

Documentation de Pleade 3 (v. 3.3.1)

Manuel d'administration

AJLSM
17 rue Vital Carles
33000 Bordeaux, France
Téléphone : +33 5 57 14 25 25 // Télécopieur : +33 5 56 44 08 47
info@ajlsm.com

11/03/2010

Avertissement

Cette documentation est fournie par AJLSM à ses clients et partenaires pour leur seul usage.

Toute redistribution publique est interdite. Toute réutilisation du contenu doit faire l'objet d'une autorisation préalable de la part d'AJLSM

© 2010 AJLSM

Table des matières

1) Introduction ... 3
2) Recherche et publication ... 4
2.1 Champs et index ... 4
2.1.1 Concepts ... 4
2.1.2 Champs « standards » ... 4
2.1.3 Index ... 5
2.1.4 Champs de recherche ... 7
2.2 Champs combinés ... 7
2.3 Champs dupliqués ... 8
2.4 Propriétés ... 9
2.4.1 Propriétés de publication ... 9
2.4.2 Propriétés d'affichage ... 11
2.4.3 Propriétés OAI ... 12
2.5 Redémarrer l'application ... 12
3) Menus ... 12
3.1 Structure générale ... 12
3.2 Élément « menu » ... 13
3.3 Élément « item » ... 14
4) Formulaires de recherche ... 16
4.1 Introduction ... 16
4.2 Structure générale ... 16
4.3 Attributs du formulaire ... 17
4.4 Les critères de recherche ... 18
4.4.1 Principales caractéristiques communes ... 18
4.4.2 Types de critères ... 19
4.4.3 Fonctions avancées ... 23
5) Rubriques ... 27
5.1 Approche générale ... 28
5.2 Publication ... 31
6) Statistiques ... 32
6.1 Présentation ... 32
6.1.1 Statistiques de recherche ... 32
6.1.2 Journaux de publication ... 33
7) Pages statiques ... 33
7.1 Gestion de fichiers ... 34
7.2 Ajouter un fichier dans le dossier ... 34
Annexes ... 36
1.Fichier de configuration des champs « standards » ... 36
2.Fichier de configuration des index ... 38
3.Fichier de configuration des champs combinés ... 42
4.Fichier de configuration des champs dupliqués ... 42
5.Fichier de configuration des menus ... 42
6.Fichier de configuration du formulaire : ead-default.xconf ... 45

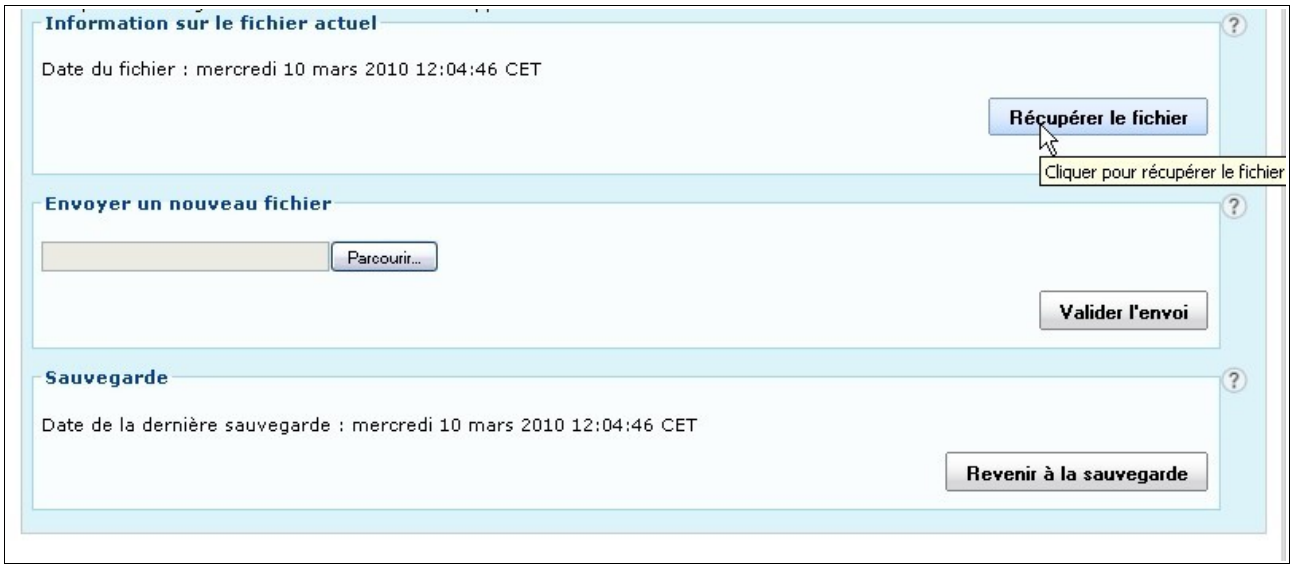
© 2010 AJLSM

1) Introduction

Cette documentation concerne l'administration de Pleade accessible dans l'onglet « *Administration* ». Pour pouvoir y accéder, votre profil d'utilisateur doit contenir le rôle « `eadeac-admin` ».

L'interface d'administration de Pleade permet de paramétrer la méthode de publication des documents EAD par défaut, de pouvoir créer des champs et des index, de créer des formulaires de recherche personnalisés, de configurer l'apparence et l'ergonomie de l'application (menus, rubriques, pages de contenu statique) et de pouvoir consulter les statistiques d'utilisation de Pleade.

Dans chaque page de l'interface d'administration, le paramétrage des différentes fonctionnalités est effectué simplement en éditant et en modifiant le fichier XML associé. Ces modifications peuvent être effectuées à l'aide d'un éditeur XML ou d'un éditeur de texte, en faisant toutefois attention dans ce dernier cas à ne pas briser la structure XML des documents et à respecter l'encodage des fichiers (UTF-8).



Voici les étapes à réaliser pour modifier un fichier de configuration depuis l'interface :

1. Récupérer le fichier de configuration.
   Chaque page propose un bouton pour récupérer le fichier de configuration. Il permet de sauvegarder sur le poste de travail une copie du fichier afin de pouvoir l'éditer.
2. Modifier la copie du fichier de configuration. **Ne pas modifier le nom du fichier.**
3. Envoyer la copie du fichier de configuration modifiée.
   Chaque page contient un formulaire pour renvoyer le fichier modifié vers l'application. Le nouveau fichier sera automatiquement intégré dans l'application. Pour les configurations concernant la recherche et la publication, il est nécessaire, par la suite, de redémarrer l'application via l'onglet « *Redémarrer l'application* ».

Chaque fichier de configuration est automatiquement sauvegardé avant d'être remplacé par un nouveau. Il est ainsi possible, si elle existe, de restaurer une version antérieure du fichier de configuration.

Ce manuel détaille l'utilisation et les possibilités de chacune des sections de l'onglet « *Administration* ».

© 2010 AJLSM

2) Recherche et publication

2.1 Champs et index

2.1.1 Concepts

Champ

Dans Pleade, un ***champ*** est lié à la recherche : il s'agit d'un contenu qui peut être recherché individuellement. Toutes les recherches s'effectuent dans des champs, y compris la recherche « *plein texte* » qui s'effectue dans le champ nommé « `fulltext` » et qui n'a rien de particulier.

C'est au moment de la publication que Pleade « remplit » les champs avec des valeurs issues des documents EAD. Donc toute modification à la configuration des champs **nécessite une republication** pour en voir les effets.

Il y a différents types de champs, les plus courants étant les champs de type « mot » (« `word` ») ou les champs de type « valeur » (« `field` »).

Les champs de type « mot » sont destinés à des recherches de type « libre », c'est-à-dire où une analyse des mots est effectuée (suppression des mots vides, des majuscules, des accents, etc.) et où on exprime des requêtes constituées de un ou plusieurs mots ou expressions.

Les champs de type « valeur » sont destinés à une recherche exacte, la plupart du temps depuis une liste de valeurs possibles.

Par exemple, si l'on considère la chaîne de caractères « François-René de Chateaubriand », on obtient :

- pour un champ de type « mot » : `francois`, `rene`, `chateaubriand`
- pour un champ de type « valeur » : `François-René de Chateaubriand`

Index

Les ***index*** dans Pleade servent à la navigation dans le cadre de gauche de la fenêtre de consultation des documents EAD. Pour que les index fonctionnent, des champs de recherche associés sont également nécessaires.
Les ***index*** peuvent aussi servir à la recherche. Dans ce cas ils sont considérés comme des champs comme tous les autres.

2.1.2 Champs « standards »

Pleade propose une liste de champs « standards », accessible dans l'onglet intitulé « *Champs* ». Cette liste n'est pas réellement destinée à être modifiée. Toutefois, dans certains cas, on peut supprimer des champs afin d'alléger la publication des documents. Mais ces contenus ne seront plus cherchables, donc à faire avec précaution.

La liste des champs standards se trouve dans le fichier `module-eadeac/conf/ead-fieldlist-base.xconf` (voir en Annexe 1. Fichier de configuration des champs « standards »). On y retrouve des définitions de champs qui respectent cette forme :

```
<field name="" type="word|field|date|unindexed" brief="true|false"/>
```

Chaque champ doit avoir un nom unique (y compris parmi les champs définis par d'autres méthodes, telles que documentées dans les sections suivantes) qui doit être dans l'attribut `name`. Ce nom ne doit pas comprendre d'espaces, ni de caractères spéciaux ou accentués. Par usage dans Pleade, on donne des noms de champs préfixés par `w` pour champs de type « mot » (word) si par défaut ils sont de type « valeur » ou préfixés par `f` si par défaut ils sont de type « mot ». Par exemple : `frepository`, `wunitid`. Il est ainsi plus facile de savoir quels champs utilisés dans quelles circonstances.

Ensuite, l'attribut `type` indique le type, avec les valeurs suivantes :

© 2010 AJLSM

word
Un champ de type « mot ». Voir la définition dans la section « Concepts » (2.1.1 Concepts).

field
Un champ de type « valeur ». Voir la définition dans la section « Concepts » (2.1.1 Concepts).

date
Un champ de type « date ». Ces champs sont traités de manière particulière et permettent de faire des recherches par intervalle de dates. Il doit s'agir de valeur au format ISO 8601.

unindexed
Un champ qui n'est pas cherchable. Il peut toutefois servir à l'affichage des résultats de recherche.

En effet, l'attribut `brief` permet d'indiquer si le champ doit être présent dans les résultats de recherche. Par défaut (si l'attribut n'est pas présent ou si sa valeur est `false`), il n'est pas dans les résultats. Pour l'obtenir, il faut mettre sa valeur à `true`. Pour pouvoir trier sur un champ, il faut aussi que celui-ci soit présent dans les résultats de recherche.

Toute modification à cette liste doit être suivie d'un **redémarrage de l'application** pour en constater l'effet.

2.1.3 Index

La configuration des index s'effectue dans le fichier `module-eadeac/conf/ead-index.xconf` (voir en Annexe 2. Fichier de configuration des index). On y définit les index disponibles dans l'application, et pour chaque index la manière de les remplir depuis les documents EAD ainsi que les champs de recherche qui y seront associés. Toute modification à ces index doit être suivie d'un **redémarrage de l'application** pour en constater l'effet.

Pour bien comprendre la configuration des index, nous allons dans un premier temps documenter comment ils sont déclarés, ensuite comment ils sont remplis, et enfin comment on y associe des champs de recherche et pourquoi.

2.1.3.1 Déclaration

Un index est déclaré par l'ajout d'un élément `index` à l'intérieur de l'élément racine `index-list` du fichier de configuration, et en attribuant un identifiant unique `id` à cet index. Par exemple :

```
<index-list>
  <index id="auteur" display-order="" ...>
    ...
  </index>
  ...
</index-list>
```

N.B. : Pour les identifiants, éviter les espaces, les accents et autres caractères spéciaux.

Dans cet exemple, un index identifié par « auteur » est déclaré. **L'ordre de déclaration des index est très important**, car un contenu EAD ne peut faire partie que d'un seul index. Ainsi, lorsque Pleade tente d'associer un contenu EAD à un index, le premier index approprié dans le fichier sera utilisé, les suivants seront ignorés. Ainsi, les index prioritaires doivent être placés en premier dans le fichier.

On peut toutefois définir un ordre d'affichage différent des index en ajoutant un attribut `display-order` à l'élément `index`. Cet attribut doit contenir un nombre entier, les index seront affichés dans l'ordre numérique dans le cadre de gauche utilisé pour la navigation dans la fenêtre EAD.

2.1.3.2 Contenu

Le contenu d'un index est contrôlé par les sous-éléments que l'on insère à l'intérieur de l'élément index correspondant. Ces sous-éléments indiquent quelles sont les structures EAD qui vont correspondre à l'index et vont donc l'alimenter.

À noter que Pleade considère uniquement certains éléments EAD comme pouvant faire partie d'un index : `unitid`, `publisher`, `corpname`, `famname`, `geogname`, `name`, `occupation`, `persname`, `subject`, `genreform`, `function`, `title`. Tous les autres éléments ne peuvent pas participer à la construction d'un

© 2010 AJLSM

index. Par ailleurs, **si ces éléments sont à l'intérieur du `eadheader` ou du `frontmatter`, ils ne renseigneront pas les index**.

Pour définir le contenu d'un index, on utilise une structure XML telle que :

```
<index id="test" ...>
  <content element="" role="" source="" type="" ancestor=""/>
  ...
</index>
```

On peut donc retrouver une série de sous-éléments `content`, chacun définissant une structure EAD qui vient alimenter l'index si elle est présente dans le document. Ces structures sont cumulatives, c'est-à-dire que plus vous ajoutez d'éléments `content`, plus l'index sera alimenté.

L'élément `content` peut prendre cinq attributs. Ils sont tous facultatifs, mais il doit y en avoir au moins un pour que l'élément serve à l'alimentation du contenu. Si plusieurs attributs sont présents, l'élément EAD doit respecter toutes les règles pour alimenter l'index.

Donc plus on ajoute d'éléments `content`, plus on alimente l'index. Mais plus on ajoute d'attributs à l'élément `content`, moins on alimente l'index.

Ces attributs sont :

`element`

Le nom de l'élément EAD qui alimente le contenu. Par exemple, pour la valeur `geogname`, seuls les éléments EAD `geogname` viendront alimenter le contenu.

```
<index id="test">
  <content element="geogname"/>
</index>
```

`role`

L'attribut `role` de l'élément EAD. Par exemple, pour la valeur `auteur`, seuls les éléments EAD (qui participent aux index) qui ont un attribut `role` avec la valeur `auteur` viendront alimenter le contenu.

```
<index id="test">
  <content role="auteur"/>
</index>
```

`source`

L'attribut `source` de l'élément EAD. Par exemple, pour la valeur `thesaurus-w`, seuls les éléments EAD (qui participent aux index) qui ont un attribut `source` avec la valeur `thesaurus-w` viendront alimenter le contenu.

```
<index id="test">
  <content source="thesaurus-w"/>
</index>
```

`type`

L'attribut `type` de l'élément EAD. Par exemple, pour la valeur `filtre`, seuls les éléments EAD (qui participent aux index) qui ont un attribut `type` avec la valeur `filtre` viendront alimenter le contenu.

```
<index id="test">
  <content type="filtre"/>
</index>
```

`ancestor`

L'attribut `ancestor` permet de définir le contexte dans lequel doit se trouve un élément EAD pour alimenter un index. Par exemple, si on indique la valeur `imprint` pour `ancestor`, un élément viendra alimenter un index seulement s'il se situe dans un élément `imprint`, directement ou non.

```
<index id="test">
```


© 2010 AJLSM

```
  <content ancestor="imprint"/>
</index>
```

2.1.4 Champs de recherche

Les index peuvent être utilisés pour la navigation ou la recherche. Dans ces deux cas de figure, Pleade a besoin d'un champ de recherche associé à l'index, et le nom de ces champs doit être renseigné. Ils doivent aussi être uniques parmi tous les noms de champs (standards, dupliqués, combinés, ...) définis dans l'application.

Les noms de champs sont ajoutés en attribut de l'élément index, de cette manière :

```
<index id="..." local-sdxfield="" global-sdxfield="" word-sdxfield="" display-order=""...>
  ...
</index>
```

Il y a donc trois attributs possibles, et ils sont tous trois facultatifs.

`local-sdxfield`

Le nom du champ qui sera utilisé par Pleade pour la navigation dans le cadre de gauche. En effet, lorsque plusieurs unités de description d'un même document EAD sont indexées par un terme, une recherche est effectuée pour en présenter la liste. Par usage dans Pleade, on préfixe ce nom de champ par `lf`, comme « local field » (champ de type « valeur » pour la navigation). Par exemple : `lfpersname`, `lfsubject`. Il s'agit d'une astuce pour se souvenir plus facilement du nom des champs.

Ce champ sera systématiquement de type « valeur » dans Pleade.

Si cet attribut est absent, alors cet index ne sera pas utilisé en navigation.

`global-sdxfield`

Le nom du champ de type « valeur » qui permettra d'effectuer des recherches sur la totalité des valeurs de cet index, c'est-à-dire issues de tous les documents EAD. Il est utile dans un formulaire de recherche par exemple. Par usage dans Pleade, on préfixe ce nom de champ par `f`, comme « field » (champ de type « valeur »). Par exemple : `fpersname`, `fsubject`. Il s'agit d'une astuce pour se souvenir plus facilement du nom des champs.

Ce champ sera systématiquement de type « valeur » dans Pleade.

Si cet attribut est absent, alors cet index ne sera pas utilisé en recherche globale.

`word-sdxfield`

Le nom du champ de type « mot » qui permettra d'effectuer des recherches sur la totalité des valeurs de cet index, c'est-à-dire issues de tous les documents EAD. Il est utile dans un formulaire de recherche par exemple.

Ce champ sera systématiquement de type « mot » dans Pleade.

Si cet attribut est absent, alors cet index ne sera pas utilisé en recherche globale par des critères de type saisie libre.

2.2 Champs combinés

Pleade offre un mécanisme très simple pour créer des champs « combinés », c'est-à-dire dont le contenu est obtenu par l'agrégation des valeurs d'autres champs. Par exemple, si on peut définir un champ de type « mot » nommé « `analysetitre` » par l'agrégation des champs « `unittitle` » et « `scopecontent` », alors le champ « `analysetitre` » contiendra tous les mots que l'on trouve dans les champs « `unittitle` » et « `scopecontent` », et ce pour toutes les unités de recherche.

À noter que pour un fonctionnement optimal, les champs combinés doivent **être de même type que les champs utilisés** pour former leur contenu. Leurs noms doivent aussi être uniques parmi tous les noms de champs (standards, dupliqués, index, ...) définis dans l'application.

© 2010 AJLSM

La configuration des index s'effectue dans le fichier `module-eadeac/conf/ead-aggregated-fields.xconf` (voir en Annexe 3. Fichier de configuration des champs combinés). Le contenu du fichier de configuration est très simple, on y trouve des structures du genre :

```
<aggregated-field name="{nom du nouveau champ}" type="word|field|date">
  <field name="{nom du premier champ fournisseur de contenu}"/>
  <field name="{nom du second champ fournisseur de contenu}"/>
  ...
</aggregated-field>
```

Ainsi, pour chaque champ agrégé on crée un élément `aggregated-field` ; on précise son nom dans l'attribut `name` et son type dans l'attribut `type` (trois valeurs possibles `word`, `field`, `date`). On y insère autant d'éléments `field` qu'il y a de champs qui contribuent au contenu de ce champ agrégé : pour chacun on indique le nom dans l'attribut `name`.
Comme pour les autres champs, chaque champ combiné doit avoir un nom unique (y compris parmi les champs définis par d'autres méthodes). Ce nom ne doit pas comprendre d'espaces, ni de caractères spéciaux ou accentués.

Pour la définition du champ « `analysetitre` » cité en exemple, on aurait donc :

```
<aggregated-field name="analysetitre" type="word">
     <field name="unittitle"/>
     <field name="scopecontent"/>
  </aggregated-field>
```

Toute modification à cette liste doit être suivie d'un **redémarrage de l'application** pour en constater l'effet.

2.4 Propriétés

Cet onglet permet de définir les propriétés par défaut utilisées lors de la publication des documents. La définition des propriétés se trouve dans le fichier `module-eadeac/conf/ead-documents-property-sets.xconf`.

Ces valeurs pourront être éventuellement modifiées au cas par cas, au moment de la publication d'un document.

Il a trois ensembles de propriétés : les propriétés de publication, les propriétés d'affichage des documents publiés.

Les tableaux suivants décrivent les propriétés, leur valeur par défaut dans la distribution de Pleade ainsi que leur signification. La signification archivistique de ces différentes propriétés est documentée dans le manuel de Gestion de contenu ; nous invitons le lecteur à consulter cette documentation en parallèle. Si une mention `[auto]` apparaît dans la colonne « Valeur par défaut », cela signifie que ce paramètre est renseigné automatiquement par Pleade et ne doit donc pas être indiqué dans le fichier de configuration.

2.4.1 Propriétés de publication

Nom	*Valeur par défaut*	*Description*
`eadid`	`[auto]`	L'identifiant du document EAD. Cette propriété aura toujours une valeur, elle est insérée par la première étape dans le processus de publication. **On ne doit donc pas la renseigner dans le fichier de configuration.**
`frag-method`	`all-components`	La méthode de fragmentation du document en unités de recherche. La valeur par défaut `all-components` consiste à fragmenter sur tous les composants. La valeur `_none` signifie qu'on ne crée aucun fragment. Pour connaître les valeurs correspondant aux différentes méthodes de fragmentation se reporter au manuel

© 2010 AJLSM

Nom	Valeur par défaut	Description
		Gestion de contenu, partie 2.2 Paramètres de publication. Pour chaque méthode, la valeur est indiquée entre crochets.
frag-method-info		Une information complémentaire associée à la méthode de fragmentation.
notsearchable-method	_none	La méthode d'identification des unités non cherchables, c'est-à-dire non retournées dans les résultats de recherche. La valeur _none signifie que toutes les unités sont cherchables. Pour connaître les valeurs correspondant aux différentes méthodes de fragmentation se reporter au manuel Gestion de contenu, partie 2.2 Paramètres de publication. Pour chaque méthode, la valeur est indiquée entre crochets.
notsearchable-method-info		Une information complémentaire associée à la méthode d'identification des unités non cherchables.
toc-method	_same	La méthode d'identification des entrées de table des matières. La valeur _same indique qu'il faut utiliser la même méthode que pour la fragmentation en unités de recherche. Pour connaître les valeurs correspondant aux différentes méthodes de fragmentation se reporter au manuel Gestion de contenu, partie 2.2 Paramètres de publication. Pour chaque méthode, la valeur est indiquée entre crochets.
toc-method-info		Une information complémentaire associée à la méthode d'identification des entrées de table des matières.
title-method	unittitle	La méthode d'identification des titres. La valeur par défaut unittitle consiste à utiliser les intitulés. Pour connaître les valeurs correspondant aux différentes méthodes de fragmentation se reporter au manuel Gestion de contenu, annexe Méthode pour les titres générés. Pour chaque méthode, la valeur est indiquée entre crochets.
title-separator	•	Le séparateur utilisé entre les différentes parties du titre. Il est aussi inséré entre les différentes occurrences d'un élément. La valeur par défaut représente la chaîne • (espace insécable, puce, espace). Les caractères sont indiqués avec leur code Unicode.
title-limit	150	Le nombre maximum de caractères à insérer dans un titre généré.
inheritance-always		Les champs à hériter des unités de description supérieures, et ce systématiquement. Les valeurs renseignées ici sont des noms de champs, séparés par des espaces. Par défaut, ce paramètre n'a aucune valeur.
index-global	_all	Les index à inclure dans l'indexation de toute la base. La mention _all indique que tous les index disponibles

© 2010 AJLSM

Nom	Valeur par défaut	Description
		seront inclus ; il s'agit de la valeur par défaut. **Attention :** Pour les instruments de recherche non cherchables, il faut supprimer la valeur `_all`.
`inheritance-empty`	`unittitle bdate edate wunitdate unitdate origination forigination`	Les champs à hériter des unités de description supérieures mais seulement s'il n'est pas défini pour l'unité de description courante. Les valeurs renseignées ici sont des noms de champs, séparés par des espaces. Pour connaître les champs disponibles se reporter au manuel Gestion de contenu.
`inheritance-index`		Indique quels éléments d'indexation à hériter. Les valeurs possibles sont « `global` » et/ ou « `local` », avec l'espace comme séparateur. Ce paramètre peut en général être ignoré, au profit de l'utilisation des « champs dupliqués », documentés ailleurs dans ce manuel.
`attached-extensions`	`pdf doc rtf`	La liste des extensions de fichiers (séparées par une espace, sans le « . ») qui doivent être considérés comme des documents attachés lorsqu'ils sont la cible d'un lien hypertexte relatif. **Attention** : cette option est sensible à la casse.
`dao-attached-extensions`	`jpg jpeg`	La liste des extensions de fichiers (séparées par une espace, sans le « . ») qui doivent être considérés comme des documents attachés lorsqu'ils sont associés comme objet numérique.
`dao-implicit-series`	`false`	Si la valeur est `true`, tous les liens vers des objets numériques seront considérés comme des séries d'images. À utiliser avec précaution.
`subset-list`		La liste des rubriques dont fait partie le document. On doit renseigner l'identifiant de la ou des rubriques, séparés par des espaces. Par défaut, ce paramètre n'a aucune valeur.
`docUrl`	`[auto]`	L'URL du document, renseignée automatiquement par Pleade. **On ne doit donc pas la renseigner dans le fichier de configuration.**
`show-illustrated`	`false`	Si la valeur est `true`, dans l'affichage du sommaire, du cadre de classement et des résultats de recherche, les entrées pour lesquelles il existe des documents numérisés sont précédés d'une icône spécifique, afin d'indiquer la possibilité de consulter un document numérisé. Cette icône peut varier en fonction du type de document numérisé.
`delete-internal`	`true`	Si la valeur est à `true`, les éléments EAD protégés par l'attribut `audience` avec la valeur `internal` sont supprimés de l'EAD lors de la publication pour des raisons de sécurité. Leur contenu n'est donc pas cherchables, ni affichables : il ne paraît pas non plus dans les index.

© 2010 AJLSM

Nom	Valeur par défaut	Description
		Si la valeur est à `false`, les éléments sont conservés mais leur contenu n'est pas cherchable et consultable uniquement sous certaines conditions (droits de l'utilisateur). L'usage de l'attribut `audience` doit donc être réfléchi.

2.4.2 Propriétés d'affichage

Les propriétés pour lesquelles la valeur par défaut indiquée dans le tableau suivant est `[auto]` ne doivent pas être renseignées dans le fichier de configuration.

Nom	Valeur par défaut	Description
`eadid`	`[auto]`	L'identifiant du document EAD. Cette propriété aura toujours une valeur, elle est insérée par la première étape dans le processus de publication.
`titleproper`	`[auto]`	Le titre propre de l'instrument de recherche.
`subtitle`	`[auto]`	Le sous-titre de l'instrument de recherche.
`author`	`[auto]`	Le ou les auteurs de l'instrument de recherche.
`sponsor`	`[auto]`	Le commanditaire de l'instrument de recherche.
`archdesc-id`	`[auto]`	L'identifiant de l'unité de description racine.
`search-form`	`simple`	Le formulaire de recherche associé au document, qui sera donc disponible dans la fenêtre de consultation du document EAD, dans le cadre de gauche.
`index-local`	`_all`	La liste des index à utiliser pour la navigation dans le cadre de gauche dans la fenêtre de consultation du document EAD. La valeur `_all` signifie tous les index disponibles.
`ead-link-extensions`	`xml`	Les extensions de fichiers à considérer comme des liens hypertextes vers d'autres documents EAD publiés dans Pleade.
`display-navigation`	`true`	

2.5 Redémarrer l'application

Toutes les modifications apportées aux fichiers de configuration de la section « *Recherche et publication* » nécessitent un redémarrage de l'application.

Cette opération ne prend qu'un instant.

3) Menus

Pleade propose des menus de navigation qui sont construits dynamiquement à partir du fichier de configuration `module-eadeac/conf/menus.xconf` (voir en Annexe 5. Fichier de configuration des menus). Chaque menu et chaque item y est défini de manière déclarative, sans programmation. La structure XML de ce fichier est très simple. Nous allons l'aborder par étape.

3.1 Structure générale

La structure générale du fichier est la suivante :

```
<menus>
  <menu id="" type="">
```


© 2010 AJLSM

```
      <item id="" type="" .../>
      <item id="" type="" .../>
      <item id="" type="" ...>
         <item id="" type="" .../>
         <item id="" type="" .../>
         ...
      </item>
      ...
   </menu>
   <menu>
      ...
   </menu>
</menus>
```

Ainsi, au premier niveau, le fichier contient un élément `menus` qui contient un ou plusieurs élément(s) `menu`. Ces derniers contiennent des éléments `item`, qui peuvent eux-mêmes contenir d'autres éléments `item`, ce qui constitue un *sous-menu* en quelque sorte.

Nous allons documenter d'abord la définition d'un `menu`, ensuite des `items`.

3.2 Élément « menu »

La définition d'un menu est fait à l'aide de la structure XML suivante :

```
<menu
         id=""
         type=""
         depends=""
         reserve=""
         class="pl-accordion"
>
   <item.../>
   <menu.../>
   ...
</menu>
```

Un élément `<menu>` peut contenir un sous-élément `<menu>` et / ou `<item>`.

Les informations importantes sont donc définies dans les quatre attributs, définis ci-dessous.

`id`

Obligatoire. L'identifiant du menu. Il doit être unique parmi tous les identifiants du fichier, menus ou items. Il sert notamment à déclarer les dépendances entre menus.

`type`

Obligatoire. Le type de menu, qui détermine notamment son emplacement dans la page. Les valeurs possibles sont : `htop`, pour un positionnement horizontal en haut de page, avant même l'en-tête de la page ; `htabs`, pour un positionnement sous l'en-tête de la page, sous la forme d'onglets à l'horizontal ; `vleft`, pour un menu de navigation vertical à gauche de la page.

`depends`

Optionnel. La dépendance du menu par rapport à un autre. Si un menu est dépendant d'un autre, il ne sera affiché que si le menu dont il est dépendant est lui-même actif. Le contenu de l'attribut est l'identifiant du menu dont il dépend. Si l'attribut n'est pas présent, il n'y a aucune dépendance (le menu est toujours affiché).

`class`

Optionnel. Permet de donner une classe particulière pour ajuster la mise en forme dans une feuille de style CSS. Dans le cas, d'un menu vertical (type `vleft`), si on donne la classe `pl-accordion` à un menu, celui-ci s'affichera sous la forme d'un accordéon, c'est-à-dire que l'on peut plier ou déplier ses sous-entrées.

© 2010 AJLSM





Un attribut `default` avec la valeur `true` permet de préciser le menu à afficher par défaut. Il ne doit être mis que sur un seul menu par type. Il est surtout utile pour les menus verticaux ; il peut y en avoir plusieurs car on peut associer à menu vertical à un onglet.

3.3 Élément « item »

La définition des attributs des items permet d'automatiser la création de plusieurs types de liens.

Pour ce qui est de son contenu, un item doit fournir une indication à l'utilisateur (un « texte ») et, dans la plupart des cas, un lien hypertexte.

La structure générale d'un item de menu est la suivante :

```
<item
        id=""
        type=""
        href=""
        context=""
        title=""
        field=""
        reserve=""
        reserve-uri=""
>
     Texte affiché du menu
     <item.../>
     ...
</item>
```

Un `item` peut contenir d'autres items, qui définissent ainsi des sous-menus, ou encore des items de niveau inférieur. La structure des éléments `item` est la même peu importe le niveau.

Le texte affiché pour l'item est celui que l'on trouve à l'intérieur de l'élément `item`. Dans le fichier par défaut de Pleade, ce texte utilise le mécanisme `i18n` de Cocoon, mais cela n'est pas obligatoire et n'est pas documenté ici. On peut y insérer du texte normal.

Ensuite, les informations qui définissent le comportement de l'item sont renseignées en attribut de l'élément `item`. Nous les définissons ici.

`id`

Obligatoire. L'identifiant de l'item. Il doit être unique parmi tous les identifiants du fichier, menus ou items.

`type`

Obligatoire. Le type d'item. Plusieurs valeurs sont possibles, et le contenu de l'attribut `href` sera interprété différemment selon le type d'item.

- `link-normal` : un lien normal, c'est-à-dire dont la cible sera affichée dans la même page que celle où se trouve l'item.

© 2010 AJLSM

- `link-ead` : un lien vers un document EAD, qui ouvrira la fenêtre de consultation d'un document EAD. L'attribut `href` doit alors contenir l'identifiant du document EAD.
- `link-window` : un lien qui ouvrira une autre fenêtre de navigation dans le navigateur Internet. Si l'on souhaite que la fenêtre ait un nom spécifique dans le navigateur (ce nom n'est pas visible, mais cela permet d'éviter plusieurs fenêtre différentes), on peut renseigner ce nom après un « : », par exemple `link-window:ext`.
- `txt` : un item qui n'est pas un lien, mais uniquement un texte affiché.
- `grp` : un item qui contient uniquement d'autres items. S'il contient directement du texte, celui-ci sera ignoré.
- `simple-search` : un formulaire de recherche simple sera inséré par Pleade à l'emplacement de cet item. Si ce type est utilisé, on peut ajouter quatre attributs à l'élément item : `action`, pour indiquer la page de résultats de recherche (par défaut `results.html`) ; `method`, pour indiquer la méthode de soumission du formulaire (par défaut `get`) ; `base`, pour indiquer la ou les base(s) de documents à rechercher (par défaut `ead`), séparées par un espace ; `@field` pour modifier le champ de recherche (par défaut `fulltext`).



`href`

Obligatoire, sauf pour les liens de type `txt` ou `simple-search`. Indique l'adresse de la cible du lien représenté par cet item.

Pour les items de type `link-window` ou `link-normal`, `href` doit contenir l'adresse URL, qui peut être relative ou absolue (les protocoles supportés sont : http, mailto.). Dans le cas d'adresse relative, on doit la renseigner par rapport à l'URL racine de l'application Pleade.

Pour les items de type `link-ead`, c'est l'identifiant du document EAD uniquement qui doit être renseigné dans cet attribut.

`onclick`

Optionnel. Cet attribut et son contenu sont reportés dans le HTML généré.

`context`

Optionnel. Le contexte où cet item est « actif », c'est-à-dire le patron d'URL qui, s'il est respecté par la page en cours d'affichage, rend l'item actif. Le patron est une expression régulière valide selon la norme XSLT. Le contexte peut aussi être donné par l'attribut `href`, dans ce cas, il faut indiquer `@href` comme contexte.

Par exemple, si le contexte est `edit/.*`, cela signifie que l'item sera actif lorsque la page affiché a une URL qui comme par `edit/`.

`field`

Optionnel et uniquement pour un item de type `simple-search`. Précise le nom du champ de recherche pour la recherche simple, par défaut celui-ci est `fulltext`.

`title`

Optionnel, mais recommandé pour des raisons d'accessibilité et de facilité d'utilisation. Le texte qui sera affiché en bulle d'aide lorsque l'utilisateur passe la souris au-dessus du lien. Par défaut, le

© 2010 AJLSM

contenu de cet attribut est une clé i18n dont la valeur est définie dans un fichier de message. Cette méthode simplifie l'internationalisation de l'application. Mais ce n'est pas obligatoire, on peut y mettre le texte directement.

`reserve`

Optionnel. Cet attribut peut contenir un ou plusieurs rôles (séparés par une espace), et si c'est le cas l'item ne sera visible que si l'utilisateur est identifié et joue l'un de ces rôles. Par défaut, c'est-à-dire si l'attribut n'est pas renseigné, l'item est visible par tous.

`reserve-uri`

Optionnel. Cet attribut peut contenir un ou plusieurs patrons d'URL (séparés par une espace), et si c'est le cas l'item ne sera visible que si l'URL demandée correspond à l'un des patrons. Par défaut, c'est-à-dire si l'attribut n'est pas renseigné, l'item est visible pour toutes les URL.

4) Formulaires de recherche

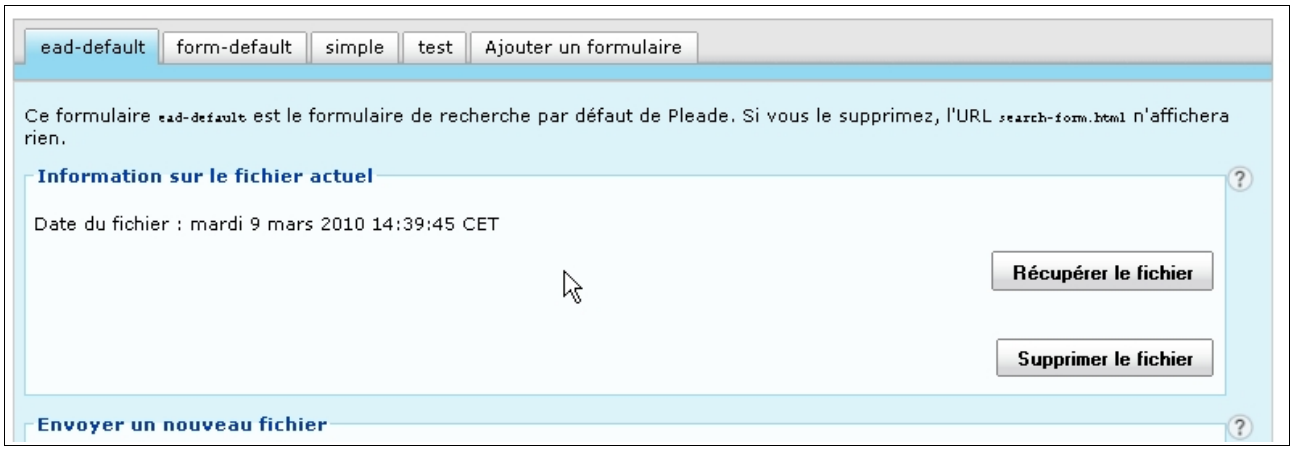
4.1 Introduction

Dans une application Pleade, on peut proposer aux utilisateurs des formulaires de recherche, qui permettent d'exprimer une requête et de lancer la recherche. Dans un projet d'intégration Pleade, la préparation de ces formulaires de recherche est en général une activité importante.

On peut afficher dans Pleade un formulaire de recherche avec une URL qui utilise le nom du fichier de formulaire. Par exemple, pour afficher le formulaire défini dans le fichier `cadastre.xconf`, il faut utiliser l'URL `search-form.html?name=cadastre`. Donc dans le paramètre `name`, on indique le nom du fichier du formulaire, mais sans son extension. Tous les fichiers de configuration de formulaires sont dans le répertoire `module-eadeac/conf/search-forms/`.

Le formulaire nommé `ead-default.xconf` est un cas particulier (voir en Annexe 4) Formulaires de recherche), il peut être appelé directement par l'URL `search-forms.html`, sans paramètre `name`. Il s'agit du formulaire par défaut de Pleade.

Dans l'interface d'administration des menus, un formulaire est présenté comprenant autant d'onglets que de formulaires configurés dans l'application, plus un permettant d'ajouter un nouveau formulaire.



4.2 Structure générale

La structure générale du document XML qui décrit un formulaire est schématisée ainsi :

```
<form ...>
  <key.../>
  <key.../>
  ...
</form>
```


© 2010 AJLSM

Cette structure est donc très simple : un élément `form` qui matérialise le formulaire et ensuite une suite d'éléments `key` qui représentent chacun un critère de recherche.

On constate donc une caractéristique importante des formulaires Pleade : il s'agit nécessairement d'un formulaire de recherche « à plat », c'est-à-dire constitué d'une suite de critères, mais sans regroupement, sans sous-critères, etc.

4.3 Attributs du formulaire

L'élément racine `form` dans le fichier de définition du formulaire peut prendre ces différents attributs :

```
<form
        action=""
        max-chars=""
        documents=""
        auto-expand=""
>
  ...
</form>
```

La signification des différents attributs est :

`action`

Obligatoire. L'URL qui doit être appelée par le formulaire lorsque l'utilisateur valide sa recherche. Pour l'instant, les seules valeurs supportées par Pleade doivent être de la forme `*-results.html`. La valeur par défaut est `results.html`. Les types d'affichage des résultats possibles sont :

- `results.html` : les résultats sont affichés selon le cadre de classement. Affichage par défaut.
- `table-results.html` : les résultats sont affichés sous forme de tableau
- `list-results.html` : les résultats sont affichés en liste sans hiérarchie

`max-chars`

Optionnel. Dans les listes de valeurs, le nombre maximal de caractères à afficher en largeur, afin d'éviter des débordements (listes trop longues en largeur). Cette indication peut également être fournie dans la définition d'un critère de recherche.

`documents`

Obligatoire. La base de documents à rechercher, dans une installation standard de Pleade, les valeurs possibles sont : `ead` ou `eac`. Plusieurs valeurs peuvent être indiquées séparées par une espace.

`auto-expand`

Optionnel. Les résultats de recherche étant affichés dans l'ordre du cadre de classement, on peut indiquer avec cet attribut comment les entrées du cadre de classement seront affichées, c'est-à-dire si elles seront automatiquement « ouvertes » ou non. Si cet attribut n'est pas renseigné, c'est le paramètre d'installation correspondant qui fait foi.

Cet attribut peut prendre différentes valeurs :

- `_all` : tous les niveaux seront automatique ouverts.
- `n` (un nombre, par exemple 3) : les entrées jusqu'au niveau indiqué seront automatiquement ouvertes.
- `docs:n` (la mention « docs » suivie d'un nombre) :une entrée sera ouverte uniquement si elle contient au plus `n` documents résultats.
- `units:n` (la mention « units » suivie d'un nombre) : une entrée sera ouverte uniquement si elle contient au plus `n` unités de recherche résultats.

© 2010 AJLSM

4.4 Les critères de recherche

4.4.1 Principales caractéristiques communes

Chaque critère de recherche est exprimé par un élément `key`. Cet élément peut prendre de nombreux attributs mais également des sous-éléments. Ces attributs ou sous-éléments sont parfois communs à l'ensemble des critères, soit spécifiques à un type de critère.

Nous allons dans un premier explorer les principales caractéristiques communes à tous les types de critères. La structure générale de l'élément `key` est la suivante :

```
<key
        id=""
        type=""
        field=""
        fieldh=""
        connector=""
        format=""
        max-chars=""
        terms-browse=""
        help=""
>
  ...
</key>
```

Les attributs et leur définition sont :

`id`

Optionnel. L'identifiant unique du critère qui va permettre de pouvoir distinguer le critère d'un autre dans la mise en forme par feuille de style CSS.

`type`

Obligatoire. Le type de critère ; ce qui va déterminer comment il sera affiché (zone de saisie, liste, etc.) et aussi quels sont les autres attributs qui pourront lui être associés. Les valeurs possibles sont : `cdc`, `text`, `suggest`, `list`, `hidden`, `date`, `boolean`.

`field`

Obligatoire pour la plupart des types de critères. Le champ dans lequel s'effectue la recherche avec ce critère. Si l'attribut `fieldh` est spécifié, l'attribut `field` indique le champ utilisé lorsqu'un seul critère de recherche est renseigné par l'utilisateur.

Pour connaître la liste des champs définis par défaut dans Pleade, se reporter au manuel Gestion de contenu.

`fieldh`

Optionnel, mais fortement conseillé pour la plupart des types de critères. Le champ dupliqué dans lequel s'effectue la recherche si au moins deux critères de recherche sont renseignés par l'utilisateur.

`connector`

Optionnel. Des informations sur l'opérateur booléen entre ce critère et le précédent dans le formulaire. La valeur de cet attribut est un nombre :

- `1` : l'opérateur est forcément le ET, il ne sera pas affiché. C'est également le comportement si l'attribut `connector` n'est pas renseigné.
- `2` : l'opérateur est forcément le OU, il ne sera pas affiché.
- `3` : l'opérateur peut être le ET ou le OU, et le choix sera laissé à l'utilisateur, par l'affichage d'une liste déroulante à gauche du critère.
- `4` : l'opérateur est forcément le SAUF, il ne sera pas affiché.

© 2010 AJLSM



format

Optionnel. Le format accepté pour le critère, particulièrement utile pour les dates ou les zones de texte, dans certains cas ; il déclenche la validation par javascript du critère à la soumission du formulaire. Si l'utilisateur saisit une valeur non permise par le format, le formulaire ne sera pas validé et un message sera affiché. Le contenu de cet attribut doit être une expression régulière valide en Javascript.

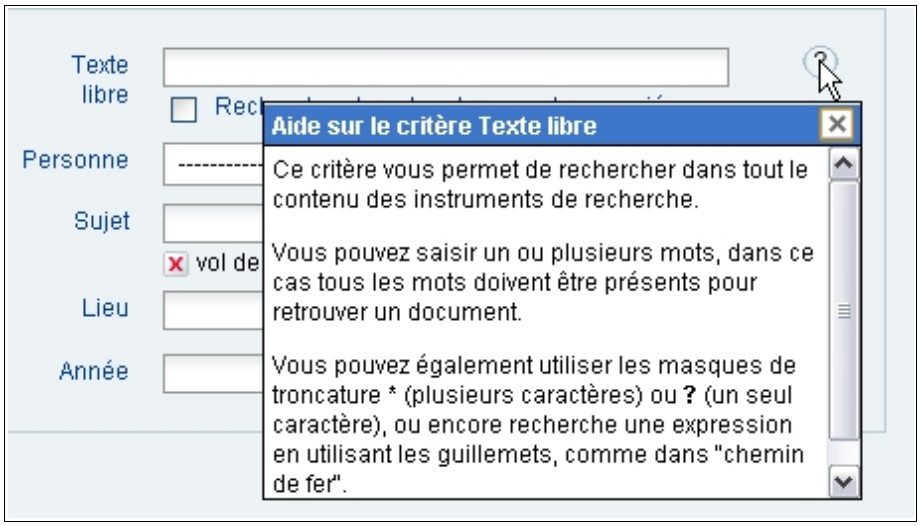
max-chars

Optionnel. Dans les listes de valeurs, le nombre maximal de caractères à afficher en largeur, afin d'éviter des débordements (listes trop longues en largeur). Cette indication peut également être fournie dans la définition d'un critère de recherche.

Si cet attribut n'est pas renseigné, c'est la valeur de l'attribut défini au niveau de l'élément `form` qui s'applique sauf dans le cas des dates, où la valeur est automatiquement fixée à quatre (recherche par année seulement).

help

Optionnel. Si sa valeur est `true`, un bouton d'aide est affiché à droite du critère.



Les critères peuvent prendre d'autres attributs, qui sont expliqués soit dans les explications sur les différents types de critères dans la section suivante soit dans les fonctions avancées un peu plus loin (4.4.3 Fonctions avancées).

4.4.2 Types de critères

Un formulaire de test reprenant les types de critères courants et des commentaires explicatifs est livré par défaut. Il peut servir de modèle pour créer ou modifier un formulaire. Ce formulaire de test est visible à l'URL `/search-forms.html?name=test`.

4.4.2.1 Critère caché (`hidden`)

Les critères cachés permettent d'ajouter systématiquement un critère de recherche de manière transparente pour l'utilisateur. On doit utiliser la valeur `hidden` pour l'attribut `type`. Il supporte un attribut spécifique :

value

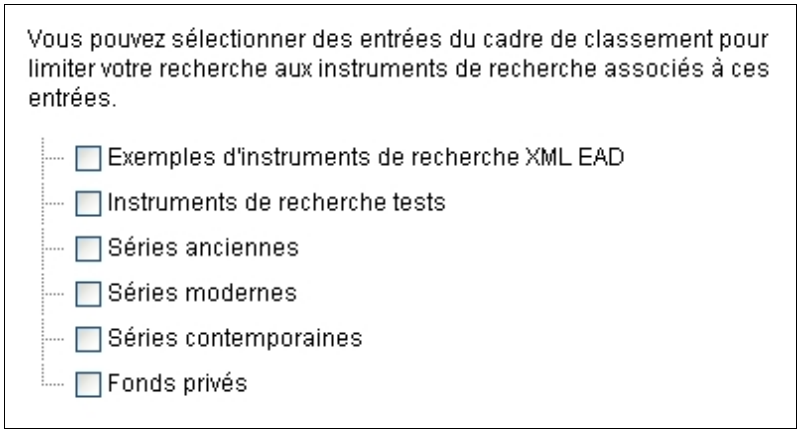
Obligatoire pour utiliser ce critère. Il s'agit de la valeur que doit avoir le champ indiqué dans l'attribut `field`.

© 2010 AJLSM

```
<key type="hidden" connector="2" field="unitid" value="2002W" />
```

4.4.2.2 Cadre de classement (`cdc`)

Ce critère se comporte de manière particulière : il permet d'afficher le cadre de classement à droite du formulaire, et l'utilisateur pourra choisir une ou plusieurs entrées du cadre de classement à l'aide de cases à cocher.



Il ne supporte aucun attribut spécifique.

```
<key type="cdc"/>
```

Pour un critère de ce type, les attributs `field` et `fieldh` ne sont pas utiles, car Pleade utilise toujours les mêmes champs de recherche. De même, l'attribut `help` n'a aucun impact.

4.4.2.3 Zone de texte (`text`)

Ce critère se présente sous la forme d'une zone de texte où l'utilisateur peut saisir librement des mots, expressions, etc. On l'utilise normalement pour des recherches sur des champs de type « mot ».



Pour un tel critère, on peut ajouter des attributs afin de permettre à l'utilisateur de choisir un autre champ de recherche, un champ « alternatif ». Cela est particulièrement utile pour proposer à l'utilisateur d'étendre la recherche aux documents associés, comme le fait le formulaire par défaut de la distribution standard de Pleade.

Quand est un champ alternatif est défini pour un critère, une boîte à cocher est affichée sous le critère offrant à l'utilisateur un choix.

```
<key type="text" field="fulltext" fieldh="hfulltext" help="true" alternate-field="extfulltext"
alternate-fieldh="hextfulltext" alternate-field-checked="false"/>
```

Pour réaliser cela, il faut ajouter les attributs suivants à l'élément `key` :

`alternate-field`

Obligatoire pour utiliser cette fonctionnalité. Le nom du champ à rechercher si l'utilisateur choisit cette option. Si l'attribut `alternate-fieldh` est également renseigné, ce champ sera utilisé uniquement s'il y a un seul critère de recherche.

`alternate-fieldh`

Optionnel mais fortement recommandé. Le nom du champ dupliqué alternatif à rechercher si l'utilisateur renseigne au moins deux critères de recherche.

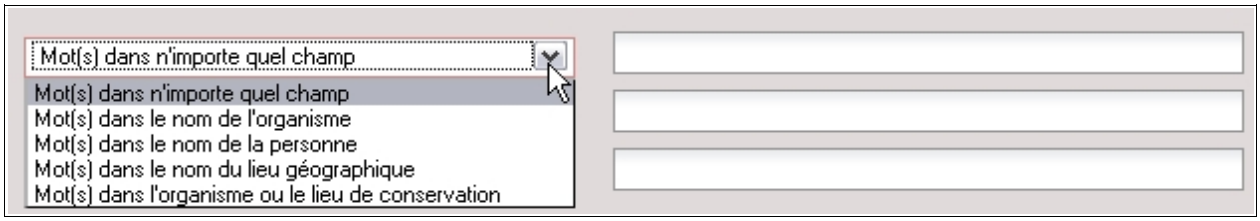
© 2010 AJLSM

`alternate-field-checked`

Optionnel, la valeur par défaut est `false`. Si la valeur est `true`, l'option pour le champ alternatif sera déjà sélectionnée au moment d'afficher le formulaire.

À noter que les **champs alternatifs sont supportés uniquement pour les critères de type texte** dans Pleade.

On peut également omettre l'attribut `field` et plutôt proposer à l'utilisateur de choisir un champ de recherche parmi une liste.



Pour ce faire, il faut donc ne pas renseigner l'attribut `field` mais plutôt des sous-éléments `field` tel que dans cet exemple :

```
<key type="text" ...>
  <field name="scopecontent"/>
  <field name="unittitle" default="true"/>
  <field name="fulltext"/>
</key>
```

Dans cet exemple, trois champs seront proposés à l'utilisateur, avec une liste déroulante située à gauche de la zone de texte. La seconde valeur sera pré-sélectionnée, grâce à la présence de l'attribut `default` avec la valeur `true`.

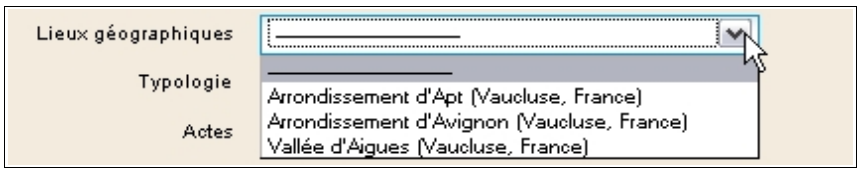
À noter que si vous utilisez cette possibilité de laisser à l'utilisateur le choix du champ, l'attribut `fieldh` n'est pas utilisé et il n'est actuellement pas possible de faire l'équivalent.

4.4.2.4 Listes (`list`)

Les listes sont destinées à présenter à l'utilisateur une liste de choix. Il peut alors sélectionner une ou plusieurs valeurs et effectuer une recherche sur ces valeurs. Les listes peuvent être présentées de différentes façons, et c'est la valeur de l'attribut `variant` qui va le déterminer :

`combo`

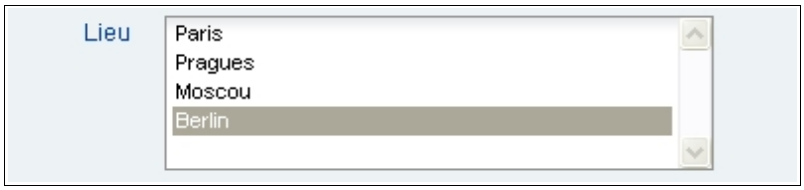
Une liste déroulante. Dans la liste, l'utilisateur ne peut sélectionner qu'une valeur, mais Pleade offre la possibilité d'ajouter d'autres valeurs.



`fixed`

Une liste à hauteur fixe, c'est-à-dire avec un nombre prédéfini de valeurs affichées. On peut directement sélectionner plusieurs valeurs.

© 2010 AJLSM



checkbox

Des cases à cocher, donc chaque item de la liste est accompagné d'une case à cocher. On peut sélectionner plusieurs valeurs.



radio

Des boutons radio, donc chaque item de la liste est accompagné d'un bouton radio. On ne peut sélectionner qu'une seule valeur.

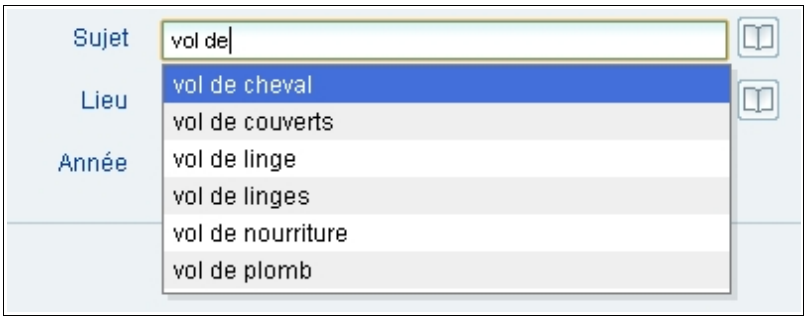


En général, les valeurs de la liste seront automatiquement renseignées par Pleade à partir des valeurs que l'on trouve dans le champ indiqué par l'attribut `field`.

On peut également pour les listes, contrôler l'opérateur intra-champ (voir 4.4.3.2 Choix de l'opérateur intra-champ)

4.4.2.5 Suggestions (`suggest`)

Les suggestions sont un cas particulier de liste : toutes les valeurs ne sont pas présentées directement à l'utilisateur, celui-ci va plutôt saisir quelques lettres et la liste sera filtrée automatiquement avec des entrées qui contiennent ces lettres, sans tenir compte des accents et des majuscules/minuscules.



Pour avoir un tel critère, il s'agit d'utiliser l'attribut `type` avec une valeur `suggest`. Par défaut, l'utilisateur peut sélectionner plusieurs valeurs, mais si l'on souhaite qu'il ne sélectionne qu'une seule valeur on peut ajouter un attribut `multiple="false"`.

```
<key type="suggest" field="fgeogname" help="true" multiple="false"/>
```

On peut également contrôler l'opérateur intra-champ (voir 4.4.3.2 Choix de l'opérateur intra-champ).

4.4.2.6 Booléens (`boolean`)

Un critère booléen est un critère qui ne peut prendre que deux valeurs, vrai ou faux. Il se présente sous la forme d'une boîte à cocher. On utilise généralement ce critère pour filtrer la recherche sur un attribut particulier : par exemple, les documents illustrés, limiter la recherche aux articles, etc.

Pour un tel critère, on peut ajouter les attributs particuliers suivants :

© 2010 AJLSM



value-on

Obligatoire pour utiliser ce critère. Il s'agit de la valeur que doit avoir le champ indiqué dans l'attribut `field`.

default

Si la valeur de cet attribut est `on`, la case sera cochée par défaut.

```
<key type="boolean" field="fgeogname" value-on="Afrique" default="on" connector="3" />
```

4.4.2.7 Dates (`date`)

La recherche par dates est traitée de manière particulière dans Pleade et fait l'objet d'un type de critère bien particulier. Il n'est pas nécessaire de préciser le champ de recherche car Pleade va toujours utiliser les mêmes champs associés aux dates extrêmes (`bdate` et `edate`). De plus, il n'est pas nécessaire d'utiliser le mécanisme des champs dupliqués et de l'attribut `fieldh` car les dates sont déjà héritées par défaut dans Pleade.



Donc pour effectuer une recherche par dates, on doit utiliser une syntaxe telle que celle-ci :

```
<key
        type="date"
        id="annee"
        variant="exact interval"
        max-chars="4"
        format="(\d)+"
        help="true"
/>
```

Les attributs `type`, `format`, `max-chars` et `help` ont déjà été évoqués, les autres attributs sont :

id

Obligatoire. L'identifiant du critère, utilisé pour le lier avec les clés `i18n` pour la génération des textes.

variant

Obligatoire. Des précisions sur les zones de texte à saisir. Si cet attribut contient `exact`, une recherche d'une date exacte sera proposée. Si cet attribut contient `interval`, une recherche par intervalle sera proposée. On peut bien sûr obtenir les deux en indiquant les deux mots, séparés par une espace comme dans l'exemple ci-dessus.

4.4.3 Fonctions avancées

4.4.3.1 Troncature implicite

Une troncature permet de faire une recherche sur une partie d'un mot. Par exemple, une troncature à droite sur le mot "bois" s'écrit "bois*". La recherche s'effectue alors sur tous les mots commençant par "bois".

Une troncature implicite peut être ajoutée aux éléments de formulaire de type `text`. Pour ajouter la troncature automatiquement à la validation du formulaire, il suffit d'ajouter l'attribut `truncate`.

truncate

Permet de déterminer où sera ajoutée la troncature implicite. Cet attribut peut prendre les valeurs `left`, `right` ou `both` :

```
<key type="text" field="fulltext" truncate="left"/>
```


© 2010 AJLSM

```
<key type="text" field="fulltext" truncate="right"/>
<key type="text" field="fulltext" truncate="both"/>
```

4.4.3.2 Choix de l'opérateur intra-champ

Par défaut, l'opérateur intra-champ pour tous les éléments de formulaire qui ne sont pas de type `list` ou `suggest`, est l'opérateur `AND`. Pour les éléments de type `list` et `suggest`, l'opérateur intra-champ par défaut est l'opérateur `OR`.

Il est possible de changer ces valeurs par défaut au moyen de l'attribut `op` dans les éléments `key`.

`op`

Cet attribut peut prendre les valeurs `AND` ou `OR`. Il permet de définir l'opérateur intra-champs.

`op-display`

Permet de gérer l'affichage de ces opérateurs dans l'interface. Il peut prendre les valeurs `true` ou `false`. Dans le cas des éléments de formulaire de type `list` avec les attributs `variant="fixed"` et `selection="multiple"`, ainsi que pour les éléments de type `suggest` multiples, les opérateurs s'afficheront lorsqu'il y a plus de deux sélections.

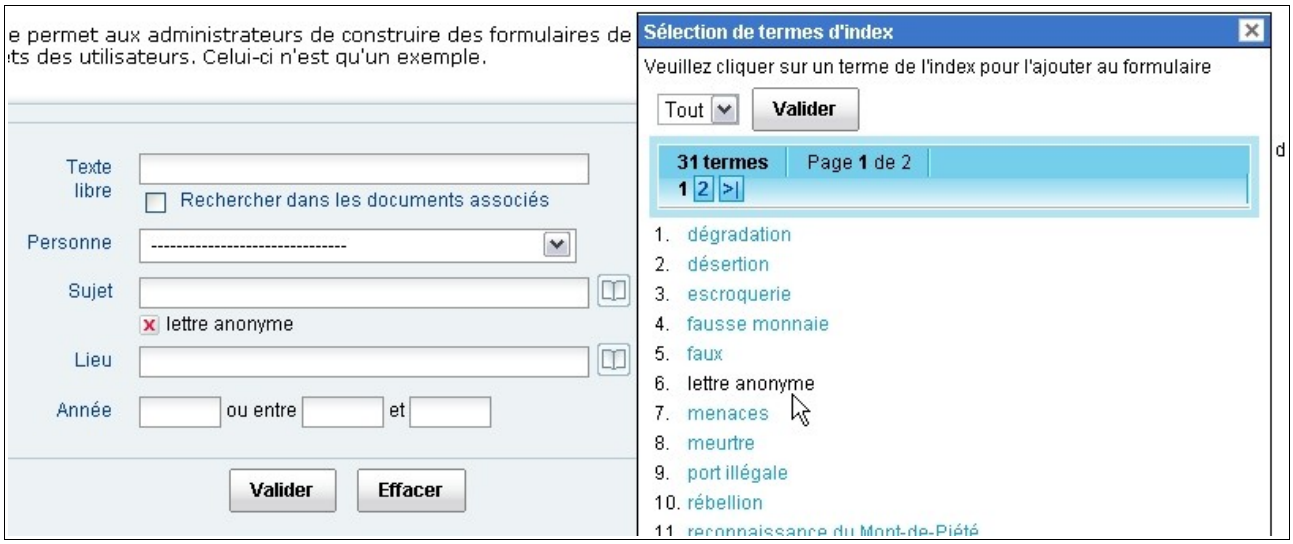


Exemple :

```
<key type="suggest" field="fgeogname" style="background:#00CCFF" help="true" op-display="true"/>
<key type="suggest" field="fgeogname" style="background:coral" op-display="true" op="AND"/>
<key type="list" variant="combo" rows="5"
      selection="multiple" op-display="true"
      display-field="fatitle" field="root-id">
      <no-choice i18n-key="form-no-choice-choix"/>
</key>
```

4.4.3.3 Accès au index

Pour chaque critère, il est possible d'ajouter un bouton permettant de parcourir dans un panneau, la liste des termes indexés, associés au champ de la suggestion.



Pour cela, il suffit d'ajouter :

© 2010 AJLSM

terms-browse

Optionnel. Permet d'activer l'aide à la recherche de type "navigation d'index" ; c'est-à-dire une icône placée à droite du critère qui déclenche l'affichage d'un panneau listant tous les termes indexés dans un champ de recherche. Cet attribut peut avoir plusieurs valeurs :

- `true` ou `1` : cette aide est activée et l'index consulté est celui indiqué comme valeur de l'attribut `field`
- vide ou `false` ou `0` : cette aide n'est pas activée. Même chose si cet attribut est absent.
- contient le nom d'un champ autre que celui indiqué comme valeur de l'attribut `field` : cette aide est activée et l'index consulté est celui indiqué comme valeur d'attribut.

terms-browse-hpp

Optionnel. Cet attribut permet de préciser le nombre d'entrées par page proposées dans la fenêtre de consultation de l'index.

4.4.3.4 Accès à un thesaurus

Cette partie ne concerne que les applications Pleade disposant du module Thesaurus.

Pour chaque critère dont le champ de recherche est lié à un vocabulaire contrôlé, il est possible d'ajouter un bouton permettant de consulter ce vocabulaire s'il est géré par ThesX.



Pour cela, il suffit d'ajouter :

thesx-browse

Optionnel. Permet d'activer l'aide à la recherche de type « consultation de thesaurus » dans un panneau ; c'est-à-dire une icône placée à droite du critère qui déclenche l'affichage d'un panneau permettant de naviguer dans le thesaurus. Cet attribut peut avoir plusieurs valeurs :

- `true` ou `1` : cette aide est activée et le thesaurus consulté est celui dont l'identifiant correspond à celui indiqué comme valeur de l'attribut `thesx-id`
- vide ou `false` ou `0` : cette aide n'est pas activée. Même chose si cet attribut est absent.

thesx-id

Optionnel mais obligatoire si l'attribut `thesx-browse` active l'aide. Il s'agit de l'identifiant du thesaurus géré par ThesX que l'on veut consulter.

thesx-relations

Optionnel. Cet attribut permet de préciser selon quel type de relations on veut étendre la recherche. Par défaut la recherche est étendue au terme sélectionné et à tous ses spécifiques (relation TS).

© 2010 AJLSM

4.4.3.5 Filtre

Il est possible à l'aide d'un critère ou plusieurs critères cachés de limiter la recherche à l'aide d'un formulaire de recherche à une rubrique (voir section suivante 5) Rubriques), à une liste d'instruments de recherche, à une valeur particulière pour un champ. Dans ce cas, si jamais le formulaire de recherche propose des listes ou des aides à la recherche comme un accès aux index, il faut que ces listes soient elles-aussi filtrées par les mêmes critères.

Attention : Pour le moment, les filtres ne fonctionnent pas avec les critères de type Suggestions.

L'usage des filtres peut remplacer les rubriques.

Pour filtrer un critère de recherche, on doit utiliser une syntaxe telle que :

```
<key type="list" ...>
     <filters>
        <filter field=""
                value=""/>
     </filters>
  </key>
```

L'élément `filters` permet de définir un ensemble de filtre. Il contient des éléments `filter` : chaque filtre est exprimé par un élément `filter`. Cet élément peut prendre deux attributs :

`field`

Obligatoire. Le champ qui permet de filtrer le critère.

Pour connaître la liste des champs définis par défaut dans Pleade, se reporter au manuel Gestion de contenu.

`value`

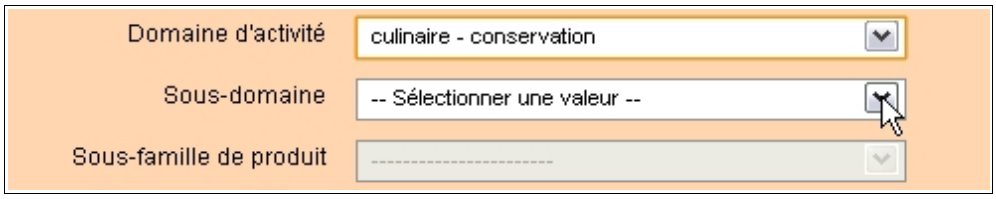
Obligatoire. La valeur que doit avoir le champ pour le filtre.

Voilà l'exemple d'un filtre sur un type de documents :

```
<filters>
     <filter field="fgenreform" value="dessins"/>
</filters>
```

4.4.3.6 Listes dépendantes

Une liste peut dépendre d'une autre c'est-à-dire que son contenu peut changer en fonction de la valeur sélectionnée dans l'autre liste. Par exemple, dans un formulaire pour des documents d'état civil, on pourrait vouloir que la liste des types de registres se mette à jour en fonction de la commune sélectionnée.



Pour obtenir ce fonctionnement, il faut utiliser l'attribut :

`updateSelect`

Placé au niveau du critère de type liste source (celle dont la sélection d'une entrée va mettre à jour la liste cible ou liste dépendante), sa valeur doit être le nom du champ (attribut `field`) de la liste cible (celle dont le contenu va changer).Dans notre exemple de formulaire d'état civil, sa valeur serait le nom du champ correspondant aux types de registres.

De plus, au chargement du formulaire, deux comportements sont possibles :

- le critère cible est rempli normalement, dans ce cas en général une indication invite l'utilisateur à choisir une entrée
- le critère cible est vide et ne contient qu'une indication invitant l'utilisateur à choisir une entrée dans la liste source.

© 2010 AJLSM

```
<key type="list" field="insee" updateSelect="type" variant="combo" [...]>
  {champ source}
</key>
<key type="list" field="type" variant="combo" [...]>
  {champ cible)

  Solution 1 : au chargement de la page, le critère est "rempli" normalement,
               son premier item invite l'utilisateur à sélectionner une valeur.
  <no-choice i18n-key="form-no-choice-choix"/>

  Solution 2 : au chargement de la page, le critère est vide et ne contient
               qu'une unique valeur : un message invitant l'utilisateur à
               sélectionner une valeur dans le champ source.
  <items>
    <item value="" i18n-key="form-no-choice-since-insee-choix" />
  </items>
</key>
```

4.4.3.7 Styles

Dans chaque élément `key` il est possible de définir un attribut `style` dont la valeur doit être valide par rapport au contenu du même attribut dans la norme XHTML, soit des règles CSS.



`style`

Permet de surcharger le style CSS de l'élément de formulaire.

```
<key type="text" style="width:10em;" field="fulltext"/>
```

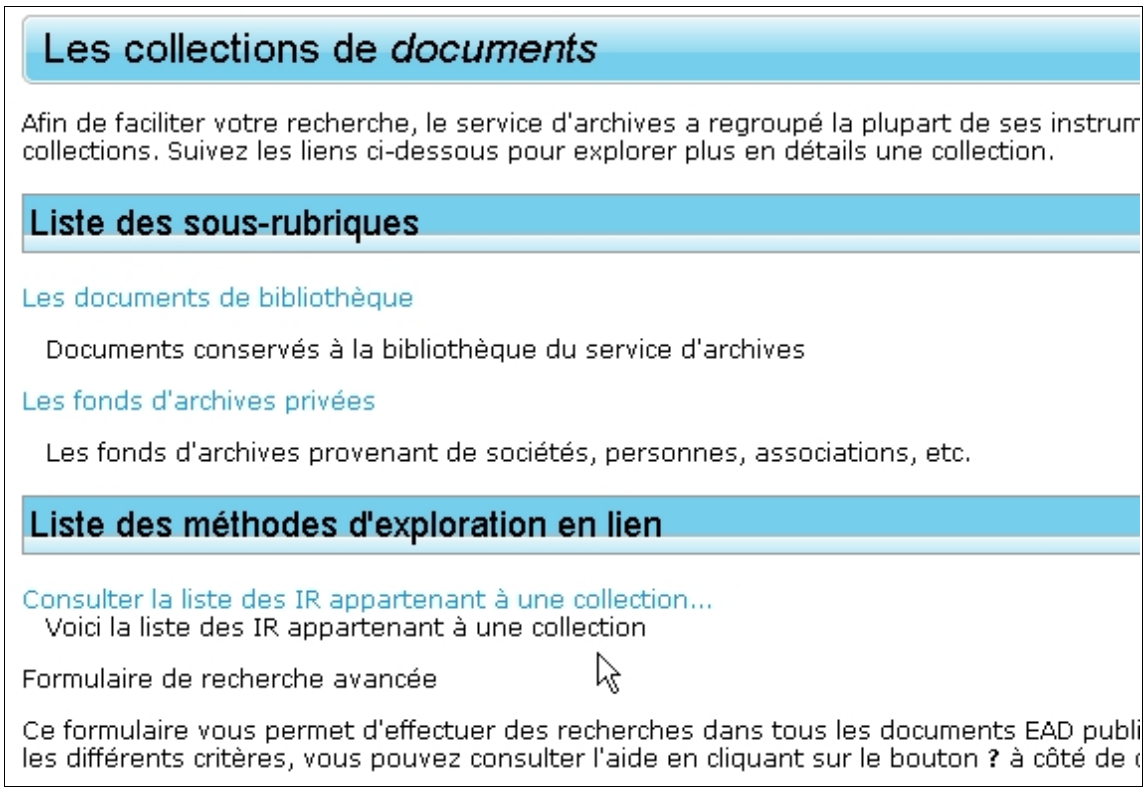
5) Rubriques

Pour organiser les documents EAD de manière à faciliter leur exploration dans le site Pleade, il est possible de créer des rubriques et sous-rubriques, dans lesquelles pourront être placés les documents.

L'affichage d'une rubrique se présente sous la forme d'une page Web avec selon sa configuration : du texte, un formulaire de recherche (la recherche s'effectue uniquement dans les documents EAD associés à la rubrique), la liste des documents EAD ou des unités de recherche associés à cette rubrique, des liens vers un formulaire de recherche ou vers la liste des documents, la hiérarchie des sous-rubriques, etc.

On peut afficher dans Pleade une rubrique avec une URL qui utilise le nom du fichier de la rubrique. Par exemple, pour afficher la rubrique définie dans le fichier `subsets-default.xconf`, il faut utiliser l'URL `subset.html?name=subsets-default`. Donc dans le paramètre `name`, on indique le nom du fichier de la rubrique, mais sans son extension. Tous les fichiers de configuration de rubriques sont dans le répertoire `module-eadeac/conf/subsets/`.

© 2010 AJLSM



5.1 Approche générale

Pleade permet d'organiser les documents avec un système de rubriques à plusieurs niveaux. Chaque rubrique de premier niveau est définie dans un document XML contenant les caractéristiques générales de cette rubrique. Ce document peut contenir également des définitions de sous-rubriques et leurs paramètres, selon le même schéma imbriqué.

Chaque rubrique ou sous-rubrique est définie par un élément `subset`. Cet élément va contenir la description de la rubrique qui se fait à l'aide d'un nom, d'un en-tête, d'un résumé, d'un ensemble de zones ainsi que d'une description. Chaque élément étant susceptible d'être utilisé dans un contexte différent, il est essentiel de tous les renseigner. La structure générale de cet élément `subset` est la suivante :

```
<subset id="">
  <name xml:lang="">Nom de la rubrique</name>
  <header xml:lang=""></header>
  <summmary xml:lang=""></summary>
  <description xml:lang=""></description>
  <zones>
     <zone role="subsets">
        <header xml:lang="fr">Liste des sous-rubriques</header>
     </zone>
     <zone role="links">
        <header xml:lang="fr">Liste des méthodes d'exploration en lien</header>
     </zone>
  </zones>
  <subset id="">
     Une sous-rubrique
  </subset>
  ...
</subset>
```

Certains éléments comme l'en-tête, le résumé et la description peuvent contenir du contenu XHTML de type *bloc*, comme des paragraphes.

L'usage de ces éléments descriptifs est le suivant :

© 2010 AJLSM

name

Obligatoire. Le nom est utilisé dans différents contextes pour identifier très sommairement la rubrique, par exemple dans le titre des pages HTML (affiché dans la barre de titre du navigateur Web), ou encore dans le formulaire de publication pour permettre d'associer le document à publier avec une ou plusieurs rubriques.

header

L'en-tête est utilisé pour *titrer* le contenu d'une page qui présente la rubrique. C'est le cas, par exemple, lorsqu'on se trouve sur la page de présentation de la rubrique, où le contenu de l'en-tête apparaît en tout début de contenu.

summary

Le résumé est utilisé pour présenter brièvement une rubrique avant que l'on affiche directement sa page. Par exemple, lorsqu'on affiche la page de la rubrique de premier niveau, le résumé (et l'en-tête) des sous-rubriques sont affichées, afin de guider l'utilisateur vers celles-ci.

description

La description est affichée dans la page principale de la rubrique, afin de bien expliquer à l'utilisateur ce qu'il trouvera, ce qu'il pourra faire, etc. La description suit immédiatement l'en-tête dans la page.

zone

Cet élément permet de donner des intitulés à différentes zones de rubriques (via le sous-élément `zone` et son attribut `role`) présentes dans la page. L'attribut `role` peut avoir les valeurs suivantes : `subsets` (zone d'affichage des sous-rubriques) ou `links` (zone d'affichage des méthodes d'exploration).

Les éléments `name`, `header`, `summary` et `description` ont un contenu de type éditorial que l'on pourrait vouloir proposer en plusieurs langues. Il suffit donc de créer autant de ces éléments que de langues souhaitées et pour chacun d'indiquer à l'aide de l'attribut XML `xml:lang` la langue du contenu. Pour indiquer la langue, utiliser les codes ISO sur deux caractères : fr pour le français, en pour l'anglais, de pour l'allemand, etc.

```
<name xml:lang="fr">Bibliothèques</name>
<name xml:lang="en">Libraries</name>
```

Après la description d'une rubrique, on doit aussi lui associer des méthodes d'exploration, pour que Pleade puisse construire les liens de navigation ou de recherche pertinents. À noter que ces méthodes sont pertinentes pour les rubriques de premier ou de second niveau. Si une rubrique de premier niveau contient à la fois des méthodes d'exploration et des sous-rubriques, alors les sous-rubriques seront présentées en premier, suivies des méthodes d'exploration.

Il y a essentiellement trois méthodes d'exploration, et les trois peuvent être utilisées conjointement :

Formulaires de recherche avancée

Il est possible d'associer un ou plusieurs formulaires de recherche avancée à une rubrique. S'il y a un seul formulaire, il peut être présenté dans la même page que la rubrique, ou encore sur une page suivante. S'il y a plusieurs formulaires, un formulaire peut être présenté sur la page de la rubrique, et un menu déroulant permet de faire afficher les autres formulaires.

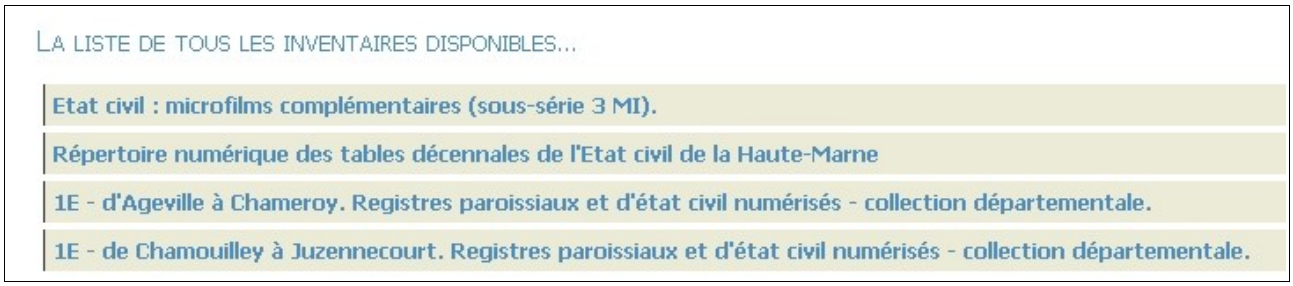
Pour cette méthode, utiliser l'élément `forms`.

Liste des instruments de recherche

La liste des instruments de recherche peut être affichée, soit directement dans la page de présentation de la rubrique, soit par un lien. De plus, le nombre d'instruments affichés par page peut être contrôlé.

Pour cette méthode, utiliser l'élément `list-document`.

© 2010 AJLSM



Liste des unités documentaires

La liste des unités documentaires peut également être affichée, avec le même type de contrôle que pour la liste des instruments de recherche.

Pour cette méthode, utiliser l'élément `list-units`.

L'exemple ci-dessous indique comment spécifier ces différentes méthodes :

```
<subset id="bib">
  <!-- Les éléments descriptifs ne sont pas repris ici... -->
  <forms>
     <form ref="form-bib" display="in-page" display-information="true">
        <title xm:lang="fr">Le titre du formulaire</title>
        <introduction xm:lang="fr">L'introduction du formulaire</introduction>
     </form>
     <form ref="form-bib-2" display="link"/>
  </forms>
  <list-documents display="link|in-page" hpp="20">
     <title xml:lang="fr">Consulter la liste des documents associés</title>
     <title xml:lang="en">Browse the list of related documents</title>
     <introduction xml:lang="fr">Description d'une  recherche</introduction>
  </list-documents>
  <list-units display="link|in-page" hpp="20">
     <title xml:lang="fr">Consulter la liste des unités associées</title>
     <title xml:lang="en">Browse the list of related units</title>
     <introduction xml:lang="fr">Description d'une  recherche</introduction>
  </list-units>
</subset>
```

L'usage de ces éléments est le suivant :

`forms`

L'élément `forms` permet d'indiquer qu'un ou des formulaires de recherche doivent être présentés.

`form`

Cet élément référence via l'attribut `ref` le nom du formulaire à afficher. Voir la rubrique 4) Formulaires de recherche pour savoir comment créer un formulaire.

`list-documents`

Cet élément permet de demander l'affichage d'une liste des documents EAD associés à cette rubrique au moment de la publication.

`list-units`

Cet élément peut avoir les mêmes paramètres que l'élément `list-documents`, mais cette fois il permet d'afficher la liste des unités documentaires des documents EAD associés à cette rubrique au moment de la publication. L'ordre d'affichage est l'ordre alphabétique des titres des unités. Cette liste est présentée comme un résultat de recherche, donc selon le cadre de classement.

Les sous-éléments communs aux éléments `form`, `list-units`, `list-documents` sont décrits ici:

`title`

Dans le cas où un lien vers la méthode est affichée, c'est le contenu de cet élément qui est utilisé comme contenu du lien.

© 2010 AJLSM

`introduction`

Cet élément permet d'afficher une information supplémentaire sur la recherche ou un formulaire.

Leurs attributs sont décrits ici:

`display`

La valeur de cet attribut définie comment et où sera affichée la méthode d'exploration. Il peut prendre les valeurs `link` ou `in-page`. Dans le premier cas, un lien vers la méthode d'exploration sera affichée. Dans le deuxième cas, la méthode sera affichée dans la page. Si plusieurs méthodes ont le même affichage, elles seront affichées les unes à la suite des autres.

Il s'agit d'un attribut commun aux éléments `form`, `list-units`, `list-documents`.

`display-information`

Cet attribut permet d'indiquer si l'on souhaite afficher ou non le titre et la description associés au formulaire. Il peut prendre la valeur « `true` » ou « `false` ». Sa valeur par défaut est '`true`'. Les éléments `title` et `introduction` permettent éventuellement de surcharger ceux du formulaire.

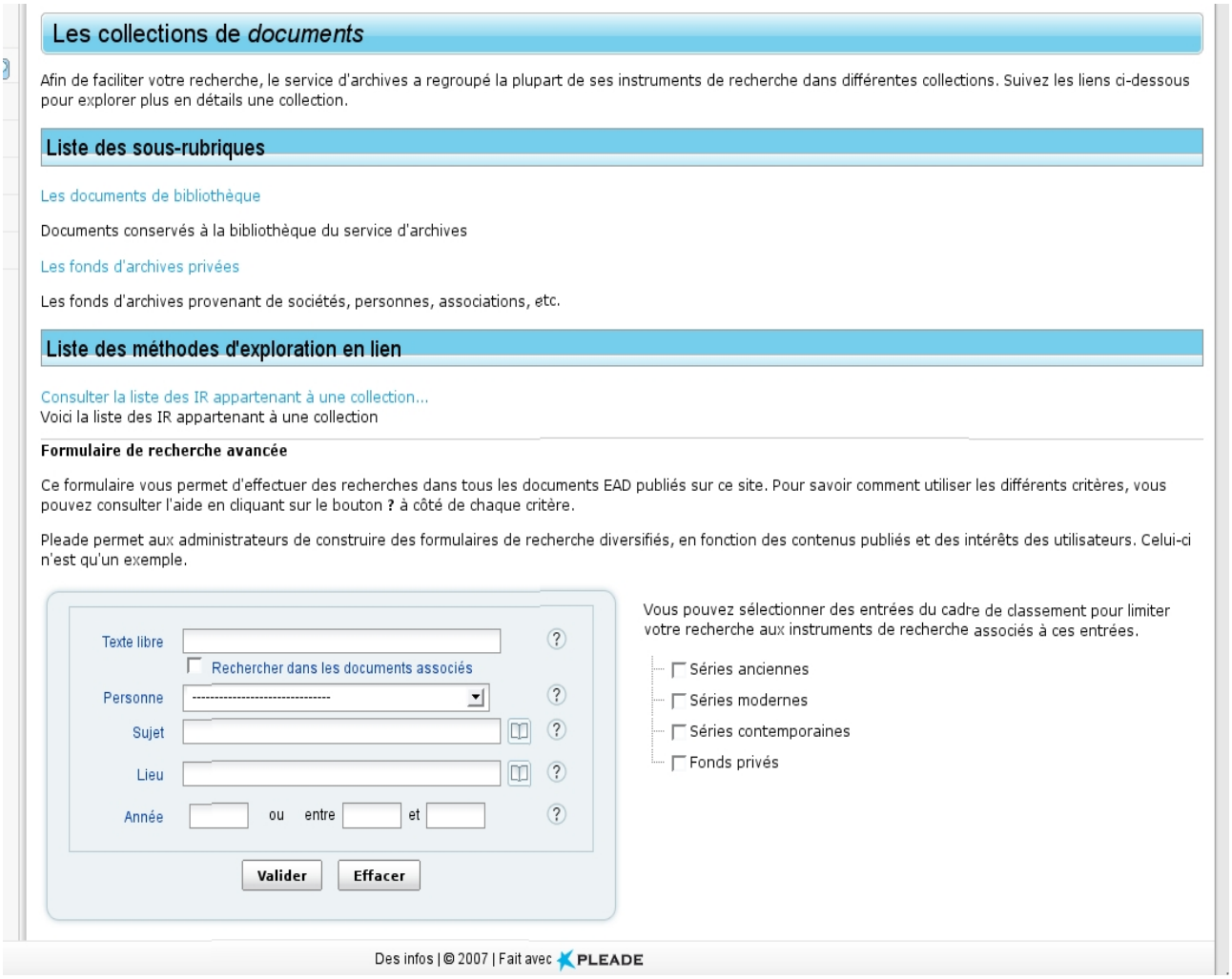
Il s'agit d'un attribut commun aux éléments `form`.

`hpp`

Cet attribut permet de contrôler le nombre de documents qui seront affichés par page à l'aide de l'attribut `hpp`, qui peut prendre un nombre entier positif quelconque. À noter que pour indiquer *tous les documents en une seule page*, il faut utiliser la valeur `-1`.

Il est utilisé pour les éléments `list-units` et `list-documents`.

© 2010 AJLSM



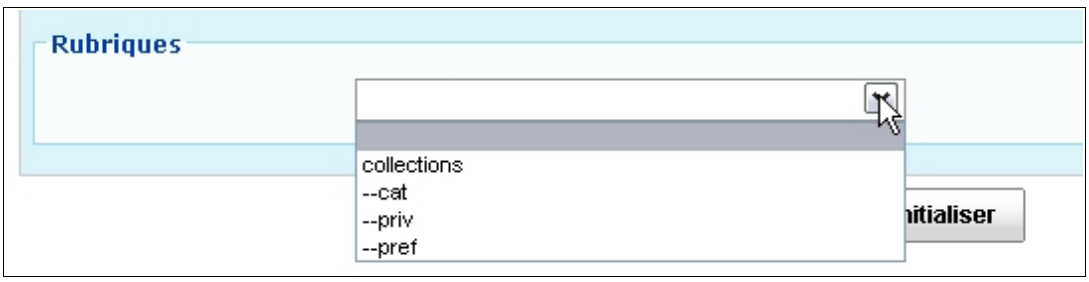
Exemple d'un affichage d'une rubrique

5.2 Publication

Une fois créées, les rubriques devront être associées à des documents EAD au moment de leur publication dans l'onglet « *Paramètres de publication* » du formulaire de publication.

La valeur du champ « *Rubriques* » de ce formulaire est à choisir dans la liste déroulante qui présente de manière hiérarchique les rubriques et sous-rubriques définies dans Pleade.

Lors de la publication d'un document EAD, il est possible de l'associer à une rubrique ou une sous-rubrique. Dans le cas d'une sous-rubrique, les unités documentaires du document publié seront visibles dans la rubrique parente. Ainsi, la rubrique de premier niveau hérite du contenu associé à toutes ses sous-rubriques.



© 2010 AJLSM

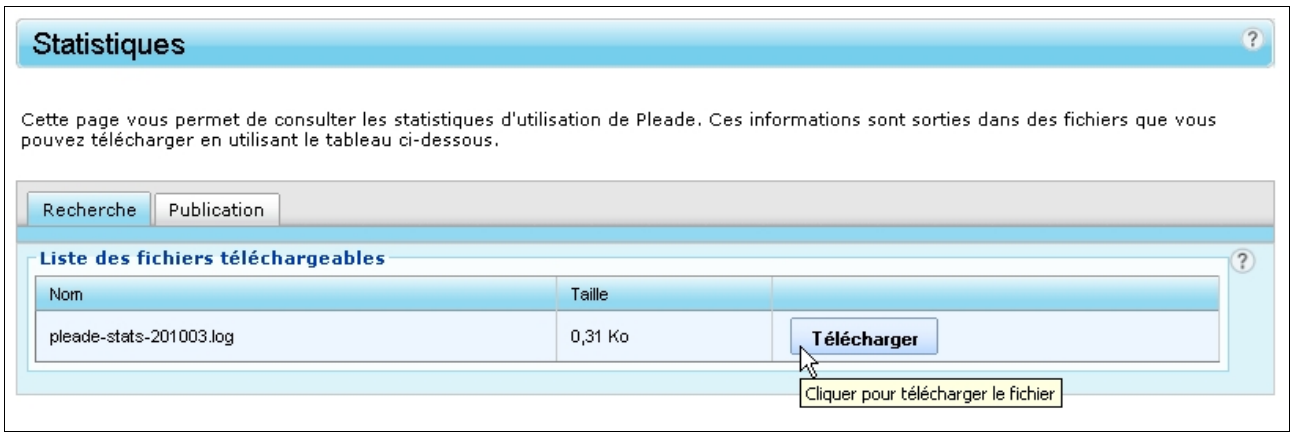
6) Statistiques

6.1 Présentation

La section « *Statistiques* » vous permet de consulter les statistiques d'utilisation et les journaux de publication de Pleade.

6.1.1 Statistiques de recherche

Ces statistiques contiennent des informations sur les recherches effectuées via les formulaires de recherche. Ces informations sont écrites dans des fichiers texte que vous pouvez récupérer.



Un fichier de statistiques par mois est automatiquement généré. Le nom du fichier se présente sous la forme `pleade-stats-{année}{mois}.log`, où `{année}` et `{mois}` sont l'année et le mois d'enregistrement des statistiques en valeur numérique.

Exemple : `pleade-stats-200806.log`

Pour récupérer un fichier de statistique pour un mois donné, vous devez simplement cliquer sur le bouton « *Télécharger* » qui lui correspond. Selon la configuration de votre navigateur Internet, il vous proposera d'enregistrer le fichier ou de l'ouvrir avec un éditeur de texte.

Remarque : le contenu de ces fichiers est écrit avec l'encodage UTF-8.

Dans le fichier, une ligne correspond à une recherche enregistrée. Une ligne contient un certain nombre de valeurs séparées par des tabulations.

Exemple :

```
2008-06-06 12:48:56 127.0.0.1       ead     ead-default    2       1       2       ( Texte libre :
Louis ) ET ( Lieu : ALSACE )
```

Voici la liste des valeurs enregistrées pour chaque ligne :

- Date (`2008-06-06`)
- Heure (`12:48:56`)
- Adresse IP de l'ordinateur qui a effectué la requête (`127.0.0.1`)
- Base de documents sur laquelle est effectuée la recherche (`ead`)
- Formulaire de recherche utilisé (`ead-default`)
- Nombre de critères de recherche renseignés avec au moins une valeur (`2`)
- Nombre de documents trouvés (`1`)
- Nombre d'unités de description trouvées (`2`)

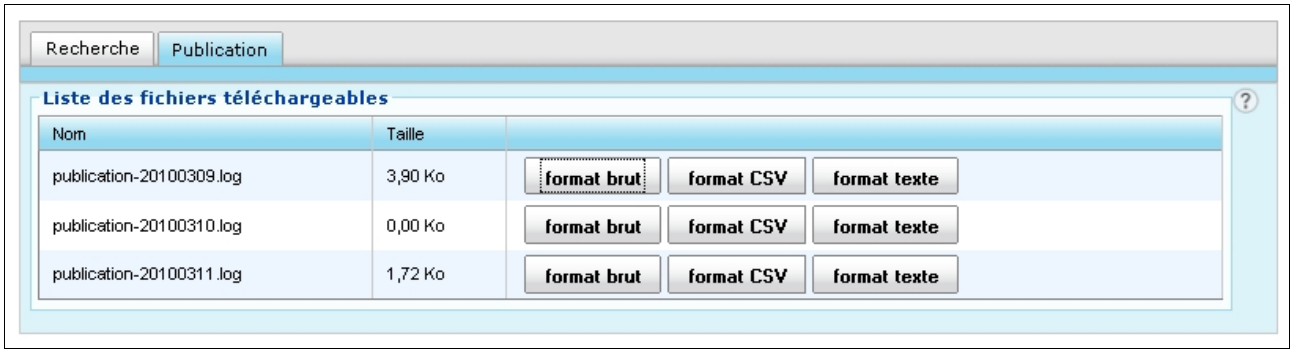
© 2010 AJLSM

- Expression textuelle de la requête de recherche, telle qu'affichée dans l'historique des résultats de recherche (`( Texte libre : Louis ) ET ( Lieu : ALSACE )`)

6.1.2 Journaux de publication

Ces journaux contiennent les rapports sur les publications effectuées soit via l'interface d'administration, soit de manière automatique. Ces informations sont écrites dans un fichier `*.log` que vous pouvez récupérer dans 3 formats :

- ***format brut*** : présente le journal tel qu'il apparaît sur le serveur. Il ne subit aucune transformation ; les données de chaque publication sont présentées par informations (i.e., plusieurs lignes pour chaque publication).
- ***format texte*** : mise en forme du journal de manière à présenter les informations par publication (ie, un seul paragraphe par publication).
- ***format CSV*** (Tableur) : propose le journal dans un format propre à être ouvert avec un tableur tel que Microsoft Excel, OpenOffice Calc, GNUmeric, etc. Les données sont présentées par document publié.



Les journaux sont quotidiens. Un journal vide indique qu'il n'y a pas eu de publication pour le jour indiqué. Le nom du fichier se présente sous la forme `pleade-stats-{année}{mois}{jour}.log`, où `{année}`,, `{mois}` et `{jour}` sont l'année, le mois et le jour d'enregistrement des statistiques en valeur numérique.

Exemple : `publication-20100309.log`

Pour récupérer un journal pour un jour donné dans un format particulier, vous devez simplement cliquer sur le bouton de téléchargement correspondant au format souhaité. Selon la configuration de votre navigateur Internet, il vous proposera d'enregistrer le fichier ou de l'ouvrir avec un éditeur de texte.

Remarque : le contenu de ces fichiers est écrit avec l'encodage UTF-8.

7) Pages statiques

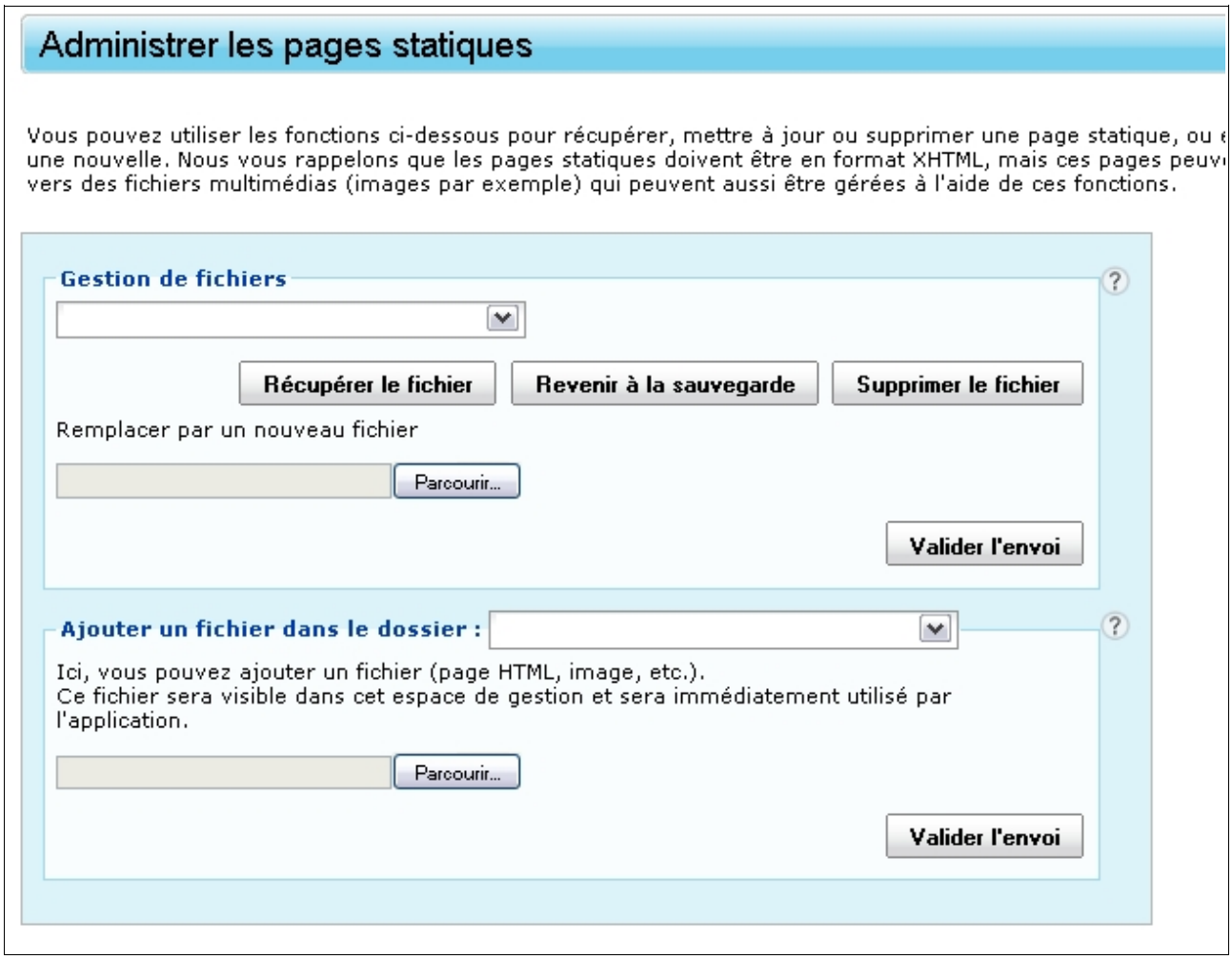
Les pages statiques sont des pages de l'application qui ne présentent aucun caractère dynamique spécifique (affichage d'un document, résultats de recherche, etc). Par exemple une page d'accueil est généralement une page statique.

Vous pouvez récupérer, mettre à jour ou supprimer une page statique, ou encore en ajouter une nouvelle. Les pages statiques doivent être en **format XHTML**, mais ces pages peuvent contenir des liens vers des fichiers multimédias (images par exemple) qui peuvent aussi être gérés à l'aide de l'interface d'administration.

On peut afficher dans Pleade une page statique avec une URL qui utilise le nom du fichier de la page. Par exemple, pour afficher la page statique définie dans le fichier `maPage.html`, il faut utiliser l'URL `pages/maPage.html`.

La page d'accueil (fichier `index.html`) est un cas particulier, elle est appelée directement par l'URL `/`.

© 2010 AJLSM



7.1 Gestion de fichiers

Ce panneau vous permet de gérer les pages statiques et leurs fichiers associés qui sont actuellement visibles dans l'application.

1. Vous devez d'abord sélectionner un fichier dans la liste déroulante.
2. Ensuite, vous pouvez effectuer l'une des actions suivantes, en cliquant sur le bouton correspondant :
   - ***Récupérer le fichier*** : permet de télécharger une copie du fichier sur votre poste de travail, pour le modifier par exemple.
   - ***Revenir à la sauvegarde*** : à chaque fois que vous effectuez une mise à jour d'un fichier, Pleade conserve une copie de sauvegarde du fichier précédent. Vous pouvez revenir à cette copie de sauvegarde à l'aide de ce bouton.
   - ***Supprimer le fichier*** : pour supprimer le fichier, il ne sera donc plus accessible dans l'application.
   - ***Remplacer par un nouveau fichier*** : en utilisant le bouton *Parcourir...*, vous pouvez choisir un fichier sur votre poste de travail pour remplacer le fichier sélectionné sur le serveur, et ainsi le mettre à jour. Une copie de sauvegarde du fichier actuel sera conservée. Une fois que vous avez sélectionné votre fichier, vous devez cliquer sur le bouton « *Valider l'envoi* » pour le placer sur le serveur.

7.2 Ajouter un fichier dans le dossier

Vous pouvez ajouter un nouveau fichier (XHTML, image, etc.) en suivant la procédure suivante :

© 2010 AJLSM

1. sélectionner d'abord le nom du dossier dans lequel vous souhaitez le déposer
2. cliquer sur le bouton *Parcourir...* pour sélectionner le fichier à ajouter qui doit être accessible sur votre poste de travail.
3. cliquer sur le bouton « *Valider l'envoi* ».

À noter que le nom du fichier sera conservé, vous devez donc le nommer correctement, avec une extension **.html** et le nom du fichier correspondant à l'URL que vous souhaitez utiliser pour y accéder.

Remarque : depuis cette interface, il n'est pas possible d'organiser les ressources liées aux pages statiques dans des répertoires car on ne peut pas créer de nouveaux répertoires.

© 2010 AJLSM

Annexes

Dans les exemples de fichiers de configuration reportés ci-dessous, les commentaires sont mis en vert : les commentaires sont des textes qui apportent des informations complémentaires ou explicatives qui facilitent la lecture par une personne d'un fichier XML.

1. Fichier de configuration des champs « standards »

Le cadre ci-dessous présente le fichier `ead-fieldlist-base.xconf` de la version standard de Pleade. La personnalisation de votre application peut faire que le vôtre diffère.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Licence -->
<!-- Ce fichier de configuration fait partie de la distribution standard
de Pleade. Vous pouvez le modifier à votre guise. -->
<!--
    Fichier de configuration qui liste les champs de base, c'est-à-dire ceux
    qui ne sont pas issus d'un index. A noter que les attributs de l'élément
    racine (sdx:fieldList) sont conservés dans le application.xconf.
-->
<fieldList id="ead-fieldslist" xmlns="http://www.culture.gouv.fr/ns/sdx/sdx"
analyzerConf="/sdx/resources/conf/analysis/fr.xml" xml:lang="fr">

 <!-- Le champ par défaut -->
 <field name="fulltext" type="word" default="true"/>

 <!-- Les champs de type word qui correspondent à des éléments EAD portant une sémantique
(archivistique ou logique) -->
 <field name="abbr" type="word"/>
 <field name="abstract" type="word"/>
 <field name="accessrestrict" type="word"/>
 <field name="accruals" type="word"/>
 <field name="acqinfo" type="word"/>
 <field name="address" type="word"/>
 <field name="altformavail" type="word"/>
 <field name="appraisal" type="word"/>
 <field name="archref" type="word"/>
 <field name="arrangement" type="word"/>
 <field name="author" type="word"/>
 <field name="bibliography" type="word"/>
 <field name="bibref" type="word"/>
 <field name="bibseries" type="word"/>
 <field name="bioghist" type="word"/>
 <field name="container" type="word"/>
 <field name="controlaccess" type="word"/>
 <field name="custodhist" type="word"/>
 <field name="date" type="word"/>
 <field name="daodesc" type="word"/>
 <field name="did" type="word"/>
 <field name="dimensions" type="word"/>
 <field name="edition" type="word"/>
 <field name="expan" type="word"/>
```


© 2010 AJLSM

```
<field name="extent" type="word"/>
<field name="extref" type="word"/>
<field name="ext" type="word"/> <!--contenu des documents externes-->
<field name="extrefloc" type="word"/>
<field name="fileplan" type="word"/>
<field name="frontmatter" type="word"/>
<field name="imprint" type="word"/>
<field name="index" type="word"/>
<field name="indexentry" type="word"/>
<field name="langmaterial" type="word"/>
<field name="language" type="word"/>
<field name="legalstatus" type="word"/>
<field name="materialspec" type="word"/>
<field name="note" type="word"/>
<field name="num" type="word"/>
<field name="odd" type="word"/>
<field name="originalsloc" type="word"/>
<field name="origination" type="word"/>
<field name="otherfindaid" type="word"/>
<field name="physdesc" type="word"/>
<field name="physfacet" type="word"/>
<field name="physloc" type="word"/>
<field name="phystech" type="word"/>
<field name="prefercite" type="word"/>
<field name="processinfo" type="word"/>
<field name="publisher" type="word"/>
<field name="ref" type="word"/>
<field name="refloc" type="word"/>
<field name="relatedmaterial" type="word"/>
<field name="repository" type="word"/>
<field name="scopecontent" type="word"/>
<field name="separatedmaterial" type="word"/>
<field name="subarea" type="word"/>
<field name="unittitle" type="word"/>
<field name="userestrict" type="word"/>

<!-- Deux exceptions car par défaut ils sont en field -->
<field name="wunitdate" type="word"/>
<field name="wunitid" type="word"/>

<!-- Des champs associés à certains éléments EAD -->

<!-- Ceux qui sont retournés avec les résultats -->
<field name="unitid" type="field" brief="true"/> <!-- FIXME: toujours utile en brief? -->
<field name="forigination" type="field" brief="true"/>
<field name="level" type="field"/>
<field name="pacomm" type="field" brief="true"/>
<field name="pacommd" type="date"/>

<!-- Ceux qui ne sont pas retournés avec les résultats -->
<field name="frepository" type="field"/>
<field name="flegalstatus" type="field" />
<field name="fphysfacet" type="field" />
<field name="bdate" type="date"/>
```


© 2010 AJLSM

```
  <field name="edate" type="date"/>
  <field name="unitdate" type="field"/>

  <!-- Des champs de gestion nécessaires pour le bon fonctionnement de l'application -->
  <field name="top" type="field" brief="true"/>           <!-- Documents racines -->
  <field name="cdcroot" type="field" brief="true"/>       <!-- Entrée du cadre de classement d'un
document racine -->
  <field name="cdcall" type="field" brief="true"/>        <!-- Toutes les entrées du cadre de
classement, en remontant vers le haut -->
  <field name="cdc" type="field" brief="true"/>           <!-- Entrée du cadre de classement -->
  <field name="pos" type="field" brief="true"/>           <!-- Position dans le document EAD -->
  <field name="subsetall" type="field"/>                  <!-- L'ensemble des rubriques, en
remontant vers le haut -->
  <field name="subset" type="field"/>                     <!-- Rubriques -->
  <field name="ancobject" type="field"  brief="true"/>    <!-- Ancêtre d'une unité illustrée -->
  <field name="object" type="field"  brief="true"/>       <!-- Unité illustrée -->
  <field name="att" type="field"  brief="true"/>          <!-- Unité avec document attaché -->
  <field name="ctitle" type="unindexed" brief="true"/>    <!-- Le titre des ancêtres -->
  <field name="cid" type="unindexed" brief="true"/>       <!-- L'identifiant des ancêtres -->
  <field name="root-id" type="field" brief="true"/>       <!-- Le eadid -->
  <field name="fatitle" type="unindexed" brief="true"/>   <!-- Le titre de l'instrument de
recherche -->
  <field name="fucomptitle" type="unindexed" brief="true"/>      <!-- Le titre de l'unité de
description -->
  <field name="uunitid" type="unindexed" brief="true"/>          <!-- La cote pour le tri -->
  <field name="udate" type="unindexed" brief="true"/>            <!-- Les dates extrêmes, pour
les résultats -->
  <field name="type" type="field" brief="true"/>          <!-- Le type d'unité : ead, eadheader,
frontmatter, , div, component... -->
  <field name="searchable" type="field"/>         <!-- Si 0, l'unité n'est pas cherchable, si 1
oui -->
  <field name="dao_href" type="field"/>           <!-- [JC] Les liens pour les dao et daoloc -->
  <field name="parent_id" type="field"/> <!-- [JC] L'id du parent pour la pagination des enfants
d'un IR dans le fenêtre EAD -->

</fieldList>
```

2. Fichier de configuration des index

Le cadre ci-dessous présente le fichier `ead-fieldlist-base.xconf` de la version standard de Pleade. La personnalisation de votre application peut faire que le vôtre diffère.

Les déclarations d'index comprises entre les chaînes `<!--` et `-->` ne sont pas prises en compte car elles sont commentées selon la norme XML.

```
<?xml version="1.0"?>
<!-- Licence -->
<!-- Ce fichier de configuration fait partie de la distribution standard
de Pleade. Vous pouvez le modifier à votre guise. -->
<!-- Définition des index pour cette instance -->
<index-list>
  <!-- Des index destinés à intercepter certains descripteurs, qui ne doivent pas
       "polluer" d'autres index -->
  <!-- Pour les producteurs, on peut soit intercepter le descripteur provenant de la zone
producteur, soit l'utiliser pour construire un index. Suivant le cas, on commentera l'un ou
l'autre des index dans-origination ou producteur (LC 20090521) -->
```


© 2010 AJLSM

```
  <!-- Un autre cas est possible : les utilisateurs ont paramétré un descripteur spécifique
producteur indépendant de la zone de saisie. Il faut alors décommenter les deux (LC 20090521) -->
  <!-- Les corpname, persname, ... dans origination pour exclusion -->
  <index id="dans-origination">
     <content ancestor="origination"/>
  </index>
  <!-- Les auteurs de description dans Arkhéïa -->
  <index id="auteur-description">
     <content role="auteur_description"/>
  </index>

  <!-- Ci-dessous pour construire un index des producteurs. A décommenter si nécessaire et
commenter alors dans-origination  (LC 20090521)  -->
  <!-- <index id="producteur" local-sdxfield="lfproducteur" global-sdxfield="fproducteur" word-
sdxfield="producteur" display-order="10">
     <content role="producteur"/>
  </index>
  -->

  <!-- Des index plus spécifiques à Arkhéïa / Aide au classement -->
  <!-- Filtre : critère de recherche caché -->
  <index id="filtre" global-sdxfield="ffiltre">
     <content element="genreform" type="filtre"/>
  </index>
  <!-- Type d'instrument de recherche pour préciser si l'on veut chercher dans un IR
d'orientation ou un IR détaillé : critère caché ou pas (LC 20090521) -->
  <index id="typir" global-sdxfield="ftypir" word-sdxfield="typir">
     <content type="typir"/>
  </index>
  <!-- Grand thème de recherche ou typologie de document pour restreindre les réponses avec un
formulaire de recherche général (LC 20090521) -->
  <index id="theme" global-sdxfield="ftheme" word-sdxfield="theme">
     <content source="liste-theme"/>
  </index>
  <!-- Type document Aide au classement. Si AJLSM ne se sert pas de ce descripeur, il serait
souhaitable de l'intercepter. Dans ce cas, commenter ci-dessous et décommenter ce qui suit  (LC
20090521) -->
  <index id="typedocac" global-sdxfield="ftypedocac" word-sdxfield="typedocac">
     <content source="liste-typedocAC"/>
  </index>
  <index id="batiment" local-sdxfield="lfbatiment" global-sdxfield="fbatiment" word-
sdxfield="batiment" display-order="60">
     <content role="batiment"/>
  </index>
  <index id="auteur" local-sdxfield="lfauteur" global-sdxfield="fauteur" word-sdxfield="auteur"
display-order="40">
     <content role="auteur"/>
  </index>
  <index id="twmatiere" local-sdxfield="lftwmatiere" global-sdxfield="ftwmatiere" word-
sdxfield="twmatiere" display-order="10">
     <content source="thésaurus_W_matière"/>
  </index>
  <index id="twaction" local-sdxfield="lftwaction" global-sdxfield="ftwaction" word-
sdxfield="twaction" display-order="20">
     <content source="thésaurus_W_action"/>
  </index>
```


© 2010 AJLSM

```
  <index id="twtypologie" local-sdxfield="lftwtypologie" global-sdxfield="ftwtypologie" word-
sdxfield="twtypologie" display-order="30">
     <content source="thésaurus_W_typologie_documentaire"/>
  </index>
  <index id="twcontexte" local-sdxfield="lftwcontexte" global-sdxfield="ftwcontexte" word-
sdxfield="twcontexte" display-order="10">
     <content source="thésaurus_W_contexte-historique"/>
  </index>
  <index id="contexte" local-sdxfield="lfcontexte" global-sdxfield="fcontexte" word-
sdxfield="contexte" display-order="10">
     <content source="contexte-historique"/>
  </index>
  <index id="association" local-sdxfield="lfassociation" global-sdxfield="fasso" word-
sdxfield="association" display-order="50">
     <content source="liste-asso"/>
  </index>
  <index id="acte" local-sdxfield="lfacte" global-sdxfield="facte" word-sdxfield="acte" display-
order="30">
     <content source="liste-acte"/>
  </index>
  <index id="lieuacte" local-sdxfield="lflieuacte" global-sdxfield="flieuacte" word-
sdxfield="lieuacte" display-order="60">
     <content role="lieu_acte"/>
  </index>
  <!-- Index des localisations des documents (LC 20090521) -->
  <index id="lieudocument" local-sdxfield="lflieudocument" global-sdxfield="flieudocument" word-
sdxfield="lieudocument" display-order="60">
     <content role="localisation_document"/>
  </index>
  <index id="rameau" local-sdxfield="lframeau" global-sdxfield="frameau" word-sdxfield="rameau"
display-order="10">
     <content source="liste-rameau"/>
  </index>
  <!-- Laval et Garnier : autres thesauri que Rameau utilisés suivant les services. Construits
sur le même modèle que Rameau (LC 20090521) -->
  <!-- <index id="laval" local-sdxfield="lflaval" global-sdxfield="flaval" word-sdxfield="laval"
display-order="10">
     <content source="liste-laval"/>
  </index>
  <index id="garnier" local-sdxfield="lfgarnier" global-sdxfield="fgarnier" word-
sdxfield="garnier" display-order="10">
     <content source="liste-garnier"/>
  </index> -->
  <!-- Deux index construits sur les séries d'archives et les séries de bibliothèques (éléments
de la cote). Utile pour les catalogues d'instruments de recherche (LC 20090521) -->
  <index id="seriearch" local-sdxfield="lfseriearch" global-sdxfield="fseriearch">
     <content source="liste-seriearch"/>
  </index>
  <index id="seriebib" local-sdxfield="lfseriebib" global-sdxfield="fseriebib">
     <content source="liste-seriebib"/>
  </index>
  <!-- Index des mois (LC 20090521) -->
  <!-- <index id="mois" local-sdxfield="lfmois" global-sdxfield="fmois" display-order="10">
     <content source="liste-mois"/>
  </index> -->
  <!-- Index des langues des documents (LC 20090521) -->
  <!-- <index id="langue" local-sdxfield="lflangue" global-sdxfield="flangue" display-order="10">
```


© 2010 AJLSM

```
    <content source="liste-langue"/>
  </index> -->
  <!-- Index des communes utile pour la recherche commune pour les services d'AD (LC 20090521)
-->
  <index id="commune" local-sdxfield="lfcommune" global-sdxfield="fcommune">
    <content source="liste-commune"/>
  </index>
  <!-- Six index des titres spécifiques. Peut donner lieu à d'autres descripteurs (LC 20090521)
-->
  <!-- <index id="titreaffiche" local-sdxfield="lftitreaffiche" global-sdxfield="ftitreaffiche"
word-sdxfield="titreaffiche">
    <content source="liste-affiche"/>
  </index>
  <index id="titrecarte" local-sdxfield="lftitrecarte" global-sdxfield="ftitrecarte" word-
sdxfield="titrecarte">
    <content source="liste-carte"/>
  </index>
  <index id="titrecarte_postale" local-sdxfield="lftitrecarte_postale" global-
sdxfield="ftitrecarte_postale" word-sdxfield="titrecarte_postale">
    <content source="liste-carte_postale"/>
  </index>
  <index id="titremonographie" local-sdxfield="lftitremonographie" global-
sdxfield="ftitremonographie" word-sdxfield="titremonographie">
    <content source="liste-monographie"/>
  </index>
  <index id="titreperiodique" local-sdxfield="lftitreperiodique" global-
sdxfield="ftitreperiodique" word-sdxfield="titreperiodique">
    <content source="liste-periodique"/>
  </index>
  <index id="titrephotographie" local-sdxfield="lftitrephotographie" global-
sdxfield="ftitrephotographie" word-sdxfield="titrephotographie">
    <content source="liste-photographie"/>
  </index> -->

  <!-- Les index de base en EAD : sur le nom seulement -->
  <index id="name" local-sdxfield="lfname" global-sdxfield="fname" word-sdxfield="name" display-
order="55">
    <content element="name"/>
  </index>
  <index id="corpname" local-sdxfield="lfcorpname" global-sdxfield="fcorpname" word-
sdxfield="corpname" display-order="50">
    <content element="corpname"/>
  </index>
  <index id="famname" local-sdxfield="lffamname" global-sdxfield="ffamname" word-
sdxfield="famname" display-order="40">
    <content element="famname"/>
  </index>
  <index id="persname" local-sdxfield="lfpersname" global-sdxfield="fpersname" word-
sdxfield="persname" display-order="40">
    <content element="persname"/>
  </index>
  <index id="geogname" local-sdxfield="lfgeogname" global-sdxfield="fgeogname" word-
sdxfield="geogname" display-order="60">
    <content element="geogname"/>
  </index>
  <index id="occupation" local-sdxfield="lfoccupation" global-sdxfield="foccupation" word-
sdxfield="occupation" display-order="20">
```


© 2010 AJLSM

```
    <content element="occupation"/>
  </index>
  <index id="function" local-sdxfield="lffunction" global-sdxfield="ffunction" word-
sdxfield="function" display-order="20">
    <content element="function"/>
  </index>
  <index id="subject" local-sdxfield="lfsubject" global-sdxfield="fsubject" word-
sdxfield="subject" display-order="10">
    <content element="subject"/>
  </index>
  <index id="genreform" local-sdxfield="lfgenreform" global-sdxfield="fgenreform" word-
sdxfield="genreform" display-order="30">
    <content element="genreform"/>
  </index>
  <index id="title" local-sdxfield="lftitle" global-sdxfield="ftitle" word-sdxfield="title"
display-order="70">
    <content element="title"/>
  </index>
</index-list>
```

3. Fichier de configuration des champs combinés

Le cadre ci-dessous présente le fichier `ead-aggregated-fields.xconf` de la version standard de Pleade. La personnalisation de votre application peut faire que le vôtre diffère.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Licence -->
<!-- Ce fichier de configuration fait partie de la distribution standard
de Pleade. Vous pouvez le modifier à votre guise. -->
<!-- La liste des champs qui sont une aggrégation de champs existants -->
<aggregated-fields>
  <!-- Un champ analyse-titre -->
  <aggregated-field name="analysetitre" type="word">
    <field name="unittitle"/>
    <field name="scopecontent"/>
  </aggregated-field>
  <!-- Un champ texte-plein + texte integral -->
  <aggregated-field name="extfulltext" type="word">
    <field name="ext"/>
    <field name="fulltext"/>
  </aggregated-field>
</aggregated-fields>
```

5. Fichier de configuration des menus

Le cadre ci-dessous présente le fichier `menus.xconf` de la version standard de Pleade. La personnalisation de votre application doit faire que le vôtre diffère.

```
<?xml version="1.0"?>
<!-- Licence -->
<!-- Ce fichier de configuration fait partie de la distribution standard de Pleade. Vous pouvez
le modifier à votre guise. -->
<!-- Fichier de configuration des menus du module EAD-EAC. Catalogue i18n par défaut = module-
eadeac -->
<menus xmlns:i18n="http://apache.org/cocoon/i18n/2.1">
  <!-- Un menu affiché en haut de page à droite -->
```


© 2010 AJLSM

```
  <menu type="htop" id="htop">
     <item id="index" type="link-normal" href="pages/index.html" title="pleade:menu.home.title"
i18n:attr="title">
        <i18n:text key="menu.home.text" catalogue="pleade">accueil</i18n:text>
     </item>
     <item id="contacts" type="link-normal" href="pages/contacts.html"
title="pleade:menu.contacts.title" i18n:attr="title">
        <i18n:text key="menu.contacts.text" catalogue="pleade">contacts</i18n:text>
     </item>
  </menu>

  <!-- Un menu d'onglets sous l'en-tête -->
  <menu type="htabs" id="htabs">
     <item id="consult" type="link-normal" context="((pages/fonds)|(search)|(cdc)|(results)|
(historic)|(basket)).*\.html" href="pages/fonds.html" title="menu.consult.title"
i18n:attr="title">
        <i18n:text key="menu.consult.text">consultation des fonds</i18n:text>
     </item>
     <item id="infos" type="link-normal" context="pages/infos.*\.html" href="pages/infos.html"
title="menu.infos.title" i18n:attr="title">
        <i18n:text key="menu.infos.text">informations pratiques</i18n:text>
     </item>
     <item id="edit" type="link-normal" context="edit/.*" href="edit/" reserve="eadeac-editor,
eadeac-web" title="module-eadeac-edit:edit.menu.edit.title" i18n:attr="title">
        <i18n:text i18n:catalogue="module-eadeac-edit"
key="edit.menu.edit.text">éditer</i18n:text>
     </item>
     <item id="ead-admin" type="link-normal" context="ead-admin/.*" href="ead-admin/"
reserve="eadeac-admin, eadeac-web" title="module-eadeac-edit:ead-admin.menu.title"
i18n:attr="title">
        <i18n:text i18n:catalogue="module-eadeac-edit" key="ead-
admin.menu.text">administration</i18n:text>
     </item>
     <item id="admin" type="link-normal" context="admin/.*" href="admin/users.html"
reserve="admin" title="pleade:admin.menu.admin.title" i18n:attr="title">
        <i18n:text key="admin.menu.admin.text" catalogue="pleade">super administration</i18n:text>
     </item>
  </menu>

  <!-- Menu de gauche lié à la consultation des fonds -->
  <menu type="vleft" id="search" default="true" depends="htabs:consult">
     <item type="txt">
        <i18n:text key="menu.simple-search.intro">recherche simple</i18n:text>
        <item id="simple-search" type="simple-search" action="results.html" method="get"
base="ead" /><!-- Ces trois attributs sont facultatifs, ici ce sont leurs valeurs par défaut -->
     </item>
     <item id="search" type="link-normal" context="@href" href="search-form.html"
title="menu.search.title" i18n:attr="title">
        <i18n:text key="menu.search.text">recherche avancée</i18n:text>
     </item>
     <item id="cdc" type="link-normal" context="@href" href="cdc.html" title="menu.cdc.title"
i18n:attr="title">
        <i18n:text key="menu.cdc.text">cadre de classement</i18n:text>
     </item>
     <item id="histo" type="link-normal" context="@href" href="historic.html"
title="menu.historic.title" i18n:attr="title">
        <i18n:text key="menu.historic.text">historique</i18n:text>
```


© 2010 AJLSM

```
    </item>
    <item id="basket" type="link-normal" context="@href" href="basket.html"
title="menu.basket.title" i18n:attr="title">
       <i18n:text key="menu.basket.text">porte-documents</i18n:text>
    </item>
    <item id="pleadeHome" type="link-window" href="http://pleade.com"
title="menu.pleade.com.title" i18n:attr="title">
       <i18n:text key="menu.pleade.com.text">Pleade.com</i18n:text>
    </item>
 </menu>

 <menu type="vleft" id="vleft_infos" depends="htabs:infos">
    <item id="access" type="link-normal" title="menu.access.title" i18n:attr="title"
context="@href" href="pages/infos-access.html">
       <i18n:text key="menu.access.text">accès</i18n:text>
    </item>
    <item id="hours" type="link-normal" context="@href" title="menu.hours.title"
i18n:attr="title" href="pages/infos-opening.html">
       <i18n:text key="menu.hours.text">heures d'ouverture</i18n:text>
    </item>
 </menu>

 <!-- Gestion de contenus -->
 <menu type="vleft" id="vleft_edit" depends="htabs:edit" reserve="eadeac-editor">
    <item type="txt">
       <i18n:text catalogue="module-eadeac-edit" key="edit.menu.ead.intro">documents
EAD</i18n:text>
       <item id="ead-publish" reserve="eadeac-editor" type="link-normal" title="module-eadeac-
edit:edit.menu.ead-publish.title" i18n:attr="title" context="edit/ead-publish(-form)?\.html"
href="edit/ead-publish-form.html">
          <i18n:text catalogue="module-eadeac-edit" key="edit.menu.ead-
publish.text">publier</i18n:text>
       </item>
       <item id="ead-manage" reserve="eadeac-editor" type="link-normal" title="module-eadeac-
edit:edit.menu.ead-manage.title" i18n:attr="title" href="edit/ead-manage.html" context="@href">
          <i18n:text catalogue="module-eadeac-edit" key="edit.menu.ead-
manage.text">gérer</i18n:text>
       </item>
       <item id="ead-cdc-manage" reserve="eadeac-editor" type="link-normal" context="@href"
title="module-eadeac-edit:edit.menu.ead-cdc-manage.title" i18n:attr="title" href="edit/ead-cdc-
manage.html">
          <i18n:text catalogue="module-eadeac-edit" key="edit.menu.ead-cdc-manage.text">cadre
de classement</i18n:text>
       </item>
       <item id="ead-export-index" reserve="eadeac-editor, eadeac-web" type="link-normal"
context="@href" href="edit/ead-export-index.html" title="module-eadeac-edit:edit.menu.export-
index.title" i18n:attr="title">
          <i18n:text catalogue="module-eadeac-edit" key="edit.menu.export-
index.text">récupérer les index</i18n:text>
       </item>
    </item>
 </menu>

 <!-- Administration de la recherche et de la publication -->
 <menu type="vleft" id="vleft_ead-admin" depends="htabs:ead-admin" reserve="eadeac-admin">
    <item id="ead-admin-ead" reserve="eadeac-admin" type="link-normal" title="module-eadeac-
edit:ead-admin.menu.ead.title" i18n:attr="title" context="@href" href="ead-admin/ead.html">
```


© 2010 AJLSM

```
        <i18n:text catalogue="module-eadeac-edit" key="ead-admin.menu.ead.text">recherche et
publication</i18n:text>
    </item>
    <item id="ead-admin-menus" reserve="eadeac-admin" type="link-normal" title="module-eadeac-
edit:ead-admin.menu.menus.title" i18n:attr="title" context="@href" href="ead-admin/menus.html">
        <i18n:text catalogue="module-eadeac-edit" key="ead-
admin.menu.menus.text">menus</i18n:text>
    </item>
    <item id="ead-admin-search-forms" reserve="eadeac-admin" type="link-normal" title="module-
eadeac-edit:ead-admin.menu.search-forms.title" i18n:attr="title" context="@href" href="ead-
admin/search-forms.html">
        <i18n:text catalogue="module-eadeac-edit" key="ead-admin.menu.search-
forms.text">formulaires de recherche</i18n:text>
    </item>
    <item id="ead-admin-subsets" reserve="eadeac-admin" type="link-normal" title="module-eadeac-
edit:ead-admin.menu.subsets.title" i18n:attr="title" context="@href" href="ead-
admin/subsets.html">
        <i18n:text catalogue="module-eadeac-edit" key="ead-
admin.menu.subsets.text">rubriques</i18n:text>
    </item>
    <item id="ead-admin-statistics" reserve="eadeac-admin, eadeac-web" type="link-normal"
title="module-eadeac-edit:ead-admin.menu.statistics.title" i18n:attr="title" context="@href"
href="ead-admin/pleade-stats.html">
        <i18n:text catalogue="module-eadeac-edit" key="ead-
admin.menu.statistics.text">statistiques</i18n:text>
    </item>
    <item id="ead-admin-pages" reserve="eadeac-admin" type="link-normal" title="module-eadeac-
edit:ead-admin.menu.pages.title" i18n:attr="title" context="@href" href="ead-admin/pages.html">
        <i18n:text catalogue="module-eadeac-edit" key="ead-admin.menu.pages.text">pages
statiques</i18n:text>
    </item>
  </menu>

</menus>
```

6. Fichier de configuration du formulaire : ead-default.xconf

Le cadre ci-dessous présente le fichier `search-forms/ead-default.xconf` de la version standard de Pleade. La personnalisation de votre application doit faire que le vôtre diffère.

```
<?xml version="1.0"?>
<!-- Licence -->
<!-- Ce fichier de configuration fait partie de la distribution standard
de Pleade. Vous pouvez le modifier à votre guise. -->
<form action="results.html" max-chars="50" documents="ead">
  <key type="cdc" help="true"/>
  <key type="text" field="fulltext" fieldh="hfulltext" help="true" alternate-field="extfulltext"
alternate-fieldh="hextfulltext" alternate-field-checked="false"/>
  <key type="list" variant="combo" field="fpersname" help="true">
     <no-choice i18n-key="form-no-choice-barre"/>
  </key>
  <key type="suggest" field="fsubject" help="true" terms-browse="true"/>
  <key type="suggest" field="fgeogname" help="true" terms-browse="true"/>
  <key type="date"  id="annee" variant="exact interval" format="(\d)+" help="true"/>
</form>
```


© 2010 AJLSM