



S-Fields XML Mapping

Insuma GmbH

26th April 2005

Insuma reserves the right to make changes in the specifications and other information contained in this publication without prior notice. In case of doubt, the reader should consult Insuma to determine whether any such changes have been made. The software described in this document is furnished under a license and may be used or copied in accordance with the terms of such license.

Trademarks

Insuma and its logo are registered trademarks of Insuma GmbH. All other product and company names are either trademarks or registered trademarks of their respective companies.

Insuma GmbH
Sand 13
72076 Tübingen
Germany
Phone: +49 (7071) 3 65 96 10
Fax: +49 (7071) 3 65 96 14
E-mail: info@insuma.de
<http://www.insuma.de>

Contents

1	Definitions	5
2	Formal definition of splitting policy	6
3	List of standard s-fields	7
4	XML query examples	8
4.1	Search for single word	8
4.2	Search for multiple words	8

Executive summary

This document is an extension to the XML handbook. It describes the support for splitted fields (so called s-fields), that offer a better search performance for complex queries. It includes a formal description of how the words are splitted and some example queries with s-fields as well. This document should be used when you are changing the queries to use s-fields in addition to (or instead of) standard queries described in the XML handbook. This document is targeted to technical specialists in the companies - clients of Insuma GmbH.

1 Definitions

When the XML queries you send to the search engine include a lot of conditions (search terms related per AND by default in your search form, frequent search in categories and similar) it is advisable to switch to using the s-fields instead of standard queries. Such s-queries can be little more complicated to program, although may bring significant improvement in performance. Using s-fields means that you gain more control over the search index of InsumaFocus, i.e. use the index more directly.

If a usual text field is called title, then the corresponding s-field will be called stitle. S stands for string. An example:

The content of the title is:

```
<title>Foo bar</title>
```

then the s-fields provided will include:

```
<stitle>foo</stitle>
```

```
<stitle>bar</stitle>
```

The s-fields allow to apply the eq predicate to the contents of such fields. Though you have to care that you split the words in the query prior to sending it to the search engine, as described in the next chapter.

2 Formal definition of splitting policy

The words in the text are splitted according to the following regular (Python) expression (see [PytRE05] for further details on Python regular expressions):

```
r"[\w*]+(?:' - [\w*]+)*"
```

where:

- [...] \equiv used for specifying a character class, which is a set of characters that you wish to match,
- \w \equiv [a-zA-Z0-9_],
- (?:foo) \equiv non-capturing group containing the subexpression foo,
- + \equiv pattern has to occur 1 - ∞ ,
- * \equiv pattern has to occur 0 - ∞ .

The closest English language wording of this policy (though not 100% correct) would be: “The text is splitted in words, every continious sequence of alphabetic characters and numbers is considered a word”.

3 List of standard s-fields

This is a list of the standard s-fields. They are derived from the standard fields coming from HTML as described in XML Handbook. Note that the *_morpho*, *_typo* and *_phonetic* fields are only available if the morpho, typo and/or phonetic modules are installed:

- stitle
- stitle_morpho
- stitle_typo
- stitle_phonetic
- sheadings
- sheadings_morpho
- sheadings_typo
- sheadings_phonetic
- skeywords
- skeywords_morpho
- skeywords_typo
- skeywords_phonetic
- sdescription
- sdescription_morpho
- sdescription_typo
- sdescription_phonetic
- sbody
- sbody_morpho
- sbody_typo
- sbody_phonetic

4 XML query examples

Here are some examples for queries that utilize the s-fields accordingly:

4.1 Search for single word

If you want to search for documents that contain the word SMS:

```
<?xml version="1.0" encoding="utf-8"?>
<query max_results="100" show_attrs="score summary title">
  <condition attr="sbody" predicate="eq" value="SMS"/>
</query>
```

4.2 Search for multiple words

If you want to search for documents that contain the words SMS *and* Internet:

```
<?xml version="1.0" encoding="utf-8"?>
<query max_results="100" show_attrs="score summary title">
  <condition attr="sbody" predicate="in" value="SMS Internet"/>
</query>
```

If you want to search for documents that contain the words SMS *or* Internet:

```
<?xml version="1.0" encoding="utf-8"?>
<query max_results="100" show_attrs="score summary title">
  <or>
    <condition attr="sbody" predicate="eq" value="SMS"/>
    <condition attr="sbody" predicate="eq" value="Internet"/>
  </or>
</query>
```

Bibliography

[PytRE05] A.M. Kuchling: *Regular Expression HOWTO*. <http://www.amk.ca/python/howto/regex/>, 2005.