

全文検索システム『ひまわり』/設定ファイルリファレンスマニュアル

[Top](#) / [全文検索システム『ひまわり』](#) / 設定ファイルリファレンスマニュアル

言語を選択 | ▼

[全文検索システム『ひまわり』](#)

はじめに [±]

この文書は、「ひまわり」(ver.1.3, 1.5)の設定ファイルのリファレンスである。

- [はじめに](#)
- [設定一覧](#)
 - [コーパス, 索引ファイル関連](#)
 - [corpora 要素](#)
 - [index_cix 要素](#)
 - [index_eix 要素](#)
 - [index_aix 要素](#)
 - [検索設定関連](#)
 - [field_setting 要素](#)
 - [length_context_kwic 要素](#)
 - [length_context_search 要素](#)
 - [preceding_context_constraint 要素](#)
 - [following_context_constraint 要素](#)
 - [GUI 関連](#)
 - [fontsize 要素](#)
 - [isIndexingEnable 要素](#)
 - [isGenerateCorpusFileEnable 要素](#)(ver.1.5で廃止)
 - [閲覧関連](#)
 - [browsers 要素](#)
 - [xsl_files 要素](#)
 - [外部データベース関連](#)
 - [external_tools 要素](#) (since ver.1.6).
 - [access_command1, access_command2 要素](#) (deprecated).
 - [ext_db1, ext_db2 要素](#)
 - [jitaidic 要素](#)
 - [一覧表示機能関連](#)
 - [corpus_fields](#)
 - [unit_fields](#)
 - [user_defined_lists](#) (since ver.1.6, 未確定).
 - [stat_fields_1, stat_fields_2, stat_fields_3](#)
 - [インポート関連](#) (since 1.5β).
 - [import / target_file_type 要素](#)
 - [import / char_normalization 要素](#)
 - [import / char_conversion_table 要素](#)
 - [import / text_transformation_definition 要素](#)
 - [import / xhtml_style_sheet 要素](#)
 - [import / xml_style_sheet 要素](#)
 - [import / as_subcorpora 要素](#) (since 1.5β04).
 - [import / not_now_indexing 要素](#) (since 1.5β04).
 - [import / source_files 要素](#) (since 1.5β04).

- [アノテーション関連](#)
 - [annotator 要素 \(since 1.5β\)](#)
- [「ひまわり」資料参照関連](#)
 - [manual 要素](#)
 - [hp 要素](#)
- [パッケージ設定ファイル](#)
 - [インストール設定ファイルの仕様](#)
 - [インストール設定ファイルの例](#)
 - [『分類語彙表』パッケージ](#)
 - [『青空文庫』パッケージ](#)

↑

設定一覧 [↑]

↑

コーパス, 索引ファイル関連 [↑]

↑

[corpora 要素](#) [↑]

コーパス集合を定義する要素

- 属性
 - name: コーパス集合名
 - dbpath: データベースへのパス(存在しない場合は、一つ目のli/@pathのディレクトリを使用) (**since ver.1.5β**)
- li (空要素)
 - @name: コーパス名
 - @path: コーパスファイル名へのパス+コーパスファイル名の本体(body)
 - @isSelected (**since ver.1.3**): 検索対象とするか否か
 - false: しない
 - true: する(default)
- 例

```
<corpora name="「太陽」コーパス">
  <li name="「太陽」" path="Corpora/Zassi/Taiyo/corpus" />
  <li name="女性雑誌" path="Corpora/Zassi/Josei/corpus" />
</corpora>
```

↑

[index_cix 要素](#) [↑]

要素内容への索引

- li (空要素)
 - @name: 索引対象の要素名
 - @label: 検索対象選択メニューに表示される文字列
 - @middle_name: 索引ファイルの第2拡張子(ファイル名の末尾から2番目の拡張子)
 - @field_name: 結果レコードに照合した文字列を格納するフィールド名
 - @type: 索引タイプ
 - normal (default) ... 指定された要素の索引を作成する(検索文字列の先頭文字列が指定した要素内にあれば、マッチングする)
 - record_based ... 指定された要素の索引を作成する(検索文字列がすべて指定した要素内になければマッチングしない)
 - null ... 検索時に索引を使用しない(正規表現が利用可能) (**since ver.1.3**)
 - restricted (**obsoleted since ver.1.3**)
 - @stop_element: ストップ要素名
 - @isEditable: 編集の可否 (**since ver.1.3**)

- true
- false (default)

• 例

```
<index_cix>
  <li label="本文" name="雑誌" middle_name="magazine"
    type="normal" field_name="キー" />
  <li label="本文(s 要素考慮)" name="雑誌" middle_name="magazine"
    type="normal" stop_element="s" field_name="キー" />
</index_cix>
```

↑

index_eix 要素 [↑]

要素への索引

- li (空要素)
 - @name: 索引対象の要素名
 - @middle_name: 索引ファイルの第 2 拡張子(ファイル名の末尾から 2 番目の拡張子)
 - @is_empty: 空要素か否か
 - true (= empty_forward)
 - false (default)
 - empty_forward (後方検索) ... マッチした文字列の後方に存在する指定要素に対する索引 (since ver.1.3)
 - empty_backward (前方検索) ... マッチした文字列の前方に存在する指定要素に対する索引 (since ver.1.3)
 - @top: 資料の最大要素
 - @isBrowsed: 閲覧対象の要素か否か

• 例

```
<index_eix>
  <li name="雑誌" middle_name="magazine" is_empty="false" top="true" />
  <li name="記事" middle_name="article" is_empty="false" isBrowsed="true" />
  <li name="引用" middle_name="quote" is_empty="false" />
  <li name="l" middle_name="ref" is_empty="true" />
</index_eix>
```

↑

index_aix 要素 [↑]

要素属性への索引

- li (空要素)
 - @name: 索引対象の要素名
 - @argument: 索引対象の属性名
 - @label: 検索対象選択メニューに表示される文字列
 - @middle_name: 索引ファイルの第 2 拡張子(ファイル名の末尾から 2 番目の拡張子)
 - @field_name: 結果レコードに照合した文字列を格納するフィールド名
 - @type: 索引タイプ
 - normal (default) ... 指定したキーで属性を検索
 - record_based ... 指定したキーで属性を検索 (前後文脈を検索条件で指定する代わりに, キー前後の文字列 (つまり属性) を正規表現で指定。例えば, 「あ」で始まる属性を検索すると行ったことが可能)
 - db ... データベースに格納されたアノテーション情報に基づき検索する
 - six ...
 - @isCompleteMatch: 完全一致検索をするか否か(このオプションを変更した場合は, 再度索引づけする必要あり)
 - true
 - false (default)

• 例

```
<index_aix>
  <li label="ルビ(rt)完全一致" name="r" middle_name="r" argument="rt"
    isCompleteMatch="true" field_name="キー" />
  <li label="ルビ(rt)部分一致" name="r" middle_name="r2" argument="rt"
    type="record_based" isCompleteMatch="false" field_name="キー" />
  <li label="外字(name)" name="外字" middle_name="gaiji" argument="name"
    isCompleteMatch="true" field_name="キー" />
</index_aix>
```

検索設定関連 [↑]

field_setting 要素 [↑]

結果レコードのフィールド定義

- li (空要素)
 - @name: フィールド名
 - @type: フィールドタイプ
 - argument: 属性検索
 - sibling: 兄弟要素検索
 - relative: 隣接要素検索
 - index: レコード索引(行番号)
 - db: データベース検索 (since ver.1.5)
 - @width: フィールド幅 (default: 40)
 - @align: 文字揃え
 - LEFT (default)
 - CENTER
 - RIGHT
 - @edit_type: 編集タイプ (since ver.1.5)
 - TEXT ... 自由記述(default)
 - SELECT ... 選択記述
 - @edit_option: 編集オプション (since ver.1.5)
 - @element: 表示対象の要素
 - 特殊要素 _EDIT ... 編集
 - @attribute: 表示対象の要素属性
 - @isEditable: 編集の可否 (since ver.1.5)
 - true ... 編集可能
 - false ... 編集不可(default)
 - @sort_direction: ソート方向
 - L ... 左から右(default)
 - R ... 右から左
 - @sort_order: ソート順位 (1~)
 - @sort_type: ソートの種類
 - string (default) ... 文字列としてソート
 - numeric ... 数字としてソート

例

```
<field_setting>
  <li name="no" type="index" width="30" align="RIGHT" sort_direction="L" />
  <li name="類似する語(前)" type="preceding_context" element="_sys"
    attribute="_preceding_context" width="180" align="RIGHT"
    sort_direction="R" />
  <li name="表記" type="key" element="_sys" attribute="_key" width="80"
    align="LEFT" sort_direction="L" />
  <li name="類似する語(後)" type="following_context" element="_sys"
    attribute="_following_context" width="160" align="LEFT"
    sort_direction="L" />
  <li name="見出し" type="argument" element="e" attribute="見出し" width="80"
    align="LEFT" sort_direction="L" sort_order="1" />
  <li name="概念パス" type="argument" element="c" attribute="概念パス"
```

```
width="80" align="LEFT" sort_direction="L" sort_order="2"/>
<li name="分類番号" type="argument" element="c" attribute="分類番号"
width="80" align="LEFT" sort_direction="L" />
<li name="行番号" type="argument" element="s" attribute="行番号" width="80"
align="LEFT" sort_direction="L" />
<li name="行内番号" type="argument" element="l" attribute="行内番号"
width="80" align="LEFT" sort_direction="L" />
</field_setting>
```



length_context_kwic 要素 [↑](#)

KWIC の文脈長を定義

- @value: 文脈長(文字)
- 例

```
<length_context_kwic value="10" />
```



length_context_search 要素 [↑](#)

正規表現検索時の照合文字列長の定義

- @value: 文脈長(文字)
- 例

```
<length_context_search value="10" />
```



preceding_context_constraint 要素 [↑](#)

前文脈の制約値を定義。この値は、前文脈欄の history 機能の履歴として登録される。

- @value: 制約値
- 例

```
<preceding_context_constraint
value="^[0-9 アーヴー a-z A-Z. / . : ]^[0-9 アーヴー a-z A-Z¥s]?¥" />
```



following_context_constraint 要素 [↑](#)

後文脈の制約値を定義。この値は、後文脈欄の history 機能の履歴として登録される。

- @value: 制約値
- 例

```
<following_context_constraint
value="^[0-9 アーヴー a-z A-Z¥s]?[0-9 アーヴー a-z A-Z. / . : ]" />
```



GUI 関連 [↑](#)



fontsize 要素 [↑](#)

GUI のフォントサイズの定義

- @value: 7 ~ 18 (pt)
- 例

```
<fontsize value="14" />
```



isIndexingEnable 要素 [↑](#)

メニュー項目 [ツール]→[インデックス生成] を表示するか否か

- @value
 - true
 - false (default)
- 例

```
<isIndexingEnable value="true" />
```

↑

isGenerateCorpusFileEnable 要素(ver.1.5で廃止) [↑](#)

メニュー項目 [ツール]→[コーパスファイル生成] を表示するか否か

- @value
 - true
 - false (default)
- 例

```
<isGenerateCorpusFileEnable value="false" />
```

↑

閲覧関連 [↑](#)

↑

browsers 要素 [↑](#)

閲覧用の WWW ブラウザの定義

- @temp_file: ブラウズ時のテンポラリファイル
 - @label: [ツール]→[閲覧]中の項目文字列
- li (空要素)
 - @name: ブラウザ名
 - @path: ブラウザの実行プログラムへのパス
 - パスに "Program Files" を含み、起動に失敗した場合, "Program Files (x86)"に置換した上で再実行
 - [[default_browser]]の場合は, OSのデフォルトブラウザを使用(since 1.6)
 - @option: ブラウザ実行時のオプション
 - @os: 対応するOS名
 - Windows
 - Mac
 - Linux
 - @omit_scheme ... **廃止**
 - true ... isBrowsed が true の要素閲覧時の URL に対して, スキーム(file://)を付けない。
 - true 以外 ... スキームを付ける。
- 例

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

↑

xsl_files 要素 [↑](#)

閲覧用の XSL ファイルに関する定義

- @root_path: XSL ファイルを格納しているディレクトリのルートパス(「ひまわり」を格納しているディレクトリからの相対パスで定義)
- @temp_dir: XSL ファイルを格納しているディレクトリのルートパス(「ひまわり」を格納しているディレクトリからの相対パスで定義)
- li (空要素)
 - @name: XSL ファイル名
 - @label:
- 例

```
<xsl_files root_path="xslt">
  <li label="横書き・段落表示" name="zassiHP.xsl" />
  <li label="横書き・行番号表示" name="zassiHL.xsl" />
  <li label="縦書き・段落表示" name="zassiVP.xsl" />
  <li label="縦書き・行番号表示" name="zassiVL.xsl" />
</file_xsl>
```

外部データベース関連 [↑]

external_tools 要素 (since ver.1.6) [↑]

外部ツールへのアクセス方法の定義。後述の[access_command1, 2要素](#)では、2個までしか定義できなかったが、3個以上定義できるようにした。また、OS別の記述も可能

- li (空要素)
 - @field: access_command1, 2と同一
 - @path: access_command1, 2と同一
 - @argument: access_command1, 2と同一
 - @name: 設定名（重複しないように設定すること）。メニューには@labelの値が用いられる。
 - @os: 対応するOS名
 - Windows
 - Mac
 - Linux

access_command1, access_command2 要素 (deprecated) [↑]

外部ツールへのアクセス方法の定義(2通り定義できる)。ver.1.6からexternal_tools 要素を推奨。

- @label: メニュー用のラベル
- @path: 実行プログラムへのパス。ただし、[[]] で囲われている場合は、次の内部コマンドを実行する。
 - soundplayer ... 音声再生用プログラム
 - xdb1, xdb2 ... 簡易データベース検索プログラム。それぞれ、ext_db1, ext_db2 要素で設定を行う。
 - browser ... ブラウザ。browser 要素で指定されたプログラムを利用する。(since 1.5)
- @argument: 実行プログラムの引数
 - (()) で囲われている場合は、検索結果中の当該フィールド値で置き換えられる。
 - @field が指定されておらず、@argument の値がフィールド指定になっていれば、それが起動用フィールドとしても利用されるになる(ただし、「((雑誌名)) 400px」のように、フィールド指定の文字列以外の文字列を含んではならない)。
 - 内部コマンドでない場合、@argument中の引数が複数ある場合、"/_"で区切る。空白は引数の区切りには使用できない(ファイル名中の空白と区別するため)
 - 内部コマンドの引数
 - soundplayer ... 第1引数: サウンドファイル
 - xdb1, xdb2 ... 第1引数: DB検索キー、第2引数: 結果表示ウィンドウの幅
 - browser ... 第一引数: URL
- @field: 実行プログラムの起動用フィールド。検索結果ウィンドウの指定されたフィールドをダブルクリックすると当該コマンドが実行される。(since 1.3)
- 例

```
<access_command1 label="著者情報" path="[[xdb1]]" argument="((著者))" />
<access_command1 label="著者DB" path="[[xdb1]]" argument="((著者)) 400px"
  field="著者ID"/>
```

ext_db1, ext_db2 要素 [↑](#)

[ツール]→[一覧]から参照される外部データベースの参照形態を定義する。また、ext_db1, ext_db2 は、それぞれ xdb1, xdb2 用の設定である。

- @name: データベース名。この名前は、[ツール]→[一覧]で表示される文字列としても使われる。
- @url: データベースファイルへの URL
- @record_name: 検索対象のレコード名
- @key: 検索対象のフィールド名
- li (空要素)
 - @name: フィールド名
 - @width: フィールド幅 (default: 40)
 - @align: 文字揃え
 - @sort_order: ソート順位 (1~)
 - @sort_type: ソートの種類
 - string (default) ... 文字列としてソート
 - numeric ... 数字としてソート
- 例

```
<ext_db name="著者DB" url="authors.xml" record_name="著者" key="氏名">
  <li name="氏名" width="100" sort_order="1" />
  <li name="氏名よみ" width="100" />
  <li name="所属" width="80" />
  <li name="分野" width="80" />
  <li name="生年" width="50" align="RIGHT" />
  <li name="没年" width="50" align="RIGHT" />
</ext_db>
```

jitaidic 要素 [↑](#)

字体辞書の定義

- @url: 字体辞書ファイルのURL
- 例

```
<jitaidic url="jitaidic.xml" />
```

一覧表示機能関連 [↑](#)

corpus_fields [↑](#)

収録しているコーパスの一覧を表示する

unit_fields [↑](#)

閲覧対象要素の一覧を表示する。閲覧対象要素は、index_eix/@isBrowsed で定義する。

user_defined_lists (since ver.1.6, 未確定) [↑](#)

指定した要素の一覧を表示する。なお、element 要素は複数記述することができる。

- @label: (廃止予定)
- element 要素
 - @name: 一覧対象の要素名
 - @label: メニュー表示用のラベル
 - li 要素

- @name: 表示用のフィールド名
- @element: 表示する属性の要素名（ただし、一覧する要素か、それを包含する要素）
- @attribute: 表示する属性名
なお、特殊な属性値として、_contents (要素内容を表示)、_length (要素内容の文字数を表示)が定義されている。
- @width: フィールド幅
- @align: 文字揃え
- @sort_order: ソート順位 (1～)
- @sort_type: ソートの種類

↑

stat_fields_1, stat_fields_2, stat_fields_3 ↑

頻度計算するのに利用するフィールドの定義

- @label: [ツール]→[統計]メニュー用のラベル
- li (空要素)
 - @name: フィールド名
 - @width: フィールド幅 (default: 40)
 - @align: 文字揃え
 - @sort_order: ソート順位 (1～)
 - @sort_type: ソートの種類
 - string (default) ... 文字列としてソート
 - numeric ... 数字としてソート
- 例

```
<li name="キー" width="40" />
<li name="年" width="50" align="RIGHT" />
<li name="号" width="50" align="RIGHT" />
<li name="頻度" width="60" align="RIGHT" sort_type="numeric" />
```

↑

インポート関連 (since 1.5β) ↑

↑

import / target_file_type 要素 ↑

インポート時のデフォルトのインポート対象ファイルを設定

- @names: 対象ファイルの種類
 - txt: テキストファイル
 - xhtml: HTML, および, XHTML ファイル
 - xml: XML ファイル
- 例

```
<target_file_type names="txt,xhtml" />
```

↑

import / char_normalization 要素 ↑

インポート時のデフォルトの文字正規化方法を設定

- @name: 正規化方法
 - none: なし
 - user_defined: ユーザ定義
 - nfkc: NFKC (Unicode)
- 例

```
<char_normalization name="none" />
```

↑

import / char_conversion_table 要素 ↑

インポート時のテキスト変換用テーブルの設定。import/char_normalization[@name="user_defined"]の時に使用される。文字の対応は @from と @to で、先頭から1文字ずつが対応するように定義する。

- @from: 変換前
- @to: 変換後
- 例 (数字の半角⇒全角変換)

```
<char_conversion_table from="0123456789" to="0 1 2 3 4 5 6 7 8 9" />
```

↑

import / text_transformation_definition 要素 [↑](#)

インポート時のテキスト置換用定義ファイルを指定

- @dir: 定義ファイルが存在するフォルダ
- @default: デフォルトの定義ファイル名
- 例

```
<text_transformation_definition dir="resources/htd" default="aozora.htd" />
```

- 定義ファイルの仕様
 - 例

```
# 改行位置に、<br />を挿入
¥n      <br />¥n
# ルビ(範囲指定あり)
[ ](.+?)《(.+?)》      <r rt="$2">$1</r>
```

- 置換規則は、先頭から順番に適用される。
- 定義ファイルの文字コードは、UTF-8 で記述する。
- 置換対象の文字列は正規表現で指定する。正規表現は Java の [java.util.regex.Pattern クラス](#) に準じる。
- 置換対象、置換文字列はタブで区切る。置換文字列には、\$1, \$2 などの前方参照値を使うことができる。詳細は、[java.util.regex.Matcher クラス](#) の解説を参照のこと。
- 先頭が # で、タブの数が一つでない行は、コメントとみなされる。

↑

import / xhtml_style_sheet 要素 [↑](#)

インポート時の XHTML ファイル用のスタイルシートを指定

- @dir: スタイルシートファイルが存在するフォルダ
- @default: デフォルトのスタイルシートファイル
- @isTidied: HTML⇒XHTML変換の可否のデフォルト値
 - true: 変換する
 - false: 変換しない
- 例

```
<xhtml_style_sheet dir="resources/xsl/xhtml" default="xhtml2xml_aozora.xsl" isTidied="true" />
```

↑

import / xml_style_sheet 要素 [↑](#)

インポート時の XML ファイル用のスタイルシートを指定

- @dir: スタイルシートファイルが存在するフォルダ
- @default: デフォルトのスタイルシートファイル
- 例

```
<xml_style_sheet dir="resources/xsl/xml" default="xml.xsl" />
```

↑

import / as_subcorpora 要素 (since 1.5β04) [↑](#)

インポート対象フォルダの直下のフォルダをサブコーパスとしてインポートする。

- @value:
 - false: サブコーパスとしない(default)
 - true: サブコーパスとする
- 例

```
<include_subcorpora value="true" />
```



import / not_now_indexing 要素 (since 1.5β04) [↑](#)

インポート時にすぐ索引づけしないように設定する。

- @value:
 - true: すぐ索引づけしない
 - false: すぐ索引づけする(default)
- 例

```
<not_now_indexing value="true" />
```



import / source_files 要素 (since 1.5β04) [↑](#)

インポート時にコピーするファイルを指定する。なお、ファイルのパスの起点は、ユーザが指定したインポート対象のフォルダである。

- @corpus_dir: コーパスのソースファイルを格納したフォルダ
- @corpus_name: コーパス名
- @config_file1: 設定ファイル 1 (必須)
- @config_file2: 設定ファイル 2 (任意)
- @xslt_dir: xslt のフォルダ
- @aux_dir: 補助フォルダ
- 例

```
<source_files corpus_dir="test_src" corpus_name="test"
  config1="config.test.xml" config2="config.test.db.xml"
  xslt_dir="xslt" aux_dir="aux" />
```



アノテーション関連 [↑](#)



annotator 要素 (since 1.5β) [↑](#)

アノテーションをするための外部プログラムの定義

- li 要素 ... 一つの外部プログラム
 - @name: 外部プログラム名
 - @os: 対象のOS
 - @annotation: アノテーション名 (field_setting/li/@type="db" のとき, field_setting/li/@element で指定する名前となる)
 - @command: 外部プログラムへのパス (パス中に「Program Files」を含み, 当該のコマンドが存在しない場合は, 「Program Files (x86)」のパスでも実行を試みます)
 - @option: 外部プログラム実行時のオプション
 - @encoding: 外部プログラムが入力として想定する文字コード (since 1.6)
- li/extract ... コーパス中でアノテーションの対象となる要素
 - @name ... 要素名
 - @attribute ... 要素の属性名
 - @value ... 属性値
 - 例

```
<extract element="" attribute="" value="" /> ... 全要素を対象とする
<extract element="s" attribute="" value="" /> ... s 要素を対象とする
```

```
<extract element="引用" attribute="種別" value="会話" />
... 「引用」要素のうち、「種別」属性が「会話」のものを対象とする
```

- 注意
 - @attribute を指定しない場合、@name で指定した任意の要素が処理対象になる
 - 指定する要素は、連続しているか、改行で区切られていなければならない。この条件が満たされない場合、アノテーションに失敗する可能性がある。
- li/chunk ... extract 要素で指定されたコーパス中の要素を外部プログラムが処理する単位を決定する。
 - @delimiter ... 単位分割するための正規表現
 - @maxlength ... 1処理単位に含まれる最大文字数（これを超えた場合、強制的に分割される）
 - 例（最大200文字で、「。」または「？」で区切られる単位を1処理単位とする場合）

```
<chunk delimiter="【. ?】+" maxlength="200" />
```

- li/result_fields ... 外部プログラムの出力の形式を定義する。
 - @delimiter ... 外部プログラムの出力結果のフィールド区切り文字
 - li/@name ... フィールド名
 - li 要素の順序は、外部プログラムの出力結果のフィールド順と対応する
 - field_setting/li/@type="db" のとき、field_setting/li/@attribute で指定する名前となる
 - "_TEXT"は、解析対象の文字列(形態素解析の場合は、出現形に相当)を表す。出力結果にはこのフィールドを必ず含まなければならない。
 - "_used"で始まる場合、辞書のフィールドとして追加しない。(since 1.6)
 - li/@isIndexed ... field_setting/li/@type="db" 用の索引の有無
 - true: 索引あり
 - false: 索引なし(default)
 - li/@contextLength ... 前後要素長 (since 1.6)
 - 当該要素の前後n個分のフィールドを検索結果に追加する
 - @name="基本形", @contextLength=2のとき、四つのフィールド「基本形-2」「基本形-1」「基本形1」「基本形2」を検索結果に追加する
 - 例

```
<li name="_TEXT" isIndexed="true" />
<li name="読み" />
<li name="基本形" />
<li name="品詞" />
<li name="細品詞" />
<li name="活用型" />
<li name="活用形" />
```

「ひまわり」資料参照関連 [↑](#)

manual 要素 [↑](#)

「ひまわり」マニュアルの URL の定義

- @url: 「ひまわり」マニュアルの URL (なお、URL にプロトコルが指定されていない場合は、ファイルとみなし、file:// とパスを付加する)
- 例

```
<manual url="manual/index.html" />
```

hp 要素 [↑](#)

「ひまわり」ホームページの URL の定義

- @url: 「ひまわり」ホームページの URL
- 例

```
<hp url="http://mimir.corpus.rdl.local/resource/index.php" />
```

パッケージ設定ファイル [±]

パッケージ設定ファイル(.himawari_package_info)は、インストール機能([ファイル]⇒[インストール], ver.1.5 以降)に対応したパッケージに同梱されるもので、インストールするファイルのリストが記述される。

インストール設定ファイルの仕様 [±]

- パッケージ設定ファイルの名前は、.himawari_package_info とし、パッケージを含むディレクトリのルートに設置するものとする。
- パッケージ設定ファイルの文字コードは、UTF-8 とする。ただし、文字コードに起因するコピー時の問題を避けるため、ISO/IEC 646 の範囲で記述することを推奨する。
- パッケージ設定ファイルには、『ひまわり』フォルダにコピーするファイル、及び、フォルダを列挙する。
 - コピーするファイル、および、フォルダの指定は、『ひまわり』フォルダを起点とする相対パスで指定する（下記の例を参照のこと）。
 - コピー先は、次の場所に限定する。
 - Corpora フォルダ内
 - resources フォルダ内
 - 『ひまわり』フォルダ直下(この場合は、『ひまわり』設定ファイルのみ)
 - フォルダを指定した場合、フォルダに含まれるすべてのファイルがコピーされる。
- 『ひまわり』設定ファイルがパッケージに同梱されていれば、インストール成功時に自動的に読み込まれる。なお、『ひまわり』設定ファイルが複数ある場合は、最初に指定されたものが読み込まれる。

インストール設定ファイルの例 [±]

『分類語彙表』パッケージ [±]

```
Corpora/Bunrui  
config_bunrui.xml
```

- パッケージの Corpora/Bunrui と config_bunrui.xml が『ひまわり』フォルダにコピーされる。
- パッケージインストール後、config_bunrui.xml が読み込まれる。

『青空文庫』パッケージ [±]

```
Corpora/Aozora  
config_aozora.xml  
config_aozora.db.xml
```

- パッケージの Corpora/Aozora, config_aozora.xml, config_aozora.db.xml が『ひまわり』フォルダにコピーされる。
- パッケージインストール後、config_aozora.xml が読み込まれる。

Last-modified: 2018-07-11 (水) 18:14:45 (50d)

Site admin: [anonymous](#)

PukiWiki 1.4.7 Copyright © 2001-2006 [PukiWiki Developers Team](#). License is [GPL](#).
Based on "PukiWiki" 1.3 by [yu-ji](#). Powered by PHP 5.1.6. HTML convert time: 0.144 sec.