut qui s'appliqueront aux différents sites indexés :

- `Default_Policy` : permet de définir comment AFS traite les URLs trouvées dans les pages indexées. En choisissant `Deny`, on indique au moteur que l'on suit l'URL seulement si celle-ci appartient à un site listé dans le périmètre. Alors qu'en choisissant `Allow` comme comportement par défaut, le moteur suit toutes les URLs et les ajoute dynamiquement même si elles appartiennent à des sites non déclarés. La valeur par défaut est `Deny`.
- `Hosts_Default_Policy` : permet de définir comment AFS traite les URLs qui se trouvent dans les pages indexées et que le Host auquel appartient une URL est dans la liste des Hosts. La valeur par défaut est `Allow`, ce qui signifie que le lien sera suivi et que la page sera indexée. Si la valeur est positionnée à `Deny`, l'URL pointée ne sera pas indexée, ce qui permet de définir une indexation page par page.
- `Max_Hosts` : permet de limiter le nombre max de Host de la configuration. Pour ne pas limiter, il suffit de ne pas définir le tag ou de mettre la valeur `0` (zéro).
- `Max_Pages` : permet de limiter la taille du périmètre. Une fois le nombre max de pages atteint, les nouvelles URLs ne seront plus prises en compte. Pour ne pas limiter, il suffit de ne pas définir le tag ou de mettre la valeur `0` (zéro).
- `Max_Pages_Per_Host` : permet de fixer un nombre maximum de pages à indexer par site. Une fois le nombre max de pages atteint pour un même Host, les nouvelles URLs du site ne seront plus prises en compte. Pour ne pas limiter, il suffit de ne pas définir le tag ou de mettre la valeur `0` (zéro).
- `Allowed_Language` : permet de définir les langues indexées pour un site donné. Les valeurs possibles sont `FRENCH`, `ENGLISH`, `GERMAN`, `SPANISH`, `ITALIAN`, `PORTUGUESE`, `DUTCH`, `NORWEGIAN`, `GREEK`, `RUSSIAN`, `JAPANESE`, `CHINESE`, `KOREAN`, ... Si aucune langue n'est définie, le comportement par défaut dépend de l'activation de la vérification du langage dans le fichier *afs.xml* (section `<Language_Checker>`). Si la vérification est activée et qu'aucune langue n'est indiquée, aucune page ne sera indexée. Inversement, si la vérification du langage n'est pas activée, toutes les langues seront prises en compte.
- `Default_Language` : permet de définir la langue par défaut si la détection n'est pas activée ou si le nombre de mots utilisés pour la détection ne permet pas une décision.
- `Min_Words_For_Language_Detection` : permet de définir le nombre de mots à utiliser pour la détection du langage. La valeur par défaut est 5. Ce paramètre n'est pris en compte que si la détection du langage est activée.
- `Allow_Loopback` : permettre aux crawlers de crawler la loopback. La valeur par défaut est `false`. Ce paramètre est utile uniquement dans le cas où le crawler AFS est installé sur le même serveur que le contenu.
- Max_Crawl_Depth : permet de limiter la profondeur de crawl. La valeur par défaut est 0, ce qui correspond à aucune limite.
- `Mime_Types` : cette section permet de définir les types de fichier qui sont indexés par défaut. Pour chaque type de fichier à prendre en compte, utiliser une balise `Support`, en indiquant le type Mime.

Exemple : `<Support>application/pdf</Support>`

ou `<Support>application/x-shockwave-flash</Support>`

- `Parameters` : cette section permet de définir comment traiter les paramètres des URLs, à savoir :
  - `Keep_Parameters` : si les paramètres dynamiques sont conservés. La valeur par

défaut est `false`.

- `Order_Parameters` : si les paramètres des URLs doivent être réordonnés. La valeur par défaut est `true`.
- `Number_Of_Parameters` : le nombre de paramètres à conserver. La valeur par défaut est `0` (zéro).
- `Session_Variables` : une liste de variable de session additionnelles à celles prises en compte par défaut par AFS.

◆ `Rules` : cette section permet de définir des règles de traitement qui permettent de conserver (balise `<Allow>`) ou d'ignorer (balise `<Deny>`), de catégoriser ou de forcer la langue d'une URL donnée lorsque celle-ci concorde (match) avec une règle. Lorsqu'il s'agit de conserver ou d'ignorer une URL, la décision est prise sur la base de la première règle qui match. Par contre, lorsqu'il s'agit de catégoriser une URL (affecter une URL à une catégorie ou à une langue), c'est la règle la plus proche (présentant la similitude la plus profonde) qui est utilisée.

Dans l'exemple ci-dessous, 3 règles sont déclarées :

- la première permet d'indiquer que pour tous les sites à indexer, si l'URL d'une page est de la forme "`http://<site>/forum/*`", alors l'URL doit être ignorée, c'est-à-dire qu'elle ne sera pas indexée.
- la deuxième règle est une règle de catégorisation qui indique que par défaut les pages (puisque l'URL "`/*`" match toutes les URLs possibles) sont affectées à la rubrique "Divers".
- la troisième règle permet de préciser que toutes les pages qui ont une URL de la forme "`http://<site>/en/*`" sont des pages en anglais (ce qui reste valable même si la fonction Language Checker est dévalidée).

L'exemple suivant permet d'illustrer la configuration des paramètres par défaut :

```
<Perimeter>
<!-- les paramètres par défaut de l'indexation -->
  <Defaults>
    <Default_Policy>Deny</Default_Policy>
    <Hosts_Default_Policy>Allow</Hosts_Default_Policy>
    <Max_Pages_Per_Host>20000</Max_Pages_Per_Host>
    <Allowed_Language>FRENCH</Allowed_Language>
    <Allowed_Language>ENGLISH</Allowed_Language>
    <Allow_Loopback>false</Allow_Loopback>
    <Parameters>
      <Keep_Parameters>true</Keep_Parameters>
      <Order_Parameters>false</Order_Parameters>
      <Number_Of_Parameters>4</Number_Of_Parameters>
      <Session_Variable>sessid</Session_Variable>
      <Session_Variable>jid</Session_Variable>
    </Parameters>
    <Rules>
      <Deny>/blogs/*</Deny>
<!-- indique que l'on ne crawle pas les URLs commençant par blogs quelque soit les sites
car la Host_Default_Policy est allow -->
      <Allow _lang='ENGLISH'>/en/*</Allow>
      <Allow rubric='div'>/*</Allow>
  <!-- par défaut une page est affectée à la catégorie rubrique=divers -->
    </Rules>
  </Defaults>
```

Comme cela va être présenté, ces différents paramètres peuvent être redéfinis au niveau de chacun de sites à indexer. En effet, une fois les paramètres par défaut définis, il convient de définir dans la section `<Hosts>`, la liste de sites à indexer. Chaque site est déclaré par une section de type `<Host>`.

```
    <!-- les sites à indexer -->
    <Hosts>
          <Host name='...'>
                    <!-- un site -->
          </Host>
          <Host name='...'>
                    <!-- un autre site -->
          </Host>
    </Hosts>
 </Perimeter>
```

Pour chaque entrée de type `<Host>`, les paramètres de configuration disponibles sont les suivants :

- l'attribut `name : URL du site à indexer.`
- `Description` : quelques mots permettant de décrire le site à indexer.
- `Enabled` : cette balise qui prend comme valeur `true` (par défaut) ou `false` permet de suspendre l'indexation d'un site en mettant la valeur à `false`, comme si l'entrée n'existait pas, ce qui permet cependant de ne pas la supprimer.
- `Comment` : permet d'associer un commentaire de type administratif à un site. A la différence de la balise `Description` qui est utilisée pour des informations de type éditorial, ce champ permet de stocker des notes plus "techniques".
- `Defaut_Policy` : balise qui prend comme valeur `Allow`, `Deny` ou `Default` (par défaut). Elle permet de modifier la valeur par défaut définie avec `Hosts_Default_Policy` et donc de définir comment AFS traite les URLs qui se trouvent dans les pages indexées.
- `Default_Category` : catégorie par défaut des URLs qui ne matchent aucune règle (ou alors si aucune règle n'est définie). Il peut y avoir autant de déclaration que de catégorie distincte.
- `Max_Pages` : le nombre max de page à indexer pour ce site. Cela permet de redéfinir localement la déclaration `Max_Pages_Per_Host`. Utiliser la valeur 0 (zéro) pour indiquer à AFS de ne pas limiter le nombre de pages, en particulier si `Max_Pages_Per_Host` est défini.
- `Allowed_Language` : pour modifier la liste des langages à indexer pour ce site par rapport à la configuration par défaut. Si aucune balise par défaut n'est définie, c'est la configuration par défaut qui s'applique.
- `Force_Language_For_Path` : le contenu attribue une langue aux urls du chemin spécifié dans l'attribut `key`. La valeur contenue doit être une des valeurs possibles de la balise `<Allowed_Language>`. Cette valeur peut aussi être définie dans l'attribut `_lang` de la balise `<Allow>`.
- `Case_Sensitive_Paths` : cette balise peut prendre comme valeur `true` (par défaut) ou `false` et lorsqu'elle est mise à faux, les URLs sont normalisées en minuscule ce qui est particulièrement utile pour éviter de boucler dans les sites sous Windows dont les URLs ont des formes multiples.
- les balises `Keep_Parameters`, `Order_Parameters` et `Number_Of_Parameters` peuvent être redéfinies localement pour le site courant.
- Avec des balises `Session_Variable`, il est également possible d'enrichir la liste des variables sessions utilisées par le site.
- `Minimal_Request_Interval_Seconds` : cette balise permet de définir le nombre de secondes

à attendre entre deux requêtes sur un même serveur afin de ne pas le saturer. La valeur par défaut est 0 (zéro) ce qui signifie que chaque fois qu'une URL est récupérée la suivante est demandée, et que si plusieurs crawlers sont activés, ils pourront demander des URLs en parallèle.

- les balises `Seed` permettent de définir des URLs par lesquelles rentrer dans le site. Cela est particulièrement utile si les liens contenus dans la première page du site sont des JavaScript qui ne peuvent être interprétés par AFS.
- de façon similaire à la configuration des paramètres par défaut, une section `<Rules>` permet de définir un ensemble de règles permettant d'inclure ou d'exclure certains morceau du site, ainsi que d'affecter des catégories à des pages ou de définir leur langue.
- `Passwords` : une section qui permet de définir une liste de couple { utilisateur / mot de passe } pour une URL donnée, permettant ainsi de s'authentifier auprès du site afin de l'indexer si celui-ci est protégé en accès. La syntaxe d'une balise `Password` est la suivante : `<Password key='URL'>login:motdepasse</Password>`

L'exemple suivant permet d'illustrer l'ensemble de ces paramètres. Il décrit l'indexation de nos trois sites : site public, intranet et mini-site produit.

```
<Hosts>
  <Host name='http://www.masociete.fr'>
<!-- par défaut toutes les pages sont accessibles à tous les profils mais les
rules permettent de dire que tout ce qui est dans /forum/* n'est accessible
qu'aux clients et aux partenaires. -->
    <Default_Category rubric='soc' prof='anon,cli,part' />
    <Rules>
      <Allow rubric='div'>/actu/*</Allow>
      <Allow rubric='support' prof='cli,part'>/forum/*</Allow>
      <Allow rubric='prod'>/*/produits/*</Allow>
      <Allow rubric='serv'>/*/services/*</Allow>
      <Allow _lang='ENGLISH'>/en/*</Allow>
    </Rules>
    <Default_Policy>Deny</Default_Policy>
    <Seed>/sitemap.php</Seed>
<!-- pour être sûr de ne rien rater on indique la page contenant le plan du
site-->
    <Max_Pages>5000</Max_Pages>
  </Host>
  <Host name='http://www.monextranet.fr'>
    <Default_Category rub='support' prof='part'></Default_Category>
    <Rules>
      <Allow rubric='tech'>/technologie/*</Allow>
    </Rules>
    <Default_Policy>Deny</Default_Policy>
    <Keep_Parameters>true</Keep_Parameters>
    <Order_Parameters>true</Order_Parameters>
    <Number_Of_Parameters>8</Number_Of_Parameters>
    <Session_Variable>sess_sid</Session_Variable>
    <Passwords>
      <!-- une liste de password permettant d'indexer -->
      <Password key='www.monextranet.com/support'>nomuser:monpasswd</Password>
      <Password key='www.monextranet.com/prix'>nomuser2:autrepasswd</Password>
    </Passwords>
    <Comment>Site protégé. Crawl qui nécessite des accès spécifiques.</Comment>
  </Host>
  <Host name='http://www.monproduit.eu'>
```

```
    <Description>Mini ... qui a procédé au lancement du produit</Description>
    <Default_Category rubric='prod' prof='anon'></Default_Category>
    <Rules>
      <Allow rubric='support' prof='part'>/download/*</Allow>
    </Rules>
    <Default_Policy>Deny</Default_Policy>
    <Case_Sensitive_Paths>false</Case_Sensitive_Paths>
    <Seed>/index/plandusite.html</Seed>
  </Host>
</Hosts>
```

2.2.1.1 Configurer la fréquence de crawl

Il est possible de définir la fréquence de mise à jour du crawl au sein du fichier *perimeter.xml*.

- `<Refresh>` : Cette balise déclarée au sein de la section `<Defaults>` permet de définir la fréquence d'indexation par défaut avec l'attribut `@when`.
- `<Host>` : Il est également possible de définir pour chaque host une fréquence d'indexation spécifique. Cette fréquence sera alors spécifiée grâce à l'attribut `@refresh_when`. La fréquence peut également être précisée pour un chemin particulier défini alors dans la balise `<Allow>` avec l'attribut `@refresh_when`.

Les valeurs possibles pour l'attribut `@refresh_when` des balises `<Host>` ou `<Allow>` ou de l'attribut `@when` de la balise `<Refresh>` sont de la forme : fréquence suivie du mois, du jour et de l'heure séparés par des `@`.

Exemple : `yearly@january@01@22h50`.

Liste des valeurs possibles :

- Tous les jours : `everyday` (== `monday`, `tuesday`, `wednesday`, `thursday`, `friday`, `saturday`, `sunday`)
- Tous les jours de la semaine : `weekday` (== `monday`, `tuesday`, `wednesday`, `thursday`, `friday`)
- Tous les soirs de la semaine : `weeknight` (== `sunday`, `monday`, `tuesday`, `wednesday`, `thursday`)
- une fois par an : `yearly@january@01@22h50` qui peut être abrégée en `january@01@22h50`
- une fois par mois : `monthly@01@22h50` qui peut être abrégée en `01@22h50`
- une fois par semaine : `weekly@monday@22h50` qui peut être abrégée en `monday@22h50`
- une fois par jour : `daily@22h50`
- une fois par heure : `hourly@15`

Plusieurs valeurs peuvent être données pour un même champ en les séparant par des virgules. Chaque valeur matchée sera appliquée.

Exemple : `(monday,sunday)@22h50` (= le lundi ou le dimanche à 22h50, le ou étant inclusif)

Exemple :

```
<AFS>
  <Defaults>
    <Refresh when="monthly@01"/><!-- mise à jour tous les mois -->
  </Defaults>
```

```
  <Hosts>
    <Host name="toto" refresh_when="monday@10H,friday@14H">
      <!-- maj 2 fois/semaine -->
      <Allow>...</Allow>
      <!-- Prend la valeur par défaut du Host -->
      <Allow refresh_when="everyday@00H10">...</Allow>
      <!--  tous les jours a 00H10 -->
      <Allow refresh_when="weekday@00h10">...</Allow>
      <!--  les jours de la semaine a 00H10 -->
      <Allow refresh_when="monday@00H10">...</Allow>
      <!--  lundi a 00H10 -->
      <Allow refresh_when="weekly@monday">...</Allow>
      <!--  tous les lundis -->
      <Allow refresh_when="monthly@01">...</Allow>
      <!-- mensuel -->
    </Host>
    <Host name=”titi”>
        <!-- prend la valeur par défaut -->
  </Hosts>
</AFS>
```

2.2.2 Prétraitement des données

Il est possible d'utiliser un plugin de traitement des données HTML en vue d'une exploitation différente. Ce plugin nécessite un développement spécifique.

Il consiste à transformer des données HTML connues (nécessite une structure des données stable – par exemple un forum) dans un autre format de sortie, comme par exemple en XML.

Cette configuration se trouve dans le fichier *afs.xml* à la section `<Crawling>` (*cf § 2.1.3*)

2.2.2.1 Définir des sites comme valeur de filtre

Il est possible de limiter la recherche à une partie d'un site afin de filtrer par exemple sur ses différentes rubriques.

Par exemple, si un site générique (http://magazines.fr) contient un sous ensemble de sites spécifiques (http://magazine-sante.fr/allergies, http://magazine-sante.fr/enfants, http://magazine-litterature.fr/selection etc...) et que l'on souhaite proposer une recherche contextuelle pour chacun de ces sites, il faudra les déclarer dans un fichier appelé *virtual_servers.xml* afin de leur attribuer une valeur pour le filtre prédéfini SITE.

Ce fichier a pour racine la balise `<Virtual_Servers>` qui contient des sections `<Virtual_Server>` permettant de déclarer chaque site pouvant faire l'objet d'une recherche ciblée. L'attribut `URI/@pattern` contient l'URL du site (sans le protocole) et l'attribut `Virtual_Server/@name` définit la nouvelle valeur pour le filtre SITE.

Exemple :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- Fichier de configuration: serveurs virtuels    -->
<Virtual_Servers>
  <Virtual_Server name="allergies"> <!-- recherche sur site:allergies -->
          <URI pattern="magazine-sante/allergies"/>
  </Virtual_Server>
  <Virtual_Server name="enfants">
          <URI pattern="magazine-sante/enfants"/>
  </Virtual_Server>
...
</Virtual_Servers>
```

2.2.3 Analyse fine des pages HTML : Les Fichiers aihtml

Le moteur d'indexation HTML d'AFS procède à une analyse de la page HTML afin d'indexer l'ensemble des contenus de la page. Considérons les deux cas suivants :

- Certains sites sont conçus de telle façon que le contenu central de la page est entouré de beaucoup d'autres informations comme la navigation ou des éléments de contenu annexes (fil d'info, ...). Ces informations ne sont pas en relation directe avec le contenu de la page et ajoute donc ce qu'on appelle communément du bruit. En effet une recherche sur le site donné pourra amener à une page ne traitant pas spécifiquement du problème, juste parce que des éléments connexes "en parle". Dans la plupart des cas, les algorithmes de pertinence d'AFS résolvent automatiquement le problème mais il reste des cas dans lesquels ce n'est pas possible.
- Comme on l'a vu précédemment, il est possible de catégoriser les pages lors de l'indexation, c'est-à-dire d'affecter chaque page à une ou plusieurs catégories. Les mécanismes proposés dans la définition d'un périmètre permettent de réaliser cette catégorisation sur la base des formes des URLs des pages grâce à des fonctions de matching. Ce mécanisme, tout performant qu'il soit, ne couvre pas tous les besoins puisque dans certains cas, la catégorisation doit être réalisée à partir d'informations contenues dans les pages elles-mêmes et non à partir de méta-données.

Pour répondre à des problèmes comme ceux décrits ci-dessus, AFS offre la possibilité de piloter le module d'analyse des pages HTML en lui indiquant des actions à réaliser comme la possibilité d'ignorer certains éléments de contenu et de se concentrer sur d'autres, ou alors d'extraire certaines informations des pages afin de générer dynamiquement des méta-données qui sont ensuite utilisées pour la catégorisation, le filtrage, la pondération ou le tri.

2.2.3.1 Structure des fichiers

Le paramétrage fin du module d'analyse se fait à travers des fichiers de configuration appelés "fichiers aihtml" pour "AFS Indexer HTML".

Ces fichiers sont aux formats XML et ils possèdent la structure suivante :

```
<?xml version="1.0" encoding="utf-8"?>
<HTML_Plugin name="Pattern Site1">
  <URI_Patterns>
    <URI_Pattern>http://autresite/wiki/</URI_Pattern>
    <URI_Patterns>
      <!-- Sections de déclaration de règles de traitement -->
      <HTML_Code>
        <Default_Policy>Remove</Default_Policy>
        <!-- ou keep -->
        <Keep_Section type="tag" begin="HEAD" end="Head"/>
        <!-- autres valeurs possibles pour type : comment, class -->
      </HTML_Code>
      <Title>
<!-- règles pour extraire et fabriquer le titre à utiliser en réponse -->
      </Title>
      <Description>
<!-- règles pour extraire et fabriquer un résumé du document -->
      </Description>
      <Date>
<!-- règles pour extraire des dates -->
      </Date>
      <Store_Items>
<!-- règles permettent d'extraire et de générer des méta données -->
```

```
      </Store_Items>
    </HTML_Plugin>
```

Ce fichier possède :

- un nœud racine `HTML_Plugin` ayant un attribut `name` qui donne un nom aux patterns définis
- une balise `<URI_Pattern>` qui contient une liste de balises `<URI_Pattern>` qui permet d'indiquer dans quel cas les règles contenues dans ce fichier doivent être appliquées en définissant un pattern d'URL. Si le fichier ne s'applique qu'à une URL, celle ci peut être donnée en valeur à l'attribut `URI_Pattern` de la racine `<HTML_Plugin>`.
- un ensemble de sections qui décrivent des règles de traitement à appliquer.

Lorsqu'une page HTML qui a été crawlée est transmise à un indexeur, celui-ci vérifie parmi tous les fichiers aihtml si l'un d'eux doit être appliqué, c'est-à-dire si l'URL de la page match un des URI_Pattern définis dans l'ensemble de fichiers aihtml ajoutés à la configuration. Si plusieurs fichiers aihtml matchent une URL, c'est le fichier qui a le pattern le plus spécifique qui est appliqué.

2.2.3.2 Les règles de traitement aihtml

Les règles de traitement qui doivent être appliquées par les indexeurs sont déclarées dans un ensemble de sections selon l'objectif recherché :

<HTML_Code>

La section `HTML_Code` permet de définir les sections de codes HTML à conserver ou à ignorer lors de l'indexation. La balise `<Default_Policy>`, qui prend la valeur `Keep` ou `Remove`, permet d'indiquer si par défaut tout le code HTML est conservé et que l'on va procéder par exclusion, ou si au contraire tout le code est ignoré et que les règles vont indiquer quel contenu conserver. Dans le cas où la politique par défaut est `Remove`, des balises `<Keep_Section>` permettent de déclarer quoi conserver. A l'inverse, si la politique par défaut est `Keep`, des balises `<Remove_Section>` indiquent quoi ignorer.

Ces deux balises fonctionnent de la même façon :

elles ont un attribut `type` qui permet d'indiquer à quel type de contenu la règle s'applique. type peut prendre les valeurs `tag`, `comment` ou `class` selon que l'on cherche un tag HTML précis, du contenu compris entre deux commentaires, ou alors appartenant à une classe particulière.

si le `type` est '`tag`' ou '`comment`', alors elles ont ensuite deux attributs `begin` et `end` qu'il convient de renseigner pour indiquer le contenu sur lequel le filtrage doit se faire. Dans le cas où l'attribut `type` a pour valeur '`tag`', `begin` et `end` doivent avoir la même valeur.

si le `type` est '`class`', alors un attribut `id` permet d'indiquer l'identifiant de la classe CSS.

Exemple :

```
<HTML_Code>
  <!-- par défaut on ne conserve aucun contenu de la page HTML -->
  <Default_Policy>Remove</Default_Policy>
  <!-- et on indique ce que l'on conserve et indexe -->
  <!-- le header HTML qui est pris en compte -->
  <Keep_Section type="tag" begin="HEAD" end="HEAD" />
  <!-- le HTML contenu entre les commentaires DEBUT CONTENU et FIN CONTENU -->
  <Keep_Section type="comment" begin="DEBUT CONTENU EDITORIAL"
                               end="FIN CONTENU EDITORIAL" />
  <!-- et le ou les titres du contenu -->
  <Keep_Section type="class" begin="gros_titre" end="gros_titre" />
</HTML_Code>
```

<Title>

Il est courant que toutes les pages d'un même site aient le même titre, c'est-à-dire le même contenu dans la balise Title du header HTML. Ce titre HTML est donc peu indicatif et ne peut pas être utilisé lors de la recherche pour présenter les réponses.

La section `Title` des fichier aihtml permet justement de définir des règles permettant d'extraire un titre de page depuis le contenu. Il suffit pour cela d'utiliser un ensemble de balise `Title_Item` qui fonctionnent de façon similaire aux balises `Keep_Section` et `Remove_Section`, c'est-à-dire avec un attribut `type` indiquant où trouver le contenu à extraire pour fabriquer le titre (`tag`, `comment`, `class`). La première règle `Title_Item` qui permet d'extraire du contenu sera prise en compte et les suivantes ignorées.

Par ailleurs, il est possible de nettoyer le contenu extrait pour enlever des parties communes à tous les titres et peu signifiants. Par exemple si tous les titres des pages commencent par le nom du site, il est intéressant de pouvoir enlever ce nom du site afin de ne conserver que le contenu signifiant pour l'indexation et la présentation de résultat. Le nettoyage se fait grâce à la balise `Remove_Pattern` qui permet dans l'attribut `value` d'exprimer un pattern à base de texte (les parties à ignorer) et d'étoiles (les parties à conserver). Il est possible de fournir plusieurs instructions `Remove_Pattern` ; elles seront toutes essayées dans l'ordre d'apparition.

Comme indiqué, le contenu éliminé par les balises `Remove_Pattern` ne sera pas non plus indexé.

Exemple:

```
  <Title>
        <Title_Item type="class" id="gros_titre_bleu"/>
        <Title_Item type="AFS"/>    <!-- sinon titre par defaut  du header -->
    <!-- nettoyage de la chaine extraite  : on enlève le début si c'est le nom
                  du site et on ne conserve que la suite -->
    <Remove_Pattern value="Intranet masociété - *"/>
  </Title>
```

<Description>

De façon similaire à ce qui a été décrit ci-dessus pour le titre, la balise `Description` permet de définir avec des balises `Description_Item` puis `Remove_Pattern` la façon dont extraire du contenu de la page, puis nettoyer les éléments textuels qui seront utilisés pour constituer le résumé qui sera présenté dans la liste des réponses.

A la différence du titre, dans le cas de la description le contenu nettoyé par la balise `Remove_Pattern` sera quand même indexé. Sa suppression n'est effective que lors de la constitution des résumés.

Exemple :

```
<Description>
  <Description_Item type="comment" begin="DEBUT CONTENU" end="FIN CONTENU"/>
      <!-- nettoyage de la chaine extraite
      <Remove_Pattern value="Home - Plan du site *"/>
  </Description>
```

<Date>

La balise `Date` permet de définir comment extraire une date depuis le contenu HTML afin de générer dynamiquement une métadonnée qui sera ensuite utilisée pour fournir des fonctions de tri ou de filtrage.

Cette balise possède les attributs suivants :

`type` : admet pour les commentaires la valeur `comment_filter`

`filter` : definit les valeurs à filtrer

`format` : donne le format de la date formatée selon les fonctions time, localtime et strftime de la librairie C.

`alt_format` : permet de donner un autre format possible

`name` : le nom de la métadonnée créée.

Exemple :

```
<Date type="comment_filter"
      filter="date: *" <!-- on ne prendra que ce qui suit 'date : ' -->
      format="%B %Y"
      alt_format_1="%Y"
      name="date_publication"/>
```

<Store_Items>

La section `Store_Items` permet de définir des règles d'extraction et de génération de métadonnées à partir du contenu de la page HTML. Chaque règle d'extraction est déclarée avec une balise `Store_Item` qui contient 3 attributs :

`type` : admet pour les commentaires la valeur comment_filter

`filter` : definit les valeurs à filtrer

`name` : le nom de la métadonnée ainsi créée.

Exemple:

```
<Store_Items>
   <Store_Item type="comment_filter" filter="categorie: *"      name="CAT"     />
   <Store_Item type="comment_filter" filter="sous_categorie: *" name="SOUS_CAT"/>
   <Store_Item type="comment_filter" filter="titre: *"          name="TITRE"   />
   <Store_Item type="comment_filter" filter="image: *"          name="IMAGE"   />
   <Store_Item type="comment_filter" filter="video: *"          name="VIDEO"   />
</Store_Items>
```

2.3 Configuration de l'indexation d'une base de données

L'indexation des sources de données structurées telles que les bases de données SQL se décompose en deux parties :

la déclaration des bases à indexer avec les données à prendre en compte ;

la description des paramètres d'indexation.

2.3.1 Récupération des données : Le fichier aidbml

La déclaration d'une base à indexer se fait au moyen d'un fichier de configuration XML (fichier d'extension `aidbml`) qui paramètre le module des crawlers conçu pour se connecter aux bases de données SQL et de lire les informations à indexer. Ce fichier doit être placé dans **`$AFS/plugins/index/`** pour être chargé automatiquement lors de l'indexation.

Ce fichier de paramétrage contient deux sections distinctes :

la première, intitulée **`<Defaults>`** permet de définir les paramètres techniques de connexion au serveur de base de données (section **`<Database>`**) et la gestion des pièces jointes (section **`<Attachements>`**) ;

la seconde, intitulée **`<Feeds>`** permet de lister les différentes données à extraire.

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<Database_Plugin>
  <!-- Paramétrage Global -->
  <Defaults>
    <Database>
      <!-- Paramètres nécessaires -->
      <Server>mydbserver</Server>
      <User>user</User>
      <Password>secret</Password>
```

```
      <Database>DBNAME</Database>
      <Driver>mysql</Driver>  <!-- mysql, sqlserver, .. -->

      <!-- Paramètres optionnel -->
      <Port>3443</Port>
      <Client>AFS</Client>
      <Connection_String>arg1=val1&amp;arg2=val2</Connection_String>
    </Database>
    <Attachements>
      <Abstract_Max_Len>8192</Abstract_Max_Len>
      <Repository_Mode>true</Repository_Mode>
    </Attachements>
  </Defaults>

  <!-- Les sources de données à indexer -->
  <Feeds>
    <!-- ICI la déclaration des sources -->
  </Feeds>
</Database_Plugin>
```

Comme on le voit ci-dessus, la section définie par la balise `<Database>` permet de définir les paramètres de connexion au serveur :

- `<Server>` : le nom ou l'adresse IP du serveur hébergeant la base de données ;
- `<User>` : le nom d'utilisateur à utiliser pour l'authentification ;
- `<Password>` : le mot de passe de l'utilisateur ;
- `<Database>` : le nom de la base de données ;
- `<Driver>` : le type de serveur de base de données (MySQL, Microsoft SQL Server) ;
- `<Port>` : le numéro de port TCP pour la connexion ;
- `<Client>` : une chaîne de caractère décrivant le client connecté afin de l'identifier avec une commande de type `sp_who` ;
- `<Connection_String>` : une chaîne de caractères que l'on peut passer en argument de la connexion dans certaines bibliothèques clientes.

La section définie par la balise `<Attachements>` permet quant à elle de définir le format des pièces jointes :

`<Repository_Mode>` : La présence de cette balise permet de maintenir un cache local afin de vérifier lors d'une nouvelle indexation si les fichiers ont été modifié, s'ils ce n'est pas le cas il ne les réindexe pas.

La section `Feeds` permet de définir les différents objets à indexer. Chaque objet est défini par une entrée de type `<Feed>`:

```
<Feeds>
  <Feed name="articles">
    <Seeds>
      <!-- Liste des objets à indexer par défaut -->
      <Seed>
        <!-- Requête utilisée pour récupérer la liste des informations à indexer -->
        <Request>SELECT id FROM NEWS.articles WHERE site='NEWS.public'</Request>
        <!-- Requête exécutée avant de récupérer la liste des objets -->
        <Pre>SELECT COUNT(*) FROM NEWS.articles WHERE site='NEWS.public'</Pre>
        <!-- Requête exécutée après avoir récupéré la liste des objets -->
        <Post>DELETE FROM NEWS_UPDATE.articles</Post>
      </Seed>

      <!-- Liste des objets à indexer par défaut -->
      <Seed name="Diff">
        <!-- Requête utilisée pour récupérer la liste des informations à indexer -->
        <Request>SELECT id FROM NEWS_UPDATE.articles
```

```
                WHERE site='NEWS_UPDATE.public' AND NEWS_UPDATE.DATE=TODAY()</Request>
      <!-- Requête exécutée avant de récupérer la liste des objets -->
      <Pre>SELECT COUNT(*) FROM NEWS_UPDATE.articles
           WHERE site='NEWS_UPDATE.public' AND NEWS_UPDATE.DATE=TODAY()</Pre>
    </Seed>
  </Seeds>
  <Get_Item>
    SELECT
    info.titre                    , /* $1 */
    UNIX_TIMESTAMP(info.parution), /* $2 */
    info_contenu                  , /* $3 */
    info_keywords                 , /* $4 */
    produit.nom                   , /* $5 */
    type.produit                    /* $6 */
    FROM NEWS.articles
    LEFT OUTER JOIN NEWS.produits produit ON info.produit=produit.id
    LEFT OUTER JOIN NEWS.types type ON info.type=type.id
    WHERE news.id=$_1
  </Get_Item>

  <XML_Template encoding="iso-8859-1" suffix=".artml">
    <ARTICLE>
      <ID>$_1</ID>
      <TITRE>$1</TITRE>
      <PARUTION>$2</PARUTION>
      <CONTENU>$3</CONTENU>
      <KEYWORDS>$4</KEYWORDS>
      <NOM_PRODUIT>$5</NOM_PRODUIT>
      <TYPE_PRODUIT id='$6'/>
      <PDF_URI_action='index'>ftp://login:password@host/$_1.pdf</PDF_URI>
    </ARTICLE>
  </XML_Template>
</Feed>

<!-- puis définir ici les autres objets -->
<Feed>
  <!-- ... -->
</Feed>
</Feeds>
```

Cet exemple qui extrait depuis une base d'articles de news, les données à indexer met en évidence les points suivants :

la balise `<Feed>` possède un attribut `name` qui permet de nommer la source. Ce nom est utilisé pour identifier la source et donc l'ensemble des enregistrements de la base de données qui la compose.

la section ayant pour balise `<Seeds>` permet de définir les requêtes SQL utilisées pour récupérer la liste des éléments à indexer.

la section ayant pour balise `<Seed>` permet de définir une requête SQL utilisée pour récupérer la liste des éléments à indexer. L'attribut `name` permet de nommer la requête de sélection des objets. La (ou les) requête(s) utilisée(s) est(sont) précisée(s) à l'indexation. Par défaut, le nom du `Seed` est: « `default` ». La balise `Seed` contient les balises:

- `<Request>`: Balise obligatoire. Cette balise contient la requête à exécuter pour récupérer la liste des objets à indexer.
- `<Pre>`: Balise optionnelle. Cette balise contient la requête exécutée avant de récupérer la liste des objets.
- `<Post>`: Balise optionnelle. Cette balise contient la requête exécutée après avoir récupéré la liste des objets.

la section de balise `<Get_Item>` permet de définir la requête SQL qui sera utilisée pour lire le contenu de chaque élément à indexer tel que défini dans `<Seed>`. Cette requête admet en paramètre la variable `$_1` qui est le premier champ extrait par le SELECT de la requête `<Seed>`. Dans ce cas, il s'agit du champ `id` des news.

enfin la section ayant pour balise `<XML_Template>` permet de définir un format XML pour chaque objet extrait de la base. Cette balise admet les attributs `encoding` et `suffix` qui seront utilisés pour générer le flux XML. En effet, le résultat du module de lecture d'une base de données est un flux XML qui sera utilisé en entrée du module d'indexation de flux structuré. `Encoding` permet de définir l'encodage du flux et `suffix` sert à définir le fichier de paramètre d'indexation à utiliser.

La balise `<XML_Template>` contient un template XML de format libre dans lequel les différents champs lus par `<Get_Item>` sont instanciables par les variables `$1`, `$2`, ...

Seule la section `<Feed>` intervient lors de la récupération des données qui s'effectue avec la commande `as_umanager –vYf <identifiant_feed>` (cet identifiant est défini par l'attribut `name` de la balise `Feed`).

Les autres sections (`Get_Item` et `XML_Template`) sont traitées lors de l'indexation. Les données contenues dans la base de données (récupérées selon les requêtes SQL de `Get_Item`) sont alors transformées en flux XML (selon le Template défini dans `XML_Template`).

Note concerant les types SQL qui sont supportés dans AFS:

Type	***Exemple***	***Commentaire***
Chaîne de caractères	`CHAR`, `VARCHAR`	*Support natif*
Entier	`SMALLINT`, `INT`, `BIT`	*Support natif*
Flottant	`DECIMAL`	*Support natif*
Date	`SMALLDATETIME`, `DATETIME`	**Attention**, ce type n'est pas nativement supporté par AFS: il convient donc de le convertir vers une chaine de caratères, par exemple avec la fonction SQL suivante: `CONVERT(VARCHAR, Annonce.DateParution, 103)` De plus, les dates d'un index AFS sont au format UNIX, c'est à dire qu'elles ne peuvent pas être inférieures à 1970. pour contourner cette limitation, vérfier si vous avez besoin de stocker une date complète, ou bien si l'année (un entier) ne suffirait pas...

L'indexation se poursuit avec les mêmes étapes que celle d'un flux XML (nécessite donc la rédaction d'un fichier `aixml`) → ***cf § 2.4***

2.4 Configurer l'indexation d'un flux structuré XML

Pour l'indexation XML, la configuration doit être modifiée, le nombre de crawlers doit impérativement être mis à zéro. (Section `<Crawler>` du fichier *afs.xml § 2.1.3*) sauf dans le cas de l'indexation de pièces jointes en http.

Les **données structurées indexables par AFS sont des flux XML** qui peuvent :

être générés par les crawlers grâce à un plugin de lecture propre à un type de source (comme le plugin de lecture des bases de données décrit ci-dessus) ;

être des fichiers XML déjà présents sur le serveur ;

des flux XML accessibles par Internet : pages web générées en XML et non en HTML, flux RSS, ...

2.4.1 *Prétraitement des données : Le fichier aixml*

A chaque source de données structurées identifiée correspond donc un ou plusieurs flux XML et pour indexer chacun d'entre eux, il suffit de définir les paramètres spécifiques dans un fichier d'indexation aixml.

Ce fichier *aixml*, au format XML, permet pour chaque flux de définir les paramètres généraux et les options, les champs à indexer, ceux à conserver à des fins de tri ou de catégorisation, la façon dont générer le titre et le résumé d'une réponse ... Le fichier doit être placé dans le répertoire `$AFS/plugins/index`.

La structure générale d'un fichier *aixml* est la suivante :

```
<?xml version="1.0" encoding="utf-8"?>
<Plugin suffix="..." mime_type="..." category="...">
  <Split>
<!-- permet de repérer dans le flux d'entrée les objets de base -->
  </Split>

  <Attachments>
<!-- Indexation de pièces jointes -->
  </Attachments>

 <Preproprecessing>
<!-- Nettoyage des balises HTML  -->
  </Preproprecessing>
  <Indexed_Tags>
<!-- déclaration des balises à indexer -->
  </Indexed_Tags>
  <Title_Items>
<!-- déclaration des balises composant le titre de la réponse -->
  </Title_Items>
  <Contents_Items>
<!-- déclaration des balises composant le résumé de la réponse -->
  </Contents_Items>
  <Store_Items>
<!-- balises à ne pas indexer mais à conserver pour les rendre dans la réponse et qui
peuvent être utilisées pour du tri ou du filtrage -->
  </Store_Items>
<!-- déclaration de la balise composant l'URL de la réponse -->
  <URL_Tag name="..."/>
</Plugin>
```

La balise racine d'un fichier *aixml* est Plugin. Elle admet trois attributs :

`suffix` : c'est grâce à ce suffixe qu'est fait le lien entre ce fichier de configuration et un fichier XML d'entrée existant (dont le nom doit avoir ce suffixe) ou un flux généré à partir d'une base de données (grâce à l'attibut `suffix` de la balise `XML_Template`).

`mime_type` : permet d'associer un type MIME interne AFS à ce type de contenu. Ce type MIME est ensuite utilisé pour générer les fichiers à indexer.

`category` : information de catégorie générale qui est restituée dans la liste des réponses et peut donc être utilisée pour le filtrage ou la présentation.

NOTATION : Dans la suite de ce paragraphe, nous désignerons par champ indifféremment une balise XML ou un attribut d'une balise XML. En prenant comme exemple le format *aixml* lui-même, la balise `<Plugin>` ou l'attribut `suffix` de la balise `Plugin` sont des champs, et nous utiliserons la notation `<BALISE@attribut>` (par exemple

<Plugin@suffix>)dans le reste de ce document pour désigner les champs de type attribut.

2.4.1.1 Les sections d'un fichier aixml

Les différentes sections d'un fichier *aixml* sont décrites ci-après. Elles sont a chaque fois illustrées par un exemple dans lequel AFS indexe le contenu d'un CMS Typo3 contenant des articles de type rédactionnel et AFS indexe ce contenu sous la forme de pages du site générées au format XML par un template spécifique. Il est à noter que la casse des paths est préservée et qu'il faut donc être vigilant à bien la respecter.

<Split>

La section Split permet de définir les informations qui permettent de découper un flux d'entrée en éléments de base. La définition de cette section est très importante car elle permet de définir la granularité des données.

Plusieurs opérations peuvent être effectuées :

- Le découpage d'un fichier en plusieurs éléments de base (balise <Item_Node>): si le flux XML est un seul fichier contenant toutes les données à indexer, il convient de le découper en autant de fiches que d'éléments contenus dans le flux.

- L'ajout de pièces jointes (balise <Attachment_XPath>) : le flux XML peut contenir des données accessibles par une URL. Ce traitement permet alors de récupérer les pièces jointes et d'en faire une pré-indexation. Le téléchargement de la pièce jointe permet une détection du mime type et donc un lancement automatique de l'indexeur approprié. Si par exemple le mime type est `text/xml`, `as_xpathindex` est déclenché et nécessite donc la définition d'un fichier aixml.

Les différentes balises disponibles dans cette section sont :

Root_Node : permet d'indiquer le nom du nœud racine de l'arbre XML complet à indexer. Laisser cette balise vide si ce nœud n'est pas défini.

Item_Node : nom du nœud de l'arbre XML désignant un élément de base à indexer.

`Keep_Toplevel_Nodes` : permet de conserver les balises (avec leurs attributs et leur contenu) présentes au-dessus de l'élément de base.

`Encoding` : encodage XML des fichiers XML résultat.

`Append_Mode` : permet de continuer la numérotation des fichiers splittés en sortie selon le contenu du répertoire de sortie.

`Repository_Mode` : la présence de cette balise permet de préciser qu'il ne faut réindexer que ce qui a été modifié.

Exemple :

```
<Split>
  <Root_Node></Root_Node>
  <Item_Node>page</Item_Node>
  <Keep_Toplevel_Nodes/>
  <!-- garder le contexte "pages", ... -->
  <Encoding>ISO-8859-1</Encoding>
  <Append_Mode/>
  <Repository_Mode/>
  <!-- Ne recrawler que ce qui a ete modifie -->
</Split>
```

<Attachments>

La section `Attachments` permet l'indexation de pièces jointes :

`On_The_Fly` : sa présence permet de déterminer le mode indexation de pièces jointes

`Attachment/@name` : définit le Xpath du champ contenant le chemin d'accès aux pièces jointes.

Exemple :

```
  <Attachments>
        <On_The_Fly/>
        <!-- Les attachements a telecharger -->
        <Attachment name="PAGE/fichier/@_link"/>
</Attachments>
```

Lors de l'indexation le contenu et le titre des pièces jointes seront stockées dans les balises _AFS_TITLE et _AFS_ABSTRACT.

Attention : Si les pièces jointes à indexer sont disponibles via des liens HTTP, il faudra déclarer un ou plusieurs crawlers dans le fichier afs.xml dans la section Crawler §2.1.3.

<Preprocessing>

La section `Preprocessing` permet de nettoyer le contenu de certaines balises, dans le cas où elles contiennent soit des ancres ou balises HTML, soit des informations inutiles :

`replace/@pattern` : Pattern à remplacer défini par l'utilisation d'expressions régulières

`replace/@with :` Le pattern sera remplacé par cette valeur

Exemple :

```
   <Preprocessing>
      <!-- Eliminer les éléments entre crochets.-->
      <Replace pattern="\[[^]]*\]" with=""/>
   </Preprocessing>
```

<Indexed_Tags>

La section `Indexed_Tags` permet de définir les champs à indexer, que ce soit des balises ou des attributs.

Chaque balise `Indexed_Tag` permet d'indiquer un champ à indexer et cette balise admet trois attributs :

`name` : le nom du champ à indexer (balise ou attribut), défini relativement à `Root_Tag`.

`score` : un pourcentage qui sera utilisé dans le calcul de pertinence afin de donner une importance relative aux différents champs. Le score maximum de 100% indiquant que celui-ci est prédominant en terme de pertinence.

`field` : un nom choisi par l'utilisateur qui permet de créer un champ virtuel s'étendant sur plusieurs champs XML. Dans l'exemple ci-dessous, on crée un champ virtuel `content` qui regroupe les balises `DESCRIPTION` et `ABSTRACT`. Il est alors possible de rechercher spécifiquement dans le champ virtuel `content` et la recherche portera alors sur les deux balises.

Il est possible d'indexer des sous-objets en les déclarant sous la balise `<Indexed_Collection>`.

Cette balise a pour attribut “object” qui définit le xpath de l'objet père et contient la liste des sous objets. L'indexation de sous-objets permet de connaître le sous-objet matché par un mot-clé lors d'une requête.

```
 <Indexed_Tags>
    <Indexed_Tag name="PAGE/TITLE" score="90"/>
    <Indexed_Tag name="PAGE/KEYWORDS" score="90"/>
```

```
    <Indexed_Tag name="PAGE/DESCRIPTION" score="80" field="content" />
    <Indexed_Tag name="PAGE/ABSTRACT" score="70" field="content" />
    <Indexed_Tag name="PAGE/TT_CONTENT/TEXT" score="60"/>
    <Indexed_Tag name="PAGE@TOPIC" score="30"/>
  <Indexed_Collection object="PAGE/fiche">
    <Indexed_Tag name="PAGE/fiche/titre" score="80"/>
    <Indexed_Tag name="PAGE/fiche/contenu score="60"/>
  </Indexed_Collection>
 </Indexed_Tags>
```

Attention : L'ordre dans lequel les champs à indexer est défini a une importance.
En effet, dans la pertinence des résultats renvoyés le pathlen a une importance majeure. Cette mesure est calculée du 1er champ au suivant.
Ainsi si on veut considérer comme pertinent la cooccurrence de deux termes provenant de champs différents, il convient de les déclarer à la suite.

Exemple :

```
 <Indexed_Tag name="Item/LIBELLE" score="90"/>
   <!-- On indexe le coloris juste apres le libelle pour minimiser la
   distance dans des recherches comme pantalon beige -->
   <Indexed_Tag name="Item/COLORI" score="40"/>
```

<Title_Items>

La section `Title_Items` permet de fabriquer des titres élaborés pour les réponses, en combinant les valeurs de plusieurs champs de l'enregistrement.

Cette combinaison prend la forme d'une liste de triplets {préfixe, champ, suffixe } où :

– le préfixe et le suffixe, qui sont définissables respectivement par les attributs `prefix` et `suffix`, sont des chaînes de caractères optionnelles qui seront placées respectivement avant et après la valeur du champ ;

– l'attribut `name` désigne le champ dont la valeur est copiée dans le titre. Le champ peut être optionnel ou obligatoire - auquel cas il devra être présent et avoir un contenu non vide faute de quoi l'enregistrement ne sera pas indexé.

Dans tous les cas, le titre ainsi constitué devra être non vide - sinon l'enregistrement ne sera pas indexé.

Il est possible de déclarer plusieurs `Title_Item` qui seront tous utilisés et composés dans l'ordre d'apparition afin de constituer le titre final.

Exemple :

```
 <Title_Items>
     <Title_Item prefix="Article" name="PAGE@ID">
     <Title_Item prefix=" - " name="PAGE/TITLE" suffix=" - publié le "/>
     <Title_Item name="PAGE/PUBLICATION_DATE"/>
 </Title_Items>
```

Et les titre des réponses auront la forme :

"*Article 132 - Le retour du printemps - publié le 02/03/2005*"

<Contents_Item>

La section `Contents_Item` permet de construire le résumé des réponses de façon tout à fait similaire à ce qui a été présenté pour les titres, selon le principe de triplets `{ préfixe, champ, suffixe }`.

Contrairement aux titres, les résumés peuvent être vides sans que cela invalide les enregistrements.

Exemple :

```
<Contents Items>
      <Contents Item préfix="Résumé : " Name="PAGE/DESCRIPTION"/>
</Contents Items>
```

<Store Items>

La section `Store_Items` permet de lister des champs qui ne seront pas indexés par AFS (c'est-à-dire que les recherches des utilisateurs ne s'appliqueront pas à ces champs), mais ils seront néanmoins pris en compte (mémorisés) afin d'être utilisés à des fin de tri, de catégorisation, de filtre ou d'affichage.

Chaque balise `Store_Item` contient un attribut `name` pour désigner le champ à mémoriser et éventuellement un attribut `@rename_node` qui permet de renommer le champ dans le rendu XML.

Exemple :

```
<Store_Items>
      <Store_Item name="PAGE@ID"/>
      <Store_Item name="PAGE/MODIFICATION_DATE" rename_node="date"/>
      <Store_Item name="PAGE/TT_CONTENT/GROUP"/>
</Store_Items>
```

<URL_Item>

La balise `URL_Item` permet d'indiquer le (ou les) champ(s) à utiliser pour constituer l'URL vers laquelle pointer.

Exemple :

```
<URL_Items>
   <URL_Item name="PAGE/NUMERO"/>
   <URL_Item name="PAGE/IDENTIFIANT"/> <!-- n elements -->
</URL_Items>
```

2.4.2 Indexation des données structurées et déclaration de filtres paramétriques : le fichier shmxml

Le moteur de recherche permet de réaliser une indexation plein texte (recherche avec tolérance) mais aussi une indexation exacte permettant un tri, un filtrage ou un comptage sur certains champs.

Pour cela, il suffit de déclarer dans un fichier chacun des champs à positionner dans l'index « SHM » avec la définition de leur indexation. Ce fichier, au format XML, appelé `fichier.shmxml` doit être placé dans le répertoire `$AFS/plugins/index`.

Exemple de fichier `fichier.shmxml` :

```
<Doc>
  <MetaDatas>
    <!-- Declaration de la racine -->
    <Root>root</Root>
    <!-- Declaration du champ de test de validité des fiches -->
    <Validity>
      <Field name="__is_valid"/>
   </Validity>
  </MetaDatas>
  <!-- Liste des champs -->
  <Fields>
    <Field name="is__valid" type="BOOL_TYPE">
      <Source type="true">
        <xpath>count(root/fiche/statut='1') &gt;0</xpath>
```

```
      </Source>
    </Field>
    <Field name="PRIX" type="FLOAT_TYPE">
      <Source type="xml">
        <xpath>root/fiche/Tarif/PRIX/@Prix_VenteTTC</xpath>
      </Source>
    </Field>
    <Field name="RAYON" type="STRING_TYPE">
      <Source type="xml">
        <xpath>root/fiche/MARQUE/@ID</xpath>
      </Source>
    </Field>
    <Field name="REFERENCE" type="STRING_TYPE">
      <Source type="xml">
        <xpath>root/fiche/REFERENCE</xpath>
      </Source>
    </Field>

    <Field name="DATE" type="DATE_TYPE">
      <Source type="xml">
        <xpath>root/fiche/date_publication</xpath>
      </Source>

      <Date_Format>%Y/%m/%d</Date_Format>
    </Field>
  </Fields>
  <!-- Liste des paramtères de filtrage -->
  <Filter_Parameters>
    <Filter_Parameter name="PRIX_FILTER">
      <Field name="PRIX" type="EQUAL_TO"/>
      <!-- Filtre sur les prix dont la valeur est égale à x -->
    </Filter_Parameter>
    <Filter_Parameter name="MIN_PRIX_FILTER">
      <Field name="PRIX" type="GREATER_THAN_OR_EQUAL_TO"/>
      <!-- Filtre sur les prix dont la valeur est supérieure ou égale à x -->
    </Filter_Parameter>
    <Filter_Parameter name="MAX_PRIX_FILTER">
      <Field name="PRIX" type="LESS_THAN_OR_EQUAL_TO"/>
      <!-- Filtre sur les prix dont la valeur est inférieure ou égale à x -->
    </Filter_Parameter>
    <Filter_Parameter name="RAYON_FILTER">
      <Field name="RAYON" type="EXACT_MATCH"/>
      <!-- Filtre sur les rayons dont la valeur matche x -->
    </Filter_Parameter>
    <Filter_Parameter name="REFERENCE_FILTER">
      <Field name="REFERENCE" type="STARTS_WITH"/>
      <!-- Filtre sur les références dont la valeur commence par x -->
    </Filter_Parameter>
  </Filter_Parameters>
  <!-- Liste des catégorisations (Parametriques) -->
  <Categorisations>
    <Categorisation name="PRIX">
      <Field name="PRIX"/>
      <!-- Filtre sur lequel on souhaite faire une catégorisation avec décompte -->
    </Categorisation>
    <Categorisation name="RAYON">
      <Field name="RAYON" separator="/">
        <!-- on souhaite faire une catégorisation avec décompte dont les différents
niveaux de hiérarchisation sont séparés par un '/'-->
        <Labels xpath="root/fiche/MARQUE/@LIBELLE">
          <Label value="Pas de marque">Autres</Label>
        </Labels>
      </Field>
    </Categorisation>
  </Categorisations>
```

```
</Doc>
```

La balise racine du fichier est la balise `<Doc>`, elle peut contenir jusqu'à 4 sections définies ci-dessous.

- Section `MetaDatas` (obligatoire)

`<Root>` : définit le nom du noeud racine

<Validity> : définit le nom du champ permettant de tester la validité des fiches

- Section `Fields` (obligatoire) : liste les champs utilisés

Chaque champ est déclaré au sein d'une balise `<Field>` qui définit son nom (`Field@name`), son type (`Field@type`), le type de sa source (`Source@type`) et son chemin d'accès (`xpath`).

Le nom donné est arbitraire; quant à l'attribut type il peut contenir les valeurs suivantes :

Pour un champ portant une valeur type de données :	***Valeur du champ « type »***
Chaîne de caractères	`STRING_TYPE`
Flottant	`FLOAT_TYPE`
Date	`DATE_TYPE`
Entier	`UINT32_TYPE`
Pour un champ ne portant pas de valeur, mais permettant d'inclure des sous champs Ici, il s'agit de définir une cardinalité N. Par exemple, un article est classé dans différentes catégories	`COLLECTION_TYPE`
Pour un champ ne portant pas de valeur, mais permettant d'inclure des sous champs Ici, il s'agit de définir une cardinalité 1. Par exemple, un contact possède une addresse, une addresse étant composée d'un ville, d'une rue, d'un numéro, ...	`AGGREGATE_TYPE`

Si le type est DATE_TYPE alors le format de la date devra être défini dans Field/Date-Format au format strftime de la librairie C.

- Section `<Filter_Parameters>` : définit les filtres attachés à chaque champ

Chaque filtre est défini au sein de la section `Filter_Parameter` où :

– `Filter_Parameter@name` définit le nom du filtre: c'est le nom que vous pourrez utiliser à l'appel du moteur AFS (paramètre d'URL). Ce nom doit toujours finir par `_FILTER`.

– `Field@name` donne le nom du champ indexé (nom précédemment déclaré dans la section `<Fields>`)

– `Field@type` donne le type d'indexation. Les valeurs possibles sont les suivantes :

Type de champ	***Type de filtre***	***Commentaire***
Numérique (nombre, date, ...)	`EQUAL_TO`	Égal
	`LESS_THAN`	Inférieur (strict)

Type de champ	***Type de filtre***	***Commentaire***
	`LESS_THAN_OR_EQUAL_TO`	Inférieur ou égal
	`GREATER_THAN`	Supérieur (strict)
	`GREATER_THAN_OR_EQUAL_TO`	Supérieur ou égal
Chaînes de caractères	`EXACT_MATCH`	Égalité sur toute la chaine
	`STARTS_WITH`	Égalité sur le début de la chaine
	`STRING_ID_MATCH_MODE`	matche l'id de la chaine, l'id est automatiquement généré par AFS

– `Field@combinator` : Permet de préciser la cardinalité du paramètre.

Type	***Commentaire***
`NO_COMBINATOR`	Pas de combinaison possible
`OR_COMBINATOR`	Combinaison possible – l'union des valeurs sera renvoyé – les valeurs alternatives sont délimitées par des tirets « `-` »
`AND_COMBINATOR`	Combinaison possible – l'intersection des valeurs sera renvoyé – les valeurs alternatives sont délimitées par des tirets « `-` »

Section `<Categorisations>` : définit les champs à catégoriser (sur lesquels compter)

Il est possible de catégoriser les données selon un champ au sein de la balise `<Categorisation>` où :

– `Categorisation@name` définit le nom de la catégorie.

– `Field@name` donne le nom du champ indexé (nom déclaré dans la section <Fields>)

– `Field@separator` permet de déclarer la présence d'un caractère séparateur de valeurs pour les catégorisations hiérarchiques (exemple pour une nomenclature de produits : Rayon_niveau1/Rayon_Niveau2/Rayon_niveau3).

– `Labels@xpath` : Pour les valeurs chaîne, cette option permet de déterminer la valeur donnée au label.

– `Labels/Label/@value` : Permet de spécifier un label pour une valeur.

3 Créer un environnement de réponse

3.1 Configuration

Les fichiers de configuration principaux de l'environnement de réponse AFS se trouvent dans le répertoire "conf" lui-même situé dans le répertoire d'installation d'AFS, donc habituellement `/usr/local/afs/conf`.

Ces fichiers sont au nombre de 2 :

- `afs.xml` contient les éléments de configuration principaux. Un template est fourni en annexe.
- `Services.conf` décrit pour chaque service de recherche les différents agents mis en oeuvre et les serveurs autorisés à participer à l'architecture de réponse.

3.1.1 Fichier afs.xml

Comme son extension l'indique, le fichier `afs.xml` est au format XML. Le noeud racine est <AFS> et les sections principales sont les suivantes :

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no" ?>
<AFS>
   <!-- - - - - - - - - - - - - - - - - - - - - - -->
   <!-- SECTIONS POUR LES PARAMETRES GENERIQUES -->
   <!-- - - - - - - - - - - - - - - - - - - - - - -->

   <Base>
          <!-- environnement de base d'AFS  -->
   </Base>

   <Aliases>
          <!-- définit des alias pour les services et les sites -->
   </Aliases>

   <Servers>
          <!-- définition précise de chacun des services de recherche -->
   </Servers>

   <Alarm>
      <!-- paramètres utilisés pour l'envoi d'alarme -->
   </Alarm>

   <!-- - - - - - - - - - - - - - - - - - - - - - - - - - -->
   <!-- SECTIONS SPECIFIQUES A LA CONFIGURATION FRONT-END -->
   <!-- - - - - - - - - - - - - - - - - - - - - - - - - - -->

   <Agents>
          <!-- configuration technique du front end et des différents agents -->
   </Agents>

   <DTD>
      <!-- options spécifiques pour les DTD de réponse des services -->
   </DTD>

   <StyleSheets>
      <!-- feuilles de style XSL à appliquer pour générer les flux de réponse -->
   </StyleSheets>
```

```
   <User_Parameters>
      <!-- définition des paramètres d'appel spécifique au service de recherche-->
   </User_Parameters>


   <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
   <!-- SECTIONS SPECIFIQUES A LA CONFIGURATION BACK-OFFICE -->
   <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->

   <Back_Office>
      <!-- Pour activer l'agent Hit-Parade -->
   </Back_Office>

   <Tracking>
      <!-- Pour activer l'agent Hit-Parade -->
   </Tracking>


</AFS>
```

3.1.1.1 Les sections pour les paramètres génériques

La section Base peut contenir les définitions suivantes :

```
<Base>
  <!-- Faux si les modules back-end sont tous situés sur le même serveur -->
  <Clustered_Service>false</Clustered_Service>

  <!-- Pour définir la langue. Valeur par défaut : fr_FR -->
  <Locale>fr_FR@euro</Locale>

  <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
  <!-- permet de fixer quelques limites et les droits au niveau du système -->
  <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
  <OS>
     <!-- la charge maximale (load) autorisée. En cas de dépassement, AFS arrête
          de lancer des agents -->
     <Max_Load>50</Max_Load>

     <Privileges>
        <!-- Permet de forcer le user et le group des process AFS à partir
          de leurs noms (setuid, setgid) -->
        <Ownership>
             <!-- Utiliser la valeur false permet de ne plus activer les droits sans
                       supprimer cette section de configuration -->
           <Force_Ownership>true</Force_Ownership>
           <User_Name>antidot</User_Name>
           <Group_Name>antidot</Group_Name>
        </Ownership>
     </Privileges>

     <!-- nombre maximum de process AFS qui peuvent être lancés. La valeur par
              défaut est 980. -->
     <Max_Processes>4096</Max_Processes>
  </OS>

  <!-- Temps d'attente maximum (en secondes) pour un ping applicatif. Défaut=3 -->
  <Ping_Timeout_Seconds>3</Ping_Timeout_Seconds>

  <!-- Temps maximum (en secondes) pour envoyer des données. Défaut=30 -->
  <Default_Timeout_Seconds>20</Default_Timeout_Seconds>

  <!-- Temps d'attente (en secondes) entre deux cycles successifs de vérification
```

```
        que tous les serveurs ont assez de composants actifs.
       A ajouter à Fork_Loop_Delay_Milliseconds. Défaut=15 -->
   <Fork_Loop_Delay_Seconds></Fork_Loop_Delay_Seconds>

   <!-- Temps d'attente (en millisecondes) entre deux cycles successifs
            de vérification que tous les serveurs ont assez de composants actifs.
            A ajouter à Fork_Loop_Delay_Seconds. Défaut=0 -->
   <Fork_Loop_Delay_Milliseconds></Fork_Loop_Delay_Milliseconds>

</Base>
```

La section Aliases permet de définir des noms de service en clair pour les identifiants internes. Dans l'exemple qui suit, AFS réalise deux services de recherche, l'un de nom "SiteWeb" (qui indexe les sites Web publics de l'entreprise) et l'autre "Intranet" (qui indexe l'intranet).

```
<Aliases>
   <Service id="100" name="SitesWeb"/>
   <Service id="103" name="Intranet"/>
</Aliases>
```

La section Servers permet de définir des informations de configuration communes à tous les services de recherche puis de raffiner ou de redéfinir les valeurs service par service.

```
  <Servers>
     <!-- Le numéro port TCP de base (de préférence un multiple de 100) à partir
          duquel les différents deamon seront à l'écoute selon un incrément défini
          dans la configuration -->
     <Base>2000</Base>

     <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
     <!-- Paramètres pour as_init, le daemon de lancement des process AFS -->
     <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
     <Init>
           <!-- PortTCP sur lequel le serveur as_init écoute, défini avec un offset
   appliqué
              à Servers/Base. Défaut=0 -->
        <Server_Port>0</Server_Port>

        <!-- Comment et à qui sont envoyées les alertes -->
        <Alert>
           <!-- le programme d'envoi à utiliser. Défaut : /bin/mail -->
           <Email_Binary_Path>/usr/local/bin/exim</Email_Binary_Path>

           <!-- Les destinataires -->
           <Send_To>support@antidot.net</Send_To>
           <Send_To>admin@client.com</Send_To>
        </Alert>
     </Init>


     <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
     <!--  Paramètres des serveurs principaux                -->
     <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
     <Queries>

        <!-- - - - - - - - - - - - - - - - - - - - - - - - - - -->
        <!-- Définition du Request Manager principal -->
        <!-- - - - - - - - - - - - - - - - - - - - - - - - - - -->
        <Master_Request_Manager>
           <!-- Son adresse IP ou nom de serveur (obligatoire) -->
           <IP>62.210.155.5</IP>
```

```
        <!-- et les ports TCP pour les services.
                    Offset par rapport à Servers/Base -->
        <Server_Port>100</Server_Port>
        <Ping_Port>105</Ping_Port>
        <Spy_Port>106</Spy_Port>

        <!-- Permet de définir l'ensemble serveurs (et agents) qui sont en contact
                   avec le Request_Manager, y compris findall -->
        <ACL>
           <Default_Policy>Deny</Default_Policy>
           <IP_Rules> <!-- De 62.210.155.0 à 62.210.155.69 -->
              <Allow>62.210.155.</Allow>
           </IP_Rules>
        </ACL>

        <!-- Le fichier de configuration des services.
                    Par défaut : Services.conf -->
        <Configuration_File>Services.conf</Configuration_File>

        <!-- Nombre de tâches de réponse. Defaut=10 -->
        <Nb_Tasks>10</Nb_Tasks>
        <!-- Nombre de tâches par service pour planifier les réponses. 1 par
défaut
             mais à augmenter en cas de forte charge -->
        <Nb_Scheduler_Tasks>5</Nb_Scheduler_Tasks>
        <!-- Durée d'inactivité après laquelle une tâche se termine.
                    Défaut=3600s -->
        <Exit_After_Idle_Seconds>3600</Exit_After_Idle_Seconds>
        <!-- Nombre de tâches par service de recherche dédiées à la construction
             d'arborescence d'agents prêts à répondre. Défaut=1 mais peut être
                   augmenté en cas de très forte charge. -->
        <Nb_Agent_Factory_Tasks>1</Nb_Agent_Factory_Tasks>

        <!-- Section qui permet de définir les paramètres de tunning avancés -->
        <Max>
           <!-- Nombre max de requêtes en attente de traitement
                       tous services confondus.
                       En cas de dépassement les requêtes sont répondues "vide".
-->
           <Pending_Requests>50</Pending_Requests>
           <!-- Si rejected_requests (nombre cumulé de requêtes rejetées) est
                       atteint, alors le Request_Manager s'arrête. Defaut=10 -->
           <Rejected_Requests>10</Rejected_Requests>
        </Max>
     </Master_Request_Manager>


     <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
     <!-- Possibilité de définir un request manager secondaire qui sera utilisé
          en cas d'arrêt du Request manager principal                         -->
     <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
     <Slave_Request_Manager>
           <!-- Seules les valeurs IP, Nb_Tasks et Exit_After_Idle_Seconds sont
             spécifiques. Les autres sont les mêmes que pour le Master. -->
        <!-- Son adresse IP ou nom de serveur (obligatoire) -->
        <IP>62.210.155.6</IP>

        <!-- et les ports TCP pour les services.
                    Offset par rapport à Servers/Base -->
        <Server_Port>110</Server_Port>
        <Ping_Port>115</Ping_Port>
        <Spy_Port>116</Spy_Port>

        <!-- Nombre de tâches de réponse. Defaut=2 -->
```

```
            <Nb_Tasks>2</Nb_Tasks>

            <!-- Durée d'inactivité après laquelle une tâche se termine.
                        Défaut=600s -->
            <Exit_After_Idle_Seconds>600</Exit_After_Idle_Seconds>
         </Slave_Request_Manager>


         <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
         <!-- Les Manager (principal et secondaire) de log des requêtes           -->
         <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
         <Log_Manager>
            <!-- Adresse ou nom du serveur principal. Defaut : 127.0.0.1 -->
            <Master_IP>62.210.155.9</Master_IP>
            <!-- Adresse ou nom du serveur secondaire. Laisser vide si il n'y en a
                 pas pour le désactiver. Defaut : 127.0.0.1 -->
            <Slave_IP></Slave_IP>

            <!-- et les ports TCP pour les services.
                        Offset par rapport à Servers/Base -->
            <Server_Port>120</Server_Port>
            <Ping_Port>125</Ping_Port>

            <!-- Nombre de tâches simultanées pour recevoir les logs. Defaut=10 -->
            <Nb_Tasks>10</Nb_Tasks>

         </Log_Manager>


        <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
        <!-- Le Document Cache Manager qui permet d'accéder aux documents cachés,
             c'est à dire tels qu'ils étaient lors de l'indexation              -->
        <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
        <Document_Cache_Manager>
        </Document_Cache_Manager>


         <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
         <!-- Le Fresh Cache Manager pour cacher les réponses fraîches            -->
         <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -->
        <Fresh_Cache_Manager>
        </Fresh_Cache_Manager>


         <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  -->
         <!-- La section Shared Memory Key permet de définir les clés de SHM -->
         <!-- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  -->
         <Shared_Memory_Key>
           <!-- SHM de Directory -->
           <Directory/> <!-- entier : 100 -->

           <!--
              Antibot  :  200
              Engine1  :  300
              Engine2  :  400
              Engine3  :  500
              Engine4  :  600
              Freshbot :  700
              User1    :  800
              User2    :  900
              User3    : 1000
              User4    : 1100
              User5    : 1200
              User6    : 1300
```

```
            User7    : 1400
            User8    : 1500
            User9    : 1600
            Mail     : 1700
          -->
        </Shared_Memory_Key>
          <!--  -->
          </> <!--  -->
          <!--  -->
          </> <!--  -->
          <!--  -->
          </> <!--  -->

    </Queries>


 </Servers>
```

La section Alarm permet de définir les paramètres permettant l'envoi d'alarme ainsi que les destinataires de ces alarmes.

```
<Alarm>
          <Gateways>
                    <Master_Host/>
                    <Slave_Host/> (localhost par défaut)
          </Gateways>
          <Recipients>
                    <Admin_Email/> <--- liste de mels
          </Recipients>
</Alarm>
```

3.1.1.3 Les sections spécifiques au front-end AFS

La section Agents définit les options :

- communes à tous les agents : Common ;
- spécifiques à un service donné : Service_Specific ;
- relatives au front end CGI : Front_End_CGI ;
- des agents :
  - web Directory : Directory ;
  - root : Root ;
  - XML : XML ;
  - Web : Web ;
  - Engine : Engine ;
  - Shop Directory : Shop_Directory ;
  - Suggestion orthographique : Hints ;
  - Liens remontés dans la réponse : Promotion ;
  - Liens associés publicitaires : Commercial ;
  - Suggestions d'expressions liées à la recherche : RTE ;
  - Recherches tranversales : See_Also ;
  - Recherche floue : Mail.
- des recherches plein texte: Full_Text_Search ;
- des services de liens sponsorisés :
  - Overture : Overture ;
  - Espotting : Espotting ;
  - PFP : PFP.
- du service de nom de domaine : DNS.

```
<Agents>
   <Common>
      <!-- Temps maximum pour envoyer des données -->
      <Timeout_Seconds/> <!-- entier : 10 -->

      <!-- Durée pendant lequel une réponse en provenance du request manager est
       attendue -->
      <Request_Manager_Failure_Timeout_Seconds/> <!-- entier : 30 -->

      <!-- Temps au bout duquel l'agent s'arrête s'il n'a pas reçu de requête -->
      <Exit_After_Idle_Seconds/> <!-- entier : 600 -->

      <!-- Nombre maximum de cycles d'attente de réponse d'un fils. Chaque cycle a
       une durée de Timeout_Seconds -->
      <Max_Child_Wait_Cycles/>  <!-- entier : 3 -->

      <!-- Nombre de réponses après lequel l'agent doit s'arrêter -->
      <Exit_After_Nb_Queries/> <!-- entier : 5 -->

      <!-- Temps pendant lequel le findall attend le rappel du root-->
      <Findall_Callback_Timeout_Seconds/> <!-- entier : 60 -->
```

```
      <!-- Redirection des utilisateurs en cas d'erreur -->
      <Redirect_To_URI_On_Error/> <!-- string : http://www.antidot.net-->
    </Common>

    <!-- Options communes à tous les agents d'un service -->
    <Service_Specific>
      <!-- Ignorer les mots vides dans la recherche ? -->
      <Ignore_Empty_Words/> <!-- booléen : true -->

      <!-- Activer les optimisations des données très volumineuses ? Vrai pour web,
       faux autrement. -->
      <Enable_Large_Optimisations/> <!-- booléen : false -->

      <!-- Activer les optimisations des données volumineuses ?(> 1M pages) -->
      <Enable_Medium_Optimisations/> <!-- booléen : false -->

      <!-- Utiliser les optimisations relatives aux mots optionnels des requêtes ?
       (Utiliser vrai sauf si vous savez ce que vous faites) -->
      <Enable_Optional_Optimisations/> <!-- booléen : true -->

      <!-- Nombre maximum de pages par serveur à envoyer à l'agent web depuis
       l'agent engine-->
      <Max_Pages_Per_Server/> <!-- entier : 100 -->

      <!-- Pour les modes d'optimisation Large/Medium, les pages possédant un score
       faible sont supprimées après que le nombre de pages suivant ait été dépassé
       -->
      <Prune_Threshold/> <!-- entier : 10000 -->

      <!-- Si les pages sont clusterisées, nombre de réponses à analyser après la
       fin de la page actuelle afin d'améliorer la précision du pager. 50 est une
       bonne valeur pour les grands volumes ; 1000 pour les moyens ; ignorée dans
       les autres cas. -->
      <Lookahead_Replies/> <!-- entier : 50 -->

      <!-- Activer le mode de parsing 'references' ? -->
      <Reference_Parsing_Mode/> <!-- booléen : false -->


      <!-- Autoriser les recherches avec des * en partie droite ? -->
      <Allow_Wildcards/> <!-- booléen : false -->

      <!-- Recherches avec des * en partie droite automatiques ? -->
      <Implicit_Wildcards/> <!-- booléen : false -->

      <!-- Si vrai, une requête vide va donner lieu à une réponse comportant
         l'ensemble des pages de la base de données. A utiliser avec précaution car
       cela peut coûter cher. Pour les agents Antibot, cette ressource doit aussi
       être spécifiée dans la section spécifique de l'agent. -->
      <Allow_Empty_Query/> <!-- booléen : false -->

      <!-- Nombre de pages stockées en cache pour chaque requête. Il est dangereux
       de mettre plus de 1 ici en général. -->
      <Store_Nb_Pages_In_Cache/> <!-- entier : 1 -->

      <!-- Parser les 'et', 'ou', 'sauf' comme des opérateurs -->
      <Parse_Word_Modifiers/> <!-- booléen : false -->

      <!-- Si Parse_Word_Modifiers, jeton pour 'and' -->
      <And_Modifier/> <!-- string : et -->

      <!-- Si Parse_Word_Modifiers, jeton pour 'or' -->
      <Or_Modifier/> <!-- string : ou -->
```

```
        <!-- Si Parse_Word_Modifiers, jeton pour 'not' -->
        <Not_Modifier/> <!-- string : sauf -->

        <!-- Produire les résultats au format XML, sans XSLT -->
        <Deliver_Raw_XML/> <!-- booléen : false -->

        <!--  Si du XML est produit par défaut, produire du HTML pour les sites
         suivants -->
        <Raw_HTML_Site/> <!-- liste d'entier -->

        <!--  Si du HTML est produit par défaut, produire du XML pour les sites
         suivants -->
        <Raw_XML_Site/> <!-- liste d'entier -->

        <!-- Tag à utiliser pour le calcul de clé -->
        <Checksum_Tag_Name/>

        <!-- Faut-il protéger le service en vérifiant le checksum -->
        <Require_Checksum/> <!-- booléen : true -->

        <!--  Liste de Bots pour lesquels le service sera refusé -->
        <Banned_Bots/> <!-- -->

        <!--  Liste d'IPs pour lesquelles le service sera refusé -->
        <Banned_IP/>

        <!-- Faut-il utiliser le cache de réponses ? -->
        <Use_Cache_Manager/> <!-- booléen : false -->

        <!-- Afficher les promotions et les PFP sur toutes les pages ? -->
        <Display_Promotions_On_Every_Page/> <!-- booléen : false -->

        <!--  -->
        <Use_UNIQUE2/> <!-- booléen : false -->

        <!-- Coder le nom unique de l'agent dans les arguments -->
        <Encode_UNIQUE_In_Args/> <!-- booléen : false -->

        <!-- Redirection des utilisateurs en cas d'erreur -->
        <Redirect_To_URI_On_Error/> <!-- http://www.antidot.net-->

        <!-- Namespace externe pour utiliser clé=nom valeur=URI -->
        <Namespace/> <!-- string_map -->

        <!-- Valeur optionnelle de l'attribut 'id' pour la balise SearchResult -->
        <SearchResultId/> <!-- string : '' -->

        <!-- Entête optionnel XML -->
        <XML_Header_String/> <!-- string : '' -->

        <!--  Pied de page optionnel XML-->
        <XML_Footer_String/> <!-- string : '' -->

        <!-- Liste de fichiers à inclure, la clé est la balise racine et la valeur est
           le nom du fichier. -->
        <Include_XML_File/> <!-- string_map -->

        <!-- OLD AGENTS -->

        <!-- Générer les informations QueryAdvertisment ? -->
        <Output_Advertisment_Keywords/> <!-- booléen : false -->

    </Service_Specific>
```

```
<!-- Web Directory Agent -->
<Directory>
   <!--  Hash size -->
   <Hash_Size/> <!-- entier : 11 -->
</Directory>

<!-- Options du front end CGI -->
<CGI_Front_End>
   <!-- Nombre maximal de réponses qui peuvent être demandées via le paramètre
    CGI NB_REPLIES. -->
   <Max_CGI_Specified_Replies/> <!-- Entier : 0 (désactivé) -->

   <!-- Annule le CGI après la durée spécifiée. 0 pour désactiver l'option. -->
   <Abort_After_Seconds/> <!-- entier : 60 -->

   <POST>
      <!-- POST autorisé sur ce service ? -->
      <Allow/> <!-- booléen : false -->

      <Plateform_Wide_Settings>
         <!-- Taille maximale du contenu en octets. -->
         <Max_Content_Length/> <!-- entier : 65536 -->
      </Plateform_Wide_Settings>
   </POST>

   <!-- Access Control List -->
   <ACL>
      <!-- Si vrai, tout le monde (sauf les IPs bannies) peut effectuer des
       requêtes. -->
      <Default_Policy/> <!-- policy -->

      <!-- Liste ordonnées d'IP à autoriser ('Allow') ou à interdire ('Deny') -->
      <IP_Rules/> <!-- rule_list -->

      <!--  URI sur laquelle sont redirigés les utilisateurs en cas d'accès
       refusé. -->
      <Redirect_To_URI_On_Access_Denied/> <!-- http://www.antidot.net -->
   </ACL>

   <!-- Specifie le nom d'un paramètre à utiliser comme paramètre de site (Copie
    la valeur de ce paramètre comme la valeur du paramètre X). -->
   <Alternate_Site_Param/> <!-- string -->

   <!-- clé='nom_du_paramètre', valeur='valeur_du_paramètre:site_id'. Si
    nom_du_paramètre a la valeur valeur_du_paramètre alors règle site à
    site_id.-->
   <Implicit_Site/> <!-- string_map -->

   <!-- Site à utiliser si Implicit_Site est défini mais qu'aucun ne correspond
    -->
   <Default_Implicit_Site/> <!-- entier : 0 -->

   <!-- Si défini, le champ referer de la table de log vaudra la valeur définie
    -->
   <Log_Parameter_Value_Instead_Of_Referer/> <!-- string : '' -->

   <Output>
      <!-- Encoding de la sortie -->
      <Encoding/> <!-- string : ISO-8859-1 -->

      <!-- Type mime de la sortie. Ne peut être que text/html, text/xml ou
       text/plain -->
      <Mime_Type/> <!-- string : 'text/html' -->
```

```
         <!-- Si défini, utilise ce type mime (et pas Mime_Type) -->
         <Declare_Mime_Type/> <!-- string : '' -->

         <!-- Si défini, ajoute le 'charset=encoding' au type mime (requis pour XML
          -->
         <Append_Encoding_Charset_To_Mime_Type/> <!-- booléen : false -->

         <!-- Pour ces sites, le résultat est une pièce jointe qui n'est pas
          destinée à être affichée dans le navigateur -->
         <Deliver_Attachment/> <!-- booléen : false -->

         <!-- Si non vide et si Deliver_Attachment est vrai alors cette valeur
          spécifie le nom du fichier. -->
         <Attachment_File_Name/> <!-- string : '' -->

      </Output>

      <!-- Paramètres spécifiques aux services -->
      <Parameters>
         <!-- Les paramètres sont propagés aux agents and copie dans le flux XML de
          sortie -->
         <Propagate_Parameter/> <!-- string_list -->

         <!-- Paramètres à utiliser pour un champ de recherche clé=nom_champ
          valeur=nom_paramètre (sera toujours suffixé par *KEYWORDS) -->
         <Field_Parameter/> <!-- string_map -->

         <!-- Si vrai, génère les paramètres F_MANDATORY/OPTIONAL/EXCLUDED utiles
          pour les recherches avancées. -->
         <Generate_Form_Parameters/> <!-- booléen : true -->
      </Parameters>

      <!-- Paramètres liés aux cookies -->
      <Cookies>
         <!-- Stocke la valeur des cookies (clé=nom_cookie) dans le paramètre
          (nom_paramètre=valeur) -->
         <Store_Cookie_Value_In_Parameter/> <!-- string_map -->
      </Cookies>

      <Monitoring>
         <!-- Si vrai, les requêtes avec _NO_LOG=true ne seront pas logguées. -->
         <Enable_Probing/> <!-- booléen : false -->

         <!-- Liste ordonnée d'IP autorisées (Allow) ou refusées (Deny) pour le
          probing. La politique par défaut est Deny. -->
         <IP_Rules/> <!-- rule_list -->
      </Monitoring>

   </CGI_Front_End>

   <Root>
      <Service_Specific>
         <!-- Nombre total de réponses numérotées sur la première page. Si 0, la
          valeur est mise par défaut à Nb_Replies_Per_Page. -->
         <Nb_Replies_On_First_Page/> <!-- entier : 0 -->

         <!-- Répartie les réponses entre agents seulement si N=1 et unique=0 et si
          nom_paramètre est positionné sur une valeur donnée par une clé -->
         <Dispatch_Replies_On_First_Page>

            <!-- Si positionné, les réponses doivent être réparties -->
            <Parameter_Name/> <!-- string : '' -->
```

```
            <!-- Si vrai, écrase Nb_Replies_Per_Page et Nb_Replies_On_First_Page
             selon les règles de répartition -->
            <Smart_Nb_Replies/> <!-- booléen : true -->

            <!-- Si vrai et si Smart_Nb_Replies est vrai, le nombre de réponses sur
             la première page peut être spécifié par l'utilisateur via le paramètre
             nb_repliy. -->
            <Scale_Nb_Replies/> <!-- booléan : false -->

            <!-- si le nom_paramètre a pour valeur [key] alors root va produire la
             valeur [replies] (si Smart_Nb_Replies est activé). -->
            <Root_Replies/> <!-- string_map -->

            <!-- si le nom_paramètre a pour valeur [key] alors root va produire la
             valeur [replies] -->
            <Antibot_Replies/> <!-- string_map -->

            <!-- si le nom_paramètre a pour valeur [key] alors root va produire la
             valeur [replies] -->
            <Engine1_Replies/> <!-- string_map -->

            [...]

            <!-- si le nom_paramètre a pour valeur [key] alors root va produire la
             valeur [replies] -->
            <Engine4_Replies/> <!-- string_map -->

            <!-- si le nom_paramètre a pour valeur [key] alors root va produire la
             valeur [replies] -->
            <Freshbot_Replies/> <!-- string_map -->

            <!-- si le nom_paramètre a pour valeur [key] alors root va produire la
             valeur [replies] -->
            <Directory_Replies/> <!-- string_map -->

            <!-- si le nom_paramètre a pour valeur [key] alors root va produire la
             valeur [replies] -->
            <User1_Replies/> <!-- string_map -->

            [...]

            <!-- si le nom_paramètre a pour valeur [key] alors root va produire la
             valeur [replies] -->
            <User9_Replies/> <!-- string_map -->

            <!-- Si vrai, l'agent web va ajouter les réponses de ses enfants pour
             construire la page de réponses. Autrement, il va afficher au plus un
             nombre de réponses égale à celui du fils ayant le plus de réponses. -->
            <Web_Add_Children_Replies/> <!-- booléen : false -->

         </Dispatch_Replies_On_First_Page>

         <Pager>
            <!-- Liste de paramètres. Si n'importe quel de ces paramètres est
             positionné, l'agent doit répondre. -->
            <Only_Reply_When/> <!-- string_list -->

            <!-- Liste de paramètres. Si n'importe quel de ces paramètres est
             positionné, l'agent ne doit pas répondre. -->
            <Do_Not_Reply_When/> <!-- string_list -->

         </Pager>
      </Service_Specific>
   </Root>
```

```
<!-- Agent XML générique -->
<XML>
   <!-- Lies le nom d'un exécutable à une fonction AFS -->
   <Mapping/> <!-- string_map -->

   <Service_Specific>
      <!-- Hash Size -->
      <Hash_Size/> <!-- entier : 11 -->

      <!-- Type de normalisation : 'fr_thesaurus', 'fr_metaphone',
       'fr_sms_metaphone'. Doit correspondre au type utilisé pendant l'indexation
       -->
      <Normalization_Type/> <!-- string : fr_thesaurus -->

      <!-- Stocke les réponses dans la réponse principale avec les réponses des
       agents engine, directory, ... ou dans une réponse séparée. -->
      <Store_Replies_In_Main_Set/> <!-- booléen : false -->

      <!-- Si vrai, les réponses ne seront pas numérotées (utilisé pour générer
       des informations qui ne seront pas comptabilisées parmis les résultats
       comme les jeux de parametrics -->
      <Disable_Numbering/> <!-- booléen : false -->

      <!-- Génère une numérotation des réponses globales (i.e. le nombre change
       pour chaque pages). Les nouveaux services devraient utiliser vrai, faux
       par défaut pour les anciens. -->
      <Global_Numbering/> <!-- booléen : false -->

      <!-- Construit des réponses privées (implique l'utilisation d'un agent
       mix_cell) ou construit des réponses standalone (par défaut) -->
      <Build_Private_Replies/> <!-- booléen : false -->

      <!-- si vrai, charge seulement les contenus des réponses à afficher. Très
       rapide mais peut mener à des erreurs (Réponses possédant des contenus
       corrompus). -->
      <Load_Only_Displayed_Contents/> <!-- booléen : false -->

      <!-- Si utilisé avec d'autres agents, fils d'une mix_cell, ne pas générer
       de RTE au niveau de l'agent mais au niveau du parent. -->
      <Disable_Related_Expressions/> <!-- booléen : false -->

      <!-- Filtre paramétrique générique -->
      <Parametric_Filters>
         <!-- Autoriser le moteur des filtres paramétriques -->
         <Enabled/> <!-- booléen : false -->

         <!-- Si vrai, inclues les données paramétriques dans le jeu de réponse
          privé envoyé au père (web_raw,...). Par défault, par compatibilité,
          faux. -->
         <Include_Parametrics_In_Private_Replies/> <!-- booléen : false -->

         <!-- Filtres paramétriques additionnels -->
         <Builtin_Parametrics>
            <!--  !TEMP! si le champ est non-vide, utilisation de cet axe
             paramétrique pour le tri. -->
            <Sort_Integer/> <!-- string : '' -->

            <!-- Filtre parametric entier. Plusieurs filtres peuvent être
             spécifiés. clé=prefix:slot, valeur=label. Slot est un nombre entre 1
             et 24. Requiert [prefix]_int_parametrics.db. -->
            <Enable_Integer/> <!--  -->
```

```
            <!-- Filtre paramétrique date. Si non vide, génère cette catégorie
             avec un format valide strftime. -->
            <Enable_Date/> <!-- string : '' -->

            <!-- Filtres de catégorie, par id. Plusieurs filtres peuvent être
             specifiés. clé=prefix:slot, valeur=label. Slot est un nombre entre 1
             et 10. Requiert [prefix]_id_parametrics.db dans le SHM et
             [prefix]_categories.xml. -->
            <Enable_IdCat/> <!-- string_map -->


         </Builtin_Parametrics>
      </Parametric_Filters>

      <Sort>
         <!-- Si vrai, les objects de même pertinence sont rangés aléatoirement.
          -->
         <Enable_Random_Sort/> <!-- booléen : false -->

         <!-- Modes Full et Fresh, les agents travaillent avec deux bases de
          données. -->
         <Differential_Mode/> <!-- booléen : false -->

         <!-- Spécifique à l'agent -->
         <User1>
            <!-- Liste de paramètres. Si un paramètre de cette liste est
             positionné alors l'agent doit répondre. Si une liste de valeurs est
             fournie alors seulement la première valeur sera utilisée. Autrement,
             l'agent restera muet. -->
            <Only_Reply_When/> <!--  -->

            <!-- Liste de paramètres. Si un paramètre de cette liste est
             positionné alors l'agent ne doit pas répondre. Si une liste de
             valeurs est fournie alors seulement la première valeur sera utilisée.
             Autrement, l'agent restera muet. -->
            <Do_Not_Reply_When/> <!--  -->
         </User1>

         [...]

         <User9>
            <!-- Liste de paramètres. Si un paramètre de cette liste est
             positionné alors l'agent doit répondre. Si une liste de valeurs est
             fournie alors seulement la première valeur sera utilisée. Autrement,
             l'agent restera muet. -->
            <Only_Reply_When/> <!--  -->

            <!-- Liste de paramètres. Si un paramètre de cette liste est
             positionné alors l'agent ne doit pas répondre. Si une liste de
             valeurs est fournie alors seulement la première valeur sera utilisée.
             Autrement, l'agent restera muet. -->
            <Do_Not_Reply_When/> <!--  -->
         </User9>

         <KWIC>
            <!-- Toujours afficher le début du contenu ? (même si ils ne
             contiennent aucun des mots recherchés) -->
            <Always_Display_Beginning/> <!-- booléen : true -->

            <!-- Nombre maximal de mots à afficher. 0 pour désactiver. -->
            <Max_Nb_Displayed_Words/> <!-- entier : 30 -->

            <!-- Nombre maximal de charactères à afficher dans le title. 0 pour
             désactiver. -->
```

```
               <Max_Title_Length/> <!-- entier : 0 -->

               <!-- Listes de tags dans lesquels KWIV sera appliqué -->
               <Tags/> <!-- string_list -->

               <!-- Si vrai, ne pas utiliser KWIC pour les titres. -->
               <Disable_Kwic_For_Title/> <!-- booléen : false -->

               <!-- Si vrai, ne pas utiliser KWIC pour les descriptions. -->
               <Disable_Kwic_For_Descriptions/> <!-- booléen : false -->

               <!-- Activer le surlignage des mots ? -->
               <Highlight_Empty_Words/> <!-- booléen : true -->

            </KWIC>

         </Sort>
      </Service_Specific>
   </XML>
   <Shop_Directory>
      <!-- Hash size -->
      <Hash_Size/> <!-- entier : 11 -->
      <KWIC>
         <!-- Toujours afficher le début du contenu ? (même si ils ne contiennent
          aucun des mots recherchés) -->
         <Always_Display_Beginning/> <!-- booléen : true -->

         <!-- Nombre maximal de mots à afficher. 0 pour désactiver. -->
         <Max_Nb_Displayed_Words/> <!-- entier : 30 -->
      </KWIC>
   </Shop_Directory>
   <Web>
      <!-- Réunit les résultats d'un même serveur. -->
      <Cluster_Results/> <!-- booléen : false -->

      <!-- Si Cluster_Results, indique le nombre de réponses par site à fournir. -->
      <Cluster_Size/> <!-- entier : 1 -->

      <!-- Si non 0, indique la dernière page pour laquelle les résultats doivent
       être regroupés. -->
      <Last_Clustering_Page/> <!-- entier : 0 -->

      <!-- Si vrai, le regroupement ne sera pas appliqué si UNIQUE est positionné.
       -->
      <Disable_Clustering_When_Unique/> <!-- booléen : false -->

      <!-- Si vrai, expected_nb_replies sera découpé entre les divers agents (si
       possible). Une réponse supplémentaire par agent sera fournies afin d'activer
       les liens du pager. -->
      <Balance_Agent_Replies/> <!-- booléen : true -->

      <KWIC>
         <!-- Toujours afficher le début du contenu ? (même si ils ne contiennent
          aucun des mots recherchés) -->
         <Always_Display_Beginning/> <!-- booléen : true -->

         <!-- Nombre maximal de mots à afficher. 0 pour désactiver. -->
         <Max_Nb_Displayed_Words/> <!-- entier : 30 -->
      </KWIC>

      <!-- Est-ce que l'agent web doit préserver ou forcer le type de réponse du
       moteur lors de l'insertion d'une réponse d'un nouveau moteur. -->
      <Preserve_Engine_Reply_Source/> <!-- booléen : true -->
```

```
      <!-- Quand les réponses d'un agent engine et directory sont mélangées, est-ce
       que le titre doit venir de l'agent directory ? -->
      <Keep_Directory_Title/> <!-- booléen : false -->

    </Web>
    <Full_Text_Search>
      <!-- Type de normalisation pour les lexèmes : 'fr_thesaurus' (utilise
       thesaurus.db), 'fr_metaphone', 'fr_sms_metaphone'. Doit correspondre au type
       utilisé pendant l'indexation. -->
      <Normalization_Type/> <!-- string : fr_thesaurus -->

      <!-- Liste des champs dans lesquelles la recherche est restreinte. Doit
       correspondre aux noms utilisés pendant l'indexation. -->
      <Search_Field_Name/> <!--  -->

      <!-- Est-ce que l'agent doit rapporter une erreur et s'arrêter ('false') ou
       est-ce qu'il doit commencer et toujours fournir des réponses vides ('true')
       quand les fichiers d'index sont absents. -->
      <Allow_Empty_Databases/> <!-- booléen : false -->

      <!-- Tuning des expansions des wildcards. Peut être coûteux ! -->
      <Wildcard_Tuning>
        <!-- Nombre maximum de candidats par mot. -->
        <Max_Candidates/> <!-- entier : 1000 -->

        <!-- Nombre maximum d'expansions conservées pendant une recherche (les
         meilleures). -->
        <Max_Expansions/> <!-- entier : 10 -->
      </Wildcard_Tuning>
      <Large_Optimisations_Tuning>
        <!-- Pour une requête d'un seul mot, seuil au-dessus duquel seules les
         requêtes avec w=1 sont considérées. -->
        <Single_Threshold1/> <!-- entier : 20000 -->

        <!-- Pour une requête d'un seul mot, seuil au-dessus duquel seules les
         requêtes avec w=1 ou w=2 sont considérées. -->
        <Single_Threshold2/> <!-- entier : 10000 -->

        <!-- Pour une requête d'un seul mot, seuil au-dessus duquel seules les
         requêtes avec w=1 sont considérées. -->
        <Multiple_Threshold1/> <!-- entier : 1000000 -->

        <!-- Pour une requête, seuil au-dessus duquel seules les requêtes avec w=1,
         w=2 ou w=3 sont considérées. -->
        <Multiple_Threshold3/> <!-- entier : 100000 -->
      </Large_Optimisations_Tuning>
      <Optional_Optimisations_Tuning>
        <!-- Ne pas prendre en considération les réponses ayant un rang (score
         d'authorité) compris entre 1 et Max_Auth_Rank (inclus). -->
        <Max_Auth_Rank/> <!-- entier : 5000 -->

        <!-- Si une requête a plus de mots que M_W_F_P_C alors pathlen ne sera pas
         calculé. -->
        <Max_Words_For_Pathlen_Computation/> <!-- entier : 10 -->

        <!-- À chaque fois que deux mots d'une requête sont dans un ordre inverse,
         détermine une pénalité à appliquer à la longueur du chemin. pathlen+=
         penalité*nombre_d'inversions. -->
        <Penalty_For_Inverted_Words/> <!-- entier : 2 -->

        <!-- Détermine une limite de réponses candidates (ie avant filtrage)
         analysées par un agent -->
        <Max_Candidates_Per_Agent/> <!-- entier : 0 -->
      </Optional_Optimisations_Tuning>
```

```
    </Full_Text_Search>
    <Engine>
      <!-- Charge la date du SHM -->
      <Enable_Date_Shm/> <!-- booléen : false -->

      <!-- Load redirection shared memory segment. -->
      <Enable_Redirect_SHM/> <!-- booléen : false -->

      <!-- Remplace la taille par la date dans les réponses. Vous devriez activer
       cette option si vous avez activé l'option Enable_Date_Shm. -->
      <Store_Date_Instead_Of_Size/> <!-- booléen : false -->

      <!-- Hash size pour l'agent Search Engine -->
      <Hash_Size/> <!--  entier : 11 -->

      <!-- Hash size pour l'agent Freshbot -->
      <Freshbot_Hash_Size/> <!-- entier : 3 -->

      <!-- Stocker les réponses dans l'ensemble de réponse principal ? -->
      <Store_Replies_In_Main_Set/> <!-- booléen : true -->

      <!-- Déclencheur additionnel nécessaire pour autoriser les requêtes vides dans
       l'agent engine. TRES coûteux, utiliser en conjonction avec des ressources
       génériques. -->
      <Allow_Empty_Query/> <!-- booléen : false -->

      <Parametric_Filters>
        <!-- Nouveau filtres paramétriques SHM (shm.dat) -->
        <Enable_Shm/> <!-- booléen : true -->

        <!-- Activer le moteur des filtres paramétriques ? -->
        <Enabled/> <!-- booléen : false -->

        <Builtin_Parametrics>
          <!-- Filtres paramétriques « Document Type ». Si non vide, génère cette
           catégorie avec ce label. -->
          <Enable_Doctype/> <!-- string : '' -->

          <!-- Filtres paramétriques « Country ». Si non vide, génère cette
           catégorie avec ce label. -->
          <Enable_Country/> <!-- string : '' -->

          <!-- Filtres de catégorie, par id. Plusieurs filtres peuvent être
           specifiés. clé=prefix:slot, valeur=label. Slot est un nombre entre 1 et
           10. Requiert [prefix]_id_parametrics.db dans le SHM et
           [prefix]_categories.xml. -->
          <Enable_IdCat/> <!-- string_map -->

        </Builtin_Parametrics>
      </Parametric_Filters>
    </Engine>
    <Hints>
      <!-- Fréquence minimale pour les candidats formés d'un seul mot. -->
      <S/> <!-- entier : 50 -->

      <!-- Ratio Fréquence/Distance. Autorise le tuning des réponses. -->
      <K/> <!-- entier : 100 -->

      <!-- Coefficient déterminant la distance maximum par rapport à un candidat.
       -->
      <ALPHA/> <!-- double : 0.2 -->

      <!-- Nombre maximale de candidats « hint » à considérer. -->
      <Max_Candidates/> <!-- entier : 10 -->
```

```
      <!-- Hash size pour la base de données word_id -->
      <Hash_Size/> <!-- entier : 63 -->

      <!-- Taille totale du cache en Mo -->
      <Cache_Size_Mb/> <!-- entier : 64 -->

      <!-- Toujours répondre sans prendre en compte UNIQUE -->
      <Always_Reply/> <!-- booléen : true -->

      <!-- Si non vide, fournit le nom du paramètre à utiliser pour la sélection de
       la base de données du langage. -->
      <Language_Parameter/> <!-- string : '' -->

      <!-- Si positionné et si Language_Parameter ne l'est pas, alors on utilise
       cette valeur -->
      <Default_Language/> <!-- string : '' -->

   </Hints>
   <Espotting>
      <!-- Identification de l'affilié -->
      <AffiliateId/> <!-- entier -->

      <!-- Identification du flux -->
      <FeedName/> <!-- string : affiliate.fr.espotting.com/search/xml/results.asp --
       >

      <!-- FeedName et AffiliatedId alternatif pour une valeur donnée d'un paramètre
       choisi. La clé est du type 'paramètre=valeur', valeur est du type
       'affiliate_id@feed_name' or 'MUTE'. Plusieurs alternatives peuvent être
       spécifiées dans ce mapping. Fournir les informations complètes incluant
       /search/xml/results.asp. -->
      <Alternate_Feed/> <!-- string_map -->

      <!-- Nombre de réponses à fournir -->
      <Hits/> <!-- entier -->

      <!-- Dernière page de réponse -->
      <Last_Page/> <!-- entier -->

      <!-- Si les réponses sont affichées sur plusieurs pages et que les pages ne
       sont pas les mêmes, c'est le nombre de réponses à afficher sur chaque page.
       Autrement, laisser cette valeur à 0 et les réponses Hits seront affichées sur
       chaque page.  -->
      <Nb_Displayed_Replies/> <!-- entier : 0 -->

      <!-- Fournir le même contenu sur chaque page (par défaut) ou naviguer à traver
       la liste des réponses complètes ? -->
      <Same_Pages/> <!-- booléen : true -->

      <!-- Temps d'attente de réponse -->
      <Timeout_Seconds/> <!-- entier : 3 -->

      <!-- Filtrer le contenu adulte -->
      <Adult_Filter/> <!-- booléen : false -->

      <!-- Toujours répondre indépendamment de UNIQUE ? -->
      <Always_Reply/> <!-- booléen : false -->

      <!-- Liste des paramètres. Si un paramètre de cette liste est postionné,
       l'agent ne doit pas répondre. -->
      <Mute_Parameters/> <!-- string_list -->

      <Log_SQL>
```

```
          <!-- Hôte SGBD -->
          <Host/> <!-- string : localhost -->

          <!-- Table de log -->
          <Table/> <!-- string : ESPOTTING.LOG -->

          <!-- Login SGBD -->
          <Login/> <!-- string : antisearch -->

          <!-- Password SGBD -->
          <Password/> <!-- string  -->
       <Log_SQL>
    </Espotting>
    <Overture>
       <!-- Identification du flux -->
       <FeedName/> <!-- string : xml.fr.overture.com/d/search/p/antidot/xml/fr/ -->

       <!-- PartnerName statique. Si vide, les partners Dynamiques sont utilisés. -->
       <PartnerName/> <!-- string : '' -->

       <!-- Alternate ParterName pour une valeur donnée d'un paramètre choisi. Clé
        est de type 'paramètre:valeur', valeur est le nom alternatif. Plusieurs
        alternatifs peuvent être spécifiés. -->
       <Alternate_ParterName/> <!-- string_map -->

       <!-- [key]=[trigger]:[use_param]
       [trigger]=([trigger_param] '=' [trigger_value])
       if trigger_value=trigger_value then use as partner_name value, i which '#' is
        replaced by use_param value -->
       <Dynamic_ParterName/> <!-- string_map -->

       <!-- [key] = :[use_param] use as partner_name value in which'#' is replaced by
        use_param value. -->
       <Default_Dynamic_ParterName/> <!-- string_map -->

       <!-- Nombre de réponses à fournir -->
       <Hits/> <!-- entier -->

       <!-- Dérnière page de réponse -->
       <Last_Page/> <!-- entier -->

       <!-- Founir le même contenu dans chaque page ? ou naviguer à traver la liste
        des réponses complètes. -->
       <Same_Pages/> <!-- booléen : true -->

       <!-- Si les réponses sont affichées sur plusieurs pages et que les pages ne
        sont pas les mêmes, spécifie le nombre de réponses à afficher pour chaque
        page. Autrement, laisser ce champ vide et les réponses des Hits seront
        affichées sur chaque page. -->
       <Nb_Displayed_Replies/> <!-- entier : 0 -->

       <!-- Si le paramètre dans une clé est fourni alors affiche le nombre de
        réponses spécifiées par paramètre ; pas Nb_Displayed_Replies. -->
       <Alternate_Nb_Displayed_Replies/> <!-- string_map -->

       <!-- Temps maximum à attendre pour une réponse (ajouter Timeout_Milliseconds)
        -->
       <Timeout_Seconds/> <!-- entier : 3 -->

       <!-- Temps maximum à attendre pour une réponse (ajouter Timeout_Seconds) -->
       <Timeout_Milliseconds/> <!-- entier : 0 -->

       <!-- Filtrer le contenu adulte -->
       <Adulter_Filter/> <!-- booléen : false -->
```

```
      <!-- Toujours répondre indépendament de UNIQUE ? -->
      <Always_Reply/> <!-- booléen : false -->

      <!-- Désactiver HTML B tags pour surligner les mots-clés de la recherche. -->
      <Disable_HTML_B_Tags/> <!-- booléen : false -->

      <!-- Si positionné, change les tags HTML B en tag 'valeur' -->
      <HTML_B_Tags_New_Name/> <!-- string : '' -->

      <!-- Si positionné, ne produire aucun log (ni texte ni SQL) -->
      <Disable_Logging/> <!-- booléen : true -->

      <Log_SQL>
         <!-- Hôte SGBD -->
         <Host/> <!-- string : localhost -->

         <!-- Table de log -->
         <Table/> <!-- string : OVERTURE.LOG -->

         <!-- Login SGBD -->
         <Login/> <!-- string : antisearch -->

         <!-- Password SGBD -->
         <Password/> <!-- string  -->
      <Log_SQL>

      <!-- Liste de paramètres. Si un paramètre de cette liste est positionné à une
       certaine valeur, l'agent doit répondre. Autrement, il ne répond rien (si la
       liste n'est pas vide). -->
      <Only_Reply_When/> <!-- string_map -->
   </Overture>
   <PFP>
      <!-- PFP provider : 'Overture', 'Mirago', 'AFS'. -->
      <Provider/> <!-- string -->

      <!-- Nom d'affilié par défaut -->
      <Default_Affiliate/> <!-- string : '' -->


      <!-- Affilié alternatif pour une valeur donnée d'un paramètre choisi. Clé est
       du type 'paramètre=valeur', valeur est le nom alternatif. -->
      <Alternate_Affiliate/> <!-- string_map -->

      <!-- Feed url. Si vide, user smart value for provider. -->
      <Default_Feed/> <!-- string : '' -->

      <!-- Feed alternatif pour une valeur donnée d'un paramètre choisi. La clé est
       de type 'paramètre=valeur'. Plusieurs alternatives peuvent être spécifiées.
       -->
      <Alternate_Feed/> <!-- string_map -->


      <!-- Si vrai, répondre même si UNIQUE est positionné. -->
      <Always_Reply/> <!-- booléen : false -->

      <!-- Liste de paramètres. Si un paramètre de cette liste est positionnée à une
       valeur donnée, l'agent doit répondre. Autrement, ne répond pas (si la liste
       est non vide). -->
      <Only_Reply_When/> <!-- string_map -->

      <!-- Liste de paramètres. Si un paramètre de cette liste est positionnée à une
       valeur donnée, l'agent ne doit pas répondre. Autrement, répond (si la liste
       est non vide). -->
```

```
        <Do_Not_Reply_When/> <!-- string_map -->

        <!-- Nombre de réponses à afficher -->
        <Nb_Displayed_Replies/> <!-- entier -->

        <!-- Si le paramètre dans la clé est fourni, affiche le nombre de réponses
         spécifié par la valeur du paramètre ; pas Nb_Displayed_Replies. -->
        <Alternate_Nb_Displayed_Replies/> <!-- string_map -->

        <!-- Fournir les mêmes contenus sur chaque page ? ou naviguer dans la liste
         complète des réponses. -->
        <Same_Pages/> <!-- booléen : true -->

        <!-- Dernière page sur laquelle les réponses seront affichées (0 pour ne pas
         limiter) -->
        <Last_Page/> <!-- entier : 1 -->

        <!-- Activer le contrôle de contenu adulte ? -->
        <Adult_Filter/> <!-- booléen : false -->

        <!-- Inclure des images dans le résultat ? -->
        <Include_Images/> <!-- booléen : false -->

        <!-- Désactiver les tags HTML B afin de surligner les mots-clés de la
         recherche ? -->
        <Disable_HTML_B_Tags/> <!-- booléen : false -->

        <!-- Si positionné, changer les tags HTML B en tags 'valeur'. -->
        <HTML_B_Tags_New_Name/> <!-- string : '' -->

        <!-- Temps maximum à attendre pour une réponse (ajouter Timeout_Milliseconds)
         -->
        <Timeout_Seconds/> <!-- entier : 3 -->

        <!-- Temps maximum à attendre pour une réponse (ajouter Timeout_Seconds) -->
        <Timeout_Milliseconds/> <!-- entier : 0 -->

    </PFP>
    <Promotion>
        <!-- Taille du cache en Mo -->
        <Cache_Size_Mb/> <!-- entier : 32 -->

        <!-- Liste de paramètres. Si un paramètre de cette liste est positionné,
         l'agent ne doit pas répondre. -->
        <Mute_Parameters/> <!-- string_list -->

        <!-- Liste de paramètres. Si un paramètre de cette liste est positionnée à une
         valeur donnée, l'agent doit répondre. Autrement, ne répond pas (si la liste
         est non vide). -->
        <Only_Reply_When/> <!-- string_map -->

        <!-- Liste de paramètres. Si un paramètre de cette liste est positionnée à une
         valeur donnée, l'agent ne doit pas répondre. Autrement, répond (si la liste
         est non vide). -->
        <Do_Not_Reply_When/> <!-- string_map -->

        <!-- Nom du paramètre numérique utilisé pour la sélection de la région. Les
         choix habituels sont X et RE. -->
        <Region_Parameter/> <!-- string : 'X' -->

        <!-- Toujours fournir des réponses même si N>1 et si unique est positionné ?
         -->
        <Always_Reply/> <!-- booléen : false -->
```

```
        <!-- Nombre de réponses par page ou 0 pour ne pas limiter. -->
        <Hits/> <!-- entier : 0 -->

        <!-- Doit-on mélanger les réponses ? -->
        <Shuffle_Replies/> <!-- booléen : false -->

        <!-- Utiliser ce paramètre pour spécifier le nom d'un paramètre CGI dont la
         valeur doit être ajoutée à la requête. -->
        <Extra_Parameter/> <!-- string_list -->

        <Log_SQL>
           <!-- Hôte SGBD -->
           <Host/> <!-- string : localhost -->

           <!-- Table de log -->
           <Table/> <!-- string : CAMPAIGN.CAMPAIGN -->

           <!-- Login SGBD -->
           <Login/> <!-- string : antisearch -->

           <!-- Password SGBD -->
           <Password/> <!-- string  -->
        <Log_SQL>

        <!-- Duplique les réponses d'un site à l'autre. Clé=site source, value=site
         destination. -->
        <Replicate/> <!-- string_map -->
     </Promotion>
     <Commercial>
        <!-- Taille du cache en Mo -->
        <Cache_Size_Mb/> <!-- entier : 32 -->

        <!-- Liste de paramètres. Si un paramètre de cette liste est positionné,
         l'agent ne doit pas répondre. -->
        <Mute_Parameters/> <!-- string_list -->

        <!-- Liste de paramètres. Si un paramètre de cette liste est positionnée à une
         valeur donnée, l'agent doit répondre. Autrement, ne répond pas (si la liste
         est non vide). -->
        <Only_Reply_When/> <!-- string_map -->

        <!-- Liste de paramètres. Si un paramètre de cette liste est positionnée à une
         valeur donnée, l'agent ne doit pas répondre. Autrement, répond (si la liste
         est non vide). -->
        <Do_Not_Reply_When/> <!-- string_map -->

        <!-- Nom du paramètre numérique utilisé pour la sélection de la région. Les
         choix habituels sont X et RE. -->
        <Region_Parameter/> <!-- string : 'X' -->

        <!-- Doit-on mélanger les réponses ? -->
        <Shuffle_Replies/> <!-- booléen : false -->

        <!-- Utiliser ce paramètre pour spécifier le nom d'un paramètre CGI dont la
         valeur doit être ajoutée à la requête. -->
        <Extra_Parameter/> <!-- string_list -->

        <Log_SQL>
           <!-- Hôte SGBD -->
           <Host/> <!-- string : localhost -->

           <!-- Table de log -->
           <Table/> <!-- string : CAMPAIGN.CAMPAIGN -->
```

```
          <!-- Login SGBD -->
          <Login/> <!-- string : antisearch -->

          <!-- Password SGBD -->
          <Password/> <!-- string  -->
       <Log_SQL>

       <!-- Stocker les réponses dans les responses principales (avec les réponses de
        engine, directory,...) [defaut, Search Engine 'Cheat' Mode] ou isolément,
        dans un ensemble séparé [Ad mode qui devrait être le mode par défaut si ce
        n'est à cause de la compatibilité avec les anciens services]. -->
       <Store_Reply_In_Main_Set/> <!-- booléen : true -->

       <!-- Toujours fournir des réponses même si N>1 et si unique est positionné ?
        -->
       <Always_Reply/> <!-- booléen : false -->

       <!-- Nombre de réponses par page ou 0 pour ne pas limiter. -->
       <Hits/> <!-- entier : 0 -->

       <!-- Fournir les mêmes contenus sur chaque page ? ou naviguer dans la liste
        complète des réponses. -->
       <Same_Pages/> <!-- booléen : true -->

       <!-- Duplique les réponses d'un site à l'autre. Clé=site source, value=site
        destination. -->
       <Replicate/> <!-- string_map -->
    </Commercial>
    <See_Also>
       <!-- Hash size utilisées par les serveurs -->
       <Hash_Size/> <!-- entier : 11 -->

       <!-- Nombre de réponses à afficher -->
       <Nb_Displayed_Replies/> <!-- entier : 3-->

       <!-- Si vrai, fournit seulement une réponse par host si possible. -->
       <Spread_Replies/> <!-- booléen : true -->
    </See_Also>
    <DNS>
       <!-- si non nul, loggue les requêtes sur cette base de données (autorises le
        remplissage des données Redirect_Service_id par les requêtes Service_id). -->
       <Redirect_service_id/> <!-- entier : 0 -->

       <Log_SQL>
          <!-- Hôte SGBD -->
          <Host/> <!-- string : localhost -->

          <!-- Table de réponses -->
          <Table/> <!-- string : DNS.REPLY -->

          <!-- Table d'administration -->
          <Table/> <!-- string : DNS.ADMIN -->

          <!-- Table de requêtes -->
          <Table/> <!-- string : DNS.QUERY -->

          <!-- Login SGBD -->
          <Login/> <!-- string : antisearch -->

          <!-- Password SGBD -->
          <Password/> <!-- string  -->
       <Log_SQL>

    </DNS>
```

```
    <Mail>
      <!-- Nombre de réponses à afficher -->
      <Nb_Displayed_Replies/> <!-- entier : 10-->

      <Tuning>
        <!-- Seuil minimum pour les candidats (entre 0 et 100) -->
        <Threshold/> <!-- entier : 75 -->


        <!-- Si vrai, les réponses exactes ne seront pas affichées. -->
        <Discard_Exact_Matches/> <!-- booléen : false -->

        <!-- Si vrai et si il y a une réponse exacte alors ne pas afficher d'autres
         correspondances. -->
        <Discard_Other_Matches_If_Exact/> <!-- booléen : false -->
      </Tuning>
      <Log_SQL>
        <!-- Hôte SGBD -->
        <Host/> <!-- string : localhost -->

        <!-- Table de réponses -->
        <Table/> <!-- string : EMAIL.ADDRESS -->

        <!-- Login SGBD -->
        <Login/> <!-- string : antisearch -->

        <!-- Password SGBD -->
        <Password/> <!-- string  -->
      <Log_SQL>
    </Mail>
    <Related_Expressions>
      <!-- Activer le moteur des sujets associés ? -->
      <Enabled/> <!-- booléen : false -->

      <N_grams>
        <!-- Longueur minimale d'un N-grams -->
        <Min_Length/> <!-- entier : 1 -->

        <!-- Longueur maximale d'un N-grams -->
        <Max_Length/> <!-- entier : 4 -->
      </N_grams>
      <!-- Si vrai, les expressions régulières sont recherchés seulement à
       l'intérieur des différents sites. Autrement, la recherche est effectuée à
       l'intérieur d'un ensemble de pages sans restriction de site. -->
      <Site_Scope/> <!-- booléen : true -->

      <!-- Si vrai, ignore les documents PDF pendant la création des expressions. --
       >
      <Ignore_PDF/> <!-- booléen : false -->

      <!-- (Temp??) Très coûteux ! Si vrai, utilise les données supplémentaires
       (store items,...) pour la génération des RTE. -->
      <Analyse_Extra_Data/> <!-- booléen : false -->

      <!-- Nombre maximum de réponses à donner -->
      <Nb_Replies/> <!-- entier : 8 -->
    </Related_Expressions>
  </Agents>
```

La section DTD

```
  <DTD>
```

```
          <!-- si vrai, génère le doctype avec DTD URI. -->
          <Generate_Doctype/> <!-- booléen : true -->

          <!-- URI DTD -->
          <URI/> <!-- string :'http://dtd.antisearch.net/SearchResult.dtd' -->
  </DTD>
```

La section Stylesheets

```
  <StyleSheets>
     <!-- URI feuille de style -->
     <XSL_URI/> <!-- string  -->

     <!-- Chemin d'accès à l'entête -->
     <Header_Path/> <!-- string : '' -->

     <!-- Chemin d'accès au bas de page -->
     <Footer_Path/> <!-- string : '' -->

     <Redirect>
        <!-- URI de feuille de style de redirection (redéfinissable par service) -->
        <XSL_URI/> <!-- string : 'http://xsl1.antisearch.net/AntiSearch/redirect.xsl'
         -->
        <!-- Racine du service, utilisée pour rediriger en cas de problème. -->
        <Service_Root/> <!-- string : 'http://www.antidot.net' -->
     </Redirect>
  </StyleSheets>
```

La section User_Parameters

```
  <User_Parameters>
     <Global_Parameters>
        <!-- Paramètre défini par l'utilisateur ; clé=nom_paramètre et
         valeur=valeur_paramètre. -->
        <Parameter/> <!-- string_map -->
     </Global_Parameters>
     <Services_Parameters>
        <!-- Paramètre défini par l'utilisateur ; clé=nom_paramètre et
         valeur=valeur_paramètre. -->
        <Parameter/> <!-- string_map -->
     </Services_Parameters>
     <Sites_Parameters>
        <!-- Paramètre défini par l'utilisateur ; clé=nom_paramètre et
         valeur=valeur_paramètre. -->
        <Parameter/> <!-- string_map -->
     </Sites_Parameters>
  </User_Parameters>
```

1.1.1.1. Les section spécifiques au back-office AFS

La section Back_Office

```
  <BackOffice>
       <SQL>
          <!-- Hôte SGBD -->
          <Host/> <!-- string : 'localhost' -->

          <!-- Login SGBD -->
          <Login/> <!-- string : 'antisearch' -->

          <!-- Password SGBD -->
          <Password/> <!-- string  -->
```

```
        <!-- Driver SGBD -->
        <Driver/> <!-- string : 'mysql' -->

        <!-- Base de données du BO -->
        <Database/> <!-- string : 'WEBUSERS' -->
      <SQL>
</BackOffice>
```

La section Tracking

```
<Tracking>
   <!-- Quantité de mémoire que peut utiliser l'application. -->
   <Max_Memory_Usage_Mb/> <!-- entier : 64 -->

   <!-- Hôte SGBD -->
   <Host/> <!-- string : 'localhost' -->

   <!-- Login SGBD -->
   <Login/> <!-- string : 'antisearch' -->

   <!-- Password SGBD -->
   <Password/> <!-- string -->

   <!-- Driver SGBD -->
   <Driver/> <!-- string : 'mysql' -->

   <!-- Base de données de modération -->
   <Moderation_Database/> <!-- string : 'MODERATION' -->
</Tracking>
```

3.1.2 Fichier Services.conf

Le fichier `Services.conf` est au format texte et permet de :

- déclarer des services de réponse
- structurer en arborescence les agents d'un service de réponse (comme expliqué dans l'architecture logique de réponse au début du Manuel Administrateur)
- répartir les agents en réseau
- dimensionner les arborescences
- redonder les arborescences pour assurer la résistance aux pannes

Déclarer un service de réponse

Les commentaires commencent par le symbole '`!`' et se termine par un retour chariot.

La déclaration du service est effectuée en précisant le numéro de service correspondant à celui déclaré dans la configuration (cf. § 3.1.1.1 section Aliases). La déclaration est de la forme :

`Service : <service_id>`

Ainsi par exemple :

```
!------------------
! 103 (Intranet)
!------------------
Service : 103
```

Structurer les agents en arborescence

Pour rappel une arborescence possède toujours un agent racine (Root). A cette racine, d'autres agents spécialisés peuvent être rattachés. L'arborescence décrite est de la forme :

<service_id>.<agent_name>.children: <child1_name>,<child2_name>...

Chaque agent composant l'arborescence et possédant des agents fils est précisé sur une nouvelle ligne. Par exemple :

```
! Agent root possede 3 enfants
103.root.children: web,hint,promotion

! Agent web, fils de root, possede 3 enfants
103.web.children:  engine1,engine2,user1
```

Chaque nom logique d'agent est associé à un nom de binaire correspondant de la forme : <agent_name>_cell. Concernant les noms logiques user1 à user9, qui sont des agents spécifiques au service de recherche, le nom du binaire associé est précisé dans la configuration (cf. 3.1.1.3 section Agents/XML).

Il est possible que pour certains services, des agents génériques aient été dérivés en agents spécifiques. Il est nécessaire pour ceux-ci de préciser le nom du binaire associé en ajoutant une ligne utilisant la forme :

<service_id>.<agent_name>.Exe: <bin_name>

```
! Agent web specifique
103.web.Exe:  monintranet_web_cell
```

Répartir les agents en réseau

On doit préciser sur quelle machine hôte, les agents et leur base de réponse associée, tournent pour assurer le service. On ajoute donc une nouvelle ligne de la forme :

<service_id>.<agent_name>: <nb>@<hostname>

Afin de répartir la charge de l'hôte du service de recherche, il est possible de distribuer les agents composant l'arborescence sur différentes machines. Cette distribution tiendra compte des capacités techniques de la machine hôte, et de la taille des bases de réponses sur lesquelles travaillent chaque agent.

En supposant par exemple, que la base de réponse de l'agent Web est deux fois plus importante que l'ensemble des autres, il est judicieux de séparer l'agent Web et ses bases sur une machine différente :

```
! Agents root et web tourne sur rep1
103.root:                       1@rep1
103.web:            1@rep1

! Agents hint, promotion, user1 tournent sur rep2
103.promotion:      1@rep2
103.user1:                      1@rep2

! Agent engine1 tourne sur rep3
103.engine1:        1@rep3

! Agent engine2 tourne sur rep4
103.engine2:        1@rep4
```

Dimensionner les arborescences

Afin de pouvoir répondre à plusieurs requêtes en même temps, plusieurs instances d'arborescence doivent être lancées. Ce nombre d'instances dépend donc du temps de réponse moyen et du trafic mensuel moyen estimé ou constaté pour votre service de recherche.

Pour un serveur dédié, on peut se référer à l'heuristique suivante :

<trafic_quotidien_moyen> ≈ <trafic_mensuel_moyen> / 30

<debit_moyen> ≈ <trafic_quotidien_moyen> / (12 * 60)

<nb> arborescences = borne_sup(<debit_moyen> / (60 / <temps_reponse_moyen>))

Ainsi par exemple, dans la déclaration des lignes précédentes, si le temps de réponse moyen est 0,3s et le trafic mensuel moyen estimé est 1.300.000 requêtes, le chiffre <nb> peut être mis à jour à 1.

Il est important de s'assurer que ce chiffre est le même pour toute l'arborescence, le cas échéant pouvant conduire à une inconsistance du service de recherche et donc une dégradation notable des performances.

Pour chaque ligne, l'agent concerné est lancé dans ce nombre d'exemplaires, augmentant la charge de la machine hôte. Le nombre d'arborescences a un rôle conséquent sur les performances du serveur ; il est donc nécessaire de le choisir avec précaution (procéder par ajustements incrémentaux en surveillant la charge des machines).

Redonder les arborescences

Pour pouvoir résister à une panne ou une opération de maintenance (arrêt d'un serveur hébergeant une partie du service de recherche), il est préférable de redonder les arborescences.

Chaque agent est alors lancé dans son nombre d'exemplaires sur chacune des machines déclarées. L'agent, sur l'une ou l'autre des machines, est utilisé si la charge du serveur n'est pas trop importante. Si l'un des serveurs n'est plus accessible, l'autre serveur assure le fonctionnement du service de recherche.

Pour cela, pour chacune des lignes de déclaration sur le réseau, il suffit d'ajouter une ligne supplémentaire où l'on ne change que le nom de l'hôte. Par exemple :

```
! Agent root tourne sur rep1 et rep2
103.root:                      6@rep1
103.root:                      6@rep2

! Agent web tourne sur rep1 et rep4
103.web:             6@rep1
103.web:             6@rep4
...
```

4 Intégration des fonctionnalités AFS

4.1.1 Application de méthodes linguistiques lors de l'indexation

Lors de l'indexation, AFS peut appliquer des dictionnaires de synonymes, des thésaurus ou encore des méthodes de normalisation telles que la lemmatisation (gestion des flexions) ou la phonétisation permettant ainsi d'élargir les requêtes.

Il est possible d'appliquer plusieurs méthodes de traitement linguistique lors de l'indexation. Cependant on ne peut appliquer qu'un seul type de normalisation (soit la gestion des flexions soit la phonetisation).

4.1.1.1 Indexation avec gestion des flexions

L'intégration d'un dictionnaire de flexions permet d'effectuer une indexation et une recherche plus générique en se basant sur la racine des mots.

Cela permet de corriger les effets des erreurs d'orthographe et de généraliser la recherche en ignorant les marques de genre ou de nombre. Pour que ce principe fonctionne le dictionnaire doit être appliqué au moment de l'indexation mais également être disponible pour les agents chargés de répondre.

Le dictionnaire contient une collection de racines classées par ordre alphabétique. Chaque ligne contient une racine et les mots qui dérivent de cette racine.

Pour ajouter un mot à une entrée déjà existante du dictionnaire, il suffit de trouver sa racine, puis de rajouter le mot sur la même ligne que la racine, après le caractère '`:`'.

Pour ajouter de nouveaux mots équivalents dont la racine est absente du dictionnaire il suffit de rajouter une ligne contenant la racine suivie de deux points, suivis des mots correspondants à cette racine. Pour faciliter la maintenance du dictionnaire, la nouvelle racine doit être insérée à l'emplacement qui lui convient (tri par ordre alphabétique).

Exemple de dictionnaire :

```
abacul : abacule abacules
abalon : abalone abalones
aband : abandon abandons
abandon : abandonne abandonnee abandonnees abandonnes
abatt : abattant abattants
abattag : abattage abattages
abattem : abattement abattements
abattoir : abattoir abattoirs
```

- **Configuration de l'indexation**

La méthode de normalisation est précisée au sein de la balise `<Normalization_Chain>` de la section `<Indexation>` du fichier de configuration `afs.xml` (cf § 1.1.4). Plusieurs méthodes de normalisation peuvent être définis en déclarant chaque mode dans la balise `<Normalizer>`. Pour la gestion des flexions, cette balise admet deux valeurs qui sont :

– stem_add : la forme normalisée est ajoutée à la forme de base au moment de l'indexation

– stem_replace : la forme normalisée substitue la forme fléchie

```
<Indexation>
  <Normalization_Chain>
    <Normalizer>stem_add</Normalizer>
  </Normalization_Chain>
  ...
</Indexation>
```

4.1.1.2 Indexation avec thésaurus

L'application de thésaurus au moment de l'indexation permet de prendre en compte des spécificités métier ou d'élargir les requêtes avec des termes associés et permet ainsi lors de la recherche une extension sémantique (autopostage).

- **Les différentes sortes d'autopostage :**

  L'autopostage ascendant sélectionnera les termes génériques. On utilisera le symbole `^` pour remonter d'un niveau dans la hiérarchie.

`Descripteur = ^^lapin` *générera une recherche utilisant les descripteurs lapin, rongeur, et mammifère.*

- L'autopostage descendant sélectionnera les termes spécifiques. On utilisera le symbole * pour descendre d'un niveau dans la hiérarchie.

`Descripteur = *média` *générera une recherche utilisant les descripteurs média, affichage, presse, édition, publicité, radio, télévision, et vidéo.*

- L'autopostage sur les termes associés utilise le symbole § pour parcourir un niveau d'association.

`Descripteur = §Afrique` *générera une recherche utilisant les descripteurs Afrique, culture africaine, immigré africain, OUA, et peuple d'Afrique.*

AFS fait de l'autopostage ascendant et par termes associés à l'indexation et de l'autopostage descendant et par termes associés au moment de la recherche.

C'est à dire qu'un document D_break qui parle d'un break sera indexé comme un document parlant :

- d'un break
- dans une moindre mesure, d'une voiture et donc par synonymie d'une automobile

De même:

- un document D_coupé parlant d'un coupé sera indexé comme parlant de voiture et d'automobile
- un document D_cabriolet parlant d'un cabriolet sera indexé comme parlant de voiture et d'automobile

Supposons qu'on dispose également de documents D_véhicule et D_automobile parlant respectivement de "véhicule" et d'"automobile"

Ainsi on obtient les résultats suivants lors des recherches:

Requête	***Documents Retournés***	***Documents non retournés***
véhicule (ou automobile)	D_break, D_coupé, D_cabriolet, D_véhicule, D_automobile	
break	D_break	D_coupé, D_break, D_véhicule, D_automobile
coupé	D_coupé	D_break, D_cabriolet, D_véhicule, D_automobile
cabriolet	D_cabriolet	D_break, D_coupé, D_véhicule, D_automobile

Ainsi cette méthode permet d'élargir le champ sémantique de la recherche (trouver des breaks quand on cherche des véhicules) sans induire de bruit (un coupé n'est pas un break).

Afin de pouvoir exploiter ces relations sémantiques, le thésaurus édité au format XML utilise le vocabulaire RDF Skos.

Elements Skos Utilisés :

Classe : Concept	
Définition	Un concept, idée ou notion abstraite.
Exemple	<skos:Concept rdf:about="http://thes.antidot.org#Automobile">
Commentaire	L'identifiant doit commencer par une majuscule
Propriété : prefLabel	
Définition	forme lexicale préférentielle utilisée pour désigner une ressource dans une langue donnée
Exemple	<skos:prefLabel xml:lang="fr">automobile</skos:prefLabel>
Commentaire	Le terme préférentiel identifie le concept de façon unique dans le cadre d'un schéma.
Propriété : altLabel	
Définition	forme lexicale alternative
Exemple	<skos:altLabel xml:lang="fr">auto</skos:altLabel>
Commentaire	Les acronymes, abréviations, variantes orthographiques, formes irrégulières du pluriel ou du singulier, peuvent être incluses dans les formes lexicales alternatives.
Propriété : broader	
Définition	concept générique
Exemple	<skos:broader rdf:resource="http://thes.antidot.org#Vehicule"/>
Commentaire	Les concepts génériques sont typiquement représentés comme des parents dans une hiérarchie de concepts (arbre).
Propriété : narrower	
Définition	concept spécifique
Exemple	<skos:narrower rdf:resource="http://thes.antidot..org#Break"/>
Commentaire	Les concepts spécifiques sont typiquement représentés comme des enfants dans une hiérarchie de concepts (arbre)

Structure du fichier RDF

Exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:rdfs="http://www.w3.org/2000/01/rdf-
schema#" xmlns:owl="http://www.w3.org/2002/07/owl#">
   <skos:Concept rdf:about="http://thes.antidot.org#Vehicule">
          <skos:narrower rdf:resource="http://thes.antidot.org#Voiture"/>
          <skos:narrower rdf:resource="http://thes.antidot.org#Moto/>
          <skos:narrower rdf:resource="http://thes.antidot.org#Autobus"/>
          <skos:prefLabel xml:lang="fr">vehicule</skos:prefLabel>
  </skos:Concept>
  <skos:Concept rdf:about="http://thes.antidot.org#Voiture">
          <skos:broader rdf:resource="http://thes.antidot.org#Vehicule"/>
                <skos:narrower rdf:resource="http://thes.antidot.org#Break"/>
                <skos:narrower rdf:resource="http://thes.antidot.org#Coupe"/>
                <skos:narrower rdf:resource="http://thes.antidot.org#Berline"/>
          <skos:prefLabel xml:lang="fr">voiture</skos:prefLabel>
          <skos:altLabel xml:lang="fr">automobile</skos:altLabel>
          <skos:altLabel xml:lang="fr">auto</skos:altLabel>
          <skos:altLabel xml:lang="fr">bagnole</skos:altLabel>
  </skos:Concept>
 <skos:Concept rdf:about="http://thes.antidot.org#Break">
               <skos:broader rdf:resource="http://thes.antidot.org#Voiture"/>
          <skos:prefLabel xml:lang="fr">break</skos:prefLabel>
  </skos:Concept>
   <skos:Concept rdf:about="http://thes.antidot.org#Coupe">
          <skos:broader rdf:resource="http://thes.antidot.org#Voiture"/>
          <skos:prefLabel xml:lang="fr">coupé</skos:prefLabel>
  </skos:Concept>
```

Utilisation dans les éditeurs Skos

Le format utilisé permet d'importer et d'éditer les fichiers dans des éditeurs dédiés comme ThManager, qui peut être utilisé pour la mise à jour du vocabulaire.

Configuration de l'indexation

De la même façon que pour l'indexation avec normalisation par métaphone ou gestion des flexions, le thésaurus doit être déclaré dans la section `<Normalization_Chain>`.

```
<Normalization_Chain>
  <Normalizer>skos-thesaurus</Normalizer>
</Normalization_Chain>
```

NB : L'ordre de déclaration des modes d'indexation est très important car chaque méthode est appliquée selon son ordre d'apparition. Ainsi si une indexation avec thésaurus est associée à une indexation avec normalisation, il faut d'abord déclarer le thésaurus puis le mode de normalisation.

Exemple :

```
<Normalization_Chain>
  <Normalizer>skos-thesaurus</Normalizer>
  <Normalizer>stem-add</Normalizer>
</Normalization_Chain>
```

Il est impératif que :

- le thesaurus skos soit nommé skos-thesaurus.rdf
- le fichier compilé de flexions (anciennement thesaurus.db) soit renommé stem.db

Il est possible d'utiliser plusieurs thésaurus en déclarant chaque nom de thésaurus (avec son extension) à la suite de skos-thesaurus@

Exemple :

```
<Normalization_Chain>
  <Normalizer>skos-thesaurus@thesaurus.rdf</Normalizer>
  <Normalizer>skos-thesaurus@thes_couleurs.rdf</Normalizer>
  <Normalizer>skos-thesaurus@thes_sport.rdf</Normalizer>
  <Normalizer>skos-thesaurus@thes_materiel.rdf</Normalizer>
  <Normalizer>stem</Normalizer>
</Normalization_Chain>
```

4.1.1.3 Indexation avec métaphone

L'indexation avec métaphone permet d'effectuer une indexation et une recherche basées sur la forme sonore des mots en utilisant un algorithme de phonétisation et permet ainsi une recherche plus large sur les graphies alternatives d'une forme (exemple : saison = saizon, seson ...)

- **Configuration de l'indexation**

De la même façon que pour l'indexation avec dictionnaire, la balise `<Normalization_Chain>` doit être renseignée au moment de l'indexation.

```
<Indexation>
  <Normalization_Chain>
    <Normalizer>fr_metaphone</Normalizer>
  <Normalization_Chain>
  ...
</Indexation>
```

4.1.2 Generation d'une base d'expressions RTE :

Les expressions RTE (Related Topic Expressions) permettent la suggestion d'expressions liées au mot-clé recherché.

4.1.2.1 Configuration de l'indexation

Le fichier de configuration doit être renseigné avant l'indexation.

```
<Classification>
   <Enabled>true</Enabled>
   <N_grams>
      <Enabled>true</Enabled>
      <Min_Length>1</Min_Length>
      <Max_Length>6</Max_Length>
   </N_grams>
</Classification>
```

- `Min_Length`: Désigne la longueur minimum d'une expression
- `Max_Length`: Désigne la longueur maximum d'une expression + 2.

Ainsi, dans la configuration précédente, les expressions recherchées seront composées de 1 à 4 mots.

NB : Le calcul des RTE ne nécessitant pas d'être lancé lors de chaque nouvelle indexation, il est recommandé de désactiver cette option une fois le fichier généré afin de réduire le temps d'indexation (sinon les N_grams seront recalculés lors de chaque indexation ce qui est très coûteux).

4.1.2.2 Configuration du service de réponse

Le fichier de configuration(afs.xml) du service de réponse doit également être renseigné de la présence de RTE avant d'être compilé.

```
<Related_Expressions>
   <Service name="Exemple">
      <Nb_Replies>6</Nb_Replies>
      <Enabled>true</Enabled>
      <N_grams>
         <Min_Length>1</Min_Length>
         <Max_Length>4</Max_Length>
      </N_grams>
   </Service>
</Related_Expressions>
```

- `Nb_Replies`: Nombre de réponses retournées par l'agent RTE
- `Min_Length`: Désigne la longueur minimum d'une expression
- `Max_Length`: Désigne la longueur maximum d'une expression.

Ainsi, dans la configuration précédente, les expressions recherchées seront composées de 1 à 4 mots. Notez la différence d'interprétation de la variable `Max_Length` entre l'indexation et le service de réponse.

4.1.3 Implantation d'agents spécifiques

4.1.3.1 Agent de suggestion orthographique (agent Hnt) :

Un agent de suggestion orthographique permet la suggestion d'une graphie alternative si celle ci est plus probable que la requête.

4.1.3.1.1 Configuration du service de réponse

Le fichier de configuration doit également contenir les paramètres de configuration de l'agent.

```
<Hints>
  <Service name="NomService">
    <ALPHA>0.25</ALPHA>
    <K>40</K>
    <S>10</S>
  </Service>
  ...
</Hints>
```

- `ALPHA` : Paramètre de calcul de proximité
- `K` : fréquence du candidat. Un candidat est retenu si sa fréquence est supérieure à K * la fréquence de la requête.
- `S` : seuil de fréquence en deça duquel l'agent répond (si S=5 l'agent ne répond que pour les mots dont la fréquence est strictement inférieure à 5).

4.1.4 Recherche transversale : ACC (Automatic Across Content)

L'agent ACC permet un calcul de similarité entre les informations permettant ainsi une recherche transversale.

Son utilisation est différente selon le type de source indexée. Ainsi pour l'indexation de sites Web, elle nécessite l'implantation d'un agent, l'agent See_Also, alors que pour l'indexation de données formatées ou semi-formatées, les ACC ne sont pas gérés par un agent spécifique.

Configuration d'indexation

Le fichier de configuration doit être renseigné avant l'indexation de la même façon que pour les RTE.

```
<Classification>
  <Enabled>true</Enabled>
  <N_grams>
    <Enabled>true</Enabled>
    <Min_Length>1</Min_Length>
    <Max_Length>6</Max_Length>
  </N_grams>
</Classification>
```

4.2 Mise à jour des index en temps réél

Le Web Service ws1.antisearch.net permet de notifier les modifications de la base à prendre en compte lors de la prochaine indexation, il s'agit d'un CGI à contacter en mode HTTP GET. Le CGI appelé update_index est hébergé par le serveur ws1.antisearch.net et log les modifivations sur le serveur SQL logs2, database : WEB_SERVICES, table : REQUEST.

Les paramètres d'appel de ce CGI sont:

1. C : Numéro de service
2. S : Clé de confirmation. Actuellement la valeur utilisée est le n° de service inversé
3. AGENT : La source de données à mettre à jour (user1, antibot ...)
4. ACTION : L'action demandée sur la fiche à mettre à jour: update, insert, delete
5. URI : L'identifiant de la fiche à mettre à jour.

Exemple d'URL :
http://ws1.antisearch.net/cgi-bin/update_index?C=152&S=251&AGENT=user1&ACTION=update&URI=1

Le CGI verifie si les URLS sont bien formées (si elles comportent chacun des paramètres obligatoires et s'ils sont renseignés) et retourne les messages suivants :

Si l'opération se déroule correctement, le message indique le statut « NO_CGI_ERROR ».

```
<Update_Index_Result>
  <Message>Operation complete</Message>
  <Status>NO_CGI_ERROR</Status>
</Update_Index_Result>
```

En cas de problème, un message d'erreur.
Exemple, si l'URI n'est pas renseignée :

```
<Update_Index_Result>
  <Message>Parameter [URI] is mandatory !</Message>
  <Status>MISSING_PARAMETER</Status>
</Update_Index_Result>
```

Le CGI vérifie uniquement si l'URL est bien formée et n'a aucune connaissance de la validité des identifiants fournis (URL), c'est donc au client de s'assurer de la validité des informations.

Côté exploitation, à chaque nouvelle indexation lancée par la crontab, la table REQUEST est consultée et réindexe selon les informations renvoyées.