SOW PHP API Quick Tutorial
Revision:

Mark J. Olesen

Date:

Copyright

1 Introduction

This document provides a quick tutorial to the SOW PHP API. The document should be used along with the overview, which contains some of the source explained in further detail here. This document briefly discusses templates, environments and stroking.

2 Template

A SOW *Template* is a document that contains SOW embedded characters. Templates may optionally contain sectioning commands to further divide the document into sections. All SOW documents contain an initial section named "START."

2.1 Syntax

Fields are surrounded by double hash/pound symbols (e.g. ##myfield##). Field names may only contain alphanumeric ASCII characters and underscores.

2.2 Example

Following is an example ASCII text document that contains SOW embedded characters.

```
##header##

##date##

Dear ##salute##

Your account ##account## is past due by ##past_due_days##.
To avoid further harassment, pay the full amount
of ##amount_due## now dammit!
```

Following are the fields SOW will recognize.

- header
- data
- salute
- account
- past_due_days
- amount_due

3 Environment

SOW gets substitution data through an environment. An environment consists of key/value pairs. A key directly corresponds to a field within a section. Fields within a document are place holders for data. When SOW encounters a field, it will attempt to retrieve it's associated value through the environment. That value will be used to replace the field. If a corresponding field can not be found, nothing will be output.

3.1 new_env

In PHP, an environment is first created with the new_env routine.

3.1.1 Syntax

int new_env([int symbols])

The optional *symbols* parameter, takes a positive integer, which indicates the number of symbols SOW should reserve for the environment.

3.1.2 Example

```
$env= new_env();
```

3.2 update_env

The easiest way to add data to an environment is to construct and populate an associative array and use the update_env routine.

3.2.1 Syntax

int update_env(resource env, array tuples|[string value, string key])

3.2.2 Example

Following is an example of an associative array, which contains the fields from the *Template* defined earlier.

```
$ar= array();
$ar['header']= ``Cruel Collection agency\nYou owe us, pay now!\n'';
$ar['date']= 'December 01, 1938';
$ar['salute']= 'Mr. Right';
$ar['account']= '01010101';
$ar['past_due_days']= 'many many years';
$ar['amount_due']= '$0.01';
update_env($env, $ar);
```

4 new_sow

new_sow creates a resource that can be used within the SOW API to control the construction of a document.

4.0.3 Syntax

int new_sow(string template)

4.0.4 Example

For our example we will assume our template was named "message.txt."

```
$sow= new_sow('message.txt');
```

5 stroke_sow

SOW relies on the developer to control the flow of input and output to construct a final document. This is accomplished through templates, environments and stroking.

Stroking combines a section and environment to form output that is committed to an internal data buffer, which constitutes the running document.

```
"START"  $env
   :       :
   :       :
   +---:---+
       :
       :
    [BUFFER]
```

5.0.5 Syntax

A method to commit data to the internal data buffer is to use stroke_sow.

> int stroke_sow(resource sow, resource env[, string section])

5.0.6 Example

Since our template contains only one section, we will omit the section using the default "START."

```
stroke_sow($sow, $env);
```

6 reap_sow

A document does not do much good unless it can be output. reap_sow outputs the internal data buffer to standard out.

6.0.7 Syntax

> int reap_sow(resource sow)

6.0.8 Example

```
reap_sow($sow);
```

7 Conclusion

Hopefully the quick guide provided some insight into the benefits of SOW. Since SOW is in it's infancy there is not much documentation. So, the best place to turn for further information is the sources and it's community of users.

The SOW team welcomes your comments and contributions. The success of open source relies on it's community of users. So, please check out the SOW team and get involved.

SOW is proudly hosted on Sourceforge.net. It can be found at http://sow2.sourceforge.net.

Mark J. Olesen