

Introducing dynamic variables into document templates

Accommodating a grey area between vocabulary and templates

<http://a2e.de/adv/09/03/dynaarbs07>

May 18, 2009

1 Situation

We needed a possibility to store unexpanded variables and expand them later only during document processing, even though this meant giving up a virtuous guiding principle of our document processing system.

This is needed for two different reasons

1. It does not make sense to store big expanded text chunks in the dbm database and in fact the SDBM.File package does not allow this at all, setting a severe limit of 1K
2. Sometimes expansion must happen in the document, using variables that could have been (re)defined within the document

We can now define the following normal dynamic variable in the template

```
tslv2 = $(lnol|${tslv2_0}|${tslv2_1}|${tslv2_2}|${tslv2_3}|${tslv2_4})
```

and expand it later in the template, using a vmethod `expand`, which expands recursively until there is no more expansion work to do, e.g.

```
[* tslv2.expand *]
```

The `expand` vmethod does not even need to know which variables are dynamic. It is up to the document template creator to decide whether the variable in question might need expansion.

The `expand` vmethod was enriched further so as to become a full equivalent to the `tb` filter and thus make that filter superfluous.

2 Methods for operations on dynamic variables

Beside the simple `expand` method, we have a few more vmethods:

value treat the argument as a variable name and return its value (simplified expansion)

expand expand the argument as a mlht/maketext (vocabulary building language) expression

call apply arguments

fill apply arguments and interpret function calls

funcall invoke a builtin function of the mlht/maketext language, with

All of these vmethods are also available as filters. A filter is slightly bulkier to use and to program, but has the advantage that it can be used not only on a variable but also directly on a string value.

Below are some sample text blocks, based on definitions which illustrate the use of these functions:

template source	compiled result
<code>var='tit';var.value</code>	Introducing dynamic variables into doc
<code>'sut' FILTER value</code>	Accomodating a grey area between voc
<code>titsut</code>	<code>#{tit}: #{sut}</code>
<code>titsut.expand</code>	Introducing dynamic variables into doc
<code>'[title] \${1}' FILTER call(tit)</code>	[title] Introducing dynamic variables i
<code>'\$(dt title): \${1}' FILTER fill(tit, 'dt=:call:sqbrackets')</code>	[title]: Introducing dynamic variables
<code>'23' FILTER funcall('dnumpad', '6')</code>	0 0 0 0 2 3

3 Document reference functions

We have moreover added some functions that allow us to retrieve properties and text chunks from other documents. We have used these to replace our menu generation system so that the menu is now entirely written using the Template toolkit.

template source	compiled result
<code>var = 'perl_a2e_pub';var.url</code>	<code>/adv/perl/A2E</code>
<code>'perl_a2e_pub' FILTER url</code>	<code>/adv/perl/A2E</code>
<code>dok.predoks.join(',')</code>	<code>adv_pub, adv_pub09, adv_pub0903, adv_pub_int0903</code>
<code>var = 'adv_pub0903';var.subdoks.join(',')</code>	<code>adv_pub_int0903, adv_pub_ont0903</code>
<code>var = 'adv_pub0903';var.postdoks.join(',')</code>	<code>adv_pub0904</code>
<code>dok.lab</code>	<code>Dynamics</code>
<code>dok.lab('en')</code>	<code>Dynamics</code>
<code>dok.tit</code>	<code>Introducing dynamic variables into document templates</code>
<code>dok.tit('de')</code>	<code>Einführung dynamischer Variablen in Dokumentvorlagen</code>
<code>dok.sut</code>	<code>Accommodating a grey area between vocabulary and text</code>
<code>dok.sut('zh')</code>	<code>文書與多語字串之間的灰色地帶已從雙面填充</code>
<code>var = 'a2e';var.lab</code>	<code>a2e</code>
<code>var.lab('zh')</code>	<code>亞通歐</code>
<code>var.des</code>	<code>We write, translate and program texts in languages</code>
<code>var.des('zh')</code>	<code>我們在此用中、日、德、法、意、俄等語文編寫、翻譯</code>
<code>var = 'tit';var.cit('adv_pub')</code>	<code>ADP Advances at A2E</code>
<code>'tit' FILTER cit('adv_pub')</code>	<code>ADP Advances at A2E</code>
<code>'tit' FILTER cit('adv_pub', 'zh')</code>	<code>亞通歐多語數據處理</code>

3.1 Generating meta info about arbitrary documents

Languages can be listed in order of preference, meaningful defaults are chosen.

1. dok: oas_pub
 - url: /oas
 - tit: A2E Language Services
 - des: Authoritative and sophisticated translating, interpreting and multilingual computing services, involving languages of Asia and Europe, by an experienced team of geeks, operating from an office in the center of Munich.
 - lab: Services
 - meta sut: Chinese, Japanese, German, English et al
 - meta filter sut: Chinese, Japanese, German, English et al
2. dok: eupat
 - url:
 - tit: Patentability and Democracy in Europe
 - des: The European Patent Organisation (EPO) is continuing to grant thousands of monopolies on “computer-implemented” algorithms and business methods every year, against the letter and spirit of the written law, and is now, for the third time, pushing for European legislation that would impose this practise on reticent national courts. Patent interests in combination with

inter-governmental lawmaking have formed a caricature of “innovation policy” and thereby are helping us to see clearly what is going wrong in Europe today and what the alternatives are.


- lab: EU Patent
- meta sut: How can industrial property be subordinated to modern economic policy?
- meta filter sut: How can industrial property be subordinated to modern economic policy?


3. dok: apr_pub


- url:
- tit: Freedom of Speech
- des: Personality Rights, Privacy Rights and Copyright are being emphasised by courts at least here in Germany in a way that is detrimental to the Communication Society of the digital age
- lab: Appeal
- meta sut: Communication vs Privacy
- meta filter sut: Communication vs Privacy


3.2 Example: Table of Superordinate Documents of this Node

We see the language default choice algorithm at work here too.

 ADP: ADP Advances at A2E: Not a Field of Technology: Automated Data Processing (ADP) on Asian and European texts — tools, methods and support services.

 2009: Datenverarbeitung 2009: Notizen des Jahres: Erfahrungen beim Simultandolmetschen von Chinesisch, Japanisch, Englisch und Deutsch, Übersetzungen, Programmierung vielsprachiger Dokumentation und was das Jahr sonst noch brachte.

 03: Thema: Begriff und Anwendung: Hier sammeln und destillieren wir unsere Erkenntnisse zum Thema .

 Was War: Data Processing im March 2009: Verlauf des Monats: Was diesen Monat im Bereich Data Processing passierte.


 Nachrichten: Einrichtung von RSS-Nachrichtenkanälen in A2E MLHT: Wie wir die rechte Spalte jeder Webseite mit passenden Meldungen anreichern: Mittels mehr oder weniger detaillierter Angaben in der Konfigurationsdatei des Multilingualen Hypertextsystems (MLHT) und in Vokabulardateien können wir eine detaillierte Kontrolle über die Zusammensetzung der Nachrichtenkanäle ausüben oder uns mit einigermaßen sinnvollen Voreinstellungen zufrieden geben.

Table 1: predoks

4 CAVEATS

The filter `cit`, by loading the whole vocabulary of a quoted document into memory side by side with that of the current document, provides a great mechanism for commenting a source document. This vocabulary includes the whole vocabulary of superior files and template used by that file. Use with care, i.e. only for real quotations, on one or two source documents that you may want to comment on, not for substituting the proper vocabulary building mechanisms. Otherwise our system could quickly lose its compactness and speed.

5 BUGS, LIMITATIONS

Variables defined in the document do not thereby become available to the `vmethods/filters`. The `vmethods` (e.g. `call`, `fill`, `funcall`) are aware only of the definitions made in the vocabulary file. In the above examples, the verb ‘`sqbrackets`’ must be defined in the vocabulary file.

6 Resources

- A2E Perl libraries
- `perldoc Template::Manual:VMethods`
- `perldoc Template::Context`
- `perldoc Template::Filters`
- `tmpl`

- nav_tmpl_repl: script for conversion from old to new menu source format
- Textbausteine für Patentübersetzungen
- Document Source
- Language-dependent text elements