Last update: 20 Apr 2002

File name:
health-info-format.rtf

SALUT

IST-2000-25026

REPORT D6.1A Revision 0.9

Common Health Information Format

Written by:
Jacob Palme

Contractual Date of Delivery: 30 April 2002
Last Revision: 20 Apr 2002

Nature of the Deliverable: SP

Deliverable Type: PU

Workpackage WP3
Responsible Partner: KTH

1.1 Table of contents

***Type: PU-public, LI-limited, RP-restricted**

****Nature: PR-Prototype, RE-Report, SP-Specification, TO-Tool, OT-Other**

1.2 Executive Summary

There are many different web sites providing health information to laymen. Some are specialized, and give detailed information about particular illnesses, other are general, but sometimes more shallow. Combining the information from many such sites into larger health information sites might provide better services to users. Also, different such services might reuse content developed by other services.

This document describes a standard format for exchange of such health informational pages between different services. The format is based on XML and XHTML and on the Dublin Core metadata initiative [Becket et al 2000] and on the World Wide Web Consortium Resource Description Framework (RDF) [Brickley, D. et al 2000], [Lassila, Ora et al 1999].

You may feel that the format specified here is rather complex. However, we will produce templates which make it easy to produce such pages using Dreamweaver or some other web editor, and translation programs which will translate from the Dreamweaver format to the format described here. This program will also automatically correct the markup to agree with the requirements specified here. With this program, it will thus be easy to produce documents in this format by using an ordinary web editor like Dreamweaver, Frontpage, Golive, etc. An example of such a template is given in Attachment A on page 19.

1.3 Introduction

The European Union (EU) is funding several research projects in the area of health information to non-specialists. Several of these projects will produce health informational documents. If each of these projects could easily use health informational documents developed by the other projects, more useful results could be achieved at lower costs.

The intention of this document is to specify a format for health informational pages, which makes it easy to exchange such pages between different services providing health information. This format should have the following properties:

1. Be based on XML, since XML is a widely accepted format for exchange of documents between different providers.
2. Be suitable for delivering to users both via ordinary World Wide Web (WWW) browsers on personal computers and workstations and via corresponding tools on Personal Digital Assistants (PDAs).
3. Give a simple logical description of the pages, so that different delivery gateways can format the pages in ways suitable for each delivery gateway.
4. Allow the freedom of expression provided by HTML and the WWW, including the combination of text and graphics.
5. Be compatible with the Dublin Core Metadata format [Wiebel et al 1998, Kunze 1999, Becket et al 2000] and with the Resource Description Framework [Brickley, D. et al 2000], which is a common format for searching for and describing documents which has widespread acceptance.

1.4 Attributes

The health informational format is based on the Dublin Core Metadata format [Wiebel et al 1998, Kunze 1999, Becket et al 2000], with a few additional fields and on the Resource Description Framework [Brickley, D. et al 2000].

Here is a list of all the fields in the health informational format and a description of their content (Req. indicates if this field is mandatory or not):

Field name	Req.	Description of content
`dc:language`	Yes	The language of the health informational page, using language codes according to [RFC 3066] which is based on [ISO 639]. Examples of values: en English fr French de German da Danish hu Hungarian it Italian Note: If the same text is provided in more than one language, then a <dc:language> element is placed to indicate the switch from one language to another. All fields following a <dc:language> element will be in the new language. Example: **`<dc:language>`**`en`**`</dc:language>`**

Field name	Req.	Description of content
		nl Dutch el Greek es Spanish sv Swedish
dc:title	Yes	A descriptive informational title, which tells a user whether this is the page wanted or not. (Note: This definition of dc:title is somewhat different from that in Dublin Core.)
dc:subject	No	A list of keywords, suitable to put into a <META name="keywords" content="..."> element when shown as HTML. Example: `<dc:subject>sugar, sweet, ice cream, chocolate, toffee, bonbon, honey, pastry, cake, biscuit, cookie, bun, cooky</dc:subject>`
dc:description	No	A short description of the page, suitable to put into a <META name="description" content="..."> element when shown as HTML. Should contain a suitable text which a search engine will show in its list of matches. Example: `<dc:description>A summary of medical knowledge about sugar.</dc:description>`
dc:type	No	See Dublin Core. For our purposes, the value for HTML content should be `<dc:type>text</dc:type>` A page which mostly consists of images or video might be tagged with a different value in this field.
dc:publisher	Yes	The organisation which produced this informational page. Should contain a hyperlink to a page describing this organisation. Example for a page produced by the Swedish partners of SALUT: `<dc:publisher><a href="http://salut.nu/en/swedish/"> SALUT Sweden</a></dc:publisher>`
dc:date-created	No	The date when this informational page was created, in the date format: 2001-06-19. Example: `<dc:date-created>2001-06-19</dc:date-created>`
dc:format	No	Should always have the value: `<dc:format>text/html</dc:format>`

Field name	Req.	Description of content
dc:identifier	Yes	Should contain a globally unique URL identifying this page. Example: **<dc:identifier>**http://salut.nu/en/answers/food-sugar.xml **</dc:identifier>** The part after the last "/" in the URL ("food-sugar.xml" in the example above) is the file name and the part before the extension "food-sugar" is the proper name part of this name. Note 1: This file name should not be longer than 32 characters including a four letter extension like ".htm". Note 2: This string should be different from all other health informational pages produced by the same organisation. The page might be accessible in HTML format on the WWW from the URL indicated. Note 3: In some data formats, only the file name is stored for each page. This file name can be converted to a full identifier by adding a path in front of the file name.
hi:question	No	Example of one or more questions which a user might write and to which this page would be the appropriate answer. Examples: **<hi:question>**Will sugar make me fat?**</hi:question>** **<hi:question>**Will sugar make me fat?/What should I eat?**</hi:question>**
hi:required	No	A list of search keywords for this page in the format described in chapter 1.6 on page 11.
hi:priority	No	A list of search keywords, which will cause this page to be listed in FAQ results before pages without a priority keyword, in the format described in chapter 1.6 on page 11.
hi:optional	No	A list of search keywords for this page in the format described in chapter 1.6 on page 11.
hi:forbidden	No	A list of search keywords for this page in the format described in chapter 1.6 on page 11.
hi:limit	No	The number of non-expected words in a user query which are allowed while still regarding the answer as a good hit. See chapter 1.6 on page 11.
dc:relation-alias	No	Instead of a **html:body**, you can refer to another of your help texts, and give its hi:identification here. See chapter 1.6 on page 11.
html:body	No[1]	The actual text of the informational page in XHTML format. See chapter 0 on page 8. To show this, the **html:body** tag should look like this: **<html:body xmlns='**http://www.w3.org/1999/xhtml**'>**
html:sources	No	A text indicating the sources for this page.

[1] Either **hi:relation-alias** or **hi:body** is mandatory.

Field name	Req.	Description of content
`hi:discuss-this`	No	URL of a forum, where users can discuss this page. (Not the forum for experts designing the page.)
`hi:ask-an-expert`	No	URL of a place, where users can ask questions answered by experts. The person asking the question is anonymous, the question is not published until the expert has written an answer.

Note: The Health Informational Format Documents may contain additional attributes at the same level as those described above. A processor who encounters an unknown attribute should ignore it. Additional attributes may be needed by special health informational systems which require additional, special information.

1.5 Usage of XHTML in Informational Pages

1.5.1 XHTML Features

Informational pages should use the XHTML format [XHTML]. In particular, note the following:

1.5.1.1 The XML element containing XHTML should have the attribute:

```
xmlns='http://www.w3.org/1999/xhtml'
```

Example 1:

```
<html:body xmlns='http://www.w3.org/1999/xhtml'>...</html:body>
```

Example 2:

```
<dc:publisher><a href="http://salut.nu/en/swedish/"
xmlns='http://www.w3.org/1999/xhtml'>
SALUT Sweden</a></dc:publisher>
```

1.5.1.2 Lower case tags

All HTML tags must use lower case. Example:

Right	Wrong
`<p>This is a paragraph.</p>`	`<P>This is a paragraph.</P>`

Also CSS elements and values must be lower case.

1.5.1.3 <br>, <hr> and <img> tags

The HTML `<br>` tag must be written `<br />`. Same for the `<hr>`, `<img>` tag and other tags which have no corresponding closing tag. Example:

```
<img src="/images/salad-sauce.jpg" width="55" height="113" align="left" vspace="5"
hspace="5" />
```

Here is a complete list of all tags which have this property in HTML 4.0:

```
"br", "meta", "link", "area", "img", "param", "element", "input", "col", "base",
"basefont", "hr", "frame", "isindex"
```

1.5.1.4 <p> Tag

All tags which can be closed must be closed. Example:

Right	Wrong
<p>First paragraph. <p>Second paragraph.	<p>First paragraph.</p> <p>Second paragraph.</p>
<li>First item</li> <li>Second item</li>	<li>First item <li>Second item

1.5.1.5 <pre> Tag

The <pre> tag may not contain <img>, <object>, <big>, <small>, <sub> and <sup> elements.

1.5.1.6 name and id attributes

Wherever a name attribute is specified, also specify the same value as an id attribute. Note that no two elements are permitted to have the same value of the id attribute. Example:

```
<a name="enter" id="enter"></a>
```

1.5.1.7 Always enclose attribute values in double apostrophes

Example:

Right	Wrong
<img src="/images/salad-sauce.jpg" width="55" height="113" align="left" vspace="5" hspace="5" />	<img src="/images/salad-sauce.jpg" width=55 height=113 align=left vspace=5 hspace=5 />

1.5.2 Additional HTML Recommendations

1.5.2.1 Reasons for these recommendations

Every service which uses the health informational texts should be allowed to reformat them according to its conventions. For particular, a WAP server might format text in a simpler format which suits small screens. Because of this, avoid formatting if not necessary.

1.5.2.2 Do not use style sheets

Do not use style sheets, every service should be able to apply its own style sheets. You can however use the following CSS class selectors, but every service can give their style sheet definitions of these classes:

CSS class	class=mini
Usage	For text which should be rendered especially small. Can also be used for links, so that hey will not dominate too much.
Example of possible CSS specification (but every service decides this itself)	`.mini { font-family: Verdana, Arial, Geneva, sans-serif; font-size: 10px }`

CSS class	class=list
Usage	For lists of items which should be rendered with a smaller-than-normal paragraph separation.
Example of possible CSS specification (but every service decides this itself)	`.list { line-height: 3ex }`

CSS class	class=more
Usage	For links.which should not dominate too much, for example the word "more" as a link to where more information can be found.

Example of possible CSS specification (but every service decides this itself)	`.more {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px; color: #666666}`
	`A.more:link {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px; color: #666666}`
	`A.more:visited {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px; color: #CC6633}`
	`A.more:hover {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px; color: #990000}`

1.5.2.3 Avoid fancy formatting

Avoid fancy formatting. For example, the following coding is often used in HTML to get a series of paragraphs with no space between the paragraph and text indentation at the start of the paragraph:

```
<p>First paragraph: The rain in Spain stays mainly in the plain. The big brown fox
jumped over the lazy dog.<br>
&nbsp;&nbsp;&nbsp;Second indented paragraph: The rain in Spain stays mainly in the
plain. The big brown fox jumped over the lazy dog.</p>
```

Instead just use ordinary **<p>** tags. Every service can convert **<p>** tags to **<br />&nbsp;&nbsp;&nbsp;** if it so wants, but this is something every service does as it wants, the exchange format should just use <**p**> tags.

The example above should thus instead be:

```
<p>First paragraph: The rain in Spain stays mainly in the plain. The big brown fox
jumped over the lazy dog.</p>
<p>Second indented paragraph: The rain in Spain stays mainly in the plain. The big
brown fox jumped over the lazy dog.</p>
```

1.5.2.4 Use <em> and <strong> instead of <i> and <b>

To allow every service to decide its rendering of the HTML text, use the `<em>` and `<strong>` tags instead of the `<i>` and `<b>` tags.

1.5.3 Use of Hyperlinks in Health Info Pages

You may want to use hyperlinks in health info pages for several reasons:

1.5.3.1 Hyperlinks between health info pages

One health info page wants to refer readers to another health info page in the same collection of health info pages for more information. In this case, write the hyperlink as a simple relative hyperlink with the file name part of the `dc:identifier` of the referred to page with "`.htm`" as the file extension.

Example: If you have a page with the following identifier:

```
<dc:identifier>http://salut.nu/en/answers/food-sugar.xml</dc:identifier>
```

and you want to refer to this in another health info page, you can for example write:

```
<a href="food-sugar.htm">Read more about sugar</a>
```

1.5.3.2 External hyperlinks

Hyperlinks outside of your link collection are just written as normal full absolute URLs.

1.5.3.3 Links to in-line graphics and other embedded objects

If the in-line graphic is language-independent, put it into a folder named *images*, and link to it with relative URLs of the format "`/images/pict.gif`". Example:

```
<img src="/images/salad-sauce.jpg" width="55" height="113" align="left" vspace="5"
hspace="5" />
```

If the in-line graphic is language-dependent, for example because it contains text in the graphic, then put it into a folder named en/images where en is the language, and refer to it with URLs the format "`en/images/pict.gif`". Example:

```
<img src="/images/en/salad-sauce.jpg" width="55" height="113" align="left"
vspace="5" hspace="5" />
```

In such a case, you will of course have to provide a separate image file for each language.

1.6 Format of Search Keyword Lists

Below is a description of the Search Keyword Lists of type **hi:required**, **hi:optional** and **hi:forbidden**. The use of this format is optional, but it is used by the natural-language question-answering system which SALUT [Sneiders, E. 1999A] will develop, and which is available to all the clustered partners. Partners which do not use the natural-language question-answering system need not specify these fields.

1.6.1 hi:required

In the natural-language question-answering system, these terms must exist in the user question for a good match.

Example:

```
<hi:required>sugar* sweet*; fat obese*</hi:required>
```

This says that the user query must contain either a word beginning with “sugar” or a word beginning with “sweet” plus either the word “fat” or a word beginning with “obese”. Note that space separates synonyms, and ";" separates different required terms.

1.6.2 hi:optional

These are terms, which can but need not occur in a query using the natural-language question-answering system. Example:

```
<hi:optional>good healthy non-healthy dangerous* bad harm* health* fit vigorous
well* strong beneficial recommend* nourishing wholesome unsafe* hazard* risk* safe*
damag* hurt* unsafe*</hi:optional>
```

1.6.3 hi: forbidden

These are terms which will stop a query from matching. Use sparingly. They are mostly needed if you have two different answers to similar, but not identical questions, and want to separate the two answers.

1.6.4 hi:limit

If the number of non-expected words in the query is lower than this limit, the match of the answer to the query will be higher than with more unexpected words.

1.6.5 dc:relation-alias

If you want to specify several questions with the same answer, you need specify the **`html:body`** of the answer for only one of the questions. The other questions can contain an empty **`html:body`** and instead contain an **`dc:relation-alias`** with its value equal to the dc:identifier of the answer to be used.

1.6.6 Word endings

The convention `health*` with an asterisk at the end of a term, indicates that any word which begins with “health” is to be matched.

1.6.7 Phrases

A phrase (a term consisting of more than one word) is specified in the following way:

```
[binge*; eat*]
```

where a semicolon separates the words in the phrase. Each word can have synonyms, for example:

```
[binge* spree* splurge*; eat* consum* gobbl*]
```

will match both “binge eating” and “splurge consumption”.

If other words are allowed in-between, replace the “;” with “#”. Example:

```
[many # rash*]
```

This will match both “many rashes” and “many red rashes”.

1.6.8 Synonym List

If some synonyms occur often in many questions, you can put them on a synonym list, and need then not repeat all the synonyms every time they are needed.

If, for example, many questions contain the following words, which can be regarded as synonyms when answering health questions:

```
non-healthy dangerous* bad harm* health* unsafe* hazard* risk* damag*
```

Then instead of repeating this list in every query, it can be defined in the synonym list, given, for example, the name `$bad`, and then referred to in the health informational pages.

1.6.8.1 An example of a synonym list

```
<synonym-table>
<term>
<shortcut>bad</shortcut>
<synonyms>non-healthy dangerous* bad harm* health* unsafe* hazard*
risk* damag*</synonyms>
</term>
<term>
... ...
</term>
</synonym-table>
```

1.7 The DTD for the Health Info XML format

Here is a draft DTD for the XML info format. Certain problems are still not resolved, see the list of problems in the beginning of chapter 1.8 on page 15.

This DTD can be found at http://salut.nu/health/documentation/. That location also contains health-info.dtd and health-info-big.dtd. health-info-big.dtd is an extended version with HTML syntax included.

```
<!--

  DRAFT XML DTD 20010702 for Health Info Pages version 0.1
  http://salut.nu/FAQ-entries/xml-entries/health-info.dtd

  See
    Common Health Information Format
    http://salut.nu/health/documentation/
  Authors:
    Jacob Palme <jpalme@dsv.su.se>

-->

<!-- The namespaces for RDF and DCES 1.1 respectively -->
<!ENTITY rdfns 'http://www.w3.org/1999/02/22-rdf-syntax-ns#' >
<!ENTITY dcns 'http://purl.org/dc/elements/1.1/' >
<!ENTITY hins 'http://salut.nu/health/documentation/' >

<!-- Magic - do not look behind the curtain -->
<!ENTITY % rdfnsdecl 'xmlns:rdf CDATA #FIXED "&rdfns;"' >
<!ENTITY % dcnsdecl 'xmlns:dc CDATA #FIXED "&dcns;"' >
<!ENTITY % hinsdecl 'xmlns:hi CDATA #FIXED "&hins;"' >


<!-- The wrapper element -->
<!ELEMENT rdf:RDF (rdf:Description)* >

<!ATTLIST rdf:RDF %rdfnsdecl; %dcnsdecl; %hinsdecl; >


<!ENTITY % hies "dc:language | dc:identifier | dc:title |
 dc:subject | dc:description | dc:date-created | dc:format |
 dc:type | dc:publisher | hi:question | hi:required | hi:optional |
hi:forbidden | hi:limit | dc:relation-alias | html:body" >


<!-- The resource description container element -->
<!ELEMENT rdf:Description (%hies;)* >

<!ATTLIST rdf:Description about CDATA #REQUIRED>

<!-- The elements from DCES 1.1 -->

<!-- The name given to the resource. -->
<!ELEMENT dc:title (#PCDATA)>

<!-- An entity primarily responsible for making the content of the
resource. -->
<!ELEMENT dc:creator (#PCDATA)>

<!-- The topic of the content of the resource. -->
<!ELEMENT dc:subject (#PCDATA)>

<!-- An account of the content of the resource. -->
<!ELEMENT dc:description (#PCDATA)>

<!-- The entity responsible for making the resource available. -->
<!ELEMENT dc:publisher (#PCDATA)>

<!-- An entity responsible for making contributions to the content of
the resource. -->
<!ELEMENT dc:contributor (#PCDATA)>

<!-- The creation date of the resource. -->
<!ELEMENT dc:date-created (#PCDATA)>

<!-- The nature or genre of the content of the resource. -->
<!ELEMENT dc:type (#PCDATA)>
```

```
<!-- The physical or digital manifestation of the resource. -->
<!ELEMENT dc:format (#PCDATA)>

<!-- An unambiguous reference to the resource within a given context.
-->
<!ELEMENT dc:identifier (#PCDATA)>

<!-- A Reference to a resource from which the present resource is
derived. -->
<!ELEMENT dc:source (#PCDATA)>

<!-- A language of the intellectual content of the resource. -->
<!ELEMENT dc:language (#PCDATA)>

<!-- A reference to an answer, for which this is an alias question. --
>
<!ELEMENT dc:relation-alias (#PCDATA)>

<!-- The extent or scope of the content of the resource. -->
<!ELEMENT dc:coverage (#PCDATA)>

<!-- Information about rights held in and over the resource. -->
<!ELEMENT dc:rights (#PCDATA)>

<!-- Added elements for the health info format -->

<!-- Example of a question which a user may put to get this answer. --
>
<!ELEMENT hi:question (#PCDATA)>

<!-- Required keywords in prioritized keyword matching. -->
<!ELEMENT hi:required (#PCDATA)>

<!-- Optional keywords in prioritized keyword matching. -->
<!ELEMENT hi:optional (#PCDATA)>

<!-- Forbidden keywords in prioritized keyword matching. -->
<!ELEMENT hi:forbidden (#PCDATA)>

<!-- Limit no. of non-envisaged words in prioritized keyword matching.
-->
<!ELEMENT hi:limit (#PCDATA)>

<!-- The XHTML text of the answer. -->
<!ELEMENT html:body ANY>
```

1.8 A Complete Example

Here is an example of a health information page encoded according to the standard specified in this document. This example contains the same text in two languages, English and Swedish, in order to show how texts in multiple languages can be specified.

This example can be found as http://salut.nu/FAQ-entries/xml-entries/health-info-v3.xml.

1.8.1 List of unresolved problems:

Note: This example is not yet 100 % correct. Most of the problems are probably in the DTD (see section 1.7 on page 13). The problems caused multiple error messages from the XML validator at http://www.stg.brown.edu/service/xmlvalid/.

Line	Problem description
5	Is this correct?
6	Should the about attribute refer to all language versions or be separate for each language version
7	Is this the correct way to represent multiple language versions of the same document and document description?
16-17	How can a HTML snippet be put into an XML value?
24-55	How can XHTML be put into an XML value without getting lots of error messages from the XML validator?

1.8.2 The complete example itself:

The latest updated version of this example can be found at http://salut.nu/faq-entries/xml-entries/health-info.xml.

```
1.   <?xml version="1.0"?>
2.   <!DOCTYPE rdf:RDF SYSTEM "http:salut.nu/health/documentation/health-info.dtd">
3.   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.            xmlns:dc="http://purl.org/dc/elements/1.1/"
5.            xmlns:hi="http://salut.nu/health/hi-format.html">
6.   <rdf:Description about="http://salut.nu/FAQ-entries/>
7.   <hi:translation xml:lang="en">
8.   <dc:language>en</dc:language>
9.   <dc:identifier>http://salut.nu/en/answers/food-sugar.xml</dc:identifier>
10.  <dc:title>Sugar</dc:title>
11.  <dc:subject>sugar, sweet, ice cream, chocolate, toffee, bonbon, honey, pastry,
     cake, biscuit, cookie, bun, cooky</dc:subject>
12.  <dc:description>A summary of medical knowledge about sugar.</dc:description>
13.  <dc:date-created>2001-06-19</dc:date-created>
14.  <dc:format>text/html</dc:format>
15.  <dc:type>text</dc:type>
16.  <dc:publisher><a href="http://salut.nu/en/swedish/"
     xmlns='http://www.w3.org/1999/xhtml'>
17.  SALUT Sweden</a></dc:publisher>
18.  <hi:question>Will sugar make me fat?</hi:question>
19.  <hi:priority>sugar*</hi:required>
20.  <hi:required>sugar* sweet* [ice;cream] chook* toffee* bonbon* honey* pastry*
     cake* biscuit* cookie>* bun cooky*</hi:required>
21.  <hi:optional>good healthy non-healthy dangerous* bad harm* health* fit vigorous
     well* strong beneficial recommend* nourishing wholesome unsafe* hazard* risk*
     safe* damag* hurt* unsafe*</hi:optional>
22.  <hi:forbidden></hi:forbidden>
23.  <hi:limit>1</hi:limit>
24.  <dc:relation-alias></dc:relation-alias>
25.  <html:body xmlns='http://www.w3.org/1999/xhtml'>
26.  A healthy person's body has the ability to tell you when
27.  you have eaten enough. For a person, where this ability works well, you
```

```
will feel satisfied when you have eaten what your body needs. You will then
achieve a normal, healthy and good-looking body.
<p>For some people, this ability does not work. They may then get too fat
or too thin or have other eating problems. Sugar confuses this ability,
so that it will not work correctly. Sugar will make you feel a need to
eat when you do not need more food.
<p>To achieve a healthy weight, you should avoid too much food which confuses
your ability to fell adequate hunger and satisfaction. You should thus
avoid all but small amounts of sugar and foods which include sugar.</p>
<p>Some studies indicate, however, that small amount of sugar in your
food leads to a slightly lower body weight than sugar-free food.</p>
<p>Sugar can also cause a kind of dependence, similar to depence to
drugs. This dependence is caused by the fact that sugar first raises the
amount of sugar in your body. This may cause you to feel high. Your body
then produces insulin, in order to reduce the amount of sugar. This may
cause you to feel low, and want to take another dose of sugar. If you
have this problem, switch to food which is taken up slower by the body,
such as full-grain bread instead of white bread.</p>
<p><img src="/images/salad-sauce.jpg" width="55" height="113" align="left" vspace="5" hspace="5" />Unfortunately,
a lot of foods contain unnecessary sugar. For example, bread and salad
sauces often contain sugar which need not be there. Most candies also
contain sugar. And bananas consist mostly of sugar. It is possible to
make good salad sauces, bread and other food without using sugar.
<p>There are different kinds of sugar. Common sugar, also known as sacharos,</p>
is especially dangerous to your ability to feel adequate hunger and satisfaction.
Many fruits contain other kinds of sugar which are less dangerous.</p>
<p>Recommended book: &quot;Potatoes not Prozac&quot;, by Kathleen DesMaisons,
<a href="http://www.simonsays.com/book/default_book.cfm?isbn=0684850141">Simon
&amp; Schuster</a>, 1998 ISBN 0-684-85014-1.</p>
<html:body xmlns='http://www.w3.org/1999/xhtml'>
</hi:sources>The experience of <a href=”http://www.palme.nu/gunbor/”>Gunborg Palme</a>, certified psychologist, certified psychiatrist and teacher and tutor of psychotherapy</hi:sources>
</hi:discuss-this>http://salut.nu/forum/uno/2/1/</hi:discuss-this>
</hi:ask-an-expert>http://salut.nu/forum/uno/2/2/</hi:ask-an-expert>
</hi:translation>
<hi:translation xml:lang="sv">
<dc:language>sv</dc:language>
<dc:identifier>http://salut.nu/sv/answers/food-sugar.xml</dc:identifier>
<dc:title>Socker</dc:title>
<dc:subject>socker, söt, glass, choko, karamell, godsak, snask, kaka, kex, godis</dc:subject>
<dc:description>Vad man vet om sockers hälsoeffekter.</dc:description>
<dc:date>2001-06-19</dc:date>
<dc:format>text/html</dc:format>
<dc:type>text</dc:type>
<dc:publisher><a href="http://salut.nu/sv/swedish/">
SALUT Sweden</a></dc:publisher>
<dc:date-created>2001-01-10</dc:date-created>
<hi:question>Blir man fet av socker?</hi:question>
<hi:required>socker* söt* glass* choko* karamell* godsak* snask* kaka* kex* godis*</hi:required>
<hi:optional>bra nyttig* hälso* hälsa* onyttig* ohälso* farlig* dålig* skadlig* frisk*</hi:optional>
<hi:forbidden></hi:forbidden>
<hi:limit>1</hi:limit>
<dc:relation-alias></dc:relation-alias>
<html:body>

```

```
<p>En frisk m&Auml;nniskas kropp har f&ouml;rm&aring;gan att tala om f&ouml;r dig n&Auml;r du har &Auml;tit tillr&Auml;ckligt. N&Auml;r denna f&ouml;rm&aring;ga fungerar bra, kommer du att k&Auml;nna dig m&Auml;tt, n&Auml;r du har &Auml;tit s&aring; mycket som din kropp beh&ouml;ver. Du f&aring;r d&Auml;rmed automatiskt en normal, frisk och snygg kropp.</p>
<p>Denna f&ouml;rm&aring;ga fungerar inte f&ouml;r vissa m&Auml;nniskor. Det kan leda till att de blir f&ouml;r feta eller f&ouml;r smala eller f&aring;r andra &Auml;tst&ouml;rningar. Socker f&ouml;rvirrar denna naturliga funktion i kroppen, genom att g&ouml;ra att du k&Auml;nner dig hungrig &Auml;ven n&Auml;r du inte beh&ouml;ver mera mat.</p>
<p>Man b&ouml;r undvika mat som f&ouml;rvirrar kroppens naturliga s&Auml;tt att s&Auml;ga till n&Auml;r man inte beh&ouml;ver mera mat. D&Auml;rf&ouml;r b&ouml;r man vara f&ouml;rsiktig med socker i maten.</p>
<p>Det finns dock vissa studier som visar att mindre m&Auml;ngder socker i maten leder till en l&Auml;gre kroppsvikt &Auml;n om man inte &Auml;ter n&aring;got socker alls.</p>
<p>Socker kan ocks&aring; orsaka en typ av beroende, som liknar beroende av droger. Detta beror p&aring; att socker f&ouml;rst h&ouml;jer blodsockherhalten. D&aring; b&ouml;rjar kroppen producera insulin, som s&Auml;nker blodsockerhalten. D&aring; kan den l&aring;ga blodsockerhalten g&ouml;ra att man k&Auml;nner sig kymig, och &Auml;ter socker igen f&ouml;r att h&ouml;ja sockerhalten i blodet.</p>
<p> <img src="/images/salad-sauce.jpg" width="55" height="113" align="left" vspace="5" hspace="5" />Tyv&Auml;rr finns socker i on&ouml;dan i m&aring;nga matr&aring;tter. Br&ouml;d och salladss&aring;ser inneh&aring;ller t.ex. ofta socker som inte beh&ouml;vde vara d&Auml;r. Vissa frukter inneh&aring;ller ocks&aring; mycket socker, t.ex. bananer. Snask, kakor och kex inneh&aring;ller ocks&aring; ofta socker. Det g&aring;r att g&ouml;ra bra salladss&aring;ser utan socker.</p>
<p>Det finns olika typer av socker. Vanligt socker, som ocks&aring; kallas f&ouml;r sacharos, &Auml;r speciellt skadligt, det skadar kroppens f&ouml;rm&aring;ga att k&Auml;nna igen hunger och m&Auml;ttnad s&Auml;rskilt mycket. I bananer &Auml;r det mest sacharos, medan andra frukter som &Auml;pplen och jordgubbar inneh&aring;ller mindre farliga sockerarter.</p>
<p>L&Auml;s mera: &quot;Potatis - inte prozac&quot;, av Kathleen Des Maisons, Bokf&ouml;rlaget Forum, 1999, ISBN 91-37-11495-6.</p>
</html:body>
</hi:translation>
</rdf:Description>
</rdf:RDF>
```

1.9 References

[Becket et al 2000] Using Dublin Core in XML, by Dave Becket, Eric Miller and Dan Brickley, DCMI Working Draft, http://dublincore.org/documents/2000/07/14/dcmes-xml/.

[Brickley, D. et al 2000] Resource Description Framework (RDF) Schema Specification 1.0, W3C Paper, http://www.w3.org/TR/rdf-schema/.

[ISO 639] ISO 639:1988 (E/F) - Code for the representation of names of languages - The International Organization for Standardization, 1st edition, 1988-04-01 Prepared by ISO/TC 37 - Terminology (principles and coordination). Note that a new version (ISO 639-1:2000) is in preparation at the time of this writing. http://www.oasis-open.org/cover/iso639a.html

[Kunze 1999] Encoding Dublin Core Metadata in HTML, by J. Kunze, Internet RFC 2731, ftp://ftp.sunet.se/pub/Internet-documents/rfc/rfc2731.txt.

[Lassila, Ora et al 1999] Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://purl.org/dc/elements/1.1/.

[RFC 3066] Tags for the Identification of Languages. by H. Alvestrand. January 2001. Internet RFC 3066, ftp://ftp.sunet.se/pub/Internet-documents/rfc/rfc3066.txt.

[Sneiders, E. 1998A] FAQ answering on WWW using shallow language understanding. *Information Systems in the WWW Environment. IFIP TC8/WG8.1 Working Conference*, 15-17 July 1998, Beijing, China, Chapman & Hall on behalf of IFIP, pp.298-319

[Sneiders, E. 1998B] Prioritized Keyword Matching of Natural Language Sentences in Database Querying. *Proceedings of the Second Conference on Information Management Methodologies*, April 1998, Växjö, Sweden

[Sneiders, E. 1999A] Automated FAQ Answering: Continued Experience with Shallow Language Understanding. *Question Answering Systems. Papers from the 1999 AAAI Fall Symposium.* Technical Report FS-99-02, November 5-7, North Falmouth, Massachusetts, USA, AAAI Press, pp.97-107, http://www.dsv.su.se/~eriks/Sneiders1999.pdf.

[Sneiders, E. 1999B] Question Answering by Automated FAQ Retrieval. *Proceedings of the Workshop on Futures in Information Systems and Software Engineering Research*, April 1999, Stockholm, Sweden

[Sneiders, E. 1999C] *Automated FAQ Answering on WWW Using Shallow Language Understanding.* Thesis in partial fulfilment of the requirements for the degree of Licentiate of Technology. Stockholm University / Royal Institute of Technology, Sweden

[Sneiders, E. and Larsson, K. 2001] Application and Maintenance Aspects of an FAQ Answering System. Report series No. 01-007, Department of Computer and Systems Sciences, Stockholm University / Royal Institute of Technology, Sweden

[Weibel et al 1998] Dublin Core Metadata for Resource Discovery by S. Weibel, J. Kunze, C. Lagoze, M. Wolf, Internet RFC 2413, ftp://ftp.sunet.se/pub/Internet-documents/rfc/rfc2413.txt.

[XHTML] XHTML™ 1.0: The Extensible HyperText Markup Language, by Steven Pemperton et al 2000, http://www.w3.org/TR/xhtml1/.

1.10 Attachment A: Template For Producing Pages

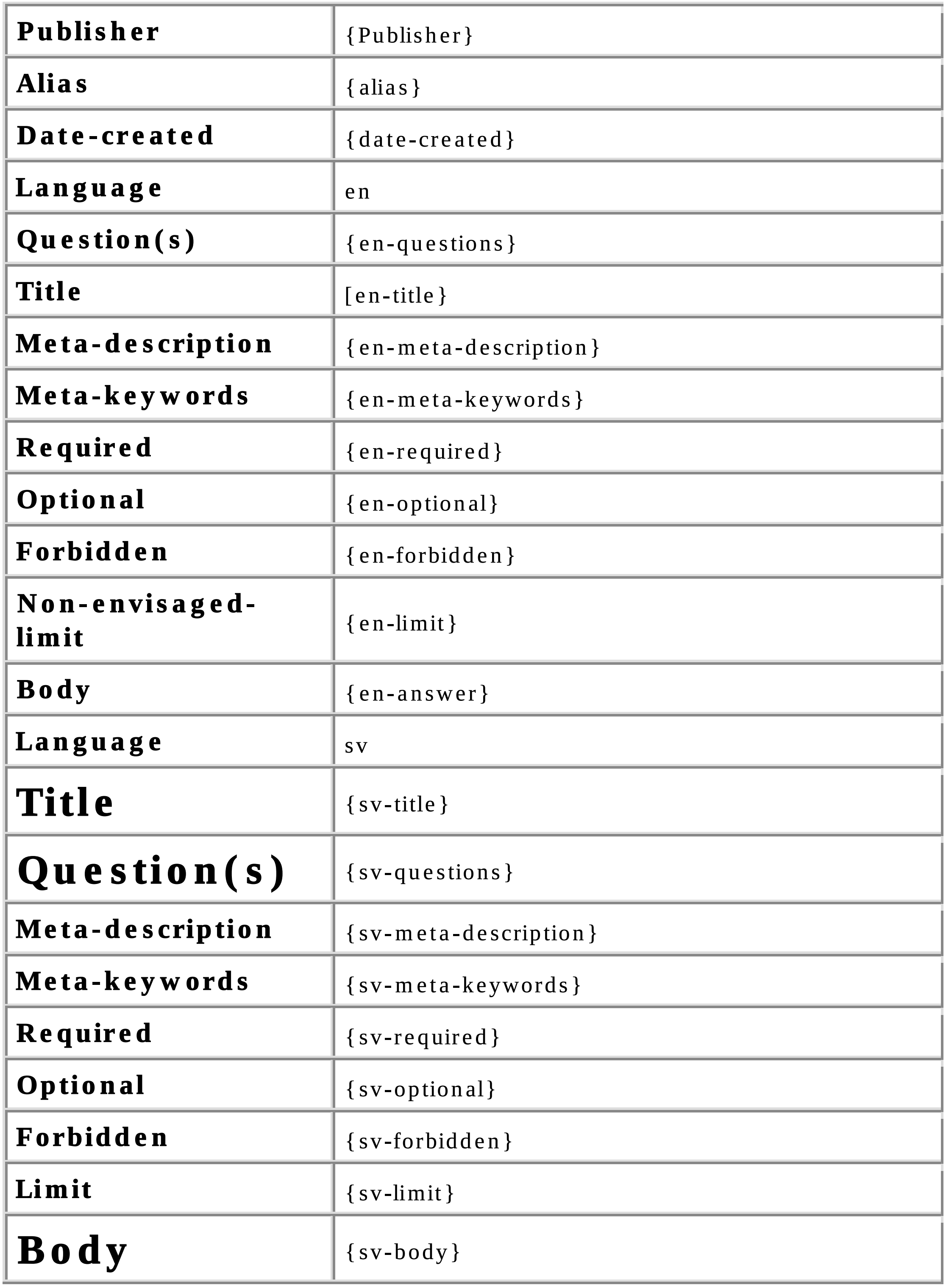
The template below will make it easy to produce pages in the format described here, if you are using a web editor like Dreamweaver, Frontpage or Golive. SALUT will produce a program to automatically convert the output with such web editors to the format described in this specification, including correction of the coding to agree with the requirements of this specification. We hope to have this program ready in September 2001.

1.10.1 An Empty Template

An empty template in Dreamweaver format can be found at http://salut.nu/Templates/FAQ-DW-entry-template.dwt. The same template in a format which will suit different web editors, not only Dreamweaver, can be found at http://salut.nu/health/documentation/FAQ-entry-template.htm.

This template only contains the languages English and Swedish, but it can easily be extended for other languages.

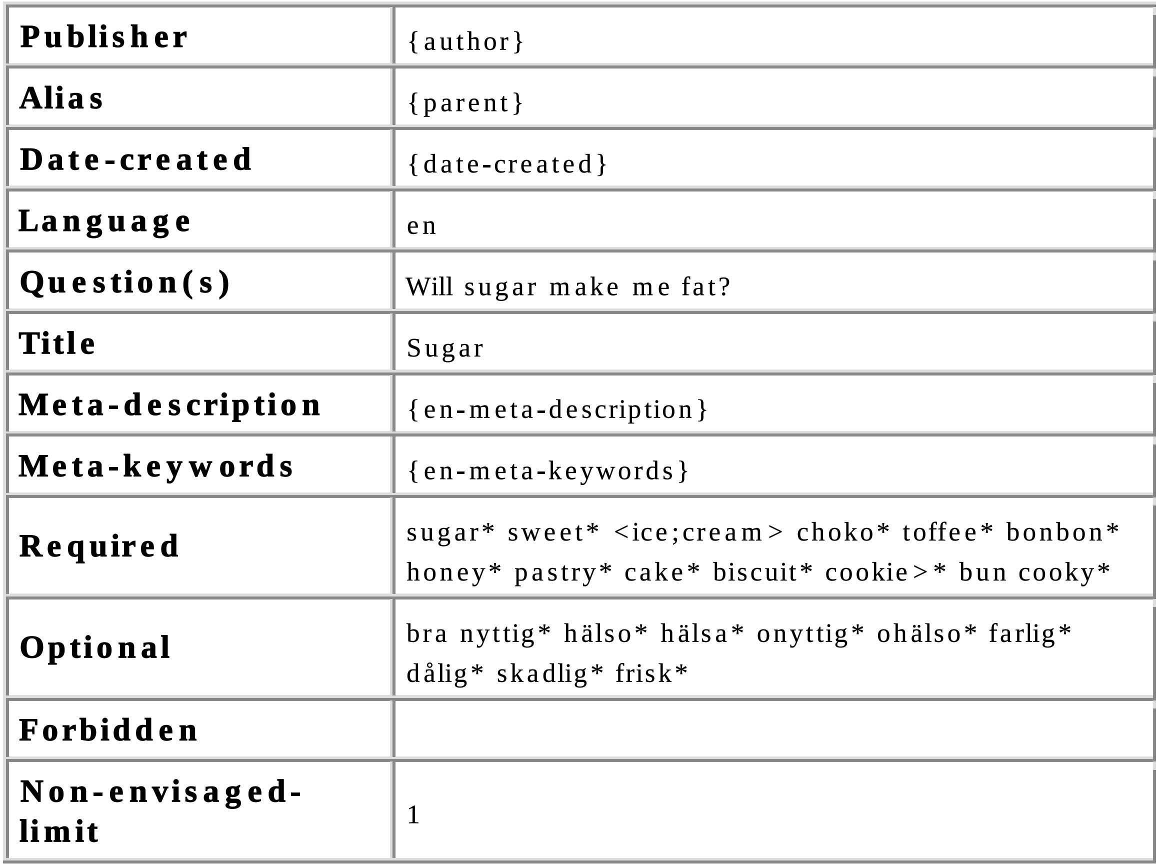
When using a web editor like Dreamweaver, this template will look something like this when nothing has been filled in.



Publisher	{Publisher}
Alias	{alias}
Date-created	{date-created}
Language	en
Question(s)	{en-questions}
Title	[en-title}
Meta-description	{en-meta-description}
Meta-keywords	{en-meta-keywords}
Required	{en-required}
Optional	{en-optional}
Forbidden	{en-forbidden}
Non-envisaged-limit	{en-limit}
Body	{en-answer}
Language	sv
Title	{sv-title}
Question(s)	{sv-questions}
Meta-description	{sv-meta-description}
Meta-keywords	{sv-meta-keywords}
Required	{sv-required}
Optional	{sv-optional}
Forbidden	{sv-forbidden}
Limit	{sv-limit}
Body	{sv-body}

1.10.2 A filled-in Template

The same template filled in with the data from the example in chapter 1.7 on page 13 can be found at http://salut.nu/en/faq-entries/food-sugar.htm. It will look something like what is shown below in a web editor like Dreamweaver:



Publisher	{author}
Alias	{parent}
Date-created	{date-created}
Language	en
Question(s)	Will sugar make me fat?
Title	Sugar
Meta-description	{en-meta-description}
Meta-keywords	{en-meta-keywords}
Required	sugar* sweet* <ice;cream> choko* toffee* bonbon* honey* pastry* cake* biscuit* cookie>* bun cooky*
Optional	bra nyttig* hälso* hälsa* onyttig* ohälso* farlig* dålig* skadlig* frisk*
Forbidden	
Non-envisaged-limit	1

Body	A healthy person's body has the ability to tell you when you have eaten enough. For a person, where this ability works well, you will feel satisfied when you have eaten what your body needs. You will then achieve a normal, healthy and good-looking body. For some people, this ability does not work. They may then get too fat or too thin or have other eating problems. Sugar confuses this ability, so that it will not work correctly. Sugar will make you feel a need to eat when you do not need more food. To achieve a healthy weight, you should avoid too much food which confuses your ability to fell adequate hunger and satisfaction. You should thus avoid all but small amounts of sugar and foods which include sugar. Some studies indicate, however, that small amount of sugar in your food leads to a slightly lower body weight than sugar-free food. Sugar can also cause a kind of dependence, similar to depence to drugs. This dependence is caused by the fact that sugar first raises the amount of sugar in your body. This may cause you to feel high. Your body then produces insulin, in order to reduce the amount of sugar. This may cause you to feel low, and want to take another dose of sugar. If you have this problem, switch to food which is taken up slower by the body, such as full-grain bread instead of white bread. Unfortunately, a lot of foods contain unnecessary sugar. For example, bread and salad sauces often contain sugar which need not be there. Most candies also contain sugar. And bananas consist mostly of sugar. It is possible to make good salad sauces, bread and other food without using sugar. There are different kinds of sugar. Common sugar, also known as sacharos, is especially dangerous to your ability to feel adequate hunger and satisfaction. Many fruits contain other kinds of sugar which are less dangerous. Recommended book: "Potatoes not Prozac", by Kathleen DesMaisons, Simon & Schuster, 1998 ISBN 0-684-85014-1.
Language	sv
Title	Socker

Question(s)	Blir man fet av socker? Får man äta socker? Vad anser du om mat som innehåller socker?
Meta-description	{sv-meta-description}
Meta-keywords	{sv-meta-keywords}
Required	socker* söt* glass* chok* karamell* godsak* snask* kaka* kex*
Optional	bra nyttig* hälso* hälsa* onyttig* ohälso* farlig* dålig* skadlig* frisk*
Forbidden	
Limit	1
Body	En frisk människas kropp har förmågan att tala om för dig när du har ätit tillräckligt. När denna förmåga fungerar bra, kommer du att känna dig mätt, när du har ätit så mycket som din kropp behöver. Du får därmed automatiskt en normal, frisk och snygg kropp. Denna förmåga fungerar inte för vissa människor. Det kan leda till att de blir för feta eller för smala eller får andra ätstörningar. Socker förvirrar denna naturliga funktion i kroppen, genom att göra att du känner dig hungrig även när du inte behöver mera mat. Man bör undvika mat som förvirrar kroppens naturliga sätt att säga till när man inte behöver mera mat. Därför bör man vara försiktig med socker i maten. Det finns dock vissa studier som visar att mindre mängder socker i maten leder till en lägre kroppsvikt än om man inte äter något socker alls. Socker kan också orsaka en typ av beroende, som liknar beroende av droger. Detta beror på att socker först höjer blodsockerhalten. Då börjar kroppen producera insulin, som sänker blodsockerhalten. Då kan den låga blodsockerhalten göra att man känner sig kymig, och äter socker igen för att höja sockerhalten i blodet.

		Tyvärr finns socker i onödan i många maträtter. Bröd och salladssåser innehåller t.ex. ofta socker som inte behövde vara där. Vissa frukter innehåller också mycket socker, t.ex. bananer. Snask, kakor och kex innehåller också ofta socker. Det går att göra bra salladssåser utan socker.
	Det finns olika typer av socker. Vanligt socker, som också kallas för sacharos, är speciellt skadligt, det skadar kroppens förmåga att känna igen hunger och mättnad särskilt mycket. I bananer är det mest sacharos, medan andra frukter som äpplen och jordgubbar innehåller mindre farliga sockerarter. Läs mera: "Potatis - inte prozac", av Kathleen Des Maisons, Bokförlaget Forum, 1999, ISBN 91-37-11495-6.	

1.11 Attachment B: The SALUT HTML Tools

1.11.1 Introduction

The SALUT HTML tools is a set of software programs under development within the SALUT project.

These tools are provided to make it easy (even for people who are not web design experts) to produce a small or large set of informational texts, and to adjust these texts to different output layouts. For example, the informational texts may be automatically formatted to the following formats:

1. A format easy to read when viewed on the screen.
2. A format easy to read when printed on paper.
3. A format to store in a data base for dynamic delivery in a natural-language question-answering system.
4. A format adjusted to the web design principles of different web sites, who may want to use the same content.
5. Versions of the same text in different languages.
6. A flexible XML format for easy exchange with XML-based software and data bases.

1.11.2 Easy to Produce Informational Texts

To make it easy to produce informational texts, these texts are written in HTML format. No fancy HTML codes need be used. Standard WYSIWYG (What You See Is What

You Get) HTML editors like Dreamweaver, Frontpage or Golive, or any other HTML editor can be used. Support for other input formats, like Microsoft Word documents, will be developed if needed.

Since informational texts are in HTML, pictures and tables can easily be included, even complex Javascript can be included if an advanced designer so wants.

1.11.3 Use of Templates to Simplify Text Production

1.11.3.1 Input template, either simple HTML or XML

To simplify text production, texts should be produced using a template, like the template shown in Attachment A on page 19. Only the following rows in this template have to be filled in:

- Language (since the same text can be supplied in multiple languages)
- Title
- Body containing the actual informational text

All the other rows in the template are optional and may be used, when needed, for producing special kinds of input, such as input for the natural-language question-answering system.

As an alternative to this template, the XML format described earlier in this report can be used. Also with the XML format, only the language, title and body tags are mandatory. The SALUT HTML tools will automatically convert between the HTML and the XML format *(Not yet implemented)*.

1.11.3.2 Output templates

For each output format, such as output according to the SALUT web design principles, or according to WEIGHT-INFO design principles, or according to some other design principles, one single output template need to be produced. This is a one-time only job, all new informational pages can then be produced according to this template. If there is a need to change the layout, only the template need to be changed, and all the informational pages can be generated in the new format.

1.11.4 Overview of Functionality Provided by SALUT HTML tools

1.11.4.1 Informational Text Input

Input is provided in one file for each informational text. This file can contain the same informational text in multiple languages. The file can either be HTML (using a template like that shown in Attachment A on page 19) or in XML (using the format described in chapter 1.4-1.8 on pages 2-18 above) *(not yet implemented)*. Input can also be an arbitrary HTML document, but then the functionalities of the program is limited. Files can also be converted between the HTML and the XML format *(not yet implemented)*.

Input can also be a set of files in a folder, one file for each health informational text, the program will automatically process all files of suitable type in a folder, and convert them to output files in another folder. This makes it easy to convert many files at once to a new output format.

1.11.4.2 Template Input

In addition to an informational text, or a folder with many informational text, a template is also needed for some modes of usage of the program. This template describes how the output is to be formatted. If output in different formats is needed, one template is needed for each output format. This is described in more detail on page 28ff.

1.11.4.3 Translation from HTML to XHTML

If wanted, the tools will translate HTML to XHTML. This is needed when producing XML, since HTML embedded in XML must be of the HTML dialect XHTML.

1.11.4.4 Overview Output

Optionally, the program can produce a list of all the informational texts in a folder. This list will show the file name, the title, and the size *(not yet implemented)*. It can be produced in a user-friendly format, or it can be produced in a format suitable for input into a data base handler or a spread sheet.

1.11.4.5 Indentation

Optionally, the program can re-indent the HTML for easy editing with a text editor. Or, the program can remove all indentation to create small, fast downloadable HTML documents.

1.11.4.6 Split into Chapters

Optionally, the program can split a large document into separate files for each chapter. This facility can be used when, for example, a book is to be converted to HTML, where each chapter of the book can be a separate web page.

1.11.4.7 Split into Chunks

Optionally, the program can split large documents into chunks. This can be used if a large text is to be delivered one separate web page at a time.

1.11.4.8 Remove Tags

Optionally, the program can remove some unwanted tags or attributes from the HTML input. This can be used to remove, for example, the fancy HTML formatting produced by many word processors when saving in HTML format.

1.11.4.9 Language Selection

The input files can contain the same informational text in multiple human languages, but the program can be instructed to produce output files using only one of these languages.

1.11.4.10 Quotes Conversion

Optionally, the program can convert book quote characters like (“) and (”) into HTML quote characters (").

1.11.4.11 Extraction

The program can extract the most important data into a simplified format, which is suitable to give to a human translator, who is to translate an informational text to different languages.

1.11.4.12 Convert Relative URLs to Absolute URLs

Optionally, the program can convert relative URLs to Absolute URLs. This is needed, for example if an informational text on one web site is to use images on another web site *(Not yet implemented)*.

1.11.4.13 Create Input for the Natural Language Query System

The program can convert the questions to the input format for the natural-language query system developed by KTH for SALUT. This conversion includes creation of alias entries and creation of a hierarchical structure of FAQ entries based on the structural information in the file names of the input data *(Not yet implemented)*.

1.11.5 More Details about the HTML Version of the Health Info Format

1.11.6 Basic Structure of the HTML Version

The main outer structure of the HTML version of the health informational format is a HTML table with two columns and a number of rows.

The first column only contains information for the person filling in the form, it is not used in the conversion process by the SalutHtmlTools program.

The second column contains a number of table cells. Each table cell has class name, for example:

```
<td class="title">
```

or

```
<td class="body">
```

Note: If the first character in the content of the HTML cell is the “{” character, this cell is regarded as empty. The reason for this is that the template, before filling the cells with content, can contain comments within “{” and “}”. This convention is taken from Dreamweaver, but the template can easily be filled in using another HTML editor than Dreamweaver. The “{” character can without problem occur anywhere except as the first character in the cell.

1.11.6.1 Conversion from the HTML or XML Format to Template-Controlled Output

Details in the description below apply to conversion from the HTML format. Conversion from the XML format is done in a similar manner.

The content of each such table cell is put into a container with the name of the class. For example, the contents of **`<td class="title">`** is put into a container with the name “title”, and the contents of **`<td class="body">`** is put into a container with the name “body”.

The template is then scanned for occurrences of the following start and end notation:

Start notation: **`<!-- #BeginEditable "title" -->`**

End notation: **`<!-- #EndEditable -->`**

If the start notation in the template contains the word "title", then the text between the start and the end notation is filled with the contents of the “title” container.

1.11.6.1.1 A simple example:

Suppose that the HTML for the health informational text contains the following HTML code:

```
<table>
  <tr>
    <td >Title</td>
    <td class="title">Drugs Causing Weight Increase</td>
  </tr>
  <tr>
    <td>Body</td>
    <td class="body">
      <p>Examples the list is not complete) of pharmaceutical drugs which can
        cause weight increase:</p>
      <ul>
      <li>Lithium, used for the treatment of bipolar (mano-depressive) disorders,</li>
      <li>Chlorpromazine, used for the treatment of psychiatric disorders,</li>
      <li>Cortisone, used for the treatment of allergies and rheumatism,</li>
      <li>Valproat, used against epilepsy.</li>
      </ul>
    </td>
  </tr>
</table>
```

It is easy to produce a template for this kind of table, which a person without knowledge of HTML can put texts into using a WYSIWYG web editor like Dreamweaver, Frontpage or Golive. The table in the example above will look like this with such an editor:

Title	Drugs Causing Weight Increase
Body	Examples (the list is not complete) of pharmaceutical drugs which can cause weight increase: · Lithium, used for the treatment of bipolar (mano-depressive) disorders, · Chlorpromazine, used for the treatment of psychiatric disorders,

	· Cortisone, used for the treatment of allergies and rheumatism, · Valproat, used against epilepsy.

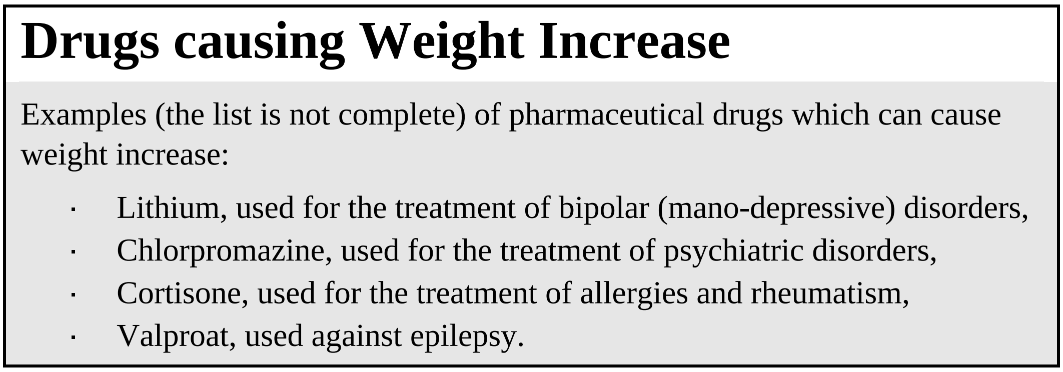
Suppose this is combined with a very simple template, containing the following code:

```
<H1><!-- #BeginEditable "title" -->{title}<!-- #EndEditable --></H1>
<TABLE WIDTH=400 bgcolor="#EEEEEE"><TR><TD>
<!-- #BeginEditable "body" -->{body}<!-- #EndEditable -->
</TD></TR></TABLE>
```

Then the output of the conversion process will be the following HTML code:

```
<TABLE WIDTH=400 bgcolor="#EEEEEE"><TR><TD>
<p>
  Examples (the list is not complete) of pharmaceutical drugs which can
  cause weight increase:</p>
<ul>
  <li>Lithium, used for the treatment of bipolar (mano-depressive) disorders,</li>
  <li>Chlorpromazine, used for the treatment of psychiatric disorders,</li>
  <li>Cortisone, used for the treatment of allergies and rheumatism.</li>
  <li>Valproat, used against epilepsy.</li>
</ul>
</TD></TR></TABLE>
```

When viewed with a web browser, it will look something like this:



Drugs causing Weight Increase

Examples (the list is not complete) of pharmaceutical drugs which can cause weight increase:

- Lithium, used for the treatment of bipolar (mano-depressive) disorders,
- Chlorpromazine, used for the treatment of psychiatric disorders,
- Cortisone, used for the treatment of allergies and rheumatism,
- Valproat, used against epilepsy.

Note that this will work also with fields not defined in the health informational standard. If the input has a table cell with

`<TD CLASS="unknown">`

and the template for the output has a section labelled

`<!-- #BeginEditable "unknown" -->{unknown}<!-- #EndEditable -->`

Then the merging of the template with the input data will still work!

1.11.6.2 Converting from the HTML to the XML format

When converting from the HTML to the XML format,
`<td class="title">Sugar</td>` is converted into
`<dc:title>Sugar</dc:title>`,
`<td class="question">Will sugar make me fat?</td>` is converted into
`<hi:question>Will sugar make me fat?</hi:question>` and
`<td class="body">` is converted into
`<html:body xmlns='http://www.w3.org/1999/xhtml'> ... </html:body>`

Any unknown class name such as
`<td class="unknown">Unknown content</td>` is converted into an XML element

such as for example
`<unknown>Unknown content</unknown>.`

This means that users of the tools can add their own additional fields and have them automatically converted from the HTML to the XML format and the reverse.

The reason for the special XML construct for the body,
`<html:body xmlns='http://www.w3.org/1999/xhtml'> ... </html:body>`
is to allow any HTML code, such as embedded tables, images, even Javascript code, to be included in the generated XML. This gives the producer of the informational texts full freedom to format the informational text using HTML layout conventions, references to CSS style sheets, etc.

1.11.7 Using the SalutHtmlTools program

Warning: All combinations of input settings will not work. The program should give an error message for non-working combinations, but such messages may not always be provided, instead, non-working combinations might cause unexpected results.

1.11.7.1 The Main Preferences Window

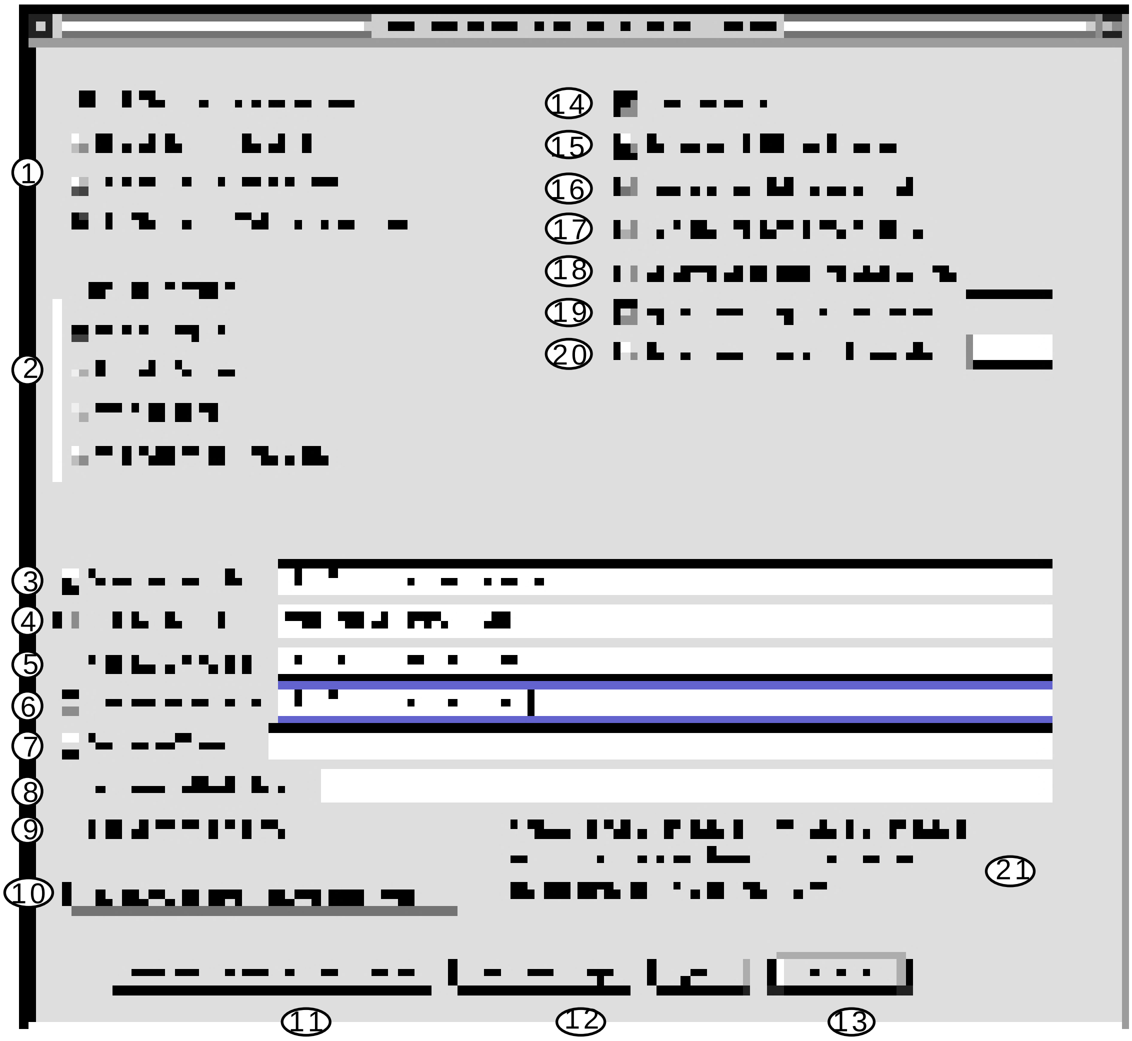


Figure 1: The main preferences window of SalutHtmlTools

1.11.7.1.1 HTML Character Conversion (1 in Figure 1)

This will get national ISO 8859-1 characters like “é” to be converted to their HTML equivalents, which for “é” is “&eacute;” *(Not yet implemented)*.

1.11.7.1.2 Output Format (2 in Figure 1)

This allows output in either HTML or XHTML format. XHTML is necessary when putting HTML code inside XML documents. Conversion from HTML to XHTML means correcting certain faulty tag ordering, like <b><i>text</b></i>, ensuring that all tags are ended and lowercased and other changes to the HTML syntax to satisfy XHTML rules.

Syntax tree means that a syntax tree of the input HTML file is produced.

Overview of files means that a list of the informational files in a folder is produced. For each file, the file name, the title, and the size of the body is listed *(size not yet implemented)*.

1.11.7.1.3 Print folder URL (3 in Figure 1)

Here an URL can be specified, which is to be added in front of the file name of the health informational file, to get an absolute URL for a printer-friendly version of this file. *(Not yet implemented)*

1.11.7.1.4 Link folder URL (4 in Figure 1)

Here an URL can be specified, which is to be added in front of the file name of the health informational file, to get an absolute URL for file with the same content, but referring to a static web page, suitable as a link target. *(Not yet implemented)*

1.11.7.1.5 Add to SRC URLs (5 in Figure 1)

Here an URL can be specified, which is to be added in front of "`SRC=`" relative links in the input file, to make them absolute, so that the images can be put somewhere else than where the informational texts are placed. *(Not yet implemented)*

Example:

If the input file contains "`<IMG SRC="abc.def">`" and the Add-to-SRC-URLs field has the value "`http://salut.nu/images/`", then this text in the input file is converted to "`<IMG SRC="http://salut.nu/images/abc.def">`". *(Not yet implemented)*

1.11.7.1.6 Add to HREF URLs (6 in Figure 1)

Similar to Add to SRC, but for relative "`HREF=`" links in the input file, which are converted to absolute URLs. *(Not yet implemented)*

1.11.7.1.7 Remove Tags (7 in Figure 1)

This field can be used to specify a comma-separated list of tags, which are to be removed from the input. This is useful to clean up the HTML produced when saving documents in HTML format from many word processing and other programs.

For example, to clean up the HTML produced by Microsoft Word 2001, the following value of "Remove tags" might be used:

```
    !--, style, meta, font. div, span, link, o:p, !
```

1.11.7.1.8 Remove Attributes (8 in Figure 1)

This field can be used to specify a comma-separated list of element attributes, which are to be removed from the input. This is useful to clean up the HTML produced when saving documents in HTML format from many word processing and other programs.

For example, to clean up the HTML produced by Microsoft Word 2001, the following value of "Remove attributes" might be used:

```
    style, lang, width, height, xmlns:o, xmlns:w, xmlns
```

1.11.7.1.9 Convert “ to ", etc. (9 in Figure 1)

Will convert English unicode double quotes (“) and (”) into HTML double apostrophes (") and single quotes to single apostrophes. Works also for Swedish, since the Swedish double quotes are a subset of the English double quotes. Could be extended to cover other languages than English and Swedish, like for example („) for German.

The actual transformation done is the following:

Before conversion	After conversion	Comment
&#8220;	&quot;	Left double quote
&#8221;	&quot;	Right double quote
&#8216;	&#39;	Left single quote
&#8217;	&#39;	Right double quote
&#8211;	--	en dash to two successive short dashes

1.11.7.1.10Remove Word Fancy Formatting (10 in Figure 1)

This button will fill the “Remove Tags” and “Remove Attributes” fields with the values indicated above to clean up the HTML produced by Microsoft Word 2001.

1.11.7.1.11Health Format Preferences (11 in Figure 1)

Will open a separate preferences window described in section 1.11.7.2 on page 35. This window contains preferences for handling input files in the health informational text format.

1.11.7.1.12Save Settings (12 in Figure 1)

Will save the settings, so that these settings will be used the next time the SalutHtmlTools are executed.

1.11.7.1.13Convert (13 in Figure 1)

Will save the settings, so that these settings will be used the next time the SalutHtmlTools are executed.

Push this button when all the preferences have been set, to start the conversion process. Depending on what kind of conversion you want, the program will ask you to locate either a file to convert, or a folder to convert all the files in this folder, and to indicate where you want to store the results, either a single file name or a folder, where the results are to be stored when converting all files in a folder.

1.11.7.1.14Indent Results (14 in Figure 1)

The output HTML code will be indented to show the structure, for example:

```
<ul>
```

```
  <li>
    Lithium, used for the treatment of bipolar (mano-depressive) disorders,</li>
  <li>
    Chlorpromazine, used for the treatment of psychiatric disorders,</li>
  <li>
    Cortisone, used for the treatment of allergies and rheumatism.</li>
  <li>
    Valproat, used against epilepsy.</li>
</ul>
```

If this box is not checked, the results will be without all indentation, for example:

```
<ul>
<li>
Lithium, used for the treatment of bipolar (mano-depressive) disorders,</li>
<li>
Chlorpromazine, used for the treatment of psychiatric disorders,</li>
<li>
Cortisone, used for the treatment of allergies and rheumatism.</li>
<li>
Valproat, used against epilepsy.</li>
</ul>
```

This reduces the size of the HTML file, but makes the HTML source text less easy to read for a human reader.

1.11.7.1.15Reduce Multiple Spaces (15 in Figure 1)

Multiple spaces in the HTML file, which will anyway only be shown as a single space to the person viewing the result in a web browser, are reduced to a single space. Contents of "**`<PRE> ... </PRE>`**" areas are not touched, since there multiple spaces are significant.

1.11.7.1.16Process all Files in a Folder (16 in Figure 1)

This checkbox will get the program to convert all files with ".htm" or ".html" extension in an entire folder. The converted results will be put in another folder chosen by the user.

1.11.7.1.17Health Format Processing (17 in Figure 1)

Check this box to do special processing for the health informational format. The preferences for this special processing are specified in the Health Format Preferences Window, which is opened with the button no. 10 in Figure 1 and described on page 35.

1.11.7.1.18Suppress Warning Messages (18 in Figure 1)

The program will write warning messages for certain kinds of input files, which it believes contain incorrect HTML. These warning messages are usually not very important, and are a nuisance when processing large files or many files, so in such cases, check this box to avoid them. Uncheck the box when you are testing new kinds of input data and want to see warning messages.

1.11.7.1.19Split Into Chapters (19 in Figure 1)

This check button will cause the program to split large HTML files into separate files for each chapter. If, for example, you have a book in Microsoft Word, and save it as HTML, you will get a very large HTML file. With this setting, you can split this into a separate HTML file for each chapter or subchapter. The number indicates the split level. "1" means split at <H1> headers," 2" split at <H1> and <H2> headings, etc.

1.11.7.1.20Split into Chunks (20 in Figure 1)

This check button will cause the program to split large HTML files into separate files for each page. Chunk size is the maximum number of characters on a page.

1.11.7.1.21Convert “<p class=pre>” (21 in Figure 1)

Will convert occurrences of “<p class=xxx>” where xxx has any of the following values: “pre address blockquote dir hr menu numberedlist bulletlist glossary dir”, into “<pre class=xxx>”. This option is included because when you save Microsoft Word 2001 documents in the "Save As Web Page" format, Word generates HTML which may look better after doing this conversion. (Or at least I think that is the reason, I am not quite sure now why this option was included in the program!)

1.11.7.2 The Health Preferences Window

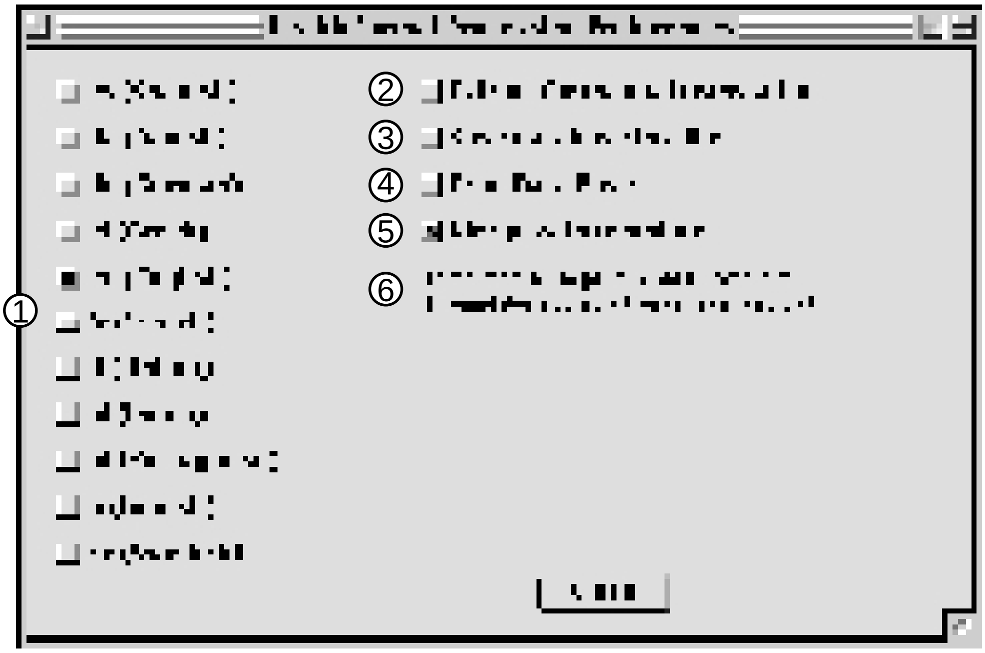


Figure 2: The Health Preferences Window

The health preferences window shown in Figure 2 contains special settings for health format processing. These settings are described below:

1.11.7.2.1 Language Selection (1 in Figure)

Here you can check which languages to process from an input file, which contains the health informational texts in multiple languages.

Note: Some kinds of processing will only work when a single language is selected. For example, merging text into a template as described in section 1.11.6.1 on page 28 will only work for one single language at a time (since the output files are to be in one language, and usually in a different folder for each language).

1.11.7.2.2 Extract for manual translation (2 in Figure)

This check box will extract those texts, which you might want to give to a human translator for translation to a new target language.

1.11.7.2.3 Save database in a file (3 in Figure)

This check box will produce a list of the HTML files in a folder in a format suitable for input for further processing using for example a spreadsheet program.

1.11.7.2.4 Print Data Base (4 in Figure)

This check box will produce a list of the HTML files in a folder in a format suitable for reading by a human.

1.11.7.2.5 Merge with template (5 in Figure 2)

Check this checkbox to indicate that you want to merge input data with a template as described in section 1.11.6.1 on page 28. When you check this checkbox, the program will ask you to locate the file. To choose a new file, uncheck the check box, and then check it again.

During the merging, HTML comments marking the start and end of an editable region in Dreamweaver document in the processed file will also be removed from the input document being processed.

1.11.7.2.6 Template file (6 in Figure 2)

Here the selected template file name and path is shown. To select a different template file, uncheck the check box numbered 5 in Figure 2 and check the same checkbox again.