

Commercial Pointers
Technical Implementation Principles
For Commercial Web Sites

Blaafisk Independent Innovations ANS

25th April 2003

Contents

1	Implementation for the Web	4
2	Icon Pointer Class Definitions	9
A	Client Standards	18
B	Client Prototype	20

Legal Notice

Copyright

Copyright © 2003 Blaafisk Independent Innovations ANS

All rights are reserved.

Disclaimer

These Terms represent the entire understanding relating to the use of the document and prevail over any prior or contemporaneous, conflicting or additional, communications.

This Document may include technical inaccuracies or typographical errors.

Many of the names used by companies to distinguish their products and services are claimed as trademarks. Where those names appear in this documentation, and Blaafisk Independent Innovations ANS is made aware of those trademarks, then the names are in capital letters or initial capital letters.

Licensing

The rights to use the techniques described in this Document commercially must be granted under the terms in the signed agreement between Blaafisk Independent Innovations ANS and the Reader of this Document.

All other rights reserved by Blaafisk Independent Innovations ANS.

This document may NOT be redistributed. Printing, scanning, digital or photostatic copying and redistribution of the complete document or textual parts of it is strictly prohibited, and restricted by the signed agreement between Blaafisk Independent Innovations ANS and the Reader of this Document.

Chapter 1

Implementation for the Web

The concept of Icon Pointers is to code mouse movement events for page elements on a web page, so that when a movement event occurs, the mouse driver translates the event, and the browser displays a custom/branded pointer icon instead of the default arrow pointer.

When the pointer is moved over various parts of a page (whose markup is specified in HTML), then the browser will change the appearance of the pointer.

When a reader's mouse focus crosses a Icon Pointer Defined Page Element, lets say, one that contains the word 'netcom', the specified icon pointer is downloaded over HTTP from a Icon Pointer Storage server and the ordinary pointer is transformed into the brand logo of Netcom, who pay for Icon Pointer impressions.

If the Netcom brand logo is later removed or temporarily unavailable from the Icon Pointer Storage server, then no change occurs to the pointer and the reader won't notice the error; the original arrow pointer will still be displayed.

Icon Pointers can be monochrome or in color, and single- or multiframe (animated). In this document we will also explain how to associate text (Pointer Text) along with the Icon Pointer.

1.1 Icon Pointer Creation

To create a new Icon Pointer for a web page, use a graphics application that is capable of saving the graphic in the standard WINDOWS(R) Icon Format, for instance the MICROANGELO application from IMPACT SOFTWARE, available from <http://www.impactsoft.com/> or ARTICONS, available from <http://www.aha-soft.com/articons/>

1.1.1 WINDOWS(R) Icon Format

The application that is used to save or convert to Icon Format must be compatible with the MICROSOFT WINDOWS(R) Icon format specification and Compatible Browser Software, such as the MICROSOFT INTERNET EXPLORER browser.

The Icon Format is described in 'Icons in Win32' by John Hornick, MICROSOFT CORPORATION (1995), available from <http://msdn.microsoft.com/library/default.asp>

Note that it is not required that the file saved in Icon Format have the .ICO, .CUR or .ANI file extension, although most applications usually require the user to specify one of those file extensions before the file is saved.

The Icon Pointer file must be at least 32x32 pixels. A minimal Icon Pointer should be 32x32 pixels large in at least 16 colors. Icon Pointers of 48x48 pixels dimension are highly recommended. It is also possible to combine icons with different screen color depths into a single file. This is recommended, since some or many of the readers may have a different screen color depth than you have.

Create the new Icon Pointer file and save it. If you need assistance, contact the software vendor for further instruction.

The explanation of how to store and publish the Icon Pointer follows.

1.2 Icon Pointer Storage

When the new Icon Pointer file has been created, using a compatible graphics software package, the file should be stored in a location that is accessible by Compatible Browser Software, such as MICROSOFT INTERNET EXPLORER 6.0. This location could either be online served from a Icon Pointer Storage server or offline in the local computer file system (for example; loaded off a promotional CD-ROM/DVD with HTML documents and various multimedia content).

1.3 Compatible Server Software

The focus in this document is on serving Icon Pointers using Compatible Server Software such as APACHE HTTPD server with APACHE JSERV from APACHE SOFTWARE CONSORTIUM. Note that it may be possible to exploit existing HTTP/1.1 compatible server infrastructure/interfaces for serving Icon Pointers in the same manner other web advertisements are now.

1.4 Icon Pointer Provider

It may be useful to establish a central Icon Pointer Provider running Compatible Server Software, such as Apache HTTPD with Apache JServ, to store and manage Icon Pointers for several commercial web site domains. A such provider server is capable of measuring impressions for each Icon Pointer Client Request (HTTP/1.1 GET).

The Icon Pointer Provider will serve the Icon Pointer Data from a static Icon Pointer Storage and serve Icon Pointer Class definitions to Clients through a server-side web application that runs queries against a relation database, matching the supplied icon parameter. A such dynamic Icon Pointer script application is prototyped later in this document.

1.5 Icon Pointer Types

1.5.1 Static Icon Pointers

Static Icon Pointers are Icon Pointers that can be enabled or disabled on a Icon Pointer Storage server for each advertiser, in time slots, and are defined statically in the document.

Lets assume that the file of a Icon Pointer is created, saved in the Windows(r) Icon format, and published on a Icon Pointer Storage server (example: `ICON.PROVIDER.NET`), and that the Icon Pointer Data for a Static Icon Pointer is accessible from the URL:

- <http://ICON.PROVIDER.NET/static/vendor-001/icon-001.ico>

1.5.2 Dynamic Icon Pointers

Dynamic Icon Pointers are defined by Icon Pointer Classes that are associated with specific page elements. Such Icon Pointers are easy to change, and the rotation settings could be be managed and looked up in a relation database on a dynamic Icon Pointer Storage server. A Dynamic Icon Pointer Definition is externally referenced inside a page from an URL such as

- <http://ICON.PROVIDER.NET/engine.php?ID=netcom>

The Icon Pointer Class definition for Dynamic Icon Pointers are selected programmatically in the PHP script 'engine.php' based on the CGI parameter 'ID=netcom' from the HTTP/1.1 GET request.

1.6 Portability

The Web implementation of Static Icon Pointers depends on a HTTP/1.1 compatible Icon Pointer Storage server.

This means it is possible to serve Static Icon Pointers from a MICROSOFT WINDOWS(R), Apple MACINTOSH(R) or UNIX(R)-like system, just as long as the platform provide a HTTP/1.1 compatible server. Hypertext Transfer Protocol, aka HTTP/1.1, is defined in RFC 2616 available from <http://www.w3.org/Protocols/rfc2616/rfc2616.txt>

The implementation of Dynamic Icon Pointers, however, is tied closely to one specific Icon Pointer Provider server software platform, and its support of web programming languages, such as Sun Java Server Pages, Java Servlets, Microsoft Active Server Pages, Perl, PHP, or Python.

1.7 Scalability

Icon Pointer Data is small in storage size (3-4 Kbytes). It means that the bandwidth demand of a Icon Pointer Provider and storage space demand is less than in traditional web banner advertisement.

1.8 Compatible Client Software

The Icon Pointer Class definition is compatible with web browser clients such as MICROSOFT INTERNET EXPLORER 6.0 from MICROSOFT CORPORATION.

Other valid client software have implemented all sections in the HTML 4.01 Specification, and the CSS level 2 standard, published by the WORLD WIDE WEB CONSORTIUM (W3C) on 12-May-1998, for full compliance.

Chapter 2

Icon Pointer Class Definitions

The proposed Icon Pointer technique for the Web is built upon a collection of standard definitions of tags and properties in the Hypertext Markup Language (HTML) and Cascading Style Sheets level 2 (CSS2) standards, as published by the WORLD WIDE WEB CONSORTIUM (W3C) on 12-May-1998.

The Cascading Style Sheets, level 2 CSS2 Specification, as published from the World Wide Web Consortium on 12-May-1998, defines the **CLASS**, **ID**, and **STYLE** Attributes.

The latest CSS2 standard document is available from

<http://www.w3.org/TR/REC-CSS2/>

The **SPAN** Tag was introduced in an early HTML Specification from the W3C. The full HTML 4.01 standard document is available from

<http://www.w3.org/TR/REC-html40/>

The **SPAN** Tag is also defined in extended versions of HTML, such as XHTML.

2.1 HTML CLASS Attribute

Most HTML Tags support the **CLASS** attribute to be defined, however we limit the usage to the **SPAN** Tag in the following examples.

The **CLASS** attribute is useful to override the Icon Pointer for a previously Icon Pointer Class defined HTML Tag.

2.2 HTML ID Attribute

The **ID** attribute is almost like the **CLASS** attribute, and is used mainly with Dynamic HTML (otherwise known as DHTML).

The big difference from the **CLASS** Attribute is the pound sign (#) instead of the dot before a **ID** name. It is possible to specify an **ID** for a specific Tag.

Note that not all tags support the **ID** Attribute, so we will avoid its use in our examples, and stick to the **SPAN** Tag combined with the **CLASS** Attribute.

2.3 HTML SPAN Tag

The **SPAN** Tag is very useful for defining Icon Pointers in specific sections on a page, especially since **SPAN** marks no layout or structure on its own.

If the **CLASS** Attribute is set, then **SPAN** allow you to define Icon Pointers in the middle of sentences and/or to place Icon Pointers inside words delineated by no existing HTML tag.

For example, suppose there's a paragraph of text and certain words should trigger different Icon Pointers, as displayed here:

```
<p>
The nominated albums in the <span class="rock">rock category</span>
are: <span class="johndoe.beats">John Doe's Beats</span>,
<span class="jeffburton.tracks">Jeff Burton's Tracks</span>, and
<span class="jimcarlington.groove">Jim Carlington's Groove</span>.
</p>
```

In this code there will be a different Icon Pointer focus for each nominated album title.

2.4 CSS2 User interface

A specific CSS2 Property is used to define a list of unique Icon Pointers that should become visible when it exists on the Icon Pointer Provider server location that the page element is defined to.

We'll start with a simple example of how to use the Icon Pointer technique with 3 different Static Icon Pointers are defined inside a Icon Pointer Class inline the HTML document.

2.4.1 Example 1: Link Anchor with Static Icon Pointer

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Link Anchor with Static Icon Pointer</title>
    <style type="text/css">
<!--
.netcom {
  cursor:
  url('http://ICON.PROVIDER.NET/ico/netcom-001.ico'),
  url('http://ICON.PROVIDER.NET/ico/netcom-002.ico'),
  url('http://ICON.PROVIDER.NET/ico/netcom-003.ico'),
  default
}
-->
    </style>
  </head>
  <body>
    <a class="netcom" title="Enkel moro med Netcom MMS
mms.netcom.no" href="http://www.netcom.no/">Netcom</a>
  </body>
</html>
```

In the example code above, if the first Static Icon Pointer (`netcom-001.ico`) doesn't exist at the URL <http://ICON.PROVIDER.NET/ico/netcom-001.ico> then the optional Static Icon Pointer at the URL <http://ICON.PROVIDER.NET/ico/netcom-002.ico> is downloaded, and so on.

The manager of the Icon Pointer Storage (in this case <http://ICON.PROVIDER.NET/>) con-

trols the queue ordering of the Static Icon Pointers in the Icon Pointer Class definition `net.com`

In order to handle special situations where no Static Icon Pointer among the listed ones is available, the generic 'default' should always be specified in the end of the Icon Pointer Class definition list, as a fallback.

Consider the situation where a vendor doesn't extend the Icon Pointer impression period, or the icon pointer provider server was down/unreachable for a moment.

Note that if the returned Static Icon Pointer Data is missing or empty, then the ordinary arrow pointer will be displayed (at least in Compatible Client Software), and the user will never experience an error.

2.4.2 Example 2: Static Icon Pointers inside STYLE Attribute

It is possible to set Static Icon Pointers using the **STYLE** Attribute in most HTML tags. The setting will hold until the closing tag is met:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Page with Icon Pointer</title>
  </head>
  <body style="cursor:
    url('http://ICON.PROVIDER.NET/ico/netcom-001.ico'),
    url('http://ICON.PROVIDER.NET/ico/netcom-002.ico'),
    url('http://ICON.PROVIDER.NET/ico/netcom-003.ico'),
    default" title="Enkel moro med NetCom MMS!
mms.netcom.no">
    ....
    <a href="http://mms.netcom.no/">
      
    </a>
    ....
  </body>
</html>
```

If the example code above is rendered, it displays a page with the Static Icon Pointer replacement as defined in the **STYLE** Attribute of the **BODY** Tag.

2.4.3 Example 3: Dynamic Icon Pointer Class Definition

With the LINK Tag, it's possible to define CSS classes outside of a HTML document, for instance on a dynamic Icon Pointer Storage server.

The Dynamic Icon Pointer Class definition (for instance generated on the fly by a PHP 'engine.php' script installed at <http://ICON.PROVIDER.NET/engine.php?ID=netcom> on the Icon Pointer Storage server) must have well-formed CSS2 syntax, such as

```
.bodypointer {
    cursor:
    url('http://ICON.PROVIDER.NET/ico/netcom-001.ico'),
    default
}
.linkpointer {
    cursor:
    url('http://ICON.PROVIDER.NET/ico/netcom-002.ico'),
    default
}
.headpointer {
    cursor:
    url('http://ICON.PROVIDER.NET/ico/netcom-003.ico'),
    default
}
```


If a page is generated on the fly, it should contain a LINK Tag (inside the HEAD Tag) with the HREF property set to the URL of the script that generates the Dynamic Icon Pointer Class Definition, such as:

```
<link rel="Stylesheet" type="text/css"
      href="http://ICON.PROVIDER.NET/engine.php?ID=netcom">
```

That way it is possible to redefine Icon Pointer Classes independent of a statically or dynamically generated HTML document:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Paragraph with External Icon Pointer Class Definition</title>
    <link rel="Stylesheet" type="text/css"
          href="http://ICON.PROVIDER.NET/engine.php?ID=netcom">
  </head>
  <body class="bodypointer">
    ...
    <a class="linkpointer" title="Enkel moro med Netcom MMS"
      href="http://www.avis.no/servlets/page?section=9&item=258007"
    >Telenor betaler NetCom</a>
    ...
    
    ...
  </body>
</html>
```

The advantage of managing Dynamic Icon Pointer Class Definitions outside the HTML document, is the ability to redefine and change one or more Icon Pointers once, and launch new campaigns on multiple documents or multiple domains immediately.

Obviously it is desired to serve Icon Pointers in time intervals for temporary and limited campaigns, and updates and rotation in more dynamical Icon Pointer campaigns.

2.5 Standard References

- Hypertext Transfer Protocol – HTTP/1.1 (RFC 2616)
W3C/MIT, June 1999
<ftp://ftp.isi.edu/in-notes/rfc2616.txt>
- Hypertext Markup Language – HTML 4.01
W3C Recommendation, 24 December 1999
<http://www.w3.org/TR/html4/>
- Cascading Style Sheets – level 2 Specifications
W3C Recommendation, 12-May-1998
<http://www.w3.org/TR/REC-CSS2/>

Appendix A

Client Standards

Our prototype is implemented in PHP with the Apache (see <http://httpd.apache.org/>) HTTP server, with XML storage.

cursor.dtd

Listing A.1: PHP Script for creating Icon Pointer Data

```
<!-- Cursorium Cursor XML 1.0 DTD

    Copyright (C) 2003 Blaafigk Independent Innovations ANS
    All rights reserved.

    Namespace = "http://www.cursorium.com/cursor/1.0"
    This DTD module is identified by the PUBLIC and SYSTEM identifier:

    PUBLIC "-//CURSORIUM//DTD_Cursor_1.0//EN"
    SYSTEM "http://www.cursorium.com/cursor/1.0"

    $Date: 2003/04/25 00:03:09 $
-->

<!ELEMENT cursor ( icon ) >
<!ATTLIST cursor
    version NMTOKEN #REQUIRED
>
<!ELEMENT icon ( id, name, text, link, created, publish, expires ) >
<!ELEMENT id ( #PCDATA ) >
<!ELEMENT name ( #PCDATA ) >
<!ELEMENT text ( #PCDATA ) >
<!ELEMENT link ( #PCDATA ) >
<!ELEMENT created ( #PCDATA ) >
<!ELEMENT publish ( #PCDATA ) >
<!ELEMENT expires ( #PCDATA ) >
```

Appendix B

Client Prototype

In the prototype, variable CSS class definitions are defined in a machine-generated CSS document and linked inside a HTML document using the LINK Tag explained in the previous chapter.

For Dynamic Icon Pointers, it is necessary to implement a server-side script on a Icon Pointer Provider server that is capable of processing one or more CGI parameters sent along with the GET request from the client to the HTTP/1.1 server.

If the server-side script is called with the ID argument, it should return the corresponding Icon Pointer Class definition as defined in the previous chapter.

Our prototype is implemented in PHP with the Apache (see <http://httpd.apache.org/>) HTTP server.

config.inc

Listing B.1: PHP Script for creating Icon Pointer Data

```
<?
include ("domain-config.inc");
$create_script="index.php";
$engine_script="engine.php";
$engine_url = "$client_url/" . "$engine_script";
$upload_script="$base_url/upload.php";
$loader_script="$base_url/admin/$domainname/0.php";
$fetch_script="$base_url/admin/$domainname/0.1.php";
$saver_script="$base_url/admin/$domainname/1.php";
$editor_script="$base_url/admin/$domainname/1.1.php";
$setup_script="$base_url/admin/$domainname/2.php";
$apply_script="$base_url/admin/$domainname/2.1.php";
$spview_script="$base_url/admin/$domainname/3.php";
$sign_script="$base_url/admin/$domainname/4.php";
$sign_confirm_script="$base_url/admin/$domainname/4.1.php";
$notif_script="$base_url/admin/$domainname/5.1.php";
$notif_confirm_script="$base_url/admin/$domainname/5.1.php";
$psign_script="$base_url/admin/$domainname/6.php";
$psign_confirm_menu="6.1.php";
$loader_menu="$base_url/admin/$domainname/0.php?template_uuid=$template_uuid";
$saver_menu="$base_url/admin/$domainname/1.php?template_uuid=$template_uuid";
$editor_menu="$base_url/admin/$domainname/1.1.php?template_uuid=$template_uuid";
$setup_menu="$base_url/admin/$domainname/2.php?template_uuid=$template_uuid";
$apply_menu="$base_url/admin/$domainname/2.1.php?template_uuid=$template_uuid";
$spview_menu="$base_url/admin/$domainname/3.php?template_uuid=$template_uuid";
$sign_menu="$base_url/admin/$domainname/4.php?template_uuid=$template_uuid";
$sign_confirm_script="$base_url/admin/$domainname/4.1.php?template_uuid=$template_uuid";
$notif_menu="$base_url/admin/$domainname/5.1.php?template_uuid=$template_uuid";
$notif_confirm_menu="$base_url/admin/$domainname/5.1.php?template_uuid=$template_uuid";
$psign_menu="$base_url/admin/$domainname/6.php?template_uuid=$template_uuid";
$psign_confirm_menu="$base_url/admin/$domainname/6.1.php?template_uuid=$template_uuid";

function get_template_id () {
    return $HTTP_COOKIE_VARS["TEMPLATE_UUID"];
}

function set_template_id ($id) {
    setcookie("TEMPLATE_UUID", $id);
}

function get_url_from_template ($template_uuid, $miscdata_path) {
    $filename = $miscdata_path . "/" . $template_uuid . ".url";
    $h = fopen($filename, "r");
    $url = fread($h, filesize($filename));
    fclose($h);
    return $url;
}

function load_content_from_template ($template_uuid, $template_path) {
    $filename = $template_path . "/" . $template_uuid; // ".html";
    $h = fopen($filename, "r");
    $content = fread($h, filesize($filename));
    fclose($h);
}
```

```

        return $content;
    }

    function save_content_in_template($content , $template_uuid , $template_path) {
        $filename = $template_path . "/" . $template_uuid; // ".html";
        $h = fopen($filename , "w+");
        fwrite($h , $content);
        fclose($h);
    }

    function debug($str) {
        echo "<pre style='font-size: 12px; color: #ff0000'><b>DEBUG</b>_{$str}</pre>\n";
    }

    function fixme($str) {
        echo "<pre style='color: #ff00ff; font-size: 12px;'><b>FIXME</b>_{$str}</pre>\n";
    }

    // <a href='$_saver_script?template_uuid=" . time() . "'>New</a>

    echo "<html>\n";
    echo "<head>\n";
    echo "<link rel='stylesheet' type='text/css' href='http://www.cursortium.com/common.css'>\n";
    echo "</head>\n";
    echo "<body>\n";
    echo "<h1>Configuration GUI</h1>\n";
    echo "<p><a href='$_create_script'>x. Upload Icon Data</a><br><a href='$_loader_menu'>0. Load</a><br><a href='$_saver_menu'>1. Save</a><br><a href='$_help'>2. Help</a></p>\n";
    echo "</body>\n";
    echo "</html>\n";
}
?>

```

domain-config.inc

Listing B.2: PHP Script for creating Icon Pointer Data

```

<?
/* Domain/ Site Configuration For Icon Server */
$domainname="cursortium.com";
$base_url = "http://www.cursortium.com";
$client_url = "$base_url/client/$domainname";
$admin_url = "$base_url/admin/$domainname";
$domain_path = "/home/petronas/www/cursortium/client/$domainname";
$template_path = "$domain_path/html";
$miscdata_path = "$domain_path/data";
$miscdata_url = "$client_url/data";
?>

```

server-config.inc

Listing B.3: PHP Script for creating Icon Pointer Data

```

<?

```

```

/* Server Configuration For Icon Server */
$cursor_version = "1.0";
/* Upload/download processing. */
$create_script_url="create.php";
$upload_script_url="upload.php";
$engine_script_url="engine.php";
/* HTML document processing. */
$filter_script_url="filter.php";
$parser_script_url="parser.php";
?>

```

create.php

Listing B.4: PHP Script for creating Icon Pointer Data

```

<?
/*
    $Id: create.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

    Copyright (C) 2003  Blaafisk Independent Innovations ANS
    Copyright (C) 2003  Ole Aamot

    All rights reserved.

    $Log: create.php,v $
    Revision 1.1.1.1  2003/04/25 00:03:09  oka
    Cursortium Version 1.0

*/
include("config.inc");
include("domain-config.inc");
include("server-config.inc");
?>

<h2>Register HTML Invocation Code</h2>

<form enctype="multipart/form-data" action="<?_echo_$upload_script;_?>" method="POST">
<table>
<tr>
<th>Domain Name:</th>
<td><select name="domainname">
<?
?>
<option value="<?_echo_$domainname;_?>"><? echo $domainname; ?> </option>
</select>
</td>
</tr>
<tr>
<th>Class Name:</th>
<td><input type="text" name="CLASS"> (no spaces allowed)</td>
</tr>
<tr>
<th valign="top">Tooltip Text:</th>
<td>
<textarea name="TOOLTIP" rows="4" cols="37"></textarea>

```



```

</td>
</tr>
<tr>
<th>Publish:</th>
<td>
<select name="publish">
<? $now = time();
    $i = $now;
    while ( $i < ($now + 86400*14)) {
        echo "<option value='$i'>" . date("Y-m-d H:i:s", $i) . "</option>\n";
        $i += 3600;
    }
?>
</select>
</td>
</tr>
<tr>
<th>Expires:</th>
<td>
<select name="expires">
<? $now = time();
    $i = $now;
    while ( $i < ($now + 86400*14)) {
        echo "<option value='$i'>" . date("Y-m-d H:i:s", $i) . "</option>\n";
        $i += 3600;
    }
?>
</select>
</td>
</tr>
<tr>
<th>Icon File:</th>
<td><input name="userfile" type="file"><input type="submit" value="Generate"></td>
</tr>
</table>
<input type="hidden" name="MAX_FILE_SIZE" value="40000">
</form>

```

upload.php

Listing B.5: PHP Script for creating Icon Pointer Data

```

<?
/*
$Id: upload.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

Copyright (C) 2003  Blaaflsk Independent Innovations ANS
Copyright (C) 2003  Ole Aamot

All rights reserved.

$Log: upload.php,v $
Revision 1.1.1.1 2003/04/25 00:03:09 oka
Cursortium Version 1.0

```

```

*/

include("domain-config.inc");
include("config.inc");
include("server-config.inc");

$camp-uuid = md5(uniqid(""));
$icon-uuid = md5(uniqid(""));
$link-uuid = md5(uniqid(""));

if (!preg_match("/\.ico$|\.bmp$|\.ani$|\.cur$/i", $HTTP_POST_FILES['userfile']['name'])) {
    exit("<p>You cannot upload this type of file.</p><p><a href='create.php'>Try again</a></p>");
}

if (move_uploaded_file($HTTP_POST_FILES['userfile']['tmp_name'], $miscdata_path . "/" . $HTTP_POST_FILES['userfile']['name'])) {

    symlink($HTTP_POST_FILES['userfile']['name'], $miscdata_path . "/" . $icon-uuid);

    $handle = fopen ($miscdata_path . "/" . $link-uuid, "a");
    $cursor = "<!--$CLASS_{cursor:url('engine_url?ID=$icon-uuid')}-->";
    fwrite($handle, $cursor);
    fclose($handle);

    echo "<h2>HTML Invocation Code</h2>\n";

    echo "<p><b>0.Rendered Display</b></p>\n";
    echo "<SPAN title='TOOLTIP' class='$CLASS'>\n";
    echo "<p>This is a interface element with the '$CLASS' override enabled.</p>\n";
    echo "</SPAN>\n";

//  echo "<table cellpadding=10>\n";
//  echo "<tr><th>CampaignID</th><th>ClassID</th><th>Tooltip Text</th><th>Filename</th></tr>\n";
//  echo "<tr><td>$camp-uuid</td><td>$CLASS</td><td>$TOOLTIP</td><td>\n";
//  echo "</td></tr></table>\n";
    echo "<p><b>1.Inline HTML Object Source Code (Static Tooltip):</b></p>\n";
    echo "<STYLE type='text/css'>";
    echo "<!--@import url('engine_url?ID=$link-uuid');-->";
    echo "</STYLE>";
    echo "<pre style='background: yellow; font-size: 12px;'>\n";
    echo "&lt;STYLE type='text/css'&gt;\n";
    echo "&lt;!--\n@import url('engine_url?ID=$link-uuid');\n--&gt;\n";
    echo "&lt;/STYLE&gt;\n";
    echo "\n";
    echo "&lt;SPAN title='TOOLTIP' class='$CLASS'&gt;\n";
    echo "This is a interface element with the '$CLASS' override enabled.\n";
    echo "&lt;/SPAN&gt;\n";
    echo "</pre>\n";

    echo "<p><b>2.Inline JavaScript Source Code (Dynamic Tooltip):</b></p>\n";

    echo "<pre style='background: yellow; font-size: 12px;'>\n";
    echo "&lt;SCRIPT src='engine_url?ID=$CLASS.on'&gt;&lt;/SCRIPT&gt;\n";
    echo "This is a interface element with the '$CLASS' override enabled.\n";
    echo "&lt;SCRIPT src='engine_url?ID=$CLASS.off'&gt;&lt;/SCRIPT&gt;\n";
    echo "</pre>\n";

    $xml = fopen("$miscdata_path/" . $icon-uuid . ".xml", "w");
    fwrite($xml, "<?xml version='1.0'?>\n");
    fwrite($xml, "<cursor version='$cursor-version' xmlns='http://www.cursorium.com/cursor/1.0'>\n");

```

```

fwrite($xml, "<icon>\n");
fwrite($xml, "<id>$icon.uuid</id>\n");
fwrite($xml, "<name>$CLASS</name>\n");
fwrite($xml, "<text>$TOOLTIP</text>\n");
fwrite($xml, "<link>$engine_url?ID=$link.uuid</link>\n");
$timestamp = date("Ymd\THis");
fwrite($xml, "<created>" . date("Ymd\THis", time()) . "</created>\n");
fwrite($xml, "<publish>" . date("Ymd\THis", $publish) . "</publish>\n");
fwrite($xml, "<expires>" . date("Ymd\THis", $expires) . "</expires>\n");
fwrite($xml, "</icon>\n");
fwrite($xml, "</cursor>\n");
fclose($xml);

echo "<p><a href=' $engine_url?ID=$icon.uuid.xml'>XML Data</a><a href=' http://www.cursorium.com/cursor/1.0'>XML DTD</a></p>\n";

$js = fopen("$miscdata_path/$CLASS.on", "w");
fwrite($js, "document.write(\"<STYLE type='text/css'>\");\n");
fwrite($js, "document.write(\"<!--@import url('$engine_url?ID=$link.uuid');-->\");\n");
fwrite($js, "document.write(\"</STYLE>\");\n");
fwrite($js, "document.write(\"<SPAN title='$TOOLTIP' class='$CLASS'>\");\n");
fclose($js);

$js = fopen("$miscdata_path/$CLASS.off", "w");
fwrite($js, "document.write(\"</SPAN>\");\n");
fclose($js);

echo "\n<p><a href=' create.php'>Register New Code</a></p>\n";
echo "</body>\n";
echo "</html>\n";
} else {

switch($HTTP_POST_FILES['userfile']['error']) {

case 0: /* No error; possible file attack! :( */
    echo "There was a problem with your upload.";
    break;
case 1: /* Uploaded file exceeds the upload_max_filesize directive in php.ini */
    echo "The file you are trying to upload is too big.";
    break;
case 2: /* Uploaded file exceeds the MAX_FILE_SIZE directive that was specified in the html form */
    echo "The file you are trying to upload is too big.";
    break;
case 3: /* Uploaded file was only partially uploaded */
    echo "The file you are trying to upload was only partially uploaded.";
    break;
case 4: /* No file was uploaded */
    echo "You must select an image for upload.";
    break;
default: /* A default error, just in case! :) */
    echo "There was a problem with your upload.";
    break;

}

echo "<p><a href=' create.php'>Try again</a></p>\n";
}

```

engine.php

Listing B.6: PHP Script for serving Icon Pointer Data

```
<?
/*
 $Id: engine.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

Copyright (C) 2003  Blaafisk Independent Innovations ANS
Copyright (C) 2003  Ole Aamot

All rights reserved.

$Log: engine.php,v $
Revision 1.1.1.1 2003/04/25 00:03:09  oka
Cursortium Version 1.0

*/

header("Cache-Control: no-cache");

$e = error_reporting();
error_reporting($e & (E_ALL-E_WARNING));

$referer = $_SERVER["HTTP_REFERER"];
$domain = explode("/", $referer);

if ( ($domain[2] == "www.cursortium.com")
    || ($domain[2] == "cursortium.com") ) {

    $fp=fopen("data/$ID", "rb");
    if ($fp) {
        fpassthru($fp);
    } else {
        header("HTTP/1.0 404 Not Found");
    }
} else {
    header("Location: ../index.php");
}
?>
```

cursor-option.php

Listing B.7: PHP Script for serving Icon Pointer Data

```
<?
$open-tags = array(
    'ID' => '<ID>',
    'ICON' => '<ICON>',
    'NAME' => '<NAME>',
    'TEXT' => '<TEXT>',
    'LINK' => '<LINK>',
    'CREATED' => '<CREATED>',
```

```

    'PUBLISH' => '<PUBLISH>',
    'EXPIRES' => '<EXPIRES>'
);

$close.tags = array(
    'ID' => '</ID>',
    'ICON' => '</ICON>',
    'NAME' => '</NAME>',
    'TEXT' => '</TEXT>',
    'LINK' => '</LINK>',
    'CREATED' => '</CREATED>',
    'PUBLISH' => '</PUBLISH>',
    'EXPIRES' => '</EXPIRES>'
);

function startElement($parser, $name, $attrs='') {
    global $open.tags, $temp, $current_tag;
    $current_tag = $name;
    if ($format = $open.tags[$name]) {
        switch ($name) {
            case 'ICON':
                // echo 'ICON: ';
                break;
            default:
                break;
        }
    }
}

function endElement($parser, $name, $attrs='') {
    global $close.tags, $temp, $current_tag;
    if ($format = $close.tags[$name]) {
        switch ($name) {
            case 'ICON':
                preview_fragment($temp);
                $temp = '';
                break;
            default:
                break;
        }
    }
}

function characterData($parser, $data) {
    global $current_tag, $temp, $catID;

    switch ($current_tag) {
        case 'ID':
            $temp['id'] = $data;
            $current_tag = '';
            break;
        case 'NAME':
            $temp['name'] = $data;
            $current_tag = '';
            break;
        case 'LINK':
            $temp['link'] = $data;
            $current_tag = '';
            break;
    }
}

```

```

case 'TEXT':
    $temp['text'] = $data;
    $current_tag = '';
    break;
case 'CREATED':
    $temp['created'] = $data;
    $current_tag = '';
    break;
case 'PUBLISH':
    $temp['publish'] = $data;
    $current_tag = '';
    break;
case 'EXPIRES':
    $temp['expires'] = $data;
    $current_tag = '';
    break;
default:
    break;
}
}

// function cursor_retrieve_xml_from_path()
// {
//     global $temp;
//     if ($temp['publish'] < date("Ymd\THis")) {
//         if ($temp['expires'] > date("Ymd\THis")) {
//             return ($temp['name'], $temp['id'], $temp['text']);
//         } else return "EXPIRED";
//     } else return "FUTURE";
// }

function preview_fragment() {
    global $temp;
    if ($temp['publish'] < date("Ymd\THis")) {
        if ($temp['expires'] > date("Ymd\THis")) {
            echo "<option value=\"" . $temp['id'] . "\">" . $temp['name'] . " („" . $temp['publish'] . " – " . $temp['expires'] . ")</option>\n";
            // echo '<style type="text/css"><!-- @import url("' . $temp['link'] . '");--></style>';
            // echo '<span title="' . $temp['text'] . '" class="' . $temp['name'] . '">';
        } else; // echo "EXPIRED";
    } else; // echo "FUTURE";
    // echo 'ID: ' . $temp['id'] . '<br/>';
    // echo 'Name: ' . $temp['name'] . '<br/>';
    // echo 'Text: ' . $temp['text'] . '<br/>';
    // echo 'Created: ' . $temp['created'] . '<br/>';
    // echo 'Publish: ' . $temp['publish'] . '<br/>';
    // echo 'Expires: ' . $temp['expires'] . '<br/>';
    // return list($temp['id'], $temp['name']);
}

function cursor_option_from_xml_file($xml_file) {

    // declare the character set – UTF-8 is the default
    $type = 'UTF-8';

    // create our parser
    $xml_parser = xml_parser_create($type);

    // set some parser options

```

```

xml_parser_set_option($xml_parser, XML_OPTION_CASE_FOLDING, true);
xml_parser_set_option($xml_parser, XML_OPTION_TARGET_ENCODING, 'UTF-8');

// this tells PHP what functions to call when it finds an element
// these functions also handle the element's attributes
xml_set_element_handler($xml_parser, 'startElement', 'endElement');

// this tells PHP what function to use on the character data
xml_set_character_data_handler($xml_parser, 'characterData');

if (!$fp = fopen($xml_file, 'r')) {
    die("Could not open $xml_file for parsing!\n");
}

// loop through the file and parse baby!
while ($data = fread($fp, 4096)) {
    if (!$data = utf8_encode($data)) {
        echo "ERROR!". "\n";
    }
    if (!xml_parse($xml_parser, $data, feof($fp))) {
        die(sprintf("XML error: %s at line %d\n",
            xml_error_string(xml_get_error_code($xml_parser)),
            xml_get_current_line_number($xml_parser)));
    }
}

xml_parser_free($xml_parser);

return $ret;
}

function cursor_retrieve_data_from_xml_file($xml_file)
{
    // declare the character set — UTF-8 is the default
    $type = 'UTF-8';

    // create our parser
    $xml_parser = xml_parser_create($type);

    // set some parser options
    xml_parser_set_option($xml_parser, XML_OPTION_CASE_FOLDING, true);
    xml_parser_set_option($xml_parser, XML_OPTION_TARGET_ENCODING, 'UTF-8');

    // this tells PHP what functions to call when it finds an element
    // these functions also handle the element's attributes
    $ret = xml_set_element_handler($xml_parser, 'startElement', 'endElement');

    // this tells PHP what function to use on the character data
    xml_set_character_data_handler($xml_parser, 'characterData');

    if (!$fp = fopen($xml_file, 'r')) {
        die("Could not open $xml_file for parsing!\n");
    }

    // loop through the file and parse baby!
    while ($data = fread($fp, 4096)) {
        if (!$data = utf8_encode($data)) {
            echo "ERROR!". "\n";
        }
    }
}

```

```

        if (!xml_parse($xml_parser, $data, feof($fp))) {
            die(sprintf( "XML error: %s at line %d\n",
                xml_error_string(xml_get_error_code($xml_parser)),
                xml_get_current_line_number($xml_parser)));
        }
    }

    xml_parser_free($xml_parser);
}

?>

```

0.php

Listing B.8: PHP Script for serving Icon Pointer Data

```

<?
/*
 $Id: 0.php,v 1.1.1.1 2003/04/25 00:03:08 oka Exp $

 Copyright (C) 2003  Blaafisk Independent Innovations ANS
 Copyright (C) 2003  Ole Aamot

 All rights reserved.

 $Log: 0.php,v $
 Revision 1.1.1.1 2003/04/25 00:03:08 oka
 Cursorium Version 1.0

 */
include("config.inc");
$template_uuid = md5(uniqid(""));
?>

<h2>0. Load Template</h2>

<h3>0A. New Template from URL over HTTP/1.1</h3>

<p>Enter the URL to a new template page to load and parse for cursor configuration:</p>

<form method="POST" action="<?_echo_" $fetch_script?template_uuid=$template_uuid";_?>">
<input type="text" name="pageurl" value="http://www.cursorium.com/cursor/dtdtest.html" size="50"/>
<input type="submit" value="Open URL"/>
</form>

<h3>0B. Modify Existing Template</h3>

<p>
<?
if ($dir = @opendir($template_path)) {
    while (( $file = readdir($dir)) != false) {
        if ($file != "." && $file != ".." && $file != ".htaccess") {
            list ($file_uuid, $ext) = split('.', $file);
            $file_uuid = substr($file, 0, 32);

```



```

        $lastmodtime = filemtime($template_path . "/" . $file_uuid);
        $lastmod = date("Y-m-d H:i:s", $lastmodtime);

        echo "<tt>[" . $lastmod . "]<a href='\$setup_script?template_uuid=$file_uuid'>" . get_url_from_template($file_uuid, $miscdata_path) . "</a><br>"
    }
}
closedir($dir);
}
?>
</p>

<p>To load and parse an existing template , enter the template UUID:</p>

<form method="POST" action="<?_echo_$setup_script;_?>">
<input type="text" name="template_uuid" value="" size="40"/>
<input type="submit" value="Open ID"/>
</form>

```

0.1.php

Listing B.9: PHP Script for serving Icon Pointer Data

```

<?
/*
    $Id: 0.1.php,v 1.1.1.1 2003/04/25 00:03:08 oka Exp $

    Copyright (C) 2003  Blaafisk Independent Innovations ANS
    Copyright (C) 2003  Ole Aamot

    All rights reserved.

    $Log: 0.1.php,v $
    Revision 1.1.1.1 2003/04/25 00:03:08  oka
    Cursortium Version 1.0

*/
include("config.inc");

// Function to strip headers and leave just page content
function strip_response_header($source) {
    $headerend = strpos($source, "\r\n\r\n");
    if (is_bool($headerend)) {
        $result = $source;
    }
    else {
        $result = substr($source, $headerend+4, strlen($source) - ($headerend+4));
    }
    return $result;
}

// error_reporting(E_ALL);

echo "<h2>0A. Load Template over HTTP</h2>\n";

$domain = explode("/", $pageurl);

```

```

$firstpart = $domain[0] . "/" . $domain[1] . $domain[2];
$baseurl = str_replace("$firstpart", "", $pageurl);
$hostname = $domain[2];

echo "TEMPLATE:␣$template_uuid<br/>";
echo "INITIAL:␣$firstpart<br/>";
echo "BASE:␣$baseurl<br/>";
// echo "SITE: $hostname<br/>";
$ports = explode(":", $hostname);
$host = $ports[0];
$port = $ports[1];

if (!isset($port)) {
    $port = "80";
}

echo "HOST:␣$host<br/>\n";
echo "PORT:␣$port<br/>\n";

$fp = fsockopen ("$host", $port, $errno, $errstr, 30);

if (!$fp) {
    echo "Errstr␣($errno)<br/>\n";
} else {
    $buff='';
    fputs ($fp, "GET␣$baseurl␣HTTP/1.0␣\r\nHost:␣$host\r\n\r\n");
    while (!feof($fp)) {
        $page = fgets ($fp,5000);
        $page = preg_replace("/<\/head>/si", "<link␣rel='stylesheet'␣type='text/css'␣href='␣$engine_url?ID=$template_uuid'/>\n<base␣href='' . $firstpart . $baseurl' />\n");
        $buff .= $page;
    }
    fclose ($fp);
}

$content = strip_response_header($buff);
?>

<h2>0.1. Edit/Save Template with unique identifier </h2>

<form method="POST" action="<?␣echo␣$saver_script␣␣␣"?template_uuid="␣␣␣$template_uuid;␣?>">
<input type="Submit" value="Save␣Template␣Content"/>
<textarea name="content" cols="90" rows="10"><? echo $content; ?></textarea><br/>
<input type="hidden" name="pageurl" value="<?␣echo␣$pageurl;␣?>" />
</form>

<? save_content_in_template($content, $template_uuid, $template_path); ?>
<? save_content_in_template($pageurl, $template_uuid . ".url", $miscdata_path);

```

1.php

Listing B.10: PHP Script for serving Icon Pointer Data

```

<?
/*
$Id: 1.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

```

```

Copyright (C) 2003  Blaafisk Independent Innovations ANS
Copyright (C) 2003  Ole Aamot

All rights reserved.

$Log: 1.php,v $
Revision 1.1.1.1  2003/04/25 00:03:09  oka
Cursortium Version 1.0

*/
include ("config.inc");?>

<h2>1A. Save modified template with unique id.</h2>

<?
    $content = stripslashes($content);
    save_content_in_template($content, $template_uuid, $template_path);
    save_content_in_template($content, $template_uuid . ".html", $template_path);
    debug("Saved template as <a href=' $pview_menu'>$template_uuid </a>");
echo  "<p>You may want to<b><a href=' $editor_menu'>Edit </a></b> or<b><a href=' $setup_menu'>Setup </a></b> the actual configuration for this template";

?>

```

1.1.php

Listing B.11: PHP Script for serving Icon Pointer Data

```

<?
/*
$Id: 1.1.php,v 1.1.1.1 2003/04/25 00:03:09  oka  Exp  $

Copyright (C) 2003  Blaafisk Independent Innovations ANS
Copyright (C) 2003  Ole Aamot

All rights reserved.

$Log: 1.1.php,v $
Revision 1.1.1.1  2003/04/25 00:03:09  oka
Cursortium Version 1.0

*/
include ("config.inc");
    $content = load_content_from_template($template_uuid, $template_path);
?>

<h2>Template Editor</h2>

<form method="POST" action="<?echo $saver_script . "?"template_uuid=" . $template_uuid;?>">
<input type="Submit" value="Save Content"/>
<input type="Reset" value="Reset Content"/>
<textarea name="content" cols="93" rows="20"><? echo $content; ?></textarea><br/>
<input type="hidden" name="pageurl" value="<?echo $pageurl;?>" />
</form>

```

2.php

Listing B.12: PHP Script for serving Icon Pointer Data

```
<?
/*
    $Id: 2.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

    Copyright (C) 2003  Blaafisk Independent Innovations ANS
    Copyright (C) 2003  Ole Aamot

    All rights reserved.

    $Log: 2.php,v $
    Revision 1.1.1.1 2003/04/25 00:03:09  oka
    Cursortium Version 1.0

*/
include("config.inc");
    include("cursor-option.php");
    $content = load_content_from_template($template_uuid , $template_path);
?>

<h2>2. Modify Config for Template with unique ID</h2>

<h2>2.1. Display current Template/Coding config.</h2>

<?
fixme("Display current config in editable HTML form.");
?>

<h2>2.2. Reconfiguration of Template/Coding config.</h2>

<?

// preg_match('/<cursor[^\>]*>(.*?)<\/cursor>/im' , $content , $matches);
// preg_match("/<cursor:(.*?)><\/cursor:(.*?)>/si" , $content , $tags);

preg_match_all("/<cursor:(.*?)>/s" , $content , $starttags , PREG_SET_ORDER);
preg_match_all("/<\/cursor:(.*?)>/s" , $content , $closetags , PREG_SET_ORDER);

$i = 0;
$j = 0;

// debug("starttag 0,1 = ". $starttags[0][1]);
// // debug("starttag 1,1 = ". $starttags[1][1]);
// debug("closetag 0,1 = ". $closetags[0][1]);
// debug("closetag 1,1 = ". $closetags[1][1]);

while ( $starttags[$i][1] != NULL) {
    // if ( $starttags[$i][1] == $closetags[$i][1]) {
    $tags[$j] = $starttags[$i][1];
    ++$j;
    ++$i;
    // }
}
}
```

```

$K = 0;

if ( $tags[0] == NULL) {
    echo "<p><b>No template element tags found!</b></p>\n";
    echo "<p>Perhaps you want to add some element tags by <a href='$_EDITOR_MENU'>editing </a> the template?</p>\n";
    echo "<p>Use the syntax &nbsp;&nbsp;&nbsp;<tt><!-- cursor:ELEMENTNAME&gt;...&lt;!-- cursor:ELEMENTNAME&gt;</tt> to markup a new element.</p>\n";
    exit(1);
} else {
    echo "<table cellpadding='10'>\n";
    echo "<tr><th>Element Name</th>\n";
    echo "<th>Icon ( Publish Date - Expires Date)</th></tr>\n";
    echo "<form method='POST' action='$_APPLY_MENU'>";

    while ( $tags[$K] != NULL) {

        echo "<tr><td valign='top'>$tags[$K]\n";
        echo "</td><td valign='top'>\n";
        echo "<select name=' $tags[$K]'>\n";
        echo "<option value=' default'>None</option>";
        if ( $dir = @opendir($miscdata_path)) {
            while (( $file = readdir($dir)) != false) {
                if ( $file != "." && $file != "..") {
                    if ( substr($file, 33, 36) == "xml") {
                        cursor_option_from_xml_file($miscdata_path . "/" . $file);
                    }
                }
            }
        }
        closedir($dir);
    }
    echo "</td></tr>\n";
    $K++;
}
echo "<tr><td><input type='submit' value='Save'></td></tr>\n";
echo "</form>\n";
echo "</table>\n";
}
?>
<br/>

```

2.1.php

Listing B.13: PHP Script for serving Icon Pointer Data

```

<?
/*
$Id: 2.1.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

Copyright (C) 2003  Blaafisk Independent Innovations ANS
Copyright (C) 2003  Ole Aamot

All rights reserved.

$Log: 2.1.php,v $
Revision 1.1.1.1 2003/04/25 00:03:09 oka

```

```

Cursorium Version 1.0

*/
include("config.inc");
    $content = load_content_from_template($template_uuid , $template_path);
?>

<h2>2.3: Apply New Config</h2>

<?
$filename = $miscdata_path . "/" . $template_uuid;
$fh = fopen($filename , "w+");
foreach ($HTTP_POST_VARS as $key => $value)
{
    fwrite($fh, "<!--_{$key}_cursor:_url('{$engine_url?ID=$value'});_-->\n");
    $items[$key] = $value;
}
fclose($fh);
?>

<p>New configuration for template <? echo $template_uuid; ?>
has been saved in <? echo $filename; ?>. You may want to
preview it.</p>

<?
$content = preg_replace_callback('/<cursor:(.*?)>/' , lookup_start , $content);
$content = preg_replace_callback('/<\/cursor:(.*?)>/' , lookup_close , $content);

save_content_in_template($content , $template_uuid . ".html" , $template_path);
debug("Saved/applied new config in template<a href='{$view_menu}'>$template_uuid </a>");

function lookup_start ($key)
{
    global $HTTP_POST_VARS;
    global $engine_url;
    global $template_uuid;
    // return "<!-- Start Definition for " . $key[1] . " -->\n<SCRIPT src='http://www.cursorium.com/client/cursorium.com/engine.php?ID=" . $key[1]
    return "<!--_Start_Definition_for_" . $key[1] . "_-->\n<style _type='text/css'><!--_@import _url(' " . $engine_url . "?ID=" . $template_uuid . "');_-->";
}

function lookup_close ($key)
{
    return "</SPAN>\n<!--_Close_Definition_for_" . $key[1] . " -->\n";
    // return "<SCRIPT src='http://www.cursorium.com/client/cursorium.com/engine.php?ID=" . $key[1] . ".off'></SCRIPT>\n<!-- Close Definition for "
}

```

3.php

Listing B.14: PHP Script for serving Icon Pointer Data

```

<?
/*
    $Id: 3.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

    Copyright (C) 2003  Blaafisk Independent Innovations ANS

```

```

Copyright (C) 2003 Ole Aamot

All rights reserved.

$Log: 3.php,v $
Revision 1.1.1.1 2003/04/25 00:03:09 oka
Cursortium Version 1.0

*/
include("config.inc");
$content = load_content_from_template($template_uuid . ".html", $template_path);

// For users prior to PHP 4.3.0 you may do this:
function unhtmlentities ($string)
{
    $trans_tbl = get_html_translation_table (HTML_ENTITIES);
    $trans_tbl = array_flip ($trans_tbl);
    return strtr ($string, $trans_tbl);
}
?>

<?
fixme("Connect_the_template_config_and_link_tag_in\_ $content.");
?>

<h2>3. Page Preview with the new config applied</h2>

<?
echo "<b><a href='\" . $client_url . "/html/\" . $template_uuid . ".html'>View this template in a separate window</a></b>"; // . ".ht
?>

<?
fixme("Frame_this\_ $content.");
?>

<? echo $content;

// unhtmlentities($content); ?>

```

4.1.php

Listing B.15: PHP Script for serving Icon Pointer Data

```

<?
/*
$Id: 4.1.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

Copyright (C) 2003 Blaaflsk Independent Innovations ANS
Copyright (C) 2003 Ole Aamot

All rights reserved.

$Log: 4.1.php,v $
Revision 1.1.1.1 2003/04/25 00:03:09 oka
Cursortium Version 1.0

```

```

*/
include("config.inc");
    $url = get_url_from_template ($template_uuid , $miscdata_path);
?>

<?

if ($username == "ola" && $password == "secret") {

?>

<h2>Confirmation of Successful Signing</h2>

<p>You have successfully signed the
new template configuration ( <? echo "<a href='$pview_menu'$template_uuid </a>"; ?>)
for the page <a href='<? echo $url; ?>'><? echo $url; ?> </a>.</p>
A notification mail below has been sent to the publisher. Goodbye!</p>

<h2>Notification Mail (Sent to Publisher)</h2>

<pre style='font-size:12px;'>
To: Publisher
From: <? echo $username; ?>@cursortium.com
Subject: BLAAFISK NOTIFICATION #1 <? echo $url; ?>

This mail is just to inform you that the new modifications
to the configuration of the template

<? echo $url; ?>

has been proposed by the user '<? echo $username; ?>'.

To acknowledge the modification , visit
<a href='<? echo $sign_script . ".php"?template_uuid=$template_uuid; ?>'><? echo $sign_script . ".php"?template_uuid=$template_uuid; ?> </a>

Have a nice day!
</pre>

<?
} else {
    exit("Bad_username/password.<a href='$sign_menu'$>Try again.</a>\n");
}

```

4.php

Listing B.16: PHP Script for serving Icon Pointer Data

```

<?
/*
    $Id: 4.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

    Copyright (C) 2003  Blaafisk Independent Innovations ANS

```



```

Copyright (C) 2003 Ole Aamot

All rights reserved.

$Log: 4.php,v $
Revision 1.1.1.1 2003/04/25 00:03:09 oka
Cursortium Version 1.0

*/
include("config.inc");
$url = get_url_from_template($template_uuid, $miscdata_path);
?>

<h2>4. Creator Configuration Signature </h2>

<p>Confirm of the final configuration
for the URL <a href='<?_echo_$url_?>'><? echo $url; ?></a> (<a href='<?_echo_$pview_menu;_?>'>preview </a>):</p>

<? debug("Username is 'ola' and password is 'secret'"); ?>

<form method="POST" action="<?_echo_$csign_confirm_script;_?>">
<input type="hidden" name="template_uuid" value="<?_echo_$template_uuid;_?>">
Username: <input type="username" name="username"/><br/>
Password: <input type="password" name="password"/><br/>
<input type="submit" value="Yes, I want to sign it."/>
</form>

```

5.php

Listing B.17: PHP Script for serving Icon Pointer Data

```

<?
/*
$Id: 5.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

Copyright (C) 2003 Blaafisk Independent Innovations ANS
Copyright (C) 2003 Ole Aamot

All rights reserved.

$Log: 5.php,v $
Revision 1.1.1.1 2003/04/25 00:03:09 oka
Cursortium Version 1.0

*/
include("config.inc");
$url = get_url_from_template ($template_uuid, $miscdata_path);
?>

<h2>6. Publisher Notification </h2>

<p>To manually notify the publisher about a
new template configuration (<? echo "<a href='$pview_menu'$template_uuid </a>"; ?>)
for the page <a href='<?_echo_$url;_?>'><? echo $url; ?></a>, press the
'Notify Publisher' button below.</p>

```

```

<form method="POST" action="<?_echo_<_notif_confirm_menu;_?>">
<input type="hidden" name="template_uuid" value="<?_echo_<_template_uuid;_?>">
Username: <input type="username" name="username"/><br/>
Password: <input type="password" name="password"/><br/>
<input type="submit" value="Notify_<_Publisher">
</form>

```

5.1.php

Listing B.18: PHP Script for serving Icon Pointer Data

```

<?
/*
    $Id: 5.1.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

    Copyright (C) 2003  Blaafisk Independent Innovations ANS
    Copyright (C) 2003  Ole Aamot

    All rights reserved.

    $Log: 5.1.php,v $
    Revision 1.1.1.1 2003/04/25 00:03:09 oka
    Cursortium Version 1.0

*/
include("config.inc");
    $url = get_url_from_template ($template_uuid , $miscdata_path);
?>

<h2>5.1 Notification Mail (Sent to Publisher)</h2>

<pre style='font-size:12px;'>
To: Publisher
From: <? echo $username;?>@cursortium.com
Subject: BLAAFISK NOTIFICATION FOR <? echo $url;?>

This mail is just to inform you that modifications
to the configuration of the template

    <? echo $url;?>

has been suggested by the user 'ola'.

To acknowledge the modification , visit
<a href='<?_echo_<_psign_script_<_?>"?template_uuid=$template_uuid;_?>'><? echo $psign_script . "?"template_uuid=$template_uuid";?></a>

Have a nice day!
</pre>

```

6.php

Listing B.19: PHP Script for serving Icon Pointer Data

```
<?
/*
    $Id: 6.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

    Copyright (C) 2003  Blaafisk Independent Innovations ANS
    Copyright (C) 2003  Ole Aamot

    All rights reserved.

    $Log: 6.php,v $
    Revision 1.1.1.1 2003/04/25 00:03:09 oka
    Cursortium Version 1.0

*/
include("config.inc");
    $url = get_url_from_template($template_uuid, $miscdata_path);
?>

<h1>6. Configuration Signature (for Publishers)</h1>

<p>You, the publisher, have accepted the new
template configuration ( <? echo "<a href='$_spview.menu'>preview </a>";
?>) for the URL <a href='<?echo $_url;?>'><?echo $url;?> </a> and
that these modifications will be official.</p>

<? debug("Username is '$_per' and password is '$_secret'"); ?>

<form method="POST" action="<?echo $_psign_confirm_menu;?>">
<input type="hidden" name="template_uuid" value="<?echo $_template_uuid;?>">
Username: <input type="username" name="username"/><br/>
Password: <input type="password" name="password"/><br/>
<input type="submit" value="Sign The New Config"/>
</form>
```

6.1.php

Listing B.20: PHP Script for serving Icon Pointer Data

```
<?
/*
    $Id: 6.1.php,v 1.1.1.1 2003/04/25 00:03:09 oka Exp $

    Copyright (C) 2003  Blaafisk Independent Innovations ANS
    Copyright (C) 2003  Ole Aamot

    All rights reserved.

    $Log: 6.1.php,v $
    Revision 1.1.1.1 2003/04/25 00:03:09 oka
```

```

Cursortium Version 1.0

*/
include("config.inc");
$url = get-url-from-template($template.uuid , $miscdata-path);
?>

<?
if ($username == "per" && $password == "secret") {
?>

<h2>Confirmation of Signature (by Publisher)</h2>

<p>Thanks for confirming the new template
configuration (<a href='<?_echo_"$pview-menu";_?>'>preview</a>)
for the URL<a href='<?_echo_"$url";_?>'><? echo $url; ?></a>

<p>This configuration will soon be permanently activated.</p>

<?
} else {
exit("The_confirmation_failed...Bad_username/password...<a href='$psign-menu'>Try_again</a>\n");
}
?>

```