全文検索システム『ひまわり』/設定ファイル作成の手引き

全文検索システム『ひまわり』

- はじめに
- 抽出する属性を追加，変更するには
- 属性を抽出する要素を指定するには
- 全文検索対象の要素を指定するには
- 複数の言語資料をまとめて検索するには
- 好みのブラウザで閲覧するには
- 閲覧用の表示形式を変更するには
- 補足説明

はじめに

このページでは，設定ファイルで記述できる項目のうち，よく使うと思われる項目について解説します。ただし，この後の説明は，『ひまわり』や XML について，次のことを前提として書いています。

- 本手引きは，「簡単な検索用データの作成方法」の内容を発展させる形で書いています。まずは，「簡単な検索用データの作成方法」をお読みください。
- 『ひまわり』では，検索対象の資料が XML で記述されていることが前提となっており， XML に関するある程度の知識を必要とします。この後の説明を読んで，難しいと感じたかたは，次の資料をまずご参照ください。
  - 本ページ中に頻繁に現れる「要素」，「タグ」，「属性」などの XML の用語を補足説明で説明しています。こちらもあわせてご覧ください。
  - XML 自体についての解説は，各種の入門書・Web ページをご覧ください。ここでは，「たのしいXML」を参考用の Web ページとして挙げておきます。

↥

抽出する属性を追加，変更するには

『ひまわり』は，検索結果の文字列をマークアップしているタグの属性を，検索結果として抽出することができます。例えば，「簡単な検索用データの作成方法 」では，検索された文字列の「著者」と「タイトル」を抽出しています。

まず，抽出する属性と『ひまわり』の設定ファイルとの関係について説明します。例として，「簡単な検索用データの作成方法」の config_simpledoc.xml を見てみましょう。config_simpledoc.xml を「秀丸」などのテキストエディタで開いて，field_setting 要素を探してください。次の部分です(見やすくするために，適宜改行しています)。

```
<field_setting>
  <li name="no" type="index" width="30" align="RIGHT" />
  <li name="前文脈" type="preceding_context" element="_sys"
      attribute="_preceding_context" width="180" align="RIGHT"
      sort_direction="R" />
  <li name="キー" type="key" element="_sys" attribute="_key"
      width="80" sort_order="1" />
  <li name="後文脈" type="following_context" element="_sys"
      attribute="_following_context" width="160" sort_order="2" />
  <li name="著者" type="argument" element="simpledoc"
      attribute="著者" width="80" />
  <li name="タイトル" type="argument" element="simpledoc"
      attribute="タイトル" width="80" />
</field_setting>
```

li 要素の name 属性に注目してください。これを見てわかるとおり，field_setting 要素中の li 要素は，検索結果の各列(フィールド)の情報を表しています。ちなみに，li 要素の順序が，検索結果の列の表示順序を決定します。

li 要素の属性のうち，本題に関係する四つの属性を見ていきましょう。

name 属性：
検索結果の列名です。この値を変更すれば，列の名前が変わります。

type 属性：
ここでは，「argument」としてください。

element, attribute 属性：
それぞれの属性値で，抽出対象の要素とその属性名を指定します。例えば，上記の最後の行の li 要素は，element の値が「simpledoc」，attribute が「タイトル」となっています。これは，「simpledoc」要素の「タイトル」属性を検索結果として表示することを意味します。

まず，抽出する属性の変更ですが，上で述べたように，li 要素の element 属性と attribute 属性を変更することによって行います。

次に，抽出する属性を追加してみます。例として，「簡単な検索用データの作成方法」の simpledoc 要素に「作成日」属性をつけます。具体的な追加例を次に示します。

```
<simpledoc タイトル="蜘蛛の糸" 著者="芥川龍之介" 作成日="2005-01-16">
```

資料に対して，「作成日」属性を追加したら，config_simpledoc.xml の field_setting 要素には，次の li 要素を追加します。

```
<li name="作成日" type="argument" element="simpledoc"
    attribute="作成日" width="80" />
```

attribute 属性が「作成日」となっているのに注意してください。name 属性は，検索結果の列名になります。field_setting 要素中のどこに追加してもかまいませんが，すでに述べたように，検索結果の列の表示順に影響します。

↥

属性を抽出する要素を指定するには

「簡単な検索用データの作成方法」では，simpledoc 要素の属性だけが検索結果として抽出されていました。ここでは，属性を抽出する要素を指定する方法の例として，複数の要素の属性を抽出する方法を説明します。

具体的な状況として，複数の章からなる文書を考えてみます。例えば，次のような構造を持った文書です。新たに設定した要素は，「章」要素です。

```
<simpledoc タイトル="全文検索システムについて" 筆者="国語太郎">
  <章 タイトル="第１章 はじめに">
    この文章では，... について述べます。
                :
                :
  </章>
  <章 タイトル="第２章 発表の手順">
    本発表の手順は，次のとおりです。
                :
                :
  </章>
</simpledoc>
```

ここで，第２章に含まれる「本発表」を検索したとします。config_simpledoc.xml の設定だと，simpledoc 要素の属性(「タイトル」属性と「著者」属性)しか，検索結果として抽出できません。どの章に含まれるかを知るには，「章」要素の「タイトル」属性を抽出する必要があります。

まず，抽出対象の属性を含んだ要素を設定ファイルに指定する方法を説明します。config_simpledoc.xml の index_eix 要素を見てください。

```
<index_eix>
  <li name="simpledoc" middle_name="sd" is_empty="false"
      top="false" isBrowsed="true" />
</index_eix>
```

index_eix 要素中の個々の li 要素で属性抽出対象の要素を指定しています。それぞれの属性の意味は，次のとおりです。

name 属性：
属性抽出対象の要素名です。上の例の場合，li 要素の name 属性で，simpledoc 要素が設定されています。

middle_name 属性：
他の li 要素の middle_name 属性値と重ならない値を半角文字で設定してください。どのような文字列でも構いません。索引ファイル名の一部となります。

isBrowsed 属性
「true」の場合は，閲覧対象の要素([ツール]→[閲覧]の対象)であることを示します。閲覧対象の要素でない場合は，指定する必要はありません。なお，現在のところ，index_eix 要素中の一つの li 要素でしか指定できません。

is_empty, is_top 属性
ここでは，詳しく説明しません。false としてください。

「章」要素を属性抽出対象の要素とするには，次の li 要素を index_eix 要素に追加してください。

```
<li name="章" middle_name="section" is_empty="false" top="false" />
```

li 要素の追加が終わったら，[ツール]→[インデックス生成]で索引を生成してください。ただし，すでに索引ファイルが作成されている場合は，生成を行う前に，索引ファイルを削除してください。索引ファイルは，corpora 要素中の li 要素の path 属性で指定された場所に作られます。例えば，次の設定の場合，Corpora/Simpledoc フォルダ中(Corpora フォルダ中の Simpledoc フォルダ)に作られます。なお，今回生成される索引ファイルは，middle_name 属性に「section」を指定していますので，corpus.section.eix となります。

```
<corpora name="simpledoc">
  <li name="simpledoc" path="Corpora/Simpledoc/corpus"/>
</corpora>
```

以上で，index_eix 要素のほうの準備は，完了です。後は，抽出する属性を表示するために，次のように field_setting 要素を修正してください。修正方法は，「抽出する属性を追加，変更するには」で説明したとおりです。

```
<li name="章のタイトル" type="argument" element="章"
     attribute="タイトル" width="80" />
```

全文検索対象の要素を指定するには

全文検索対象の要素は，index_cix 要素で指定します。config_simpledoc.xml を見てみましょう。

```
<index_cix>
  <li name="simpledoc" label="本文" middle_name="sd" type="normal"
      field_name="キー" />
</index_cix>
```

li 要素の内容は，次のとおりです。

name 属性：
全文検索対象の要素名です。上の例の場合，li 要素の name 属性で，simpledoc 要素が設定されています。

label 属性：
検索対象の選択メニューに表示される文字列です。

middle_name 属性：
索引ファイル名の一部となります。他の li 要素の middle_name 属性値と重ならない値を半角文字で設定してください。どのような文字列でも構いません。

type 属性：
「normal」としてください。

field_name 属性：
検索結果を表示する列名を指定します。この列名は，field_setting 要素で指定した列名と対応しています。上の例では，「キー」の列に全文検索の結果が入るようになっています。

index_cix 要素内の li 要素は，次のように，複数指定することができます。この例だと，「章」要素内だけ，「あとがき」要素内だけのように，特定の要素内だけを検索対象とすることができます。検索対象の選択は，選択メニューから行うことができます。

```
<index_cix>
  <li name="章" label="本文" middle_name="sd" type="normal"
      field_name="キー" />
  <li name="あとがき" label="あとがき" middle_name="sd" type="normal"
      field_name="キー" />
  <li name="前書き" label="前書き" middle_name="sd" type="normal"
      field_name="キー" />
</index_cix>
```

複数の言語資料をまとめて検索するには

検索対象の言語資料は，corpora 要素で指定します。config_simpledoc.xml では，次のように定義されています。

```
<corpora name="simpledoc">
  <li name="simpledoc" path="Corpora/Simpledoc/corpus" />
</corpora>
```

li 要素の内容は，次のとおりです。

name 属性：
言語資料名を指定します。ver.1.3 では，検索対象のコーパスリストに表示される名前になります。一意に定まる文字列を指定してください。ver.1.2 では，利用されていません。

path 属性：
言語資料の格納場所を指定します。上の例の場合，「Corpora/Simpledoc/」までが言語資料を格納するフォルダです。その後の「corpus」が言語資料のファイル名本体を表します。

複数の言語資料を検索するには，次のように li 要素を複数指定します。

```
<corpora name="simpledoc">
  <li name="simpledoc1" path="Corpora/Simpledoc1/corpus" />
  <li name="simpledoc2" path="Corpora/Simpledoc2/corpus" />
</corpora>
```

好みのブラウザで閲覧するには

ブラウザの設定は，設定ファイルの browsers 要素で指定します。config_simpledoc.xml の browsers 要素を次に示します。

```
<browsers temp_file="__searched_tmp.xml" label="作品全体">
  <li name="Microsoft Internet Explorer"
      path="c:¥progra~1¥intern~1¥iexplore" />
  <li name="Mozilla" path="mozilla" />
</browsers>
```

一つの li 要素が一つのブラウザの設定を表します。一番始めに設定したブラウザが，『ひまわり』起動時のデフォルトのブラウザとして使用されます。li 要素の内容は，次のとおりです。

name 属性：
[ツール]→[オプション]→[ブラウザ]で表示される文字列です。

path 属性：
path 属性では，起動するブラウザのコマンドを指定します。

好みのブラウザが設定されていない場合は，上記を参考にして，li 要素を追加してください。また，設定されていてもうまく起動できない場合は，li 要素の path 属性をチェックしてみてください。

なお，『ひまわり』 ver.1.2β03 では，li 要素に option 属性が追加され，ブラウザのオプションを指定できるようになります。これにより，次のようにすれば，Mac OS X でも閲覧機能を利用できます。

```
<li name="Mozilla(Mac)" path="open" option="-a Mozilla"/>
```

閲覧用の表示形式を変更するには

ブラウザによる閲覧用の表示形式は，設定ファイルの xsl_files 要素で XSL スタイルシートを指定します。XSL スタイルシートの記述方法については，参考図書や Web ページ(「たのしいXML」など)を御覧ください。

次は，config_simpledoc.xml の例です。xsl_files 要素の root_path 要素には，スタイルシートを格納しているフォルダを指定します。

```
<xsl_files root_path="Corpora/Simpledoc/xslt">
  <li label="標準" name="simpledoc.xsl" />
</xsl_files>
```

li 要素は，複数指定することができます。複数指定した場合は，[ツール]→[オプション]→[ブラウザ] で選択します。先頭の li 要素で指定した設定がデフォルトとなります。

li 要素の内容は，次のとおりです。

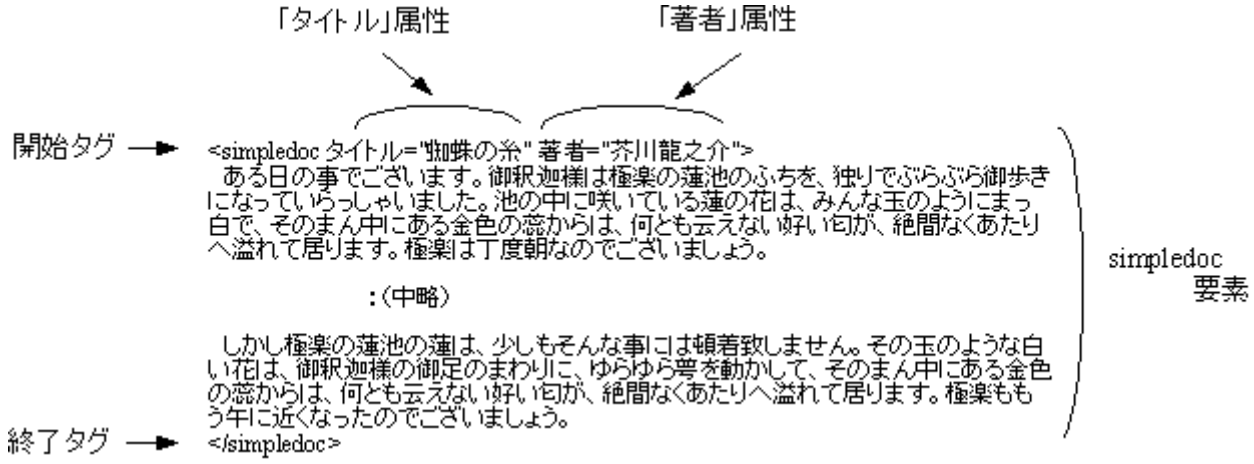
label 属性：
[ツール]→[オプション]→[ブラウザ] の表示される文字列となります。

name 属性：
XSL スタイルシートのファイル名を指定します。なお，スタイルシートの格納場所は，xsl_files 要素の root_path 属性で指定します。

補足説明

ここでは，XML に関する用語のうち，「要素」，「タグ」，「属性」など，上記説明中で頻繁に使用されている用語について補足説明を行います。

- 「開始タグ」から「終了タグ」までの部分を「要素」といいます。下の例は，simpledoc 要素の例です。
- 「開始タグ」と「終了タグ」は必ずペアになっています。



- ただし，「開始タグ」と「終了タグ」で囲われている部分（「要素内容」といいます）がない，「空要素」という「要素」もあります。例えば，上記の説明中で頻繁に出てくる li 要素です。空要素は，<... /> の形式で記述します。

```
<li name="simpledoc" path="Corpora/Simpledoc/corpus" />
```

- 上の simpledoc 要素のように，「要素」に「属性」を付与することができます。 「属性」は「開始タグ」に記入します。上の例では，simpledoc 要素の「開始タグ」に「タイトル」属性と「著者」属性を付与しています。

Last-modified: 2010-01-18 (月) 12:22:29 (3147d)

Site admin: anonymous