

Creator – My Way to create Hompages

Gebhard Greiter, Jan 2011

Creator is Freeware – my personal way to create a homepage the easy way. Creator is free to use for everyone. You use it as follows:

- Download <http://greiterweb.de/Creator.zip> and unpack it.
- What you get is a folder **Creator** consisting of
 - Creator/**empty**
 - Creator/**index.htm**.
- Create Creator/**home** as a copy of Creator/**empty** (instead of **home** you may choose any other project name) . Now you have
 - Creator/home/**cms**
 - Creator/home/**core**
 - Creator/home/**web**

Files cms/**frame_*** are templates for the Look & Feel of the homepage you want to create. These files (usually only one: cms/**frame_Master**.TXT) need to be maintained manually.

Creator/cms/**frame_Master** seems quite messy but is so only because there are lots of comments conserving knowhow I do not want to forget. A clean version of **frame_Master** is **frame**, the so called *clean Master Template* (you are to maintain frame_Master – **frame** will be updated automatically; read **frame** if you want to see the currently active templates.

In exceptional cases you may need, in parallel to the master template, variants thereof. Create them by creating and customizing copies of cms/**frame**. Their name must have the form **frame_X**, X not containing spaces.

Creator/cms/**web** is the folder you will – later on – copy, rename and then transfer to your web server. HTML or PHP files go to this folder automatically. Other files referred to by such code need to be stored by yourself in subfolders of cms/**web**. They may have any name, but their name should be short. By default we have

- web/**css** to store CSS files; there is already a **screen.css**
- web/**pix** to store pictures
- web/**dox** to store documents, usually in PDF format

All the files **web/*.htm** and **web/*.php** are created, updated and checked for consistency automatically as soon as you doubleclick **core/_UPDATE_core_and_web**.

UPDATE_web is to create for each **core**/*.htm a corresponding **web**/*.htm file, and also for each **core**/*.php a corresponding **web**/*.php file. This will be done by

- first slightly streamlining the core files
- and then expanding include commands found therein.

Include commands in this sense are all the lines

```
<!-- sec/+N, frameName -->
<!-- sec/- -->
<!-- sec/footer/?
<!-- sec/footer/DD.MM.YYYY
```

Parameter `frameName` must be `frame` or the name of an another existing template file `core/frame_*` whereas `DD.MM.YYYY` is the date you want to be given as the day when the page content (under a semantical point of view) was updated the last time. Replacing it by a question mark has the effect that it will become automatically the day on which `home/web` was updated the last time.

The number `N` in `sec/+N` refers to a section of the given frame file (a frame in this sense is NOT a frame in the sense of HTML, it is a template). A section in this sense is starting with a `sec/+` line and ending with the last line not containing either `sec/+` or `sec/-`.

Please note: Every `sec/+` must be followed by the line containing `sec/-` (future versions of Creator/tools/cms.exe will be able to add them automatically).

Very important: None of these lines is allowed to contain leading spaces.

All text fragments needed for more than only one of your `web`/*.htm or `web`/*.php files should be made a specific section of a file `core/frame_*` (usually a section of `core/frame` maintained in `core/frame_Master`).

Variants `core/frame_*` of the master template `core/frame` are needed only if – depending on which page the user is – certain submenus of the navigation menu should be opened automatically or should be hidden away.

At this point you may already be able of getting started with using Creator.

Simply see the `core`/*.htm files already there (or **samples**/core/*.htm).

Note: All the executables activated directly or indirectly by `core/_UPDATE_core_and_web` are contained in **Creator/tools** (which should be seen by you as a black box). If you insert the absolute path to this tools folder into your PATH environment, your web folder can also be updated out of a batch file by a call

```
cms publish Creator/home
```

Updating the **web** folder will always end in popping up a file allowing you to configure keywords and a **sitemap** generated automatically on the page `web/suche.htm` (die sog. **Stichwortliste**).

Why and how to use Creator

Creating and maintaining a homepage is simply too time-consuming and too error-prone to be done with only an HTML editor.

Most people therefore use an Open Source CMS (or even a commercial one). To do so has disadvantages:

- A CMS can be complex as well. I did not want to tolerate this because most functionality of such a product is not what I really need.
- CMS systems are usually PHP applications, not however self-contained programs.
- When I want to have functionality not supported by such a CMS, it is hard to generalize the system to support such a new feature.

Advantages of using Creator are:

- It was created by myself, and so I can easily generalize it in some few hours whenever I feel the need for an additional feature: to create useful new features can be fun.
- Using Creator is – for me as well as for every user who can handle and edit files – very easy: All the documentation you need to study is this manual here (6 pages).
- Using Creator could even be explained in school (Gymnasium, Volkshochschule).
- Version control – of Creator including all my applications or any one homepage project – can be done just by renaming and zipping one of the subfolders of Creator.

To start working on a new project,

- copy Creator/**empty** to, say, Creator/home
- customize home/cms/**frame_Master**
- customize text in home/core/(**index, Kontakt, Impressum**).htm
- and then create additional content files **core/anyName.htm**
- Copy any other files needed (pictures, documents), to subfolders of home/**web**.
- By double clicking on core/_UPDATE_core_and_web all files core/*.htm will be expanded to corresponding code web/*.htm
- Automatically created is also a file **web/zz.htm** which you can check via the online HTML validator <http://www.validome.org/> as soon as you have transferred a copy of the **web** folder (**under this or a different name**) to the root of your web server.

The file Creator/index.htm

is the only HTML file you should put into the root of your web server. The URL therein is the address of the first page a visitor of your site is meant to see.

How to Debug your HTML

After opening the online validator **Validome**, click on “Erweiterte Einstellungen” und then check the box “Quellcode anzeigen”. Via the drop down menu “Zeichenkodierung” select “ISO-8859-1 (Western Europe)” from the set “European Languages”. You forgot to do so if Validome is reporting a fatal error.

Click on the radio button “URL” and enter there <http://yourDomain/yourWebName/zzz.htm>.

Then click the button “Validieren”. One or two seconds later you will see a listing telling you all the still existing problems with your code (**web/zzz.htm** is the sum of all your **web/*.*htm** corrected to the effect that validators will not show false alarms).

- Correct your code,
- doubleclick core/_UPDATE_core_and_web,
- transfer web/**zzz.htm** to the web server,
- and then click again on button “Validate” on the Validome page.

Repeat this as long as Validome is describing at least two problems (the first one described is not a problem at all – Validome does not yet support HTML5, the HTML version our templates use):

The screenshot shows the Validome website interface. At the top, it says "HTML / XHTML / WML / XML Validator" and "validome". Below that are navigation links: "Forum | Ressourcen | Sidebar | Technologien | Bugmeldung | Kontakt | About | Weblog".

The main form is titled "Erweiterte Einstellungen" (Advanced Settings). It has several sections:

- Anzeige** (Display):
 - Quellcode anzeigen (Show source code)
 - Dokumentenbaum anzeigen (Show document tree)
 - Headerdaten anzeigen (Show header data)
 - Bereinige HTML mit Tidy (Clean HTML with Tidy)
- HTTP-Header**:
 - User-Agent: Mozilla/4.0 (compatible; MSE 6.0; Windows NT 5.1)
 - Accept-Language: Nicht vorhanden
 - Accept-Encoding: gzip
- Dokumententyp** (Document type):
 - Dropdown menu: automatische Erkennung
- Zeichenkodierung** (Character encoding):
 - Dropdown menu: ISO-8859-1 (Western Europe)
- Quelle** (Source):
 - Radio button selected: URL
 - Text input: http://greiterweb.de/web21/z.htm
 - Radio button: Upload
 - Button: Browse...

At the bottom of the form are buttons for "Hilfe" (Help), "Erweiterte Einstellungen" (Advanced Settings), and "Validieren" (Validate).

The validation result is shown in a red box: "Das Dokument ist nicht valide." (The document is not valid).

Below that, it says: "Benutzte Zeichenkodierung: iso-8859-1" (Used character encoding: iso-8859-1) and "Quelle: Ausgewählte Zeichenkodierung" (Source: Selected character encoding).

Then, it provides a link to re-validate the document with the current settings: "Wenn Sie einen Link auf Validome setzen möchten, mit dem das soeben validierte Dokument inkl. Ihrer individuellen Einstellungen, erneut validiert werden soll, dann verwenden Sie bitte den folgenden Link." (If you want to set a link on Validome with which the just validated document including your individual settings should be validated again, please use the following link.)

The link is: <http://www.validome.org/validate/?uri=http://greiterweb.de/web21/z.htm&extended=1&viewSource=1&charset=ISO-8859-1&userAgent=Mozilla/4.0%20compatible%3b%20MSE%206.0%3b%20Windows%20NT%205.1%29>

Below the link, it says "Fehler (1)" (Error (1)).

The error details are shown in a table:

Zeile	Spalte: 15
2	Fehler: Keine interne oder externe Teilmenge der Dokumenttypdeklaration, deshalb keine Validation möglich
	Fehlerstelle: <!DOCTYPE html><HTML>

Even better than **Validome** – and usable locally – is **TotalValidator** (available at <http://www.totalvalidator.com/downloads/index.html>). It will be called automatically if Creator/tools/TotalValidator/TotalValidator.exe exists.

Specific Features of Creator

After creating my first homepage with the help of Creator I was happy because the burden of creating a new file **web/***.htm was now reduced to simply editing my text section in form of the corresponding (much shorter) file **core/***.htm.

Soon however I realized that even the quite trivial formatting I had to do could further be simplified and made more generic. What I learned was:

HTML Macros – a concept only supported by Creator – can be very helpful.

HTML Macros exist in different forms: CMS, User, and Power Macros.

CMS macros are shorthand HTML as follows:

- `<X>`, `<Xe>`, `<eX>` (**X** a positive integer) are expanded to a sequence of `X` `
` tags, to `<X>`, or to `<X>` respectively.
- `<span/X>` will be expanded to `` (**X** a class identifier).
- An empty tag `<>` will be expanded to `­` (the potential line break marker)
- `<-tr` is short for `</tr><tr`
- `<-td` is short for `</td><td`
- `<./X>` will be expanded to the content of the file **core/X**, **X** a path
- `<@X>` is short for ``

User macros, also shorthand HTML, are specified by yourself in **cms/Macros** as tag pairs

`<X>:<anyHTML>`

A `<tag>` in this sense is a non-empty short sequence of printable characters, preferably digits or **case-sensitive** letters. In core files, such a pseudo tag `<X>` may be given attributes meant to be attributes of the last tag in `<anyHTML>`.

I myself use a lot `<X>` and a User Macro of the form `</h2>:</h2>

` (to ensure suitable layout for all my content is now no longer a pain in ass).

The above abbreviation for `­` also turned out to be helpful.

By the way: Creator is currently supporting the **code set ISO-8859-1** only. This is to ensure that characters used very often in German (Umlaute, ß, quotes, u.A.) need not be written in the form of HTML entities. As far as **core/***.htm files contain non-mandatory HTML entities, these file will be simplified automatically (and permanently).

To creating text for a homepage, I usually first create a draft in a Word file. Only after – from a semantical point of view – a text section is acceptable I copy it as unformatted text to one or more **core/***.htm file.

Using CMS Macros I can then easily break it up into a suitable sequence of paragraphs and keep my text readable. Sequences `</p><p>` are to be avoided but. They are necessary only if the two `<p>` objects need to have a different class attribute.

In short: HTML Macros, simple as they are, help me a lot. Thanks to them my core HTML became more readable, faster to create, and easier to maintain.

My favorite editor is **Gizmo Manager**: a simple text editor with syntax highlighting for HTML. It comes as freeware, download from <http://arainia.com/software/gizmo/download.html>.

Here is an example: To see whether text looks nicer if using `–` instead of `—` or vice versa was hard as long as I had this HTML entity hard-coded in my text. Much better is to use a macro `<->` defined as, e.g.

```
<->:&ndash;
```

so that the dash implementation can be replaced in all `core/*.htm` files just by reconfiguring this macro.

Because User Macro values can be any text – including quite long character sequences, or even class IDs – many more applications will certainly come up. The longer the macro's value is, the more helpful such a mechanism will be.

Macro tags may occur nested in the set of attributes of HTML object tags.

Power Macros are really powerful:

A section macro is content created and maintained automatically by Creator.

Stichwortliste, a kind of semantical site map added automatically to `web/suche.htm`, is one example:

When updating `web/*.htm`, the publish function of Creator is also looking for text T between `` and `` tags. If the first character of T is a capital letter and if T does not contain another HTML tag, T is considered a **keyword candidate**.

Keyword candidates are shown on `web/suche.htm` in form of a hyperlink pointing to the page containing this keyword. Before copying the `web` folder to the web server, please read this page to see if there are entries you do not want to have there. To remove such a keyword entry, replace in core files the `` in front of it by a ``.

If, when HTML formatting your text in `core/*.htm`, you know already that some specific text in bold should not go to the Stichwortliste, write `` instead of ``.

Not having a Stichwortliste is also possible: Add a command

```
<!-- sec/no_sitemap
```

immediately preceding the footer section in the file `core/suche.htm`.

Capitalized words found between `<h2>` and `</h2>` are called *header keyword candidates*.

They are written to `home/cms/kwf` and are classified there via a plus sign, a minus sign or a question mark. Each update of the `web` will automatically pop up `kwf` opened in Notepad thereby reminding you that reclassifying or adding header keyword candidates manually is always possible. Header keywords shown in `kwf` with a plus sign go from then on into the content attribute of the page's meta data describing to search engines what the page is about.

Note: The files `core/(index, suche, Kontakt, Impressum).htm` are mandatory. As far as missing, Creator will re-create them automatically by copying the corresponding files found in the `core/reuse` folder. When doing a homepage, these four files are the first you should design: Copy them from the `reuse` folder, customize them, and then copy them back to `reuse`.

Folder `core/not` is to hide currently not relevant core files we may want to keep (e.g. drafts).

Placeholder pages are another example for Power Macros:

For files referred to from the navigation menu but not yet existing, a placeholder page similar to core/reuse/**Baustelle**.htm is created automatically (Precondition: you have to provide a template in form of a page core/Baustelle.htm).

Furthermore: Include commands of the form

<./X>

are replaced by the contents of the file **core/X**, **X** any path into the tree **core/**. This will allow you to maintain all your forms as separate files stored under core/**forms**. Descriptions of locations e.g. can be stored under core/**Location** etc.

Files **core/z_X**.htm are fully maintained by power macros and may vanish automatically. You need to open them only if you suspect bugs in Creator.

How to check your web's Integrity

Creating the **web** via the batch file core/_UPDATE_core_and_web will also create a file core/_**check_hrefs**. A double mouse click on this batch command will tell you whether all URLs found in web/*.**htm** or web/*.**php** actually address currently existing targets.

Dear potential user of Creator:

If you find a bug or have a suggestion for improving Creator or this manual, please feel free to send me a corresponding email.

I promise to correct all bugs reported. You will be informed by me as soon as a corrected version of the Creator software is available.

If you get lost, please send me a zip file containing your complete **Creator** folder. Help from me should arrive soon (because I am eager to learn what problems exactly beginning homepage builders might encounter when using Creator).

Note also: The distribution www.greiterweb.de/Creator.zip may have been updated already after your last visit. So, if problems occur, please try first the latest version (calling Creator/tools/**cms.exe** without parameters will tell you the number YYYY.MM.DD.XXXX of the latest version).

Thank you.

Gebhard Greiter

ggreiter@gmx.de