

<The>

Gerolf

<Markup>

Shredder

<Handbook>

Version 0.08a

</Handbook>

2008-01-07

</Markup>

HTML/PDF-conversion with TeX

</The>

G. D. Brettschneider:
The Gerolf Markup Shredder Handbook
Version 0.08a (2008/01/07)
Bremen/Germany (1999–2008)
All rights reserved.
<http://www.Gerolf.org>
[MarkupShredder\(at\)Gerolf.org](mailto:MarkupShredder(at)Gerolf.org)

Index

Begin

Screen | Print | Interface | Work

Quit

Create

Open

View

Edit

Browse

Analyse

Typeset

Errors | Images | Styles | Hyphens | Ancestors

Read

Split

Select

Animation | Colors | Programs

Debug | Encoding | Codepage

Write

Glyphs | Fonts

Init

Download

Archives | Server | Unzip | Setup

Launch | Integration | System

Feedback

Appendix

*Gerolf Markup Shredder
Copyright (c) 1999–2008 by
Gerolf Diethelm Brettschneider,
Luchtbergstr. 27, D–28237 Bremen.
This software comes without any warranty.
You may freely distribute and use it.*

Begin

Screen

Documents to be published at the **World Wide Web** must be built as **Markup** lists, as shown in the left frame below: The main heading gets enclosed by `<h1>` and `</h1>` **Delimiters**, the words belonging to a paragraph are put between a `<p>` start tag and a `</p>` end tag. `
` makes a line break, `<hr />` draws a horizontal line. Special **Characters** like German umlauts or quotation marks can be written in mnemonic or numerical form, like `ö` or `“`. Using this additional information, the web browser renders the document on the screen in an appropriate way (right):

```
<html>
<head>
<title>A Short Story</title>
</head>
<body>
<hr /><br /><br /><br />
<h1 align = "center">
  A SHORT STORY</h1>
<p align = "center">
  <i>by D. E. Knuth</i></p>
<p style = "text-indent:0mm;
margin-bottom:0mm"><br />
  Once upon a time, in
  a distant galaxy called
  &Ouml;&ouml;&ccedil;, there
  lived a computer named
  R.&nbsp;J. Drofnats.</p>
<p style = "text-indent:20pt;
margin-top:0mm">
  Mr.&nbsp;Drofnats&#8212;or
  &#8220;R. J.,&#8221;
  as he preferred to be
  called&#8212;was happiest
  when he was at work
  typesetting beautiful
  documents.</p>
<br /><br /><br /><hr />
</body>
</html>
```

A SHORT STORY

by D. E. Knuth

Once upon a time, in a distant galaxy called Ööç, there lived a computer named R. J. Drofnats.

Mr. Drofnats—or “R. J.,” as he preferred to be called—was happiest when he was at work typesetting beautiful documents.

A writer cannot predict exactly how the content of the document, which is marked up in that way, gets rendered by the readers' various internet browsers. For instance, line breaking often depends on window width, and some special characters may not be available on the user's system.

Print

Documents for high-quality **Print** are stored in files of a different kind, filled with information about the shape and precise position of every single letter. These files are not readable for humans directly. From the World Wide Web Consortium (W3C), for example, you can download the Extensible Hypertext Markup Language specification (XHTML) in Portable Document Format (**PDF**) for print with Adobe Acrobat Reader – and you can use Gerolf Markup Shredder (GMS) to create such portable documents out of your hypertext files.

However, Markup Shredder still is buggy and incomplete. Until now, it cannot handle auto-width tables, and it supports only a subset of **HTML** element tags, attributes and Cascading Style Sheet (CSS) properties. Users should acquire basic knowledge of the markup language because it is often necessary to modify an input file to get a satisfactory PDF conversion. The typesetting is done by an old program called **TeX**, which is said to have an error-free kernel and to be able to produce high-quality print files.

Interface

Markup Shredder runs under *Linux*, *Dos* and *Windows* on Intel-386 compatible machines. It comes with three different user interfaces for the menu items, which form the index of this handbook:

- A **Web Browser** interface – slow, but user-friendly –, which allows to execute the GMS functions with just one mouse click from the button bar. Everything can be displayed within the same framework, even the typesetting result, using the Adobe Acrobat Reader plug-in. This interface is written in PHP language and requires installation of the XAMPP package on the computer.
- A **Text Mode** interface, looking pretty old-fashioned, which is used by default on all supported operating systems. The input file can be displayed in an internal viewer, but editor, HTML browser and PDF reader are executed as external programs. This interface, as well as the third, is implemented as a redundant system of shell scripts (*Linux*) and batch files (*Dos*, *Windows*).
- A **Command Line** interface, fast and simple, which is used behind the scenes by the text mode interface. It may be the best choice on old *Dos* systems, and it can also be used to integrate Gerolf Markup Shredder in advanced editors like (X)Emacs (*Linux*) or HTML-Kit (*Windows-32*).

Work

When started for the first time, GMS will **Open** the file `default.htm`, a small, but non-trivial markup file. Now you can do the following:

- **View** the text of the file. **Edit** it. **Browse** through the rendered content of the file. **Analyse** the usage of markup, running HTML-Tidy. **Typeset** the file, running the TeX engine. In Adobe Acrobat **Reader**, you can enjoy the PDF that has been produced.
- **Open** the other markup files collected in the `[GMS_ROOT]/doc` directory. Each one, placed in its own sub-directory with the same name, was tested successfully with GMS. **Create** a new file, based on `maximal/maximal.htm` as a template, for example, and change it according to your needs. **Learn** how to use GMS and study `handbook/handbook.htm` (this document) as well as `guide/guide.htm` and `primer/primer.htm`, if you're new to HTML & CSS.
- **Select** the external programs to associate with GMS and the main encoding of characters. Adding fonts to sub-folders of `[GMS_ROOT]/fonts` might require you to modify `font.cfg`, `encoding.cfg` and `alias.cfg` files in the `[GMS_ROOT]/etc` folder. **Write** a

new font map while computing glyph metrics. **Initialize** the TeX format file `gerolf.efmt` in `[GMS_ROOT]/bin/[system]`.

Quit

Web Browser

If you click on **[Q]**, your *document* will be *removed* from the *server*. Your browser still may hold a copy of that file in its *cache* allowing to recall the document (with the help of the **[Back]** button) if you have not *saved* it yet.

Text Mode

If you press **[Q]** in the *text mode* interface to *quit* Gerolf Markup Shredder, the batch processor shell with *GMS* environment variables will be removed from memory.

Command Line

If you enter `gms -q` or `gms /q` into the *command line* to *quit* Gerolf Markup Shredder, the batch processor shell with *GMS* environment variables will be removed from memory.

Create

Web Browser

You will have most success working with Markup Shredder, if you start to *create* your document on the basis of a template file that is tested to be convertible into PDF by GMS. Replace the old content with new text written by yourself, leaving intact parts of the markup, which tells the browser (and GMS) how to render the content.

Click on **[C]** to see the list of templates which are available. Select one and click on **[Accept]**. GMS then will *create* a copy of it, give it a random name and save it temporarily in the server's folder for guests' files. You get a short acknowledgement, if you are given access to this file, so you can *edit* it.

You still have to download this file and save it under an appropriate name on your local computer. Your file will be removed from the server after a certain time, depending on creation date.

Text Mode

If you press **[C]** to *create* a new document, you see the simple file selection dialog that is described in the *Open* section. This window shows the content of `[GMS_ROOT]/doc`, the template directory.

1. Change into one of the sub-folders and select the HTML file that is given the same name.
2. Change into another existing directory where you want to save a copy of the template file and press **[Q]** to *quit* the dialog.
3. Enter an appropriate name for this new file. You will be prompted if a file with that name already exists.

On *Windows 32*, use Explorer to create a copy of any template. Click on *Copy* and *Paste* in the context menu which is accessible with the right mouse button. In KDE for *Linux*, select *Copy to* in the context menu.

Command Line

Enter `gms -c /myfolder/newfile.htm default/default.htm` (*Linux*) or `gms /c x:\myfolder\newfile.htm default\default.htm` (*Dos, Windows*) to *create* a markup document that is based on `[GMS_ROOT]/doc/default/default`.

You can shorten the template name from `default\default.htm` to `default`, but if you say `default.htm`, an existing file with that name in the current directory, instead of `[GMS_ROOT]/doc/default`, might be used as a template.

The given example is essentially the same as the command `copy %GMS_ROOT%\doc\default\default.htm x:\myfolder\newfile.htm` (*Dos or Windows*), respectively `cp $GMS_ROOT/doc/default/default.htm /myfolder/newfile.htm` (*Linux*), if the `[GMS_ROOT]` environment variable is set to the GMS installation folder.

Open

Web Browser

Click on **[O]** to *open* an existing HTML file for conversion into PDF.

(1) If GMS runs on a remote network machine, it displays an input form where you can enter the file name. You may also **[Browse]** through your computer's local directory tree to search this file. Clicking on the **[Accept]** button starts an upload of this file into a temporary folder. You have to re-download this file to save it whenever it was changed.

If your HTML document includes other files like images, style sheets or markup files which are chained to the first, you have to upload them too. Only the following data file formats are supported by Markup Shredder: CSS, HTML, JPG, PDF and PNG. All these files will be stored in the same directory, so you have to change links to other levels.

(2) If GMS runs on your local computer, then there is no need for file up- and downloading. In this case GMS displays a selection box to **[Browse]** through your local directory tree. Click on the **[Accept]** button to change the working directory and to select a HTML file therein. On *Linux*, this works only in subdirectories of `$Document_Root`, so you may have to link your home folder to it.

Text Mode

If you press **[O]**, a simple file selection dialog is displayed to *open* an existing markup document. Pressing **[N]** or **[P]** will list the *next* or *previous* files in the current folder, **[H]** takes you to a *higher* directory level, and **[Q]** lets you *quit* this dialog without opening a file.

Numbers **[1]** to **[9]** will either *select a file*, or *change into a sub-folder* if the corresponding item is marked with two slashes, **[\]** (*Dos, Windows*) or **[/]** (*Linux*), symbolizing the level separator. You can also use the **[up]** and **[down]** arrow keys and **[space]** or **[enter]** to select an entry. GMS then works in the directory that you have chosen. On *Dos*, this dialog can only display the first 127 files of a directory.

The recommended way to open a markup file with the GMS text mode interface under *Windows 32* is clicking on its name with the right mouse button in Explorer. Select *Send to gerolf.bat* in the context menu.

On *Linux* too, you can configure your KDE or Gnome desktop to *open* HTML documents *with* GMS, using the `/usr/bin/gerolf` or `$GMS_ROOT/gerolf` links respectively the `$GMS_ROOT/htdocs/texmf/etc/gerolf` launcher script to *run* GMS *in a terminal* window. In Konqueror, click on a markup file with the right mouse to open the context menu.

Command Line

To *open* a markup document, enter `gms -o /myfolder/myfile.htm` (*Linux*) respectively `gms /o x:\myfolder\myfile.htm` (*Dos, Windows*).

View

Web Browser

If you click on **[V]** to *view* the HTML file that you have *opened* or *created*, additional markup will be inserted into the source file to perform syntax highlighting. Then you can find false tags or missing delimiters more easily because some of these mistakes are indicated by irregular color changes.

Errors detected during file *analysis* or *typesetting* are often mentioned together with the line number, which is therefore added to the left margin.

You can save the file by clicking with the right mouse button on the file name that is displayed on top of the frame content. Select *Save target as* or *Open in a new window*, where you can choose *File/Save* in the main menu.

Text Mode

If you press **[V]** in the *text mode* interface, you get a simple plain text *view* of the source file. GMS uses `browse.com` (*Dos, Windows*) and `less` (*Linux*) as internal viewers, showing the text inside the same window.

To quit the viewer, press **[Esc]** (*Dos, Windows*) or **[Q]** (*Linux*). To browse through the text, use the *arrow keys* and **[Pg Up]**, **[Pg Dn]**, **[Home]** or **[End]** keys.

This may not work on some *Linux* systems unless *terminal settings* are changed. In KDE Konsole, for example, changing *Preferences/Keyboard* to *Linux Console* may help. If not, use letter keys to move **[F]**oreward, **[B]**ackward, **[U]**p or **[D]**own; **[G]** and **[Shift-G]** let you go to the beginning or end of the text.

You can *select* another viewer if you press **[S]** and **[P]** enter enter the name (without path). The viewer binary must be located in the search path or in a sub-directory of `[GMS_ROOT]/bin`.

Command Line

In the *command line*, enter `gms -v /myfolder/myfile.htm` (*Linux*) or `gms /v x:\myfolder\myfile.htm` (*Dos, Windows*) to *view* a markup document. If the file was *opened* or *created* before, it is sufficient to call `gms -v` or `gms /v`.

Alternatively, you can execute the command `less /myfolder/myfile.htm` (*Linux*), `%GMS_ROOT%\bin\dos\browse x:\myfolder\myfile.htm` (*Dos*) or `%GMS_ROOT%\bin\win\browse x:\myfolder\myfile.htm` (*Windows*), where the `[GMS_ROOT]` environment variable must be set to the GMS installation folder.

Edit

Web Browser

If you click on **[E]**, you can *edit* the HTML file that you have *opened* or *created*.

- Change the *font size*, given as a number, and grow or shrink the *text area* by changing its *height* and *width*, given as lines and characters per line. Click on **[Accept]** to redraw the page. The text area right should fit into the frame, without overlapping scroll bars. The font size change applies to the view, *analyse* and *typeset* functions as well. You can select a smaller font in a *split* frame.
- Delete a few characters within the text area and click on **[Reset]**. Your browser will undo this change immediately and restore the original state, but it cannot remember multiple steps of editing.
- Add a few letters to the text and click on **[Accept]**. The file will now be uploaded to the web server again, and your browser has to redraw the frame content. If the file change that you just have made is lost now, then the browser has loaded an old copy of the file from its hard disk cache instead of an updated version delivered by the server. You should change the browser settings then.

You can save the file by clicking with the right mouse button on the file name which is displayed on top of the frame content. Select *Save target as* or *Open in a new window*, where you can choose *File/Save* in the main menu.

Your browser's word search function works in the text area like on any other web page, but line numbers are missing, and there is no syntax highlighting. So you will prefer to use an external editor.

If you click into the text area with the left mouse button and then open the context menu with the right one, you can select *Mark all* and *Copy*. Then open another editor like Notepad, FrontPage or *HTML-Kit (Windows)* respectively *Kwrite*, *Nedit* or *(X)Emacs (Linux)*, where you can *paste* the content from the clipboard. Similarly, you can re-transmit any modified text into the GMS text area in the browser after you have deleted its old content by pressing **[Ctrl+A]** to *mark all* and then **[del]**.

Text Mode

If you press **[E]**, an external program to *edit* the source file is executed. By default, it is *edit.com (Dos)* or *notepad.exe (Windows)*. On *Linux*, it is *vi*, if none of the following is found in the `/usr/bin` directory: *gedit*, *kedit*, *emacs*, *xemacs*, *quanta*, *kwrite*, or *nedit*.

You can *select* another *editor* if you press **[S]** and **[P]**. The editor binary, or a link to it, should be found in the search path or in `[GMS_BINARIES]`, a sub-directory of `[GMS_ROOT]/htdocs/texmf/bin`.

Command Line

In the *command line*, say `gms -e /myfolder/myfile.htm (Linux)` or `gms /e x:\myfolder\myfile.htm (Dos, Windows)` to *edit* a markup document. If the file was *opened* or *created* before, it is sufficient to call `gms -e` or `gms /e`.

Alternatively, you can execute the command `kedit /myfolder/myfile.htm (Linux)`, `edit x:\myfolder\myfile.htm (Dos)` or `notepad x:\myfolder\myfile.htm (Windows)`.

Browse

Web Browser

If you click on [\[B\]](#), you can *browse* through the HTML file that you have *opened* or *created*.

Your browser does not simply display all the letters in this file like the text *viewer* does: The characters are parsed into content and markup, and the markup is used as an instruction to render the content. So what you see is an interpretation of the file, not the file itself. The rendering changes if you resize the browser window, and another browser gives a slightly different rendering of the same file.



HTML, developed by Tim Berners-Lee around the year 1990 to retrieve information from various sources all over the world, was not intended for *typesetting*, but can be used for this task too, since style sheets were introduced which allow to specify the precise appearance of a document with respect to the media. Common web browsers still do give such a nice print as Word, failing in hyphenation, for instance.

There are techniques to produce both a PDF and an HTML version of a document for print and online publishing other than the GMS approach of converting HTML to PDF. For example, you can start with: 1. a Word or OpenOffice document (save it as a web page and print it to a file using a PDF driver), 2. a LaTeX file (convert it to PDF using the pdfTeX typesetting engine, like GMS, and to HTML with LaTeX2HTML or TeX4ht), 3. an XML file (convert it to PDF using the xmlTeX or passiveTeX, and the LaTeX or ConTeXt macro packages, and the pdfTeX typesetting engine, like GMS; if a style sheet is given, modern web browsers can also render that XML file).

If an HTML file is produced by conversion, it may be very large, but not optimized for screen display. If you start to do this manually, you have to update more than one file whenever you change the document content. Markup Shredder uses a single source together with different style sheets for screen and print. So you can use a *what-you-see-is-what-you-get* editor to create a multi-purpose document in a human-readable file format which is designed for data exchange with many other applications, and if it turns out that GMS cannot properly render your HTML file, you still can open it with Word or OpenOffice to get a nicer print.

Text Mode

If you press [\[B\]](#), an external program is executed to *browse* through the markup document. By default, it is `netscape.exe` (*Dos, Windows 16*) or `iexplore.exe` (*Windows 32*). On *Linux*, it is `lynx`, if none of the following is found in the `/usr/bin` directory: `links`, `opera`, `konqueror`, `netscape`, or `mozilla`.

You can *select* another *browser* if you press [\[S\]](#) and [\[P\]](#). The browser binary should be found in the search path or in a sub-directory of `[GMS_ROOT]/bin`.

Command Line

In the *command line*, enter `gms -b /myfolder/myfile.htm` (*Linux*) or `gms /b x:\myfolder\myfile.htm` (*Dos, Windows*) to *browse* through a markup document. If the file was *opened* or *created* before, it is sufficient to call `gms -b` or `gms /b`.

Alternatively, you can execute the command `mozilla /myfolder/myfile.htm` (*Linux*), `win ncscape x:\myfolder\myfile.htm` respectively `arachne x:\myfolder\myfile.htm` (*Dos*) or `iexplore x:\myfolder\myfile.htm` (*Windows*).

Analyse

Web Browser

If you click on **[A]**, the markup in the HTML file that you have *opened* or *created* will be *analysed* using HTML-Tidy, a software written by Dave Raggett and a group of volunteers.



HTML-Tidy performs a syntax check of your file's markup. It returns a list of line numbers whenever violations of the language rules, as defined in the HTML specification by the World Wide Web Consortium, have been detected. You should correct these errors in the *editor* because the GMS *typesetter* then will have fewer problems to render a nice portable document.

Especially if you have produced your HTML file with a *what-you-see-is-what-you-get* editor like FrontPage, Fusion or DreamWeaver, the markup may be incomplete, verbose, proprietary or deprecated. If the file was converted from a Word document, it will be inflated with masses of information that is only relevant for Microsoft Office products.

If there are no severe markup errors in the input file, HTML-Tidy also produces a cleaned up version, where less important mistakes are corrected automatically and the layout is made uniform. That file is given the same base name with an OK extension.

Text Mode

You can *analyse* the markup of a given HTML file by pressing **[A]**. The result of the syntax check will be displayed in the text *viewer*. Press **[Q]** (*Linux*) or **[Esc]** (*Dos, Windows*) to *quit*.

If there are multiple HTML files in one folder, you can *analyse* them all at once, if no special file is selected. Press **[O]** to display the *open file* dialog and select an empty line, or change into the *higher* directory, return to the folder that you just have left and *quit* the dialog before pressing **[A]** in the GMS main menu.

The precise behaviour of HTML-Tidy is defined in the `[GMS_ROOT]/etc/tidy.cfg` configuration file. For instance, if there is a line `write-back: yes`, HTML-Tidy will replace the input file with its corrected version instead of writing it to an OK file. This may result in data losses, e.g. if parts of a text are marked as an attribute by mistake.

You can *select* another syntax checker, if you press **[S]** and **[P]**. The Tidy binary should be found in the search path or in a sub-directory of `[GMS_ROOT]/bin`.

Command Line

In the *command line*, enter `gms -a /myfolder/myfile.htm` (*Linux*) or `gms /a x:\myfolder\myfile.htm` (*Dos, Windows*) to *analyse* the markup of a document. If the file was *opened* or *created* before, it is sufficient to call `gms -a` or `gms /a`.

Alternatively, you can execute the commands `cd [GMS_ROOT]/doc/default` and `[GMS_ROOT]/bin/[dos|win|linux]/tidy -config [GMS_ROOT]/etc/tidy.cfg -f default.err default.htm > default.ok`.

Typeset

Web Browser



If you click on [\[T\]](#), you can *typeset* the HTML document that you have *opened* or *created*, converting it into high-quality *Portable Document Format* with the help of the TeX engine, a software developed by Professor Donald Ervin Knuth and his students at Stanford University around the year 1980. The kernel of TeX version 3.141592x is frozen and considered to be error-free, in contrast to most other software of that size, but the system still is flexible enough to be extended.

Between 1996 and 1998 support for right-to-left typesetting was implemented by Peter Breitenlohner, and Hàn Thế Thành integrated a post-processor which creates PDF output and embeds PostScript and TrueType fonts as well as JPG and PNG images. The additional code of both extensions is contained in the `pdfetex` binary.



Markup Shredder – which can use this wonderful piece of well-tested open source software – only adds 300 kilobytes of TeX macros and twice that amount of shell, batch and PHP scripts in order to realize an HTML parser and a font metric processor together with three simple user interfaces.

Traditionally, writers had to learn hundreds of proprietary commands to run the TeX typesetting engine. Markup Shredder wants to be *TeX made easy*, as basic HTML and CSS knowledge should do now.

Like HTML-Tidy, which is described in the *analyse* section, GMS performs a syntax check of the input document's markup and creates a protocol file that is given the same base name with a LOG extension. Each tag that has been processed is listed herein, and error messages are inserted wherever the GMS macro layer or the TeX engine itself detect any irregularity.

HTML-Tidy and GMS do not always agree in their syntax check results. For instance, Markup Shredder might mention missing end tags or missing quotation marks around attribute values where HTML-Tidy gives a strict HTML rating without warnings, because GMS tries to help authors to write valid XHTML documents. Another reason is that Markup Shredder does not test whether lower case is used for all element and attribute names in documents that claim to be of type XHTML. GMS also tolerates characters `€` to `Ÿ`, which are not allowed in the specifications, though important things like dashes, Euro sign, French œ ligatures and German „gaensefuesschens“ could be placed there by Microsoft.

It is instructive to compare both test results with the HTML and XHTML specifications.

At the end of the log, you find a link to the PDF output file. If GMS runs on a remote network computer, you can download and save it, if you click on the link with the right mouse button. In the context menu select *save target as*.

Text Mode

You can *typeset* a given HTML file by pressing [\[T\]](#). The result of the syntax check will be displayed in the text viewer; press [\[Q\]](#) (*Linux*) or [\[Esc\]](#) (*Dos, Windows*) to quit.

You can *select* another TeX binary (`tex`, `etex`, `pdfetex`, or `pdfetex`), if you press [\[S\]](#) and [\[P\]](#). The TeX binary and its associated message pool file should be found in the search path or in `[GMS_BINARIES]`, a sub-directory of `[GMS_ROOT]/bin`.

Command Line

Say `gms -t /myfolder/myfile.htm` (*Linux*) or `gms /t x:\myfolder\myfile.htm` (*Dos, Windows*) to *typeset* a markup document. If the file was *opened* or *created* before, it is sufficient to call `gms -t` or `gms /t`.

Alternatively, you can execute the command `[GMS_BINARIES]/pdfetex -progname=gerolf [GMS_ROOT]/doc/default/default.htm`, if the `[TEXINPUTS]` variable is set to `[GMS_ROOT]/etc`.

Errors

Thanks to the operating system's disk buffer, the second typesetting run of an HTML input will be a bit faster than the first. In the *text mode* interface, the process may be accelerated on *Windows NT/XP*, if the window is hidden. GMS will prompt you in the task bar whether it is still busy, or how long the run has taken. In the *web browser* interface, as well as in the other interfaces on *Windows 9x*, you do not get any feedback about the typesetting progress before it is finished, though it might take some time.

Sometimes you may discover an error in your input file while TeX is still working. In the *text mode* interface or in the *command line*, you can *cancel* the process by pressing `[Ctrl+C]`. If a question comes what to do next, answer `[X][Enter]` to *exit* at this point or `[Ctrl+C]` again to *cancel* output production. On *Dos and Windows*, you are also asked: *Cancel batch process (Y/N)?* Answer `[N][Enter]` to return to the GMS menu.

If TeX should break with an *exhausted memory* message on a large document, change the value that is assigned to the `main_memory` variable in `texmf.cnf` and *initialize* the TeX format file.

Images

The pdfTeX engine can embed images, but only those of type JPG (for photographs), PNG (for graphics with a reduced number of colors), or PDF (e.g. single pages created with pdfTeX and LaTeX). The popular GIF image file format is not supported, but you do not have to modify your HTML document: Just provide a JPG or PNG file with the same base name for every inserted GIF image and place it into the same directory.

So you can have low-resolution GIF images which are displayed by browsers, while GMS, when processing GIF requests, will look for matching JPG or PNG files that may have a higher resolution. If such an image file cannot be found, however, the replacement function may return a different file with the same name from another directory within the document search path.

You may change your document to use only PNG images, avoiding to ship them in two different data formats, but Internet Explorer 3x and early Netscape Navigator 4x do not render PNG images. Using a JPG replacement for a GIF image will usually lead to a larger file size or loss of quality. GMS treats $1\text{in} = 25.4\text{mm} = 72\text{pt} = 72\text{px}$, or $1\text{px} = 1\text{pt}$, just for ease of page design, though this is not recommended by the CSS2 specification.

Styles

Here's the main trick to fine-tune an HTML document for print via GMS and Acrobat Reader without changing its appearance in a browser on screen:

- Create two different style sheets `screen.css` and `print.css` and link them to the document's `<head>` element, saying: `<link rel = "stylesheet" type = "text/css" href`

```
= "screen.css" media = "screen" /><link rel = "stylesheet" type = "text/css" href = "print.css" media = "print" />
```

- To dimension a *table cell* for GMS, for instance, add a class attribute to its definition tag, `<td class = "td1">`, and an entry like `.td1 {width: 8cm}` to `print.css` (then you can still rely on your browser's auto-width function for screen rendering), or add `.td1 {width: 50%}` to `screen.css`.
- As a consequence, it is even possible to render different content on screen and in print. You need the definitions `.noscreen {display: none}` in `screen.css` and `.noprnt {display: none}` in `print.css`. Now, if you start your document body with `Good morningevening`, the rendering reads: *Good evening!* –The media attribute applies to the `<style>` element as well, so you do not need external CSS files.
- The problem is that old browsers that do not support style sheets, as well as Microsoft Word, will display both the content for screen and print. Internet Explorer 3x only interprets the last style sheet in the document head, ignoring the media value and thus forcing authors to optimize their pages for Microsoft products. So it may be necessary to load an empty dummy file as last style sheet.

Hyphens

You can select the language of your document and the corresponding hyphenation rules by saying `<html lang = "en-UK">`, for example. The codes for the representation of names of languages are defined in ISO 639.2. If you discover wrong or missing hyphenation in the output PDF produced by GMS, use soft hyphens, like `man­u­script` or `ap­pen­dix`. Old browsers like Internet Explorer 3x and Netscape Navigator 4x, however, display `­` as a dash, being always visible.

Between 1996 and 2006, reformed spelling and hyphenation rules for German were established. In GMS, you can enable *reformed* German hyphenation rules by saying `<html lang = "de-rf">` in your document. For *traditional* texts, the `<html lang = "de">` declaration, however, is insufficient because there are extra rules to modify the spelling of some words when they are split between lines. Thus you have to write `Bäcker` and `Bettuch` to tell GMS to take a *sonderweg* leading to *Bäcker/Bäk-ker* and *Bettuch/Bett-tuch*.

In any language, if a word appears to be split in a way that you do not like, you can restrict hyphenation to the desired places by inserting soft hyphens `­`. A word will *not* be split if it is enclosed by an inline level element like ``, unless a trailing space or punctuation mark is enclosed too.

Ancestors

Since version 0.06a, Gerolf Markup Shredder supports genealogical data markup according to the Gedcom XML 6.0 Specification. While you can open the example file `gedcom60.xml` directly with the *text mode* interface, the *web browser* interface will refuse to do so. Therefore a copy of this file named `gedcom60.htm` still is required.

For other languages than English, you have to modify the generated content which is defined at the end of `gedcom60.css`, e.g. `'Gender:before {content: "Geschlecht: "}'` for German. Internet Explorer does not generate this content, so use Mozilla Firefox or Opera for browsing. The `gedcom60.pdf` output file produced by GMS includes images as generated content for `<URI>` elements, if JPG or PNG files can be found locally.

Read

Web Browser

If you click on **[R]**, you can *read*, *print* and *save* the PDF document that you have *typeset*.

Acrobat Reader, if installed after the browser, will be run therein as a plug-in, as if the browser could display portable documents by itself, like it can render markup files. In plug-in mode, the Acrobat Reader menu is not visible. Its functions are accessible using the icon bar, but shortcuts work as well. For example, you can press **[Ctrl+D]** to see the document information or **[Ctrl+Alt+F]** for the font list.

If you do not get access to the PDF file here, click with the right mouse button on the link at the end of the *typesetting* log. In the context menu, select *save target as* to download and *save* the PDF, if GMS is running on a remote network computer.

If your browser does not start the reader plug-in though the preferences are set correctly, you may have to copy `nppdf32.dll` from the `%ProgramFiles%\Adobe\Acrobat X.Y\reader\browser` folder (*Windows 32*) to your browser's `plugins` sub-directory.

Acrobat Reader 3x for *Windows 16* tends to crash when a PDF file containing images is magnified. It also cannot display embedded TrueType fonts, which are used for Unicode characters by GMS, and it fails completely to display the PDF output from some of the GMS template files that can be displayed without problems by Acrobat Reader 4x or later on their operating systems.

Acrobat Reader 4x for *Linux* sometimes mixes up red and blue color when changing pages. Resizing the document view may help. In print, this version defaults to fit the PDF document to US letter paper.

Text Mode

If you press **[R]**, the PDF renderer is executed as an external program to let you *read* your document. On *Linux*, there are other applications besides Acrobat Reader that can do this, e.g. `xpdf`, `Kpdf`, `ghostview`, `Kghostview`, `gv`, and `Kgv`.

You can *select* another *reader* if you press **[S]** and **[P]**. The reader binary should be found in the *search path* or in a sub-directory of `[GMS_ROOT]/bin`.

Command Line

Enter `gms -r /myfolder/myfile.htm` (*Linux*) or `gms /r x:\myfolder\myfile.htm` (*Dos, Windows*) to *read* the portable document. If the file was *opened* or *created* before, it is sufficient to call `gms -r` or `gms /r`.

Alternatively, execute the command `acroread /myfolder/myfile.htm` (*Linux*), `winacord16 x:\myfolder\myfile.htm` (*Dos, Windows 16*) or `%ProgramFiles%\Adobe\Acrobat X.Y\reader\acord32 x:\myfolder\myfile.htm` (*Windows 32*).

In the *command line* and *text mode* interfaces, you have to quit Acrobat Reader by pressing **[Ctrl+Q]** respectively **[Ctrl+W]**, before you can re-*typeset* your HTML source file; otherwise the reader may not update its display.

Split

Web Browser

You can *split* the workspace into two frames by clicking on the **[S]** button; then you are allowed to perform two actions independently.

For instance, you may *learn* how to use Markup Shredder by loading this handbook into the right frame, while you can still *view*, *browse*, *edit*, *analyse*, *typeset* or *read* your document at the left-hand side. Similarly, you may study the *analysis* or *typesetting* log on one side and *view* or *edit* your markup file on the other.

Press **[J]** to *join* both frames again.

Select

Text Mode

You can *select* a few of Markup Shredders settings if you press **[S]** and one of the keys mentioned in the following sub-sections.

To edit any item, press **[Enter]** to jump into the input line, and the **[Up]** or **[Down]** arrow key to leave it. Typing **[Backspace]** removes undesired characters. Press **[Q]** to *quit* the *selection* dialog.

All changes that you have done here are assigned to variables like `[GMS_CODE-PAGE]` in `$GMS_ROOT/etc/gerolf` (*Linux*) respectively `%GMS_ROOT%\etc\gerolf.bat` (*Dos, Windows*), the launcher and configuration script.

Animation

[A]: The menu *animation* can be switched on or (nearly) off. GMS makes use of the `reply` binary, which draws a menu box line after line. GMS provides a fast drawing mode and another one that looks a bit nicer. The effect is nearly invisible on modern computers, but on a slow *Dos* machine, menu building may last too long; so the command line interface should be used instead.

Colors

[C]: You can set the interface *colors* as you like. GMS will open a color pane and another input box where you can enter numbers between 0 (black) and 15 (white) for the foreground colors (text, hotkey, and pattern). The numbers for background colors (banner, shadow, and desktop) must be in range 0 to 7 (gray). The desktop pattern letter is defined by its index number in the US-American code page (ASCII); it must be in range 32 (blank) to 126 (tilde).

By default, all these values are set to `R`, meaning that they are assigned a *random* number after every viewer call or execution of external programs (chameleon mode). If you want to set the interface to fixed colors, try the following numbers: 15, 15, 15, 2, 4, 1, 47, or 0, 12, 15, 7, 0, 1, 92.

Programs

[P]: GMS opens another input box where you can enter the name of the *program* binaries to be associated with the main GMS menu functions (*view*, *edit*, *browse*, *analyse*, *typeset*, and *read*). These executables – or startup shell scripts, batch files, links (`*.sh`, `*.bat`, `*.lnk`) – should be found in the search path or in `[GMS_BINARIES]`, a sub-directory of `[GMS_ROOT]/bin`. Do not enter paths, spaces and parameters into the input line, only the file name is allowed.

For example, if you want to use HTML-Kit (*Windows NT/XP*) as *editor* to be called from GMS, open Explorer, change into the `%ProgramFiles%\Chami\HTML-Kit\bin` directory and click on `HTML-Kit.exe` with the right mouse button. In the context menu select *create link*. Move this link to `%GMS_ROOT%\bin\win` and rename it to `HTML-Kit`; so the corresponding link file is named `HTML-Kit.lnk`. Enter `HTML-Kit.lnk` into the *editor* input line in the *program selection* dialog. Do not add a search path or a parameter here.

On *Windows 9x*, LNK files are not executable; so in `%GMS_ROOT%\bin\win`, create a `HTML-Kit.bat` batch file instead, consisting of just the line `"%ProgramFiles%\Chami\HTML-Kit\bin\HTML-Kit.exe" %1 %2 %3 %4 %5`.

On *Dos*, omit the quotation marks when calling a program from a batch file. To access your work file, you can also make use of the variables %GMS_REMODRV% (drive letter), %GMS_FOLDER%, %GMS_FILE%, %GMS_SHORT% (8.3-format) and %GMS_BASE% (no extension). – Similarly, on *Linux*, use \$GMS_FOLDER, \$GMS_FILE and \$GMS_BASE.s

Debug

[D]: If you want to engage in *debugging* Markup Shredder, you can set this item to other values than 0 (*no debugging*): Z creates a [GMS_ROOT]/etc/gmsdebug.log file, listing all *internal calls* to the *command line* and *text mode* interface modules, together with parameters and values of important environment variables (not on *Dos* and *Windows 9x*). X traces these module calls on additional banners inside the *text mode* interface. Y does the same in slow motion.

Encoding

[E]: Here you can enter the *main 8-bit encoding*, which fits for most of the fonts used with Markup Shredder, usually one of a Latin alphabet. Encodings that cover only a small number of fonts should be defined in the GMS configuration files font.cfg, encoding.cfg and alias.cfg in [GMS_ROOT]/etc.

Every letter in a HTML file is stored as a number indicating its index in a table of characters. Since every code page can only hold $2^8 = 256$ characters, there are systems of related code pages covering a wider range: First, the *iso-8859-x* or *i8859-x* series by ISO. Second, the *windows-125x* or *cp125x* series by Microsoft, which makes a more efficient use of the table space. Third, the Unicode system that defines hundreds of consecutive code pages to enumerate every character of every language on earth. But there are also code pages that have been developed independently to fit a particular language or script, for example ISCII (Indian) or VISCII (Vietnamese).

Markup Shredder will employ a matching code page designation, if you simply enter one of the following keywords: Arabic, Baltic, Central, Cyrillic, Greek, Hebrew, Indian, Latin, Thai, Turkish, Vietnamese, or Western.

You cannot enter the word Unicode (or UTF-8) here. Markup Shredder works with 8-bit input files, and you have to decide for one single code page, not a whole series. However, GMS gives you access to fonts with a different encoding, if Unicode characters are escaped according to the HTML specification, writing the mathematical infinity symbol ∞ as `∞` or `∞`, for example. You can change a file's encoding with your browser's menu function *File/Save as*.

Whenever you have changed the standard code page here, you still must re-compute the TeX font metrics (in order to *write* the font map) and re-*initialize* the TeX format file for GMS.

Codepage

In the *text mode* interface, Markup Shredder can only show half of the characters from the current code page.

- [U]: GMS displays the *upper half* of the code page which is used for the terminal font in the *text mode* interface. On *Windows NT/XP*, you can change this font if you click on the title bar of the terminal window with the right mouse button. Select *Preferences* in the context menu and change the font to Lucida Console with bold characters. If the GMS main encoding is set to one of the *windows-125x* or *cp125x* series, Markup Shredder can change the terminal font code page accordingly.

- *[L]*: GMS displays the *lower half* of the code page which is used for the terminal font in the *text mode* interface. Usually it is almost identical with the American Standard Code for Information Interchange (ASCII), a 7 bit code page defined around the year 1960 that only covers the characters of an American typewriter. Table positions 0 to 31 and 127 are reserved for functions like bell, carriage return, line feed or paper feed. If characters are displayed on these positions, then they are system-dependent. Most code pages – but not the Unicode pages beyond the first – are based on ASCII, using the upper half for non-English characters.

Command Line

Enter `gms -s` to open the launcher and configuration script, and `gms -e` to *edit* it, directly *selecting* GMS properties by assigning new values to the GMS variables therein.

Write

Web Browser, Text Mode and Command Line

If you click on *[W]* in the *web browser* or press *[W]* in the *text mode* interface or say `gms -w` respectively `gms /w` in the *command line*, then GMS will *write* the *font map*, a list of available fonts which is evaluated by GMS during *initialisation* of the format file, and later is used again by the TeX engine for *typesetting*. This font map, however, is only a by-product of a metrics computation process that may last several minutes, depending on computer speed and the number of fonts.

Glyphs

TeX comes from a time before PostScript (PFA/PFB) and TrueType (TTF) fonts were introduced. The TeX approach of typesetting is to place empty boxes accurately on the page, one box for every letter, and to leave it to the screen or printer driver to add the character shape information. So the TeX engine's *device-independent* output file (DVI) could only be used on systems that already had installed the same fonts.

The pdfTeX extension written by Hàn Thế Thành enables TeX to embed all glyph data into a portable document, but the old TeX 3x kernel does not want to know what the glyphs look like; it still needs TeX font metric files (TFM) containing data about the letters' heights, depths, and widths. Moreover, if an encoding (ENC) other than TeX's proprietary 7-bit code page is used, a *virtual font* file (VF) must be provided.

The font map keeps together the information about which files are related to any font. A typical entry looks like this: `_putr Utopia-Regular " CP1252-Encoding ReEncodeFont "<CP1252.enc <putr.pfb`, denoting 1. the base name of the font files (without leading underscore), 2. the PostScript font name mentioned in the font's AFM file, 3. the PostScript encoding name (quoted), 4. the encoding file, and 5. the glyph file.

While encoding files can be found in `[GMS_ROOT]/data/enc`, font files are located in sub-directories of `[GMS_ROOT]/fonts`. To minimize search time, AFM files (Adobe or PostScript font metrics), PFA/PFB (PostScript Type 1 glyph files), TFM, TTF, and VF files are kept in separate trees. Within every tree, typesetting systems following the TeX directory specification create one level of folders named after the type foundries and a second level named after the font families. There is, however, no technical need for this pedantry.

As reported in `fontmap.log`, the GMS metric processor creates TFM and VF files from AFM, and AFM from TTF if necessary, running `ttf2afm`, `afm2tfm` and `vptovf` converters. In simple cases, if the GMS main encoding can be used that you have selected and if the PostScript font name must not be corrected, GMS will do this automatically; otherwise you have to modify three configuration files in `[GMS_ROOT]/etc`: `font.cfg`, `encoding.cfg` and `alias.cfg`.

Fonts

Now let us have a closer look on how GMS was given access to certain fonts. Then you will be able to use your favorite fonts with Markup Shredder too:

- **Bitstream Charter.** Created folders `[GMS_FONTS]/afm/bt/charter` and `[GMS_FONTS]/type1/bt/charter`. Downloaded four AFM and four PFB files from CTAN into the corresponding directories. Shortened the base name of the font files from `bchr8a` to `bchr` etc. Opened `bchr.afm` to find out the PostScript font name: `CharterBT`. Added

`\addFONTalias Charter (CharterBT) to alias.cfg`; so you can simply use the name Charter as HTML font-family name in markup documents.

- **IBM Courier:** As above, except that only two AFM and two PFA files are needed, for regular and bold weight, because the slanted variants can be computed by pdfTeX. Since Courier is a wide, non-proportional font, a narrower variant would be useful for technical documents. To get this, four lines like `ncrr %GMS_CODEPAGE% embed 0.75 0.25 nccrco` had to be added to `font.cfg`, defining four typefaces which were grouped to a font-family named CourierNarrow by saying `\addFONTfamily CourierNarrow (ncrrc, nccrco, ncrbc, ncrbco)` and `\addFONTalias CourierNarrow (IBMCourier)` in `alias.cfg`.
- **Monotype Arial:** This font is among those TTF files that GMS setup copies from `%windir%\Fonts` to `%GMS_ROOT%\fonts\ttf` to access standard TrueType fonts on [Windows 32](#). You can use this font as ArialMT in HTML documents, or as Arial, because `\addFONTalias Arial (ArialMT)` and `\addFONTfamstd Arial (arial)` were added to `alias.cfg`. The `\addFONTfamstd` command corrects irregularities in the PostScript font naming, as you can see in `font.map`: The italic font face is named Arial-ItalicMT instead of ArialMT-Italic, for example.

By default, metrics are processed only for the GMS main encoding that you have selected; GMS setup proposes CP1252 (West Europe, Africa, America, Australia). The Monotype Arial font files, however, contain data for several hundred characters, covering the *windows-125x* or (CP125x) code page series.

To access Cyrillic letters from Unicode page U+0400, the line `arial G0400 embedfamily 1.0 0.0 none 04` was added to `font.cfg`, and `\addFONTfamuni Arial (arial 04)` to `alias.cfg`. Alternatively, to support the Cyrillic code page *windows-1251*, the line `arial CP1251 embedfamily 1.0 0.0 none W1` was added to `font.cfg`, and `\addFONTfamily ArialCyrillic (arialW1, arialiW1, arialbdW1, arialbiW1)` to `alias.cfg`. Here you have to use the font-family name ArialCyrillic in HTML documents. In the `<head>` element, write `<meta http-equiv = "content-type" content = "text/html; charset=windows-1251" />`. You can, however, leave the GMS main encoding set to CP1252.

- **Monotype Andalus:** This is one of the fonts that come with [Windows XP](#) but may be uninstalled. In such a case, insert the system CD and say: `expand X:\i386\andlso.ttf %windir%\Fonts\andlso.ttf` in the *command line*, where you have to replace X with your CD-ROM drive letter. GMS comes with a script to extract all XP fonts, just enter `%GMS_ROOT%\batch\xpfonts X:\i386 %windir%\Fonts`.

This Arabic font, which only contains the ASCII subset of Latin letters, should not be processed with code page CP1252 (West Europe), if this is the GMS main encoding. Therefore, the line `andlso: ArabicMT` was added to `encoding.cfg`.

The encoding file ArabicMT.enc is the same as `cp1256.enc` (Arabic), except that it uses Monotype glyph names which differ from the GMS glyph list, which is a compromise on the Adobe and Windows glyph lists.

- **Shusha:** This font, downloaded from GeoCities to `[GMS_ROOT]/fonts/ttf/pub/shusha`, uses glyph names from the Latin code page `cp1252.enc`, but replaces them with the corresponding Hindi or Marathi letters. No changes had to be done to the configuration files to access this font with GMS, but you still may have to copy `shusha.ttf` to `%windir%\fonts`, if you want to use it on [Windows](#) with your browser.
- **Courier, Helvetica, Times, Symbol, ZapfDingbats:** As you can see in `font.map`, these core fonts must not be embedded in the PDF document, because the corresponding glyph files are present on every installation of Acrobat Reader. With core fonts, you can produce very small PDF files for the internet, if you are mainly using letters from code

page CP1252 (West Europe). Symbol and ZapfDingbats need font-specific encoding (`psy.enc` and `pzd.enc`).

- *Computer Modern*: The traditional TeX 7-bit fonts designed by Donald Ervin Knuth are excluded from the GMS metric processing. GMS comes with the required TFM files and a separate map file named `cm.map`.

After re-computing of font metrics and re-*writing* of the `font.map`, you still have to re-*initialize* the TeX format file for Markup Shredder.

If a TeX font metric file (TFM) is corrupted, pdfTeX may give you the following error message: `! Font =_GENR at 12pt not loadable: Bad metric (TFM) file ... line 666. I wasn't able to read the size data for this font, so I will ignore the font specification.` This error should disappear if font metrics are re-processed; otherwise, you have to remove the corresponding font face from your document and `font.map`.

Init

Web Browser, Text Mode and Command Line

If you click on [\[!\]](#) in the *web browser* or press [\[!\]](#) in the *text mode* interface or say `gms -i` respectively `gms /i` in the *command line*, then the TeX engine will *initialize* a so-called format file for Markup Shredder, which is given the base name `gerolf` and the extension `efmt` (*Linux, Windows 32*) or `efm` (*Dos*) in case of the `pdfetex` engine, and `fmt` for `pdftex`. In binary format, this file contains information taken from the following program texts or data structures, as reported in `gerolf.log`:



- `prologue.cfg`: Commands to be executed before initializing Markup Shredder.
- `plain.tex`: The well-tested plain TeX macros (44 kB) by Donald Ervin Knuth, as described in appendix B of *The TeXbook* (1984).
- `gerolf.tex` and other TeX files in `[GMS_ROOT]/tex/gerolf`: The Markup Shredder TeX macros (about 300 kB) by G. D. Brettschneider.
- `typeset.cfg`: User-editable typesetting parameters, as described in *The TeXbook*, pages 272–275, 348–349 and 451.
- `alias.cfg`: User-defined font names, as described in the *write* font map chapter.
- `markup.tex`: Definition of supported HTML elements (or “tags”), attributes and CSS properties, as listed in `gerolf.log`, section 1d.
- `plugin.cfg`: List of installed data structures.
- `font.map`: List of fonts, as described in the *write* font map chapter; see `gerolf.log`, section 2a.
- `phv.krn` and other margin kerning files for character protruding of selected fonts in `[GMS_ROOT]/data/krn`; see `gerolf.log`, section 2b.
- `en-US.tex` (originally named `hyphen.tex`, by Donald Ervin Knuth) and other hyphenation pattern lists for various languages in `[GMS_ROOT]/tex/hyphen`; see `gerolf.log`, section 2c.

These pattern files, downloaded from CTAN and re-encoded according to code pages supported by GMS, are human-readable. Maybe you can contribute a new one for your native language – or check the existing file? Here’s a hint for understanding them: “A hyphen is considered to be acceptable between two letters if the associated interletter value is odd. A large odd value forces desirable hyphen points, a large even value suppresses undesirable hyphens, for example: `.hy3phe2n5at2io2n.`” (*The TeXbook*, page 450).

If you discover wrong hyphens in the GMS output PDF, you can define an exception list at the end of the pattern file for your language, like this one: `\hyphenation {man-uscript ap-pen-dix}`.

- `u0000.row` and other Unicode row definitions (see `gerolf.log`, section 2d), which are excerpts of 256 code points from the character space as specified in `UnicodeData.txt`. This database file is described in `UCD.html`.

If you want to use fonts with alphabets that are not supported initially by GMS, say Tibetan, then you have to extract the required `u0F00.row` file by yourself, comprising characters on Unicode points U+0F00 to U+0FFF. GMS then will produce `u0F00.enc`, the required encoding file to support a font with Unicode glyph names like `/uni0700` etc. It is up to you to provide such a font.

- `HTMLlat1.ent` and other definition files for named entities or character references in markup documents (like `ß` for German β), as defined in the HTML 4.01 specification; see `gerolf.log`, section 2e.
- `gms.gly` and other glyph lists associating Unicode points with PostScript glyph names. A lot of Unicode characters were given multiple glyph names while encoding files must decide for a single; so `gms.gly` is a compromise on the Adobe and Windows glyph lists. If encoding files with other glyph names like `arabicmt.enc` are needed, they must be created manually.

In `gerolf.log`, section 2f, you may read warnings like `!Undefined unicode row 3000`, indicating that the corresponding file `u3000.row` (Hiragana, Katakana) was moved to `[GMS_ROOT]/data/row/_disable`, because GMS does not provide a matching font.

- `cp1252.txt` (West Europe) and definition files for other code pages which are recognized by Markup Shredder (see `gerolf.log`, section 2g), if the `<head>` element of a markup file contains a tag like `<meta http-equiv = "content-type" content = "text/html; charset=windows-1252" />`.

These files, representing a selection of Unicode characters numbered arbitrarily from 0 to 255, are used by GMS to extract PostScript glyph names from `gms.gly` when writing encoding files like `cp1252.enc`.

- `epilogue.cfg`: Commands that shall be executed after initializing Markup Shredder, for example to generate encoding files.

You have to *re-initialize* the TeX format file for GMS whenever you made changes to the information mentioned above (for instance by adding hyphenation patterns for another language, or by adding new fonts and *re-writing* the `font.map`), or when you have downloaded a newer `pdfetex` binary for *Windows* or *Linux*, together with the corresponding message pool file, from Sebastian Rahtz's TeXlive distribution.

There is a mailing list where problems concerning the `pdfetex` binary can be discussed. Please do not expect members of this list to be familiar with Markup Shredder.

Download

Archives

You can download Gerolf Markup Shredder for Intel-386-compatible computers running under *Windows*, *Dos* or *Linux*, for all these operating systems in common, in one self-extracting *Windows-32* archive (12 MB) called `gerolf.exe`. Below follows the list of files contained therein:

- *Required on all systems*: `gmsintro.txt`, `gms008a.zip`, `texhyph.zip`.
- *Required on Dos*: `gmsunzip.bat`, `unzipdos.exe`, `bindos1.zip`, `bindos2.zip`.
- *Required on Linux*: `gmsunzip`, `gmssetup`, `binlinux.zip`.
- *Required on Win32*: `gmsunzip.bat`, `unzipwin.exe`, `binwin.zip`.
- *Recommended*: `docdemo1.zip`, `docw3c.zip`, `docweb.zip`, `fontst1p.zip`, `fontst1u`, `fontsttf.zip`.
- *Optional*: `fontst1c.zip`, `fontst1d.zip`, `fontst1e.zip`, `fontsin.zip`, `fontsth.zip`.

So you can download all these files independently, omitting those which are not required, if your computer runs under *Dos* or *Linux* only. Click on the names of these files with the *right* mouse button to save them. All ZIP packages are smaller than 1.44 MB.

Server

During execution of the self-extracting archive on *Windows-32*, you are asked for a directory to install GMS. Possible names are `C:\gerolf` or `C:\gmsNNNx`, if you want to use only the *text mode* or *command line* interfaces of Markup Shredder. This is the folder where the ZIP files are copied to and GMS will be installed. Since you are free to choose the installation target, that folder is given the symbolical name `[GMS_ROOT]` within this handbook.

If you want to make use of the GMS *web browser* interface, you may still have to download the XAMPP package, which contains a PHP server. Then you can install Markup Shredder in the `[Document_Root]` directory, where HTML- and PHP-files are taken from when there is a browser request to `http://localhost`. This folder may be named `C:\wamp\htdocs` (*Windows-32*) or `/var/www` (*Linux*). Please backup its existing content.

Unzip

Execution of the self-extracting archive and download of all ZIP files to the `[GMS_ROOT]` folder ends with a message telling you that you *either* have to run `gmsunzip.bat` there in the *command line* to continue with installation (you can do this on *Linux* too, saying `gmsunzip`), *or* to call the file `http://localhost/gerolf.php` in the *web browser*, if XAMPP server is running. GMS then unpacks the non-executable ZIP archives and creates the following directory structure:

- `[GMS_ROOT]` – `../gmsNNNx`, `../gerolf`
 - `batch` – *Dos/Windows* scripts
 - `bin` – *Dos/Windows/Linux* executable files
 - `data` – code pages and hyphenation patterns
 - `doc` – HTML template files, tested with GMS
 - `etc` – Configuration, setup and start files
 - `fonts` – *PostScript*, *TrueType* and *TeX* fonts
 - `shell` – *Linux* scripts
 - `fonts` – *TeX* macro scripts

- tmp – Temporary files

In the *command line*, you can interrupt the extraction process by pressing [\[Ctrl+C\]](#). When finished, GMS Unzip starts GMS Setup.

Setup

When you call `gmssetup.bat` (*Dos, Windows*) respectively `gmssetup` (*Linux*) in the `[GMS-ROOT]/etc` folder, the GMS *command line* interface creates `gerolf.bat` (*Dos, Windows*) or `gerolf` (*Linux*), the system-dependent launcher and configuration script.

On *Windows*, GMS Setup copies the standard True Type fonts (Arial, Courier New, Georgia, Tahoma, Times New Roman and Verdana) to the `%GMS_ROOT%\fonts\ttf` folder. These fonts should be present for Unicode and non-Latin code page support in GMS, because they are used in the template documents. You can download them from many internet sites, just enter e.g. `index of arial.ttf` into the search engine window.

On *Linux*, GMS Setup asks you for the path to the *Windows* True Type directory, e.g. `/hda1/WINDOWS/Fonts` (if drive C: is mounted to `/hda1`). GMS setup also writes `font.map` and initializes the TeX-format file.

At the end of GMS setup, Markup Shredder will start up for the first time, running the *text mode* interface.

Launch

Once GMS setup is finished, you can start Markup Shredder from the *command line*, saying `gerolf` or `gms`.

On *Windows-32* and *Linux* (KDE/Gnome), you also find desktop links and start respectively context menu entries. To add an item for Markup Shredder in the context menu for the right mouse button, open *Windows* Explorer and select *Extras/Folder Options/Data Types*. Edit the *HTML document* entry and add a new process *Open with gerolf.bat* and the corresponding application `%GMS_ROOT%\gerolf.bat`.

On *Windows 3x*, it is better to run Markup Shredder and GMS Setup at the *Dos* level, because only then GMS can execute the HTML browser and the PDF reader, starting the graphical user interface before every call of these applications and leaving it afterwards when you press [\[Alt+F4\]](#) [\[Alt+F4\]](#) [\[Enter\]](#).

Integration

In the end, GMS then does not need its own user interface, since there are advanced editors allowing to execute Markup Shredder rather than being executed. HTML-Kit, for instance, click on *Edit/Preferences/Programs* and edit the *external program list*. Add `%GMS_ROOT%\batch\gms.bat` as *program path* and `-t "{{FILE}}"` as *parameter*. Clicking on *Tools/Programs/GMS* then starts the typesetting process. Click on the title bar of the console window with the right mouse button to open the context menu; select *Preferences/Programs* and *Close at end*.

You can call the PDF reader by adding `gms.bat` for a second time, but with `-r "{{FILE}}"` as *parameter*. HTML-Kit assigns shortcut key to these new functions, so you can typeset and read your document by simply pressing [\[Ctrl+1\]](#) and [\[Ctrl+2\]](#), for example.

System

A few system-dependent preparations may have to be done (none for *Windows XP*):

- *Dos, Windows 3x/9x/ME*: In `C:\config.sys`, set `buffers=99` and `files=255`.
- *MS-Dos*: To support the 43/50 lines text mode, add `device=C:\dos\ansi.sys` to `C:\config.sys`. To improve performance, install a disk buffer via `loadhigh C:\dos\smartdrv.exe 4096` in `C:\autoexec.bat`. Under *Dos 3x*, a 32 megabyte partition is too small for a complete font installation. Stop GMS unzip and remove ZIP archives before running GMS setup.
- *FreeDos*: Enable caching via `installhigh=C:\fdos\bin\lbacache.com 4096` in `C:\fdconfig.sys`.
- *DR-Dos*: To improve performance, say `hibuffers=512` in `C:\config.sys` or `superpck /EP` or `nwcache 7670` in `C:\autoexec.bat`.
- *Windows 9x/ME*: There are a few minor limitations: GMS cannot display the list of supported HTML font families and entity names in `gerolf.log`. Creation of corrected versions of the input markup files by HTML-Tidy is omitted. You do not get a feedback on progress during TeX typesetting run.
- *Unix* systems other than *Linux* for Intel 386 compatible computers are not tested for compatibility with Markup Shredder by the author. You may have to modify the shell scripts to get it running. In any case, you must call `mk_reply` in `[GMS_ROOT]\doc\reply`, executing the command `gcc -o reply reply.c -Lstring 2> reply.log` to compile the `reply` binary which is required for the *text mode* interface. Then download `tidy` as well as `pdfetex` and `pdfetex.pool` for your operating system to `[GMS_ROOT]\bin\linux`. – The `gcc`, `sed` and `unzip` binaries are expected to be already present on your system.

Feedback/Rückmelden

Gerolf Markup Shredder got listed at many software download sites on the web, among them: 1888 Software, 3D2F, 4 Software Downloads, 5p Soft, [5 Star Share](#), 5 Star Soft, AB Archive, A List Downloads, All App, [All Soft Share](#), Anne Soft, AZ Downloads, [Best Freeware Download](#), [Best Software 4 Download](#), Best Software Library, Best Vista Downloads, Be Updated, Blue Programs, Blue Softs, Brother Soft, By 32, Cab Files, Canadian Content, Cariboo Planet, Cat Download, C-Download, Clean Softs, Creabit, Daily Softs, Dev Galaxy, DL Tube, [Do Download](#), Down Broad, Download 25, Download 2 Easy, Download 3000, [Download 3k](#), Download 4 You, Download By Net, Download-CC, Download Fast, Download Feed, Download Game Demo, Download Get, Download Hotfile, Download Junction, Download Line, [Download Pipe](#), Download Programs, Download Ready, Download Rol, Download Softs, Download Software 4 Free, Download Software Freeware, Downloads Portal, Download Tip, Download Tube, Download-Ware, Down Location, Driver Guide, Efree Down, Efree Soft, Email Hound, [Euro Download](#), Exe Find, Exe Files, Fast Download, [File Award](#), [File Basket](#), File Hungry, File Light, Files Land, Files Repository, Find Soft Online, Free Biz Files, Free DDLs, Free Download 24h, Free Download Audio, Free Download Business, Free Download Manager, Free Downloads Place, Free Fun Files, Free Shareware Center, Free Soft, Freeware, Freeware Box, Freeware Park, Fresh Folder, Get Free Softs, Get Soft, Get Software, Giveaway Of The Day, Gold Download, Go Loads, Gonna Soft, Hot Downloads, [Hot Lib](#), Inteli News, Internet Files Land, IT Location, Ivertch, Kazhost, Key Downloads, Kirupa, Last Download, Mail Archive, MP3 CD Software, Multimedialne-Programy, My Free Download, My Freewares, [My Web Memo](#), Necromancers, Net Sharing, New Free Downloads, New Softs, Olean Marketplace, [One Kit](#), Opossum Soft, PAD Pile, PAD Repository, PC 24 Hours, PC Download World, PC File Advisor, PC Magazine, Peach Seed, Perfect Downloads, Perfect Freeware, Personal Information Organizer, Programs Home, Program URL, Quality Shareware, QTDD, R Bytes, Real Free Download, Recent Soft, Red Softs, Red Tram, Runterladen, Rush, Safe Site, [Script Freebies](#), Seek Freeware, Serial Downloads, Seven Buck Software, Share 32, Shareware, [Shareware 54](#), Shareware Atlas, Shareware Bay, Shareware Download, [Shareware Island](#), Shareware List, [Shareware Ratings](#), [Shareware River](#), Siet Sell, Simple Freeware, Simple Shareware, [Simtel](#), Soft 20, [Soft 32](#), Soft 3k, Soft 411, Softalizer, Soft-All-Ware, Soft Am, Soft Basket, Soft Hypermarket, [Softizer](#), Soft Jamboree, Soft Low, [Soft Lookup](#), Soft Most, [Softpedia](#), [Soft Picks](#), Soft Pile, Softs, Soft Samba, Softsia, Softs Land, [Softs List](#), Soft Tester, Software Abyss, Software Archives, [Software Dc](#), Software Dungeon, Software Horizon, Software In Search, Software KB, Software Plaz, Software Sizzle, Software Vault, Soft-Web-Ware, Stuff Mate, Surf Pack, [TeX User Group](#) (announcements of GMS 0.01a, GMS 0.02a), The Best Soft, Timely Web, Tool 32, Top Shareware, Traffic Statistic, Updates, US Computer, VicMan, Wannadown, Web Net, Willing Software, [Win Site](#), World SSP, [Zonshare](#).

“We have reviewed your product Gerolf Markup Shredder 0.02a and added it to our website. Your software product awarded and rated 5 stars.” – *One Kit Software Magazine*



“Gerolf Markup Shredder 0.02a has been approved in Brother Soft. Your new entry has been reviewed and accepted in our Business – Word Processing listings and get a rating 5 out of 5. Brother Soft Review Staff give one product the maximum rating because this product is easy to use, it has a professional-looking interface, it is excellent compared to other programs in this section and so on.” – *Brother Soft*

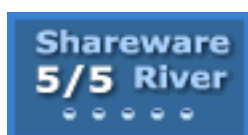
“Congratulations! Your product Gerolf Markup Shredder 0.02a has been tested by the Softpedia labs and found to be completely clean of adware/spyware components. We are impressed with the quality of your product and encourage you to keep this high standards in the future. Softpedia guarantees that Gerolf Markup Shredder 0.02a is 100% clean which means is does not contain any form of malware, including but not limited to: spyware, viruses, trojans and backdoors. This software product was tested thoroughly and was found absolutely clean, therefore it can be installed with no concern by any computer user.” – *Softpedia*



“Congratulations! Your software was rated 5 stars.” – *Exe Files*

“I am happy to be able to list your software at Best Soft Order. It really is a valued contribution that I and the rest of my team sincerely appreciate. Thank you very much for this opportunity.” – *Best Soft Order*

“Your Software Gerolf Markup Shredder 0.02a got a 5 Star award.” – *Shareware Island* – “Congratulations! Gerolf Markup Shredder has been selected and listed. Your software has been awarded 4 Stars based on our software rating guidelines.” – *Ivertch*



“We are pleased to inform you that your software has been awarded by our editors. Rating: 5 of 5 points.” – *Shareware River* – “We are glad to inform that your software Gerolf Markup Shredder 0.05a has been tested by our team and found to be safe to install.” – *New Free Downloads*

“Your fantastic software Gerolf Markup Shredder was reviewed by our software testing team and got ‘excellent’ award (5 stars).” – *Do Download*



“We have reviewed your software and want to congratulate for this outstanding and usefull software. We want to give you an award for your dedication to software development.” – *Download 2 Easy*

“We have recognized your site as interesting, informative resource what makes you prominent in comparison with subject-like Internet resources. Nowadays, in the overflowing software market quality becomes of preliminary importance, that is reliability, functionality, interface, ease of use and convenience for both beginners and professional users. Your software is ranked 5 stars by all these criteria.” – *Download Software 4 Free*

Appendix (Anhang)

GMS Program Files

Installation Folder:

[GMS_ROOT]

Subdirectories:

- batch
- data/cp
- data/enc
- data/ent
- data/gly
- data/krn
- data/lang
- data/row
- doc/reply
- etc
- shell
- tex/gerolf
- tex/hyphen

[GMS_ROOT]

Installation Files (Dos, Windows, Linux):

gerolf.css gerolf.php gmsintro.txt gms2exe.sed gmsunzip gmsunzip.bat

[GMS_ROOT]/batch

Batch Files (Dos, Windows):

browser.bat compiler.bat editor.bat g_code.bat g_color.bat g_dos.bat g_file.bat g_font.bat
g_good.bat g_launch.bat g_list.bat g_menu.bat g_palet.bat g_plug.bat g_prog.bat
g_rain.bat g_save.bat g_select.bat g_vars.bat g_wel.bat gms.bat l_banner.bat l_box.bat
l_code.bat l_color.bat l_desk.bat l_file.bat l_gms.bat l_good.bat l_list.bat l_menu.bat
l_prog.bat l_rain.bat l_save.bat l_select.bat l_wel.bat reader.bat shredder.bat viewer.bat
xpfonds.bat

[GMS_ROOT]/data/cp

Code Page Files:

cp437.txt cp850.txt cp860.txt cp863.txt cp865.txt cp874.txt cp1250.txt cp1251.txt
cp1252.txt cp1253.txt cp1254.txt cp1255.txt cp1256.txt cp1257.txt cp1258.txt
i8859-1.txt i8859-2.txt i8859-3.txt i8859-4.txt i8859-5.txt i8859-6.txt i8859-7.txt
i8859-8.txt i8859-9.txt i8859-10.txt i8859-11.txt i8859-13.txt i8859-14.txt i8859-15.txt
iscii.txt iso646.txt us-ascii.txt viscii.txt

[GMS_ROOT]/data/enc

Encoding Files, Hand-Written from Code Page:

arabicbh.enc arabicmt.enc psy.enc pzd.enc

Encoding Files, Generated from Code Page:

cp437.enc cp850.enc cp860.enc cp863.enc cp865.enc cp874.enc cp1250.enc
cp1251.enc cp1252.enc cp1253.enc cp1254.enc cp1255.enc cp1256.enc
cp1257.enc cp1258.enc i8859-1.enc i8859-2.enc i8859-3.enc i8859-4.enc
i8859-5.enc i8859-6.enc i8859-7.enc i8859-8.enc i8859-9.enc i8859-10.enc
i8859-11.enc i8859-13.enc i8859-14.enc i8859-15.enc
iscii.enc iso646.enc us-ascii.enc viscii.enc

Encoding Files, Generated from Unicode Row, with Glyph Names:

g0000.enc g0100.enc g0200.enc g0300.enc g0400.enc g0500.enc g0600.enc g0e00.enc
g1e00.enc g2000.enc g2100.enc g2200.enc g2300.enc g2400.enc g2500.enc g2600.enc
g2700.enc gf800.enc

Encoding Files, Generated from Unicode row, with Unicode Names:

u0000.enc u0100.enc u0200.enc u0300.enc u0400.enc u0500.enc u0600.enc u0e00.enc
u1e00.enc u2000.enc u2100.enc u2200.enc u2300.enc u2400.enc u2500.enc u2600.enc
u2700.enc uf800.enc

[GMS_ROOT]/data/ent

Entity Lists:

HTMLlat1.ent HTMLspec.ent HTMLsymb.ent

[GMS_ROOT]/data/gly

Glyph Lists:

gms.gly dingbats.gly

[GMS_ROOT]/data/krn

Kerning Tables:

phv.krn ptm.krn

[GMS_ROOT]/data/lang

Language Strings (**web browser** interface):

english.txt french.txt german.txt

[GMS_ROOT]/data/row

Unicode Rows:

u0000.row u0100.row u0200.row u0300.row u0400.row u0500.row u0600.row
u0a00.row u0e00.row u1e00.row u2000.row u2100.row u2200.row u2300.row
u2400.row u2500.row u2600.row u2700.row u3000.row u3100.row u3200.row u3300.row
uf600.row uf700.row uf800.row ufb00.row ufc00.row ufd00.row ufe00.row uff00.row

[GMS_ROOT]/doc/reply

Ansi-C Source (**Text Mode** Interface):

reply.c mk_reply

[GMS_ROOT]/etc

Shell Scripts (**Linux**):

folder gerolf gms_memo gmssetup

Batch Files (**Dos, Windows**):

folder.bat gerolf.bat gms_memo.bat gmssetup.bat

Configuration Files:

alias.cfg encoding.cfg epilogue.cfg files.cfg font.cfg pdftex.cfg
plugin.cfg prologue.cfg tidy.cfg typeset.cfg texmf.cnf

Log Files:

gerolf.log gmssetup.log

Font Maps:

cm.map font.map

Text Screens:

desktop.scn goodbye.scn launch_1.scn launch_2.scn launch_3.scn launch_x.scn
menu.scn welcome.scn writing.scn readme.txt

Dos System Files:

drv.sys pwd.sys

TeX Exit Files:

exit.tex q.tex quit.tex x.tex

[GMS_ROOT]/shell

Shell Scripts (Linux):

g_code g_color g_file g_font g_good g_launch g_list g_menu g_palet g_plug
g_prog g_rain g_save g_select g_vars g_wel gms l_banner l_box l_code l_color
l_desk l_file l_gms l_good l_list l_menu l_prog l_rain l_save l_select l_wel shredder

[GMS_ROOT]/tex/gerolf

TeX Macro Scripts:

gerolf.tex

[GMS_ROOT]/tex/hyphen

Hyphenation Patterns:

ca.tex cs.tex da.tex de.tex de-rf.tex el.tex en-UK.tex en-US.tex es.tex et.tex eu.tex fi.tex
fr.tex ga.tex hr.tex hu.tex ia.tex id.tex is.tex it.tex la.tex nl.tex no.tex pl.tex pt.tex ru.tex
sr.tex sv.tex tr.tex uk.tex wen.tex

[GMS_ROOT]

Installation Files (Dos, Windows, Linux)

gerolf.css

```
/* gerolf.css
===== */

.black {color: #000000} /* black */
.blue {color: #00008B} /* darkblue */
.gold {color: #B8860B} /* darkgoldenrod */
.green {color: #228B22} /* forestgreen */
.grey {color: #708090} /* slategrey */
.red {color: #FF0000} /* red */
.white {color: #FFF0FA} /* snow */

/* a:active {color: #FFF0FA} /* snow */
a:hover {color: #FF0000; /* red */
background-color: #D7EEFE}
/* background-color: #FFF0FA} snow */
a:link {color: #006400} /* darkgreen */
a:visited {color: #B8860B} /* darkgoldenrod */

body {color: #00008B; /* darkblue */
background-color: #FFFFFF; /* lightskyblue 87CEFA */
margin: 0mm; padding: 0mm;
font-family: Arial, sans-serif; font-size: 10pt}

h1, h2, h3, h4, h5, h6 {margin-top: 0pt; margin-bottom: 0pt;
margin-left: 0pt; margin-right: 0pt;
text-align: center; font-weight: bold}
h1 {page-break-before: always; font-size: 24.8832pt}
h2 {margin-top: 10pt; margin-bottom: 10pt; font-size: 20.736pt}
h3 {page-break-before: always; margin-top: 10pt;
margin-bottom: 10pt; font-size: 17.28pt}
h4 {margin-top: 10pt; margin-bottom: 10pt; font-size: 14.4pt}
h5 {page-break-before: always;
margin-top: 10pt; margin-bottom: 10pt; font-size: 12pt}
h6 {margin-top: 10pt; margin-bottom: 10pt; font-size: 10pt}

img {margin: 0mm; border-width: 0px; padding: 0mm}
img:hover {background-color: #FFF0FA} /* snow */

p.center {text-align: center}
pre.indent {margin-left: 10px; font-family: monospace}

table {margin: 0mm; border-width: 0px; padding: 0mm}

td {color: #483D8B; /* darkslateblue */
margin: 0mm; border-width: 0px; padding: 0mm;
font-size: 10pt; font-family: sans-serif; text-align: center}
td.button {width: 50%}
td.buttonhelp {width: 50%; text-align: left}
td.center {width: 100%}
td.greeting {text-align: left; font-size: 14pt}
td.left {text-align: left}
td.right {text-align: right}

tr {margin: 0mm; border-width: 0px; padding: 0mm}
tr.button {height: 28}
tr.middle {vertical-align: middle; height: 100%}
tr.top {vertical-align: top; height: 100%}
```

gerolf.php

```
<?php
# gerolf.php
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2008). <!-- #### -->

# Configuration: =====

$GMS_VERSION = "0.08a"; # <!-- #### -->
$GMS_DATE = "2008-01-07"; # <!-- #### -->

$GMS_SCRIPT = "index.php"; # <!-- #### -->
$GMS_SCRIPT = "gerolf.php"; # default script name

$GMS_LINKS = "external"; # link to external targets (server mode) <!-- #### -->
$GMS_LINKS = "internal"; # link to internal targets (default mode)

$GMS_ANALYST = "tidy";
$GMS_BANNER_X = "180"; $GMS_BANNER_Y = "150";
$GMS_CODEPAGE = "CP1252";
$GMS_HEIGHT_ONE = "30"; $GMS_HEIGHT_TWO = "40";
$GMS_MAXFILES = "123"; $GMS_MAXSIZE = "1234567";
$GMS_SIZE_ONE = "10"; $GMS_SIZE_TWO = "8";
$GMS_TSETTER = "pdfetex";
$GMS_WIDTH_ONE = "60"; $GMS_WIDTH_TWO = "40";

# Global: =====

$codepages = array ("cp1250", "cp1251", "cp1252", "cp1253", "cp1254", "cp1255",
"cp1256", "cp1257", "cp1258", "cp437", "cp850", "cp860", "cp863", "cp865",
"cp874", "i8859-1", "i8859-10", "i8859-11", "i8859-13", "i8859-14",
"i8859-15", "i8859-2", "i8859-3", "i8859-4", "i8859-5", "i8859-6", "i8859-7",
"i8859-8", "i8859-9", "iscii", "iso646", "us-ascii", "viscii");

$content = "<html><body><h1>File is empty</h1></body></html>";

$cwd = preg_replace ("/\\\\\\/", "/", getcwd ()); # current working directory

$global_actions = array ("Begin", "Quit", "Create", "Open", "View", "Edit",
"Browse", "Analyse", "Typeset", "Read", "Learn", "Download");

$global_actions_no_file = array ("Begin", "Create", "Open", "Learn",
"Download");

$global_actions_no_file_internal = array ("Begin", "Create", "Open", "Learn",
"Download", "Write", "Init");

$global_actions_no_setup = array ("Begin", "Learn", "Download", "Write",
"Init");

$global_initialisation = array (
```

```

"codepage" => "cp1252", "frame" => "parent", "language" => "English",
"process" => "", "file" => "", "backup" => "", "extrafonts" => "",
# Setting:
"width_one" => "60", "width_two" => "40",
"height_one" => "30", "height_two" => "40",
"size_one" => "10", "size_two" => "8",
"size" => "1048576", "files" => "5",
# Page:
"left" => "first", "lower" => "first", "master" => "first",
"middle" => "first", "parent" => "split",
"right" => "first", "upper" => "first";

$global_input = $global_initialisation;

if (strtoupper (substr (PHP_OS, 0, 3)) == "WIN") {
    $GMS_BINARIES = $cwd . "/bin/win";}
else {$GMS_BINARIES = $cwd . "/bin/linux";}

$GMS_FONTS = $cwd . "/fonts";
$GMS_LANGUAGE = $cwd . "/data/lang";
$GMS_PROCESS = $cwd . "/tmp";
$GMS_ROOT = $cwd;
$GMS_SETTING = $cwd . "/etc";
$GMS_TEMPLATE = $cwd . "/doc";
$GMS_VERSIONSTR = preg_replace ("/\./", "", $GMS_VERSION);
$GMS_ZIP = $GMS_ROOT . "/zip";

$PHP_VERSION = preg_replace ("/(\.)*$/", "$1", PHP_VERSION);

$SRV_ENCODING = "./data/enc";
$SRV_HANDBOOK = "/doc/handbook";
$SRV_PROCESS = "./tmp";
$SRV_SETTING = "./etc";

$title = "Gerolf Markup Shredder";

$ZIP_BODY = $GMS_ZIP . "/";

if (strtoupper (substr (PHP_OS, 0, 3)) == "WIN") {
    $setup_extrafonts = preg_replace ("/\\\\\\/", "/", $_ENV ["windir"]) .
    "/Fonts";}
else {$setup_extrafonts = "/dual/WINDOWS/Fonts";}

# Classes:

$page = new page;
$file = new file;
$frame = new frame;
$header = new header ("", "", "", "");
$language = new language ("");
$lower = new lower;
$master = new master;
$middle = new middle;
$process = new process;

# Functions: -----
function GMS_runtime ($before, $after) {
    global $language;
    $runtime = $after - $before;
    $msg = $language->language ("runtime");
    if ($runtime < "1") {$runtime = "$msg: <1s\n";}
    else {$runtime = "$msg: $runtime . "s\n";}
    return $runtime;}

function x_copy ($pattern, $sourcedir, $targetdir) {
    if (!is_dir ($targetdir)) {x_mkdir ($targetdir);}
    if (is_dir ($sourcedir)) {
        set_time_limit ("300");
        foreach (x_scandir ($sourcedir) as $entry) {
            if (preg_match ("/$pattern/i", $entry)) {
                copy ("$sourcedir/$entry", "$targetdir/$entry");}}}}

function x_mkdir ($foldername) {
    if (!is_dir ($foldername)) {@mkdir ($foldername);}
    clearstatcache ();
    if (!is_dir ($foldername)) {
        echo ("GMS: Cannot make directory $foldername<br />");}}

function x_file_get_contents ($filename) {
    if ($PHP_VERSION < 5) {
        if (is_readable ($filename)) {
            if ($h_file = fopen ($filename, "r")) {
                while (!feof ($h_file)) {$return = $return . fgets ($h_file);}
                fclose ($h_file);}
            return $return;}
        else {return file_get_contents ($filename);}}

function x_file_put_contents ($filename, $data) {
    if ($PHP_VERSION < 5) {
        if ($h_file = fopen ($filename, "w")) {
            fwrite ($h_file, $data);
            fclose ($h_file);}
        else {file_put_contents ($filename, $data);}}

function x_make_executable ($filename) {
    if (strtoupper (substr (PHP_OS, 0, 3)) != "WIN") {
        if (!is_executable ($filename)) {@chmod ($filename, 0755);}
        clearstatcache ();
        if (!is_executable ($filename)) {echo (
            "<pre><br />GMS: No executing permission for\n " .
            "$filename</pre>\n");}}

function x_scandir ($foldername) {
    if ($PHP_VERSION < 5) {
        $h_dir = opendir ($foldername);
        while (false != ($filename = readdir ($h_dir))) {
            $dirfiles [] = $filename;}
        return $dirfiles;}
    else {return scandir ($foldername);}}

function x_shell_exec ($command, $errorfile) {
    if (strtoupper (substr (PHP_OS, 0, 3)) == "WIN") {
        $command = preg_replace ("/\//", "\\", $command);}
    if (strtoupper (substr (PHP_OS, 0, 5)) == "WIN32") {
        $errorfile = "";}
    if ($errorfile == "") {shell_exec ($command);}
    else {shell_exec ("$command 2>> $errorfile");}}

function x_unlink ($pattern, $sourcedir) {
    if (is_dir ($sourcedir)) {
        foreach (x_scandir ($sourcedir) as $entry) {
            if (preg_match ("/$pattern/i", $entry)) {
                @unlink ("$sourcedir/$entry");}}}}

# Action (main program): =====
foreach ($global_input as $name => $value) {
    $value = $_REQUEST [$name];
    if ($value) {$global_input [$name] = $value;}

    if (($GMS_LINKS == "internal") and ($global_input ["process"] == "1") {
        $GMS_FILE = $global_input ["file"];}
    else {$GMS_FILE = $GMS_PROCESS . "/" . $global_input ["process"] . "/" .
        $global_input ["file"];}

    foreach ($_REQUEST as $name => $value) {
        $name = strtolower ($name);
        # Codepage:
        if ($name == "codepage") {$global_input ["codepage"] = $value;}
        # Language:

```

```

if (($name == "english")) {$global_input ["language"] = "English";
$global_input ["upper"] = "first";
if ($global_input ["backup"] != "") {
$global_input ["backup"] = "";
$global_input ["master"] = "learn";}
if (($global_input ["master"] == "write") or
($global_input ["master"] == "init")) {
$global_input ["master"] = "learn";}}
elseif ($name == "german") {$global_input ["language"] = "German";
$global_input ["upper"] = "first";
if ($global_input ["backup"] != "") {
$global_input ["backup"] = "";
$global_input ["master"] = "learn";}
if (($global_input ["master"] == "write") or
($global_input ["master"] == "init")) {
$global_input ["master"] = "learn";}} # Fixme: duplicate
# Parent frame:
elseif ($name == "join") {$global_input ["parent"] = "join";}
elseif ($name == "split") {$global_input ["parent"] = "split";}
# Action:
elseif ($name == "analyse") {$global_input ["master"] = "analyse";
$global_input ["upper"] = "first";}
elseif ($name == "analyse_left") {$global_input ["left"] = "analyse";
$global_input ["upper"] = "first";}
elseif ($name == "analyse_right") {$global_input ["right"] = "analyse";
$global_input ["upper"] = "first";}
elseif ($name == "backup") {$global_input ["master"] = "backup";
$global_input ["backup"] = preg_replace ("/[\\\\]+/", "/", $value);
$global_input ["upper"] = "wait";
$global_input ["parent"] = "join";} ## accessible via script parameter only
elseif ($name == "browse") {$global_input ["master"] = "browse";
$global_input ["upper"] = "first";}
elseif ($name == "browse_left") {$global_input ["left"] = "browse";
$global_input ["upper"] = "first";}
elseif ($name == "browse_right") {$global_input ["right"] = "browse";
$global_input ["upper"] = "first";}
elseif ($name == "create") {$global_input ["master"] = "create";
$global_input ["upper"] = "first";
$global_input ["parent"] = "join";}
elseif ($name == "download") {$global_input ["master"] = "download";
$global_input ["upper"] = "first";
$global_input ["parent"] = "join";} ###
elseif ($name == "download_left") {$global_input ["left"] = "download";
$global_input ["upper"] = "first";}
elseif ($name == "download_right") {$global_input ["right"] = "download";
$global_input ["upper"] = "first";}
elseif ($name == "edit") {$global_input ["master"] = "edit";
$global_input ["upper"] = "first";}
elseif ($name == "edit_left") {$global_input ["left"] = "edit";
$global_input ["upper"] = "first";}
elseif ($name == "edit_right") {$global_input ["right"] = "edit";
$global_input ["upper"] = "first";}
elseif ($name == "init") {$global_input ["master"] = "init";
$global_input ["upper"] = "wait";
$global_input ["parent"] = "join";}
elseif ($name == "learn") {$global_input ["master"] = "learn";
$global_input ["upper"] = "first";
$global_input ["parent"] = "join";} ###
elseif ($name == "learn_left") {$global_input ["left"] = "learn";
$global_input ["upper"] = "first";}
elseif ($name == "learn_right") {$global_input ["right"] = "learn";
$global_input ["upper"] = "first";}
elseif ($name == "open") {$global_input ["master"] = "open";
$global_input ["upper"] = "first";
$global_input ["parent"] = "join";}
elseif ($name == "quit") {$global_input ["master"] = "quit";
$global_input ["file"] = "";
$global_input ["process"] = "";
$global_input ["upper"] = "first";
$global_input ["parent"] = "join";}
elseif ($name == "read") {$global_input ["master"] = "read";
$global_input ["upper"] = "first";}
elseif ($name == "read_left") {$global_input ["left"] = "read";
$global_input ["upper"] = "first";}
elseif ($name == "read_right") {$global_input ["right"] = "read";
$global_input ["upper"] = "first";}
elseif ($name == "typeset") {$global_input ["master"] = "typeset";
$global_input ["upper"] = "wait";}
elseif ($name == "typeset_left") {$global_input ["left"] = "typeset";
$global_input ["upper"] = "wait";}
elseif ($name == "typeset_right") {$global_input ["right"] = "typeset";
$global_input ["upper"] = "wait";}
elseif ($name == "view") {$global_input ["master"] = "view";
$global_input ["upper"] = "first";}
elseif ($name == "view_left") {$global_input ["left"] = "view";
$global_input ["upper"] = "first";}
elseif ($name == "view_right") {$global_input ["right"] = "view";
$global_input ["upper"] = "first";}
elseif ($name == "write") {$global_input ["master"] = "write";
$global_input ["upper"] = "wait";
$global_input ["parent"] = "join";}
# Work file:
elseif ($name == "file_create") {$global_input ["master"] = "accept";
$global_input ["parent"] = "join";
$global_input ["right"] = "accept";
$file->samples ($value); # template
$file->accept ($REQUEST ["file_no_1"]);}
elseif ($name == "file_open") {$global_input ["master"] = "accept";
$global_input ["parent"] = "join";
$global_input ["right"] = "accept";
$content = x_file_get_contents ($FILES ["file_no"] ["tmp_name"] ["0"]);
$file->accept ($FILES ["file_no"] ["name"] ["0"]);}
elseif ($name == "file_intern") {
if (is_file ($value)) {
$global_input ["file"] = $value;
$global_input ["process"] = "1";
$global_input ["master"] = "accept";}
else {$global_input ["file_intern"] = $value;}}
elseif ($name == "file_to_save") {
$file->save ($value);}
elseif ($global_input ["master"] == "first") {
if (!(is_file ("$_GMS_SETTING/font.map")) or
!(is_file ("$_GMS_BINARIES/gerolf.efmt"))) {
$global_input ["master"] = "setup";
$global_input ["parent"] = "join";}}}

$_GMS_CODEPAGE = $global_input ["codepage"];
ksort ($global_input);
$process->run (); # limit processes
echo $frame->page ();

class backup {#=====
# Example script call to build backup in C: "gerolf.php?backup=C:"

function archive_build ($title) {
global $_GMS_BINARIES; global $_GMS_ROOT;
global $_GMS_SETTING; global $_GMS_ZIP;
$log = x_file_get_contents ("gms2zip.log");
$log = "$log\n$title.zip";
x_file_put_contents ("gms2zip.log", $log);
chdir ($_GMS_ZIP);
set_time_limit ("300");
x_shell_exec (
"$$_GMS_BINARIES/zip -r $title . >> $_GMS_SETTING/gms2zip.log", "");
@unlink ("$_GMS_ROOT/$title.zip");
@rename ("$_GMS_ZIP/$title.zip", "$$_GMS_ROOT/$title.zip");}

function backup_build () {
global $_GMS_ROOT; global $_GMS_SCRIPT; global $_GMS_VERSIONSTR;
@unlink ("$_GMS_ROOT/~gms$_GMS_VERSIONSTR.ddf");
if (file_exists ("$_GMS_ROOT/~gms$_GMS_VERSIONSTR.ddf")) {

```

```

$return = "\n\n Cannot remove lockfile $GMS_ROOT/~gms$GMS_VERSIONSTR.ddf";}
else {
$target = preg_replace ("/[\\\\\\+\/", "/", $REQUEST ["backup"]);
if (!is_dir ($target)) (@mkdir ($target));
if (!is_dir ($target)) {$return =
"\n\n Missing target, e. g. \"$GMS_SCRIPT?backup=C:/GMS_BAK\"";}
else {
$target = "$target/gms$GMS_VERSIONSTR";
x_mkdir ($target);
$today = getdate ();
$date = $today ["year"];
if ($today ["mon"] < "10") {$date = $date . "0";}
$date = $date . $today ["mon"];
if ($today ["mday"] < "10") {$date = $date . "0";}
$date = $date . $today ["mday"];
$target = "$target/$date";
$return = "\n\n Target: " . $target;
x_mkdir ($target);
x_copy ("\zip", "$GMS_ROOT", $target);
$list = array ("gerolf.css", "gerolf.php", "gmsintro.txt",
"gms2exe.sed", "gmsunzip", "gmsunzip.bat",
"unzipdos.exe", "unzipwin.exe");
foreach ($list as $entry) {
copy ("$GMS_ROOT/$entry", "$target/$entry");}
if (strtoupper (substr (PHP_OS, 0, 5)) == "WINNT") {
chdir ("$GMS_ROOT");
# Build gmsNNNx.exe:
set_time_limit ("300");
@rename ("$GMS_ROOT/gms$GMS_VERSIONSTR.exe",
"$GMS_ROOT/gms$GMS_VERSIONSTR.ex_");
shell_exec ($ENV ["windir"] .
"/system32/exprs /n gms2exe.sed 2> gms2exe.log");
if (is_file ("$GMS_ROOT/gms$GMS_VERSIONSTR.exe")) {
copy ("$GMS_ROOT/gms$GMS_VERSIONSTR.exe",
"$target/gms$GMS_VERSIONSTR.exe");}
else {$return = $return .
"\n\n Could not build gms$GMS_VERSIONSTR.exe";}}}
return $return;}

function execute () {
global $cwd;
global $GMS_ROOT; global $GMS_ZIP;
global $GMS_SETTING; global $GMS_VERSIONSTR;
global $ZIP_BODY;
# Build log file header:
x_file_put_contents ("gms2zip.log",
"GMS: Building archives in " . $cwd . " ..\n");
# gmsNNNx archive: .....
$this->zipdir_rebuild ();
copy ("$GMS_ROOT/gmsindex.htm", "$ZIP_BODY/gmsindex.htm");
copy ("$GMS_ROOT/index.htm", "$ZIP_BODY/index.htm");
copy ("$GMS_ROOT/gerolf.css", "$ZIP_BODY/gerolf.css");
# GMS_ROOT/batch:
x_mkdir ("$ZIP_BODY/batch");
x_copy ("\bat", "$GMS_ROOT/batch", "$ZIP_BODY/batch");
# GMS_ROOT/data:
x_mkdir ("$ZIP_BODY/data");
x_copy ("\txt", "$GMS_ROOT/data/cp", "$ZIP_BODY/data/cp");
x_copy ("\enc", "$GMS_ROOT/data/enc", "$ZIP_BODY/data/enc");
x_copy ("\ent", "$GMS_ROOT/data/ent", "$ZIP_BODY/data/ent");
x_copy ("\gly", "$GMS_ROOT/data/gly", "$ZIP_BODY/data/gly");
x_copy ("\krn", "$GMS_ROOT/data/krn", "$ZIP_BODY/data/krn");
x_copy ("\row", "$GMS_ROOT/data/row", "$ZIP_BODY/data/row");
x_mkdir ("$ZIP_BODY/data/row_disable");
x_copy ("\row", "$GMS_ROOT/data/row_disable",
"$ZIP_BODY/data/row_disable");
# GMS_ROOT/doc:
x_mkdir ("$ZIP_BODY/doc");
$this->zipdoc_build ("default");
x_mkdir ("$ZIP_BODY/doc/symbol");
x_mkdir ("$ZIP_BODY/doc/unicode");
x_mkdir ("$ZIP_BODY/doc/zapfding");
x_copy ("\htm", "$GMS_ROOT/doc", "$ZIP_BODY/doc");
# GMS_ROOT/etc:
x_mkdir ("$ZIP_BODY/etc");
$list = array ("\cfg", "\scn", "\sys", "\tcx", "\tex", "folder",
"folder.bat", "gmssetup", "gmssetup.bat", "texmf.cnf", "cm.map");
foreach ($list as $pattern) {
x_copy ("$pattern", "$GMS_SETTING", "$ZIP_BODY/etc");}
@unlink ("$ZIP_BODY/etc/plugin.cfg");
# GMS_ROOT/fonts:
$this->zipfontsdir_build ();
$this->zipfonts_build ("afm", "adobe", "acrobat", "afm");
$this->zipfonts_build ("tfm", "pub", "cm", "tfm");
$this->zipfonts_build ("type1", "pub", "cm", "pfb");
$this->zipfonts_build ("afm", "pub", "gutenbg", "afm");
$this->zipfonts_build ("type1", "pub", "gutenbg", "pfb");
$this->zipfonts_build ("ttf", "bh", "lsans", "ttf"); ###
$this->zipfonts_build ("ttf", "pub", "beauti", "ttf"); ###
x_mkdir ("$ZIP_BODY/fonts/ttf/bh");
$list = array ("ignore_", "handw", "### lsans",
"lhtypew", "lucon");
foreach ($list as $pattern) {
x_mkdir ("$ZIP_BODY/fonts/ttf/bh/$pattern");}
x_mkdir ("$ZIP_BODY/fonts/ttf/bt");
$list = array ("ignore_", "basker", "bodoni", "parkave", "zapfchan");
foreach ($list as $pattern) {
x_mkdir ("$ZIP_BODY/fonts/ttf/bt/$pattern");}
x_mkdir ("$ZIP_BODY/fonts/ttf/itc");
$list = array ("ignore_", "framd");
foreach ($list as $pattern) {
x_mkdir ("$ZIP_BODY/fonts/ttf/itc/$pattern");}
x_mkdir ("$ZIP_BODY/fonts/ttf/ms");
$list = array ("ignore_", "comic", "georgia", "sans", "sylfaen",
"tahoma", "trebuc", "verdana");
foreach ($list as $pattern) {
x_mkdir ("$ZIP_BODY/fonts/ttf/ms/$pattern");}
x_mkdir ("$ZIP_BODY/fonts/ttf/mt");
$list = array ("ignore_", "aharoni", "andalus", "arial", "artro",
"cour", "david", "frank", "garamond", "gothic", "impact", "levenim",
"miriam", "narkisim", "rod", "simpo", "times", "trado");
foreach ($list as $pattern) {
x_mkdir ("$ZIP_BODY/fonts/ttf/mt/$pattern");}
# GMS_ROOT/shell:
x_mkdir ("$ZIP_BODY/shell");
x_copy ("...*", "$GMS_ROOT/shell", "$ZIP_BODY/shell");
# GMS_ROOT/tex:
x_mkdir ("$ZIP_BODY/tex");
x_mkdir ("$ZIP_BODY/tex/gerolf");
x_copy ("\tex", "$GMS_ROOT/tex/gerolf", "$ZIP_BODY/tex/gerolf");
x_mkdir ("$ZIP_BODY/tex/plain");
x_copy ("\tex", "$GMS_ROOT/tex/plain", "$ZIP_BODY/tex/plain");
# GMS_ROOT/tmp:
x_mkdir ("$ZIP_BODY/tmp");
# Build archive:
$this->archive_build ("gms$GMS_VERSIONSTR");
# Hyphenation archive: .....
$this->zipdir_rebuild ();
# GMS_ROOT/tex:
x_mkdir ("$ZIP_BODY/tex");
x_mkdir ("$ZIP_BODY/tex/hyphen");
x_copy ("\tex", "$GMS_ROOT/tex/hyphen", "$ZIP_BODY/tex/hyphen");
$this->archive_build ("texhyph");
# Binary archives: .....
# Dos binary archive 1:
$this->zipdir_rebuild ();
x_mkdir ("$ZIP_BODY/bin");
x_mkdir ("$ZIP_BODY/bin/dos");
$list = array ("cwsdpmi.exe", "pdfetex.exe", "pdfetex.poo?");
foreach ($list as $pattern) {

```

```

    x_copy ($pattern, "$GMS_ROOT/bin/dos", "$ZIP_BODY/bin/dos");
$this->archive_build ("bindos1");
# Dos binary archive 2:
$this->zipdir_rebuild ();
x_mkdir ("ZIP_BODY/bin");
x_mkdir ("ZIP_BODY/bin/dos");
$list = array ("bat2exec.com", "browse.com", "afm2tfm.exe", "mdir.com",
"reply.exe", "sed.exe", "tidy.exe", "tff2afm.exe", "vptovf.exe");
foreach ($list as $pattern) {
    x_copy ($pattern, "$GMS_ROOT/bin/dos", "$ZIP_BODY/bin/dos");
$this->archive_build ("bindos2");
# Linux binary archive:
$this->zipdir_rebuild ();
x_mkdir ("ZIP_BODY/bin");
x_mkdir ("ZIP_BODY/bin/linux");
$list = array ("afm2tfm", "pdfetex", "pdfetex.pool",
"reply", "tidy", "tff2afm", "vptovf");
foreach ($list as $pattern) {
    x_copy ($pattern, "$GMS_ROOT/bin/linux", "$ZIP_BODY/bin/linux");
$this->archive_build ("binlinux");
# Windows binary archive:
$this->zipdir_rebuild ();
x_mkdir ("ZIP_BODY/bin");
x_mkdir ("ZIP_BODY/bin/win");
$list = array ("bat2exec.com", "browse.com", "afm2tfm.exe",
"pdfetex.exe", "pdfetex.poo.?", "reply.exe", "sed.exe",
"tidy.exe", "tff2afm.exe", "vptovf.exe", "zip.exe");
foreach ($list as $pattern) {
    x_copy ($pattern, "$GMS_ROOT/bin/win", "$ZIP_BODY/bin/win");
$this->archive_build ("binwin");
# Fonts archives: .....
# Build fontstlc.zip:
$this->zipdir_rebuild ();
$this->zipfontsdir_build ();
$this->zipfonts_build ("afm", "pub", "cmsuper", "afm");
$this->zipfonts_build ("type1", "pub", "cmsuper", "pf.?");
$list = array ("fb", "ff", "fi", "r", "ss", "sx", "ti", "tt", "vt", "xc");
foreach ($list as $pattern) {
    @unlink ("ZIP_BODY/fonts/afm/pub/cmsuper/fcm$pattern.afm");
    @unlink ("ZIP_BODY/fonts/type1/pub/cmsuper/fcm$pattern.pfb");
$this->archive_build ("fontstlc");
# Build fontstld.zip:
$this->zipdir_rebuild ();
$this->zipfontsdir_build ();
$this->zipfonts_build ("afm", "pub", "cmsuper", "afm");
$this->zipfonts_build ("type1", "pub", "cmsuper", "pf.?");
$list = array ("bi", "bx", "cc", "dh", "sx", "ti", "tt", "vt", "xc");
foreach ($list as $pattern) {
    @unlink ("ZIP_BODY/fonts/afm/pub/cmsuper/fcm$pattern.afm");
    @unlink ("ZIP_BODY/fonts/type1/pub/cmsuper/fcm$pattern.pfb");
$this->archive_build ("fontstld");
# Build fontstle.zip:
$this->zipdir_rebuild ();
$this->zipfontsdir_build ();
$this->zipfonts_build ("afm", "pub", "cmsuper", "afm");
$this->zipfonts_build ("type1", "pub", "cmsuper", "pf.?");
$list = array ("bi", "bx", "cc", "dh", "fb", "ff", "fi", "r", "ss");
foreach ($list as $pattern) {
    @unlink ("ZIP_BODY/fonts/afm/pub/cmsuper/fcm$pattern.afm");
    @unlink ("ZIP_BODY/fonts/type1/pub/cmsuper/fcm$pattern.pfb");
$this->archive_build ("fontstle");
# Build fontstlp.zip:
$this->zipdir_rebuild ();
$this->zipfontsdir_build ();
$this->zipfonts_build ("afm", "adobe", "utopia", "afm");
$this->zipfonts_build ("afm", "bh", "luximono", "afm");
$this->zipfonts_build ("afm", "bt", "charter", "afm");
$this->zipfonts_build ("afm", "ibm", "courier", "afm");
$this->zipfonts_build ("type1", "adobe", "utopia", "pf.?");
$this->zipfonts_build ("type1", "bh", "luximono", "pf.?");

$this->zipfonts_build ("type1", "bt", "charter", "pf.?");
$this->zipfonts_build ("type1", "pub", "$pattern", "pf.?");
$this->archive_build ("fontstlp");
# Build fontstlu.zip:
$this->zipdir_rebuild ();
$this->zipfontsdir_build ();
$list = array ("antigua", "avantgde", "bookman", "grotesk",
"palatino", "schoolb");
foreach ($list as $pattern) {
    $this->zipfonts_build ("afm", "urw", "$pattern", "afm");
    $this->zipfonts_build ("type1", "urw", "$pattern", "pf.?");
$this->archive_build ("fontstlu");
# Build fontsttf.zip:
$this->zipdir_rebuild ();
$this->zipfontsdir_build ();
$this->zipfonts_build ("tff", "bh", "lhandw", "tff");
##$this->zipfonts_build ("tff", "bh", "lsans", "tff"); ##
$this->zipfonts_build ("tff", "bt", "parkav", "tff");
$this->zipfonts_build ("tff", "bt", "zapfchan", "tff");
$this->zipfonts_build ("tff", "pub", "ayummy", "tff");
##$this->zipfonts_build ("tff", "pub", "beauti", "tff"); ##
$this->zipfonts_build ("tff", "pub", "cardo", "tff");
$this->zipfonts_build ("tff", "pub", "champ", "tff");
$this->zipfonts_build ("tff", "pub", "garamond", "tff");
$this->zipfonts_build ("tff", "pub", "gentium", "tff");
$this->zipfonts_build ("tff", "pub", "shusha", "tff");
$this->archive_build ("fontsttf");
# Build fontsin.zip (Indian):
$this->zipdir_rebuild ();
$this->zipfontsdir_build ();
$this->zipfonts_build ("tff", "pub", "shree", "tff");
$this->archive_build ("fontsin");
# Build fontsth.zip (Thai):
$this->zipdir_rebuild ();
$this->zipfontsdir_build ();
$this->zipfonts_build ("tff", "mt", "courthai", "tff");
$this->archive_build ("fontsth");
# Documentation archives: .....
# Build docdemo.zip:
$this->zipdir_rebuild ();
x_mkdir ("ZIP_BODY/doc");
$list = array ("1center", "1cleft", "1cright", "2ceven",
"2cmenu", "2cmenu", "2cstag", "3c2stag", "3ceven", "3cmenu",
"3cmenu", "3csdibar", "4ccenter", "4cstag", "4cstag",
"align", "arabic", "bible", "book", "booklet", "codepage",
"color", "cyrillic", "entities", "epic", "greek", "handbook", "hebrew",
"indian", "kerning", "latin", "maximal", "minimal", "nesting", "play",
"polyglot", "reply", "script", "spanning", "symbol", "tale", "thai",
"unicart", "vietnam", "zapfding");
foreach ($list as $pattern) {$this->zipdoc_build ($pattern);}
@unlink ("ZIP_BODY/doc/unicart/unicar.htm");
$this->archive_build ("docdemo");
# Build docw3c.zip:
$this->zipdir_rebuild ();
x_mkdir ("ZIP_BODY/doc");
$list = array ("css1", "css2", "css2/images", "css2/style",
"html401", "html401/appendix", "html401/images", "html401/index",
"html401/interact", "html401/intro", "html401/present", "html401/sgml",
"html401/struct", "html401/style", "xhtml10", "xhtmlbas");
foreach ($list as $pattern) {$this->zipdoc_build ($pattern);}
$this->archive_build ("docw3c");
# Build docweb.zip:
$this->zipdir_rebuild ();
x_mkdir ("ZIP_BODY/doc");
$list = array ("gedcom60", "gentle", "guide",
"hixie", "htmltidy", "novel", "primer", "rotation", "tds", "unicode");

```

```

foreach ($list as $pattern) {$this->zipdoc_build ($pattern);}
$this->archive_build ("docweb");
$this->zipdir_remove ();

function run () {
    global $header;
    $before = time ();
    $return = $header->header ("HTML 4.01 Transitional", "", "nometa", "") .
        "\n<body text = \"#00008B\">\n\n<br />\n\n" .
        "<pre style = \"fontstyle\">\n\n . " . "GMS: Building archives ..." .
        $this->execute () .
        $this->backup_build () .
        "\n\n Done.\n\n ";
    return $return . "GMS_runtime ($before, time ()) .
    "</pre>\n\n" . "</body>\n\n</html>";}

function zipdir_build () {
    global $GMS_ROOT; global $GMS_ZIP;
    global $ZIP_BODY;
    chdir ("GMS_ROOT");
    x_mkdir ("GMS_ZIP");
    x_mkdir ("ZIP_BODY");}

function zipdir_rebuild () {
    $this->zipdir_remove (); $this->zipdir_build ();}

function zipdir_remove () {
    global $GMS_ROOT; global $GMS_ZIP;
    chdir ("GMS_ROOT");
    if (strlen ($GMS_ZIP) > "9") {# do not remove top level directory by mistake
        if (strtoupper (substr (PHP_OS, 0, 5)) == "WIN32") {
            shell_exec ("deltree /y \"GMS_ZIP\"");}
        elseif (strtoupper (substr (PHP_OS, 0, 3)) == "WIN") {
            shell_exec ("rd /s /q \"GMS_ZIP\"");}
        else {x_shell_exec ("rm -r $GMS_ZIP", "");}}

function zipdoc_build ($title) {
    global $GMS_ROOT; global $GMS_ZIP;
    global $ZIP_BODY;
    x_mkdir ("ZIP_BODY/doc/$title");
    x_copy ("\css", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("\gif", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("\htm.?", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("\jpg", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("\png", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("charsets.txt", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("gedcom60.xml", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("favicon.ico", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("mk_reply", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("story.txt", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("readme.htm", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("readme.txt", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("reply.c", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    x_copy ("reply.txt", "$GMS_ROOT/doc/$title", "ZIP_BODY/doc/$title");
    ## These don't go into the PDF, so leave out:
    ## if [ -f HTML4.* ]; then cp HTML4.* "ZIP_BODY/doc/$1"; fi
    ## if [ -f *.dtd ]; then cp *.dtd "ZIP_BODY/doc/$1"; fi
    ## if [ -f *.ent ]; then cp *.ent "ZIP_BODY/doc/$1"; fi
}

function zipfonts_build ($one, $two, $three, $four) {
    global $GMS_ROOT;
    global $GMS_ZIP;
    global $ZIP_BODY;
    x_mkdir ("ZIP_BODY/fonts/$one/ignore");
    x_mkdir ("ZIP_BODY/fonts/$one/$two");
    x_mkdir ("ZIP_BODY/fonts/$one/$two/ignore");
    x_mkdir ("ZIP_BODY/fonts/$one/$two/$three");
    @copy ("GMS_ROOT/fonts/$one/$two/readme.txt",
        "ZIP_BODY/fonts/$one/$two/readme.htm");
    @copy ("GMS_ROOT/fonts/$one/$two/$three/readme.txt",
        "ZIP_BODY/fonts/$one/$two/$three/readme.htm");
    x_copy ("\$four", "$GMS_ROOT/fonts/$one/$two/$three",
        "ZIP_BODY/fonts/$one/$two/$three");
    # for k in lhandw parkave zapfchan; do
    #     if [ "$k" = "$3" ]; then rm "ZIP_BODY/fonts/$1/$2/$3/$j"; fi
    # done;
}

function zipfontsdir_build () {
    global $ZIP_BODY;
    x_mkdir ("ZIP_BODY/fonts");
    x_mkdir ("ZIP_BODY/fonts/ignore");
    x_mkdir ("ZIP_BODY/fonts/afm");
    x_mkdir ("ZIP_BODY/fonts/tfm");
    x_mkdir ("ZIP_BODY/fonts/ttf");
    x_mkdir ("ZIP_BODY/fonts/type1");
    x_mkdir ("ZIP_BODY/fonts/vf");}

class file {#=====
function accept ($newfile) {
    global $content; global $global_input; global $process;
    global $GMS_PROCESS;
    $counter = 0; $newdirectory = 0;
    if ($newfile != "") {
        # Remove path:
        $newfile = preg_replace ("/[\\\\|\/].*[\\\\|\/]/", "", $newfile);
        $global_input ["file"] = $newfile;
        $global_input ["file_intern"] = "";
        # Get random process number that really does not exist yet:
        srand ((double) microtime () * 1000000);
        while (($newdirectory == 0) and ($counter < 100)) {
            $counter ++;
            $secret = rand (10000000, 99999999);
            $global_input ["process"] = $secret;
            $dir_secret = $GMS_PROCESS . "/" . $secret;
            if (!file_exists ($dir_secret)) {$newdirectory = 1;}
            # Make process directory:
            $process->limit_processes ("24");
            if (!file_exists ($dir_secret)) {mkdir ($dir_secret, 0777);}
            # Store submitted files:
            if ($REQUEST ["file_no_1"] == "") {
                foreach ($FILES ["file_no"] ["error"] as $key => $error) {
                    if ($error == UPLOAD_ERR_OK) {
                        $tmp_name = $FILES ["file_no"] ["tmp_name"] [$key];
                        $name = $FILES ["file_no"] ["name"] [$key];
                        $name = preg_replace ("/[\\\\|\/].*[\\\\|\/]/", "", $name);
                        move_uploaded_file ($tmp_name, "$dir_secret/$name");
                        @chmod ("dir_secret/$name", 0660);} ##
                    $file = $GMS_PROCESS . "/" . $secret . "/" . $newfile;
                    $file = preg_replace ("/^([-\/~:~\w. ]+)$/", "$1", $file);
                    x_file_put_contents ($file, $content);}
            }
        }
    }

function analyse () {
    global $cwd; global $global_input; global $header;
    global $GMS_ANALYST; global $GMS_BINARIES; global $GMS_LINKS;
    global $GMS_PROCESS; global $GMS_SETTING; global $GMS_SHELL;
    global $SRV_PROCESS; global $SRV_SETTING; global $SRV_HANDBOOK;
    $srv_file = $SRV_PROCESS . "/" .
        $global_input ["process"] . "/" . $global_input ["file"];
    if (!file_exists ($srv_file)) {
        $srv_file = $global_input ["file"];}
    if (strtoupper (substr (PHP_OS, 0, 3)) != "WIN") {
        $srv_file = preg_replace ("|$cwd|", "", $srv_file);}
    if ($GMS_LINKS == "internal") {
        $doc_css = $SRV_HANDBOOK . "/../css2/css2.htm";}
}

```



```

$doc_html = $SRV_HANDBOOK . "../html401/html401.htm";
$doc_tidy = $SRV_HANDBOOK . "../htmltidy/htmltidy.htm";
$doc_xhtml = $SRV_HANDBOOK . "../xhtml10/xhtml10.htm";
else {
$doc_css = "http://www.w3.org/TR/REC-CSS2";
$doc_html = "http://www.w3.org/TR/html401";
$doc_tidy = "http://www.w3.org/People/Raggett/tidy";
$doc_xhtml = "http://www.w3.org/TR/xhtml1";
clearstatcache ();
$file = basename ($global_input ["file"]);
if (($global_input ["process"] == "1" and ($GMS_LINKS == "internal")) {
$GMS_FOLDER = dirname ($global_input ["file"]);
} else {
$GMS_FOLDER = $GMS_PROCESS . "/" . $global_input ["process"];
}
if (!is_writable ($GMS_FOLDER)) {echo "<pre>";
"<br /> GMS error: No writing permission for $GMS_FOLDER</pre>";
}
if (is_dir ($GMS_FOLDER)) {chdir ($GMS_FOLDER);}
# Get basename:
$search [0] = "\.xhtml/"; $replace [0] = "";
$search [1] = "\.shtml/"; $replace [1] = "";
$search [2] = "\.html/"; $replace [2] = "";
$search [3] = "\.htm/"; $replace [3] = "";
$search [4] = "\.xml/"; $replace [4] = "";
$base = preg_replace ($search, $replace, $file);
# Run Tidy (and get run time):
$before = time ();
$GMS_BASE = $base;
if (file_exists ($GMS_BASE . ".htm")) {
$jobfile = $GMS_BASE . ".htm";
}
elseif (file_exists ($GMS_BASE . ".html")) {
$jobfile = $GMS_BASE . ".html";
}
$arg = $GMS_BINARIES . "/$GMS_ANALYST -config " . $GMS_SETTING .
"/tidy.cfg -f " . $GMS_BASE . ".err " . $jobfile .
"> " . $GMS_BASE . ".ok";
set_time_limit ("300");
if (file_exists ($jobfile)) {$output = x_shell_exec ($arg, "");}
# Make error file better readable:
$rd = "span class = \"red\""; $wt = "span class = \"white\"";
$bl = "span class = \"blue\""; $gd = "span class = \"gold\"";
$gn = "span class = \"green\""; $bk = "span class = \"black\"";
$tar = "target = \"_blank\"";
$tidy = "http://www.w3.org/People/Raggett/tidy";
$Ti = "This is <a\n $tar href = $tidy>";
$V = "\n (Version)";
$ret = " GMS: Running markup syntax checker ... \n\n";
# $parse = "<rd>Input read from</span>";
# $ctcfg = "<rd>Configuration read from</span>";
# "<a $tar href = \"$SRV_SETTING/tidy.cfg\"> " . "tidy.cfg</a>\n ";
$about = "to learn more about";
$looks = "content looks like";
$usecss = "recommended to use";
# Display cosmetics:
$search [0] = "\n"; $replace [0] = " ";
$search [1] = "\n/"; $replace [1] = "\n ";
$search [2] = "/</"; $replace [2] = "&lt;";
$search [3] = "/>"; $replace [3] = "&gt;";
$search [4] = "&lt;/"; $replace [4] = "<b><rd>&lt;/span><bk>";
$search [5] = "/&gt;/"; $replace [5] = "</span><rd>&gt;/span></b>";
$search [6] = "/(attribute) \"(.*)\"(.*)\"(.*)\"/";
$replace [6] = "$1 <gn>$2</span>$3<gd>$4</span>";
$search [7] = "/$file\s*/i"; $replace [7] = "";
$search [8] = "/Info: /"; $replace [8] = "";
# Create links:
$search [9] = "\/(http.*)\n/i";
$replace [9] = "\<a\n $tar href = $1 >$1</a>";
$search [10] = "\/s(http.*)\n/i";
$replace [10] = "<a\n $tar href = $1 >$1</a>$2";
$search [11] = "\/s(http.*)\n/i";
$replace [11] = "<a\n $tar href = $1 >$1</a>\n";
$search [12] = "/HTML Tidy(.*)\n(vers/i";

```

```

$replace [12] = "$Ti<rd>HTML Tidy</span></a>$1$V";
$search [13] = "/(appears to be)(.*)/i";
$replace [13] = "\n $1<gd>$2</span>";
# $search [14] = "/Parsing \"(.*)\"/i";
# $replace [14] = "\n $parse <a\n $tar href = \"$file\">$1</a>\n $tcfg";
$search [15] = "/(Doctype given) (.*)/i";
$replace [15] = "\n <bk>$1$2</span>";
$search [16] = "/(Document content looks) (.*)/i";
$replace [16] = "\n <bk>$1$2</span>";
$search [17] = "/(bug reports to) (.*)/i";
$replace [17] = "$1<a\n $tar href = mailto:$2 >$2</a>";
$search [18] = "/($usecss) (CSS)/i";
$replace [18] = "$1<a\n $tar href = \"$doc_css\">$2</a>";
$search [19] = "/($looks) (HTML 4.01)/i";
$replace [19] = "$1<a\n $tar href = \"$doc_html\">$2</a>";
$search [20] = "/(about) (HTML Tidy)/i";
$replace [20] = "$1<a\n $tar href = \"$doc_tidy\">$2</a>";
$search [21] = "/($looks) (XHTML 1.0)/i";
$replace [21] = "$1<a\n $tar href = \"$doc_xhtml\">$2</a>";
# Highlight error lines:
$search [22] = "/(line) (\d+)/i";
$replace [22] = "$1<rd><b>$2</b></span>";
$search [23] = "/(column) (\d+)/i";
$replace [23] = "$1<rd><b>$2</b></span>";
$search [24] = "/(Warning:)/i";
$replace [24] = "\n <rd><b>$1</b></span>";
$search [25] = "/(Error:)/i";
$replace [25] = "\n <rd><b>$1</b></span>";
$search [26] = "/(.*) warnings.*!/i";
$replace [26] = "\n<rd><b>$1</b></span>";
$search [27] = "/(.*) n(o warnings.*)/i";
$replace [27] = "\n<gn><b>$1$2</b></span>";
# Read error file:
$return = " GMS: Running markup syntax checker ... \n\n ";
"\n <rd>Input read from</span>";
"<a\n $tar href = \"$srv_file\"> " . "$jobfile </a>\n";
if (file_exists ($GMS_BASE . ".err")) {
$content = x_file_get_contents ($GMS_BASE . ".err");
$return = $return . preg_replace ($search, $replace, $output . $content);
# Link to Tidy output:
if (file_exists ("$GMS_BASE.ok")) {
$return = $return . "Configuration read from " .
"<a\n $tar href = \"$SRV_SETTING/tidy.cfg\"> " . "tidy.cfg</a>\n ";
$return = $return . "\n <rd>Output written to</span>";
"<a\n $tar href = \" " . dirname ($srv_file) .
"/$base.ok\">$GMS_BASE.ok</a>\n\n ";
}
if (file_exists ($GMS_BASE . ".err")) {
return $return . GMS_runtime ($before, time ());}
}

function edit () {
global $GMS_FILE;
if (file_exists ($GMS_FILE)) {
return preg_replace ("/&/", "&amp;", x_file_get_contents ($GMS_FILE));}
}

function init () {
global $global_input;
global $GMS_BINARIES; global $GMS_LINKS;
global $GMS_PROCESS; global $GMS_ROOT;
global $GMS_SETTING; global $GMS_SHELL;
global $GMS_TSETTER;
global $header; global $language;
global $SRV_ENCODING; global $SRV_PROCESS;
global $SRV_SETTING; global $SRV_HANDBOOK;
$dir_enc = $SRV_ENCODING; $dir_hlp = $SRV_HANDBOOK;
$dir_process = $GMS_PROCESS; $dir_pro = $SRV_PROCESS;
$dir_set = $SRV_SETTING;
if ($GMS_LINKS == "internal") {
$doc_html = $SRV_HANDBOOK . "../html401/html401.htm";
$doc_pdf = $SRV_HANDBOOK . "../pdf/latex/pdf.tex.htm";
$doc_xhtml = $SRV_HANDBOOK . "../xhtml10/xhtml10.htm";}

```

```

else {
    $doc_html = "http://www.w3.org/TR/1999/REC-html401-19991224";
    $doc_pdf = "http://www.pdfTeX.org";
    $doc_xhtml = "http://www.w3.org/TR/xhtml1";
    clearstatcache ();
    $file = basename ($global_input ["file"]);
    if (is_dir ($GMS_BINARIES)) {chdir ($GMS_BINARIES);}
# Get basename:
$search [0] = "\.xhtml/"; $replace [0] = "";
$search [1] = "\.shtml/"; $replace [1] = "";
$search [2] = "\.html/"; $replace [2] = "";
$search [3] = "\.htm/"; $replace [3] = "";
$search [4] = "\.xml/"; $replace [4] = "";
$base = preg_replace ($search, $replace, $file);
putenv ("TELFONTS=" . $GMS_ROOT . "/fonts/tfm/");
putenv ("TEFORMATS=" . $GMS_BINARIES);
putenv ("TEMFONCF=" . $GMS_SETTING);
putenv ("TEXPPOOL=" . $GMS_BINARIES);
putenv ("VFFONTS=" . $GMS_ROOT . "/fonts/vf/");
putenv ("WEB2C=" . $GMS_SETTING);
# # Set path:
if (strtoupper (substr (PHP_OS, 0, 3)) == "WIN") {
    $dump = "\\dump";
    putenv ("TEXINPUTS=.:;$GMS_ROOT/data/;." .
        "$GMS_ROOT/tex/;$GMS_SETTING;$GMS_BINARIES");
    putenv ("TEXPSHEADERS=" . $GMS_ROOT . "/data/;" .
        "$GMS_ROOT . "/tex/;" . $GMS_SETTING . ";" . $GMS_BINARIES);}
else {
    $dump = "\\\\dump";
    putenv ("PATH=" . $ENV ["PATH"] . ";$GMS_SETTING;$GMS_BINARIES");
    putenv ("TEXINPUTS=.:;$GMS_ROOT/data/;." .
        "$GMS_ROOT/tex/;$GMS_SETTING;$GMS_BINARIES");
    putenv ("TEXPSHEADERS=" . $GMS_ROOT . "/data/;" .
        "$GMS_ROOT . "/tex/;" . $GMS_SETTING . ";" . $GMS_BINARIES);}
if (strtoupper (substr (PHP_OS, 0, 5)) == "WIN32") {
if (!file_exists ("$GMS_BINARIES/sort.exe")) {
@copy ($ENV ["windir"] . "command/sort.exe",
"$GMS_BINARIES/sort.exe");}
# Run TeX (and get run time):
$before = time ();
$arg = "echo $dump | \"$GMS_BINARIES/$GMS_TSETTER\" \" .
-ini *gerolf > \"$GMS_SETTING/gms.log\"";
set_time_limit ("300");
x_shell_exec ("set > $GMS_SETTING/gmsset.log, \"");
@unlink ("$GMS_SETTING/gmserr.log");
$output = shell_exec ($arg);
@unlink ("nil");
@unlink ("$GMS_SETTING/gerolf.log");
@copy ("$GMS_BINARIES/gerolf.log", "$GMS_SETTING/gerolf.log");
@unlink ("$GMS_BINARIES/gerolf.log");
# Link to log file:
$return = $language->language ("initializing") . "\n";
if (file_exists ("$GMS_SETTING/gerolf.log")) {
$content = $language->language ("done_details") .
"<a href = \"$SRV_SETTING/gerolf.log\" type = \"text/plain\" .
>gerolf.log</a>\n ";}
$return = $return . $output . $content;
# Move encoding files to data/enc directory;
if (is_dir ($GMS_BINARIES)) {
foreach (x_scandir ($GMS_BINARIES) as $entry) {
if (preg_match ("\enc/i", $entry)) {
@unlink ("$GMS_ROOT/data/enc/$entry");
@rename ("$GMS_BINARIES/$entry", "$GMS_ROOT/data/enc/$entry");
@unlink ("$GMS_BINARIES/$entry");}}
foreach (x_scandir ($GMS_BINARIES) as $entry) {
if (preg_match ("\xst/i", $entry)) {
@unlink ("$GMS_BINARIES/$entry");}}}
$return = $return . GMS_runtime ($before, time ());
if (file_exists ("$GMS_SETTING/gerolf.log")) {
$return = $return . $language->language ("create_or_open");}
return $return;}

function info () {
global $global_input;
global $language;
global $GMS_LINKS;
global $GMS_PROCESS;
if ($global_input ["file"] == "") {
$return = $language->language ("file_cleared");}
else {
if (($global_input ["process"] == "1") and ($GMS_LINKS == "internal")) {
$file = $global_input ["file"];}
else {$file = $GMS_PROCESS . "/" . $global_input ["process"] . "/" .
$global_input ["file"];}
if (file_exists ($file_)) {$return = $return .
$language->language ("file_contains") . " " .
filesize ($file) . " " .
$language->language ("file_characters") . " ,<br />";
if (is_readable ($file_)) {$return = $return .
$language->language ("file_readable") . " " .
if (is_writeable ($file_)) {$return = $return .
$language->language ("file_writeable");}
else {$return = $return .
$language->language ("file_protected");}}
else {$return = $return .
$language->language ("file_unreadable");}}
else {$return = $return .
$language->language ("file_not_there");}}
return $return;}

function read () {
global $global_input; global $header;
global $GMS_LINKS; global $GMS_PROCESS;
global $SRV_PROCESS;
clearstatcache ();
$file = basename ($global_input ["file"]);
if (($global_input ["process"] == "1") and ($GMS_LINKS == "internal")) {;
$GMS_FOLDER = dirname ($global_input ["file"]);}
else {$GMS_FOLDER = $GMS_PROCESS . "/" . $global_input ["process"];}
if (is_dir ($GMS_FOLDER)) {chdir ($GMS_FOLDER);}
$search [0] = "\.xhtml/"; $replace [0] = ".pdf";
$search [1] = "\.shtml/"; $replace [1] = ".pdf";
$search [2] = "\.html/"; $replace [2] = ".pdf";
$search [3] = "\.htm/"; $replace [3] = ".pdf";
$search [4] = "\.xml/"; $replace [4] = ".pdf";
$base = preg_replace ($search, $replace, $file);
$pdf = $GMS_FOLDER . "/" . $base;
$html = $GMS_FOLDER . "/" . $file;
if (file_exists ($html)) {
if (($global_input ["process"] != "") and (is_readable ($pdf))) {
$fp = fopen ($pdf, 'rb'); # open file in binary mode
header ("Content-Type: application/pdf");
header ("Content-Length: " . filesize ($pdf)); ###
return (fpassthru ($fp));}
else {
$file = $global_input ["file"];
$search [0] = "\.xhtml/"; $replace [0] = ".pdf";
$search [1] = "\.shtml/"; $replace [1] = ".pdf";
$search [2] = "\.html/"; $replace [2] = ".pdf";
$search [3] = "\.htm/"; $replace [3] = ".pdf";
$search [4] = "\.xml/"; $replace [4] = ".pdf";
$pdf = preg_replace ($search, $replace, $file);
return ($header->header ("HTML 4.01 Transitional", "", "nometa", "") .
"\n<body text = \"#00008B\">\n" .
"<p align = \"center\" class = \"center\">\n" .
" <br /><br /><br /><br /><br />\n" .
" <a href = \"\" . $SRV_PROCESS . "/" . $global_input ["process"] .
"/\" . $file . "\" target = \"_blank\">\n" .
" <b><big><span class = \"green\"> .
$global_input ["file"] . "</span></big></b></a><b> &rarr;\n" .

```

```

" [ <span class = \"green\">B</span> ]</b><br />\n" .
" <b>&darr;</b><br />\n" .
" <b>[ <span class = \"green\">T</span> ]</b><br />\n" .
" <b>&darr;</b><br />\n" .
" <b><big><span class = \"red\"><s> . $pdf . \"</s></span></big>\n" .
" &rarr; [ <span class = \"red\">R</span> ]</b>\n" .
"</p>\n</body>\n</html>");}}

function samples ($file_create) {
global $content;
global $GMS_TEMPLATE;
if ($file_create == "") {$content = "";}
else {$templ = $GMS_TEMPLATE . "/" . $file_create .
"/" . $file_create . ".htm"; # Pixme: accept index.htm(1) too
if (file_exists ($templ)) {
$content = x_file_get_contents ($templ);}}

# In Internet Explorer, set "Extra/Internet Options/Common/Temporary Internet
# Files/Settings/Search new versions of saved pages" to "Automatically":

function save ($file_save) {
global $GMS_FILE;
if (file_exists ($GMS_FILE)) {
x_file_put_contents ($GMS_FILE, stripslashes ($file_save));}}

function typeset () {
global $cwd; global $global_input; global $header;
global $GMS_BINARIES; global $GMS_LINKS;
global $GMS_PROCESS; global $GMS_ROOT;
global $GMS_SETTING; global $GMS_SHELL; global $GMS_TSETTER;
global $SRV_ENCODING; global $SRV_HANDBOOK;
global $SRV_PROCESS; global $SRV_SETTING;
$dir_enc_ = $SRV_ENCODING; $dir_hlp_ = $SRV_HANDBOOK;
$dir_process = $GMS_PROCESS; $dir_pro_ = $SRV_PROCESS;
$dir_set_ = $SRV_SETTING;
if ($GMS_LINKS == "internal") {
$doc_html = $SRV_HANDBOOK . "../html401/html401.htm";
$doc_pdf = $SRV_HANDBOOK . "../pdftex/pdftex.htm";
$doc_xhtml = $SRV_HANDBOOK . "../xhtml10/xhtml10.htm";}
else {
$doc_html = "http://www.w3.org/TR/1999/REC-html401-19991224";
$doc_pdf = "http://www.pdftex.org";
$doc_xhtml = "http://www.w3.org/TR/xhtml1";}
clearstatcache ();
$file = basename ($global_input ["file"]);
if (($global_input ["process"] == "1") and ($GMS_LINKS == "internal")) {
$GMS_FOLDER = dirname ($global_input ["file"]);}
else {$GMS_FOLDER = $GMS_PROCESS . "/" . $global_input ["process"];}
if (!is_writable ($GMS_FOLDER)) {echo "<pre> .
"<br /> GMS error: No writing permission for $GMS_FOLDER</pre>";}
if (is_dir ($GMS_FOLDER)) {chdir ($GMS_FOLDER);}

# Get basename:
$search [0] = "\.xhtml/"; $replace [0] = "";
$search [1] = "\.shtml/"; $replace [1] = "";
$search [2] = "\.html/"; $replace [2] = "";
$search [3] = "\.htm/"; $replace [3] = "";
$search [4] = "\.xml/"; $replace [4] = "";
$base = preg_replace ($search, $replace, $file);

# Set environment:
putenv ("GMS_FOLDER=" . $GMS_FOLDER);
putenv ("GMS_FILE=" . $global_input ["file"]);
putenv ("GMS_BASE=" . $base);
putenv ("GMS_SHORT=" . $global_input ["file"]);
putenv ("TELFONTS=" . $GMS_ROOT . "/fonts/tfm/");
putenv ("VFONTS=" . $GMS_ROOT . "/fonts/vf/");
putenv ("T1FONTS=" . $GMS_ROOT . "/fonts/type1/");
putenv ("TTFONTS=" . $GMS_ROOT . "/fonts/ttf/");
putenv ("TEXTFORMATS=" . $GMS_BINARIES);
putenv ("WEB2C=" . $GMS_SETTING);
putenv ("TEXMFCONF=" . $GMS_SETTING);

if (strtoupper (substr (PHP_OS, 0, 3)) == "WIN") {
putenv ("PATH=" . $GMS_SETTING . " . $GMS_BINARIES);
putenv ("TEXINPUTS=../" . $GMS_SETTING);
putenv ("TEXPSHEADERS=" . $GMS_SETTING . " .
$GMS_ROOT . "/data/enc;" . $GMS_ROOT . "/fonts/");}
else {
putenv ("PATH=" . $ENV ["PATH"] . " . $GMS_SETTING; $GMS_BINARIES");
putenv ("TEXINPUTS=../" . $GMS_ROOT . "/data/";
"GMS_ROOT/tex/"; $GMS_SETTING; $GMS_BINARIES");
putenv ("TEXPSHEADERS=" . $GMS_ROOT . "/data/";
"GMS_ROOT . "/tex/"; $GMS_SETTING . " . $GMS_BINARIES");}
# Run TeX (and get run time):
$before = time ();
$GMS_BASE = $base;
if (file_exists ($GMS_BASE . ".htm")) {
$jobfile = $GMS_BASE . ".htm";}
elseif (file_exists ($GMS_BASE . ".html")) {
$jobfile = $GMS_BASE . ".html";}
$arg = "\" . $GMS_BINARIES . "/$GMS_TSETTER\" -progrname=gerolf " .
$jobfile . " > gms.log";
set_time_limit ("300");
if (file_exists ($jobfile)) {$output = x_shell_exec ($arg, "");}
# Make log file better readable:
$gn = "span class = \"green\"";
$rd = "span class = \"red\""; $wt = "span class = \"grey\"";
$bl = "span class = \"blue\""; $bk = "span class = \"black\"";
$id = "Gerolf Markup Shredder"; $GDB = "G. D. Brettschneider";
$dom = "www.Gerolf.org"; $domGDB = "www.GDBrettschneider.de";
$address = "MarkupShredder@Gerolf.org";
$otpt = "Output written on";
$in = "\n <rd>Input read from </span>"; $out = "<rd>$otpt </span>";
$pro = $GMS_PROCESS . "/" . $global_input ["process"] . "/";
if (!is_dir ($pro)) {
$pro = dirname ($global_input ["file"]) . "/";}
if (strtoupper (substr (PHP_OS, 0, 3)) != "WIN") {
$pro = preg_replace ("|\$cwd|", "", $pro);}
$star = "target = \"_blank\"";
$typ = "type = \"application/pdf\"";
$ti = "This is";
# Display cosmetics:
##$search [0] = "/^/"; $replace [0] = " "; # indent line
$search [1] = "/\n/"; $replace [1] = "\n ";
$search [2] = "/< /"; $replace [2] = "<";
$search [3] = "/</"; $replace [3] = "&lt;";
$search [4] = "/(.) >/"; $replace [4] = ">";
### $search [5] = "/>/"; $replace [5] = ">";
$search [6] = "/\{-20,40\}/"; $replace [6] = "<$wt>$1</span>";
$search [7] = "/^ (-)/"; $replace [7] = "<$wt>$1</span>";
$search [8] = "/$id/"; $replace [8] =
"<a $star href = \"$dir_hlp/handbook.htm\"><rd>$id</span></a>";
### Time bandits:
### $search [9] = "/&lt;/"; $replace [9] = "<rd><b>&lt;/b></span><b>";
### $search [10] = "/&gt;/"; $replace [10] = "</span><rd><b>&gt;</b></span>";
$search [11] = "/(Web2C/i"; $replace [11] = "\n (Web2C";
$search [12] = "/(Format=gerolf)/i"; $replace [12] = "$1)";
$search [12] = "/e(entering extended mode)/i"; $replace [12] = "E$1";
$search [13] = "/\\write18 enabled.*\n/i"; $replace [13] = "";
$search [14] = "/(.*{}/"; $replace [14] = "{"; # 2nd mention of input file
$search [15] = "/\s\s\s/"; $replace [15] = "\s";
# Highlight error lines:
$search [16] = "/(1.\d+)/"; $replace [16] = "<rd><b>$1</b></span>";
$search [17] = "/(! [w:]|s|'|\"|=|.|\/|*|.| | |n|)/i";
$replace [17] = "<rd><b>$1</b></span>";
$search [18] = "/(Warning: )/i";
$replace [18] = "\n <rd><b>$1</b></span>";
$search [19] = "/(\?|s+)/i"; $replace [19] = " "; # no user interaction
$search [20] = "/(! Emergency.+)/i";
$replace [20] = "<rd><b>$1</b></span>";
$search [21] = "/(End of file.+)/i";

```



```

}

class frame {#=====

function left () {
    global $content; global $file;
    global $global_input; global $master;
    $state = $global_input ["left"];
    if ($state == "analyse") {$return = $master->analyse ();}
    elseif ($state == "browse") {$return = $master->browse ();}
    elseif ($state == "create") {$return = $master->create ();}
    elseif ($state == "download") {$return = $master->download ();}
    elseif ($state == "edit") {$return = $master->edit ();}
    elseif ($state == "first") {$return = $master->one_left ();}
    elseif ($state == "learn") {$return = $master->learn ();}
    elseif ($state == "read") {$return = $file->read ();}
    elseif ($state == "typeset") {$return = $master->typeset ();}
    elseif ($state == "view") {$return = $master->view ();}
    else {$return = $master->accept ();}
    return $return;}

function lower () {
    global $global_input; global $lower;
    if ($global_input ["lower"] == "first" ) {$return = $lower->one ();}
    else {$return = $lower->two ();}
    return $return;}

function master () {
    global $content; global $file;
    global $global_input; global $master;
    global $GMS_LINKS; global $GMS_SETTING;
    $state = $global_input ["master"];
    if ($state == "analyse") {$return = $master->analyse ();}
    elseif ($state == "backup") {
        if ($GMS_LINKS == "internal") {
            $backup = new backup; $return = $backup->run ();}
        }
    elseif ($state == "browse") {$return = $master->browse ();}
    elseif ($state == "create") {$return = $master->create ();}
    elseif ($state == "download") {$return = $master->download ();}
    elseif ($state == "edit") {$return = $master->edit ();}
    elseif ($state == "init") {
        if ($GMS_LINKS == "internal") {
            $plugin = new plugin; $return = $plugin->build ();
            $return = $return . $master->init ();}
        }
    elseif ($state == "learn") {$return = $master->learn ();}
    elseif ($state == "open") {
        if ($GMS_LINKS == "external") {$return = $master->open ();}
        else {$return = $master->open_internal ("");}
    }
    elseif ($state == "quit") {$return = $master->quit ();}
    elseif ($state == "read") {$return = $file->read ();}
    elseif ($state == "setup") {$return = $master->setup ();}
    elseif ($state == "typeset") {$return = $master->typeset ();}
    elseif ($state == "view") {$return = $master->view ();}
    elseif ($state == "write") {
        if ($GMS_LINKS == "internal") {
            $write = new write; $return = $write->run ();}
        }
    else {$return = $master->accept ();}
    return $return;}

function middle () {
    global $global_actions; global $global_input; global $middle;
    if ($global_input ["middle"] == "first") {
        $return = $middle->one ();}
    else {
        if ($global_input ["parent"] == "split") {
            array_push ($global_actions, "Join");}
        else {array_push ($global_actions, "Split");}
        $return = $middle->two ();}
    return $return;}

function page () {
    global $global_input;
    $frame = $global_input ["frame"];
    $return = "";
    if ($frame == "left") {$return = $this->left ();}
    elseif ($frame == "lower") {$return = $this->lower ();}
    elseif ($frame == "master") {$return = $this->master ();}
    elseif ($frame == "middle") {$return = $this->middle ();}
    elseif ($frame == "parent") {$return = $this->parent ();}
    elseif ($frame == "") {$return = $this->parent ();}
    elseif ($frame == "right") {$return = $this->right ();}
    elseif ($frame == "upper") {$return = $this->upper ();}
    else {$return = "<html><body><h1>GMS error: No such frame (" .
        $frame . ")<h1></body></html>";}
    return $return;}

function parent () {
    global $page; global $global_input; global $master;
    global $GMS_BANNER_X; global $GMS_BANNER_Y; global $GMS_VERSION;
    $nborder = "border = \"0\" frameborder = \"0\" framespacing = \"0\"";
    $border = "border = \"2\" frameborder = \"1\" framespacing = \"2\"";
    $return = "<html>\n" .
        "<head><title>Gerolf Markup Shredder $GMS_VERSION</title></head>";
    if ($global_input ["parent"] == "split") {
        $return = $return . "\n<frameset " .
            $nborder . " cols = \"*, *, $GMS_BANNER_X\">" .
            $page->frame ("left", "auto") .
            $page->frame ("right", "auto") . "\n <frameset " .
            $nborder . " rows = \"$GMS_BANNER_Y, *, 13%\">" .
            $page->frame ("upper", "no") .
            $page->frame ("middle", "auto") .
            $page->frame ("lower", "auto") . "\n</frameset>\n</frameset>";}
    else {$return = $return . "\n<frameset " .
        $nborder . " cols = \"*, $GMS_BANNER_X\">" .
        $page->frame ("master", "auto") . "\n <frameset " .
        $nborder . " rows = \"$GMS_BANNER_Y, *, 13%\">" .
        $page->frame ("upper", "no") .
        $page->frame ("middle", "auto") .
        $page->frame ("lower", "auto") . "\n </frameset>\n</frameset>";}
    $return = $return . "\n</frameset>\n" .
        $master->download_no_header () . "</noframes>\n</html>";
    return $return;}

function right () {
    global $content;
    global $file;
    global $global_input;
    global $GMS_LINKS;
    global $master;
    $state = $global_input ["right"];
    if ($state == "accept" ) {$return = $master->accept ();}
    elseif ($state == "analyse") {$return = $master->analyse ();}
    elseif ($state == "browse") {$return = $master->browse ();}
    elseif ($state == "create") {$return = $master->create ();}
    elseif ($state == "download") {$return = $master->download ();}
    elseif ($state == "edit") {$return = $master->edit ();}
    elseif ($state == "first") {
        if ($GMS_LINKS == "external") {$return = $master->download ();}
        else {$return = $master->one_right ();}
    }
    elseif ($state == "learn") {$return = $master->learn ();}
    elseif ($state == "read") {$return = $file->read ();}
    elseif ($state == "typeset") {$return = $master->typeset ();}
    elseif ($state == "view") {$return = $master->view ();}
    else {$return = $master->download ();}
    return $return;}

function upper () {
    global $global_input;
    global $GMS_ROOT;
    global $master;
    $state = $global_input ["upper"];
    if ($state == "wait") {$return = $master->wait ();}

```

```

# else {$return = "<html><body style = \"margin: 0\">\n" .
#   "<img src = \".doc/rotation/180x150y.gif\" />\n" .
#   "</body></html>";}
else {
  if (is_dir ($GMS_ROOT . "doc/rotation")) {
    $rotations = array ("");
    foreach (x_scandir ($GMS_ROOT . "doc/rotation") as $entry) {
      if (($entry != ".") and ($entry != "..") and ($entry != "_ignore_")) {
        if (preg_match ("/\.htm.*\/i", $entry)) {
          array_push ($rotations, $entry);
          $rot_number ++;}}
    $file = $rotations [array_rand ($rotations)];
    $file = $GMS_ROOT . "doc/rotation/" .
      $rotations [array_rand ($rotations)];
    if (is_file ($file)) {$return = x_file_get_contents ($file);}}
  return $return;}
}

class header {#=====
function header ($flavour, $title, $meta, $style) {
  # $flavour e. g. "XHTML 1.0 Strict"
  # $title e. g. "Hallo World"
  # $meta e. g. "nometa"
  # $style e. g. "nostyle"
  global $global_input; global $title;
  $declaration =
    "<?xml version = \"1.0\" encoding = \"windows-1252\"?>\n\n";
  $namespace = " xmlns = \"http://www.w3.org/1999/xhtml\" \n" .
    " xml:lang = \"en\" lang = \"en\"";
  # Doctype:
  $doctype = "<!DOCTYPE html PUBLIC";
  if ($flavour == "XHTML 1.0 Strict") {$doctype = $doctype .
    " \"/W3C/DTD XHTML 1.0 Strict/EN\"
    \"DTD/xhtml1-strict.dtd\">\n\n";}
  elseif ($flavour == "XHTML 1.0 Frameset") {$doctype = $doctype .
    " \"/W3C/DTD XHTML 1.0 Frameset/EN\"
    \"DTD/xhtml1-frameset.dtd\">\n\n";}
  elseif ($flavour == "XHTML 1.0 Transitional") {$doctype = $doctype .
    " \"/W3C/DTD XHTML 1.0 Transitional/EN\"
    \"DTD/xhtml1-transitional.dtd\">\n\n";}
  elseif ($flavour == "HTML 4.01 Strict") {$declaration = "";
    $namespace = "";
    $doctype = $doctype . " \"/W3C/DTD HTML 4.01 Strict/EN\">\n\n";}
  elseif ($flavour == "HTML 4.01 Frameset") {$declaration = "";
    $namespace = ""; $doctype = $doctype .
    " \"/W3C/DTD HTML 4.01 Frameset/EN\">\n\n";}
  elseif ($flavour == "HTML 4.01 Transitional") {$declaration = "";
    $namespace = ""; $doctype = $doctype .
    " \"/W3C/DTD HTML 4.01 Transitional/EN\">\n\n";}
  elseif ($flavour == "HTML 3.2") {$declaration = "";
    $namespace = ""; $doctype = $doctype .
    " \"/W3C/DTD HTML 3.2/EN\">\n\n";}
  elseif ($flavour == "HTML 2.0") {$declaration = "";
    $namespace = ""; $doctype = $doctype .
    " \"/W3C/DTD HTML 2.0/EN\">\n\n";}
  else {$declaration = ""; $namespace = ""; $doctype = "";}
# Meta information:
if ($meta == "nometa") {$meta = "";}
# Link to style sheet:
if ($style == "") {$style = "
<link rel = \"stylesheet\" type = \"text/css\" .
  \" href = \"\" . # setting_SRV_STYLE () . \"/\" .
  \"gerolf.css\" />\n\n";}
elseif ($style == "nostyle") {$style = "";}
else {$style = "
<link rel = \"stylesheet\" type = \"text/css\" href = \"\"
  . \"/style\" />\n\n";}
return $declaration . $doctype . "<html$namespace>\n\n<head>\n
<title>$title</title>\n" . $meta . $style . "\n</head>\n\n";}
}

class language {#=====
var $pool_english = array (
  "language_name" => "English",
  "button_accept" => "Accept",
  "button_accept_" => "Accept modifications and go on",
  "button_action" => "Action",
  "button_begin" => "Begin",
  "button_select" => "Select",
  "button_select_" => "Select a few system properties",
  "button_analyse" => "Analyse",
  "button_analyse_" => "Analyse hypertext markup syntax",
  "button_browse" => "Browse",
  "button_browse_" => "Browse through the markup file (*.htm)",
  "button_download" => "Download",
  "button_download_" => "Download Gerolf Markup Shredder",
  "button_create" => "Create",
  "button_create_" => "Create a new markup file (*.htm)",
  "button_edit" => "Edit",
  "button_edit_" => "Edit the markup file (*.htm)",
  "button_help" => "Help",
  "button_init" => "Init",
  "button_init_" => "Init GMS format",
  "button_join" => "Join",
  "button_join_" => "Join workspaces",
  "button_learn" => "Learn",
  "button_learn_" => "Learn to create portable documents with Markup Shredder",
  "button_open" => "Open",
  "button_open_" => "Open an existing markup file (*.htm)",
  "button_open_hint" => "'For security reasons most browsers delete your ' .
    'file selection when calling this side.'",
  "button_open_hint_" => "You can use the back button to come here again.",
  "button_open_left" => "Open a markup file <br />example (*.htm)",
  "button_open_picture" => "subfiles to be included",
  "button_open_right" => "Open an existing markup file (*.htm)",
  "button_quit" => "Quit",
  "button_quit_" => "Quit the current markup file",
  "button_read" => "Read",
  "button_read_" => "Read the portable document (*.pdf)",
  "button_reset" => "Reset",
  "button_reset_" => "Discard all modifications",
  "button_save" => "Save",
  "button_save_" => "Save the markup file (*.htm)",
  "button_setting" => "Setting",
  "button_split" => "Split",
  "button_split_" => "Split workspace",
  "button_typeset" => "Typeset",
  "button_typeset_" => "Typeset a portable document according to the ' .
    'hypertext markup'",
  "button_view" => "View",
  "button_view_" => "View the markup file (*.htm)",
  "button_write" => "Write",
  "button_write_" => "Write GMS font map",
  "codepage" => "Codepage",
  "create_or_open" => "
    Now you can <b>create</b> or <b>open</b> HTML documents.
    Press button <b>[C]</b> or <b>[O]</b> to continue ...",
  "done_details" => " Done. Details: see ",
  "extraction_failed" => "GMS: Extraction of archives failed.",
  "extrafonts" => "Extra fonts",
  "file_characters" => "letters of text",
  "file_cleared" => "There is no markup file (*.htm) given at present",
  "file_contains" => "contains",
  "file_is_no_text" => "does not contain text",
  "file_not_there" => "does not exist yet",
  "file_protected" => "but write-protected",
  "file_readable" => "is readable",
  "file_unreadable" => "can not be read",
  "file_writeable" => "and can be modified",
}

```

```

"initializing" => " GMS: Initializing TeX format file ...",
"language_" => "In English, please!",
"language_abbreviation" => "en",
"own_file_" => "Open own markup file",
"running" => "GMS: Running ...",
"runtime" => "GMS run time",
"sample" => "Example",
"sample_" => "Show example",
"select_area_" => "Workspace",
"select_files_" => "Maximum <i>number of '
'files</i><br />allowed for transfer",
"select_font" => "Size",
"select_height" => "Height",
"select_joined_" => "joined",
"select_size_" => "Maximum <i>total size of '
'files</i><br />allowed for transfer<br />(number of characters)",
"select_split" => "split",
"select_width" => "Width",
"setup_init" => "
GMS setup: The TeX format file must be
<b>initialized</b>. This procedure usually
lasts several seconds.
The codepage must be the same as for
writing the TeX font map.
Press button <b>[I]</b> to continue and
please wait for the next message ...",
"setup_write_1" => "
GMS setup: The TeX font map must be <b>written</b>
while computing font metrics. This procedure
usually lasts several <b>minutes</b> up to one hour
without giving you any feedback.
Codepage 1252 (West Europe) is required
for typesetting the template documents.
To use additional fonts with GMS, input
the corresponding source folder,
for example ",
"setup_write_2" => "
Press button <b>[W]</b> to continue and
please wait for the next message ...",
"setup_unzip_en" => "unzip: [E]
The archives must be extracted.
This procedure usually lasts several seconds.
Press any button to start and please wait
for the next message ...",
"setup_unzip_de" => "unzip: [D]
Die Archivdateien m&uuml;ssen entpackt werden.
Dieser Vorgang dauert gew&uuml;hnlich mehrere Sekunden.
Dr&uuml;cken Sie einen Knopf zum Starten und
warten Sie bitte auf die n&uuml;chste Meldung ...",
"template" => "Template",
"template_" => "Choose template",
"template_myfile_" => "File name",
"welcome_welcome" => "Welcome",
"welcome_to" => "to",
"welcome_version" => "Version",
"welcome_enjoy" => "Enjoy!",
"welcome_hint" => "This is just a 'well-formed markup list'...",
"writing" => " GMS: Writing TeX font map ...");

var $pool_german = array (
"language_name" => "Deutsch",
"button_accept" => "Annehmen",
"button_accept_" => "&Auml;nderungen &uuml;bernehmen und fortfahren",
"button_action" => "Befehl",
"button_analyse" => "Pr&uuml;fen",
"button_analyse_" => "Satzbau der Auszeichnungsmarken pr&uuml;fen",
"button_begin" => "Anfangen",
"button_browse" => "Uml;berfliegen",
"button_browse_" => "Satzmarkendatei (*.htm) im Brauser &uuml;berfliegen",
"button_create" => "Erzeugen",
"button_create_" => "Neue Satzmarkendatei (*.htm) erzeugen",
"button_download" => "Herunterladen",
"button_download_" => "GMS aus dem Netz herunterladen",
"button_edit" => "Bearbeiten",
"button_edit_" => "Satzmarkendatei (*.htm) bearbeiten",
"button_help" => "Hilfe",
"button_init" => "R&uuml;cksetzen",
"button_init_" => "GMS-Format R&uuml;cksetzen",
"button_join" => "Vereinigen",
"button_join_" => "Arbeitsbereiche vereinigen",
"button_learn" => "Lernen",
"button_learn_" => "
Lernen, wie man mit GMS &uuml;bertragbare Schriftst&uuml;cke erstellt",
"button_open" => "&Ouml;ffnen",
"button_open_" => "Bestehende Satzmarkendatei (*.htm) &uuml;ffnen",
"button_open_hint" => "'Aus Sicherheitsgr&uuml;nden l&uuml;schen die '
'meisten Brauser Ihre Dateiauswahl bei erneutem Aufruf dieser Seite.'",
"button_open_hint_" => "'Verwenden Sie gegebenenfalls die R&uuml;cktaste, '
'um wieder hierher zu gelangen.'",
"button_open_left" => "
&Ouml;ffnen einer Beispiel-<br />Satzmarkendatei (*.htm)",
"button_open_picture" => "einzuf&uuml;gende Unterdateien",
"button_open_right" => "
&Ouml;ffnen einer bestehenden Satzmarkendatei (*.htm)",
"button_quit" => "Beenden",
"button_quit_" => "Arbeit an dieser Satzmarkendatei (*.htm) beenden",
"button_read" => "Lesen",
"button_read_" => "Uml;bertragbares Schriftst&uuml;ck (*.pdf) lesen",
"button_reset" => "R&uuml;cksetzen",
"button_reset_" => "Alle &Auml;nderungen verwerfen",
"button_save" => "Sichern",
"button_save_" => "Satzmarkendatei (*.htm) sichern",
"button_select" => "Ausw&uuml;hlen",
"button_select_" => "'Einige Eigenschaften der '
'Arbeitsumgebung ausw&uuml;hlen'",
"button_setting" => "Einrichtung",
"button_split" => "Teilen",
"button_split_" => "Arbeitsbereich teilen",
"button_typeset" => "Setzen",
"button_typeset_" => "
Schriftst&uuml;ck gem&uuml;ssig; der Auszeichnungsmarken setzen",
"button_view" => "Betrachten",
"button_view_" => "Satzmarkendatei (*.htm) betrachten",
"button_write" => "Schreiben",
"button_write_" => "GMS-Schriftartenliste schreiben",
"codepage" => "Zeichensatz",
"create_or_open" => "
Sie k&uuml;nnen jetzt HTML-Schriftst&uuml;cke
<b>erzeugen</b> oder <b>&uuml;ffnen</b>.
Dr&uuml;cken Sie zum Fortsetzen Knopf <b>[C]</b> oder <b>[O]</b> ...",
"done_details" => " Fertig. Einzelheiten siehe ",
"extraction_failed" => "GMS: Das Entpacken der Archive ist fehlgeschlagen.",
"extrafonts" => "Zusatzschriften",
"file_characters" => "Buchstaben",
"file_cleared" => "Keine Satzmarkendatei (*.htm) gegeben",
"file_contains" => "enth&uuml;t",
"file_is_no_text" => "enth&uuml;t nichts Lesbares",
"file_not_there" => "gibt es noch nicht",
"file_protected" => "ist aber schreibgesch&uuml;tzt",
"file_readable" => "kann gelesen werden",
"file_unreadable" => "ist nicht lesbar und",
"file_writeable" => "und ist &uuml;nderbar",
"initializing" => " GMS: R&uuml;cksetzen der TeX-Formatierungsdatei ...",
"language_" => "Auf Deutsch, bitte!",
"language_abbreviation" => "de",
"own_file_" => "Eigene Satzmarkendatei &uuml;ffnen",
"running" => "GMS: L&uuml;uft ...",
"runtime" => "GMS-Laufzeit",
"sample" => "Beispiel",
"sample_" => "Beispiel anzeigen",
"select_area_" => "Arbeitsbereich",
"select_files_" => "'H&uuml;chstzul&uuml;ssige '

```

```

    '<i>Anzahl</i><br />hinaufladbarer Dateien"',
"select_font" => "Gr&ouml;&szlig;e",
"select_height" => "H&ouml;he",
"select_joined_" => "vereinigt",
"select_size_" => "H&ouml;chstzul&ouml;ssige ".
    '<i>Gesamtgr&ouml;&szlig;e</i><br />hinaufladbarer Dateien (Zeichenzahl)"',
"select_split_" => "geteilt",
"select_width" => "Breite</i>",
"setup_init" => "
    GMS-Einrichtung: Die TeX-Formatierungsdatei
    muss <b>initialisiert</b> werden. Dieser Vorgang
    dauert gew&ouml;hnlich mehrere Sekunden.
    Der Zeichensatz muss derselbe sein wie
    beim Schreiben der Schriftenliste.
    Dr&uuml;cken Sie zum Fortsetzen Knopf <b>[I]</b> und
    warten Sie bitte auf die n&auml;chste Meldung ...",
"setup_write_1" => "
    GMS-Einrichtung: Die Buchstabenma&szlig;e m&uuml;ssen berechnet
    und die TeX-Schriftenliste muss in die Datei \"font.map\"
    <b>geschrieben</b> werden. Dieser Vorgang beansprucht
    gew&ouml;hnlich mehrere <b>Minuten</b> bis zu einer Stunde,
    ohne Ihnen eine R&uuml;ckmeldung zu geben. Der
    Zeichensatz 1252 (Westeuropa) wird zum \"Setzen\"
    (zur PDF-Umwandlung) der HTML-Vorlagendateien
    ben&ouml;tigt. Um weitere Schriften mit GMS zu benutzen,
    tragen Sie das entsprechende Quellverzeichnis
    ein, beispielsweise ",
"setup_write_2" => "
    Dr&uuml;cken Sie zum Fortsetzen Knopf <b>[W]</b> und warten
    Sie bitte auf die n&auml;chste Meldung ...",
"template_" => "Vorlage",
"template_" => "Vorlage ausw&auml;hlen",
"template_myfile_" => "Dateiname",
"welcome_welcome" => "Willkommen",
"welcome_to" => "bei",
"welcome_version" => "Fassung",
"welcome_enjoy" => "Viel Vergn&uuml;gen!",
"welcome_hint" => "Dies ist eine 'wohlgeformte Auszeichnungsliste'...",
"writing" => "    GMS: TeX-Schriftenliste wird geschrieben ...";

function language ($index) {
    global $global_input;
    if ($global_input ["language"] == "German") {
        $return = $this->pool_german [$index];
    } else {
        $return = $this->pool_english [$index];
    }
    return $return;
}

class lower {#=====

function form () {
    global $page; global $GMS_SCRIPT; global $SRV_HANDBOOK;
    $return = "\n\n<form method = \"post\" target = \"_parent\" ".
    "action = \"$GMS_SCRIPT\"> \n". $page->fields ().
    "\n <table summary = \"Language\" width = \"100%\" ".
    "cellpadding = \"0\" cellspacing = \"0\">\n".
    " <tr class = \"button\">\n".
    " <td class = \"button\" align = \"center\" \n".
    " width = \"50%\" style = \"width: 50%\">\n".
    " <input type = \"submit\" value = \" E \" \n".
    " name = \"English\" title = \"en\" /></td>\n".
    " <td class = \"buttonhelp\" align = \"left\" \n".
    " width = \"50%\" style = \"width: 50%\">\n".
    " <a href = \"$SRV_HANDBOOK/handbook.htm#learn\" \n".
    " target = \"left\">English</a></td></tr>\n".
    " <tr class = \"button\">\n".
    " <td class = \"button\" align = \"center\" \n".
    " width = \"50%\" style = \"width: 50%\">\n".
    " <input type = \"submit\" value = \" D \" \n".
    " name = \"German\" title = \"de\" /></td>\n".
    " <td class = \"buttonhelp\" align = \"left\" \n".
    " width = \"50%\" style = \"width: 50%\">\n".
    " <a href = \"$SRV_HANDBOOK/handbuch.htm#learn\" \n".
    " target = \"left\">Deutsch</a></td></tr>\n".
    " </table>\n\n".
    "</form>\n\n";
    return $return;
}

function one () {
    global $header; global $global_input;
    $global_input ["lower"] = "second";
    $global_input ["middle"] = "second";
    $return = $header->header ("HTML 4.01 Transitional", "", "nometa", "").
    "\n<body class = \"middle_second\">". $this->form (). "</body>\n\n</html>";
    return $return;
}

function two () {
    global $header;
    $return = $header->header ("HTML 4.01 Transitional", "", "nometa", "").
    "\n<body class = \"middle_second\">". $this->form (). "</body>\n\n</html>";
    return $return;
}

}

class master {#=====

function accept () {
    global $cwd; global $file;
    global $header; global $global_input;
    global $SRV_PROCESS;
    $file_ = $SRV_PROCESS . "/" .
    $global_input ["process"] . "/" . $global_input ["file"];
    if (!file_exists ($file_)) {$file_ = $global_input ["file"];}
    if (file_exists ($file_)) {
        $file_ = preg_replace ("|\"$cwd|", "", $file_);
        $return = $header->header ("HTML 4.01 Transitional", "", "nometa", "").
        "\n<body text = \"#00008B\">\n".
        "<p align = \"center\" class = \"center\">".
        "<br /><br /><br /><br />".
        "<a href = \"\" . $file_ . \"\" target = \"_blank\"> ".
        "<b><b></b></b>". basename ($global_input ["file"]) .
        "</b></b></a><br /><br />".
        $file->info ().
        "</p>\n\n</body>\n\n</html>";
        return $return;
    }
}

function analyse () {
    global $file; global $header; global $global_input;
    $pt = "pt";
    if ($global_input ["parent"] == "join") {
        $font_size = $global_input ["size_one"]; $width = "50%";
    } else {
        $font_size = $global_input ["size_two"]; $width = "100%";
    }
    $fontstyle = "font-size: $font_size$pt; font-family: monospace;";
    $return = $header->header ("HTML 4.01 Transitional", "", "nometa", "").
    "\n<body text = \"#00008B\">\n\n<br />\n\n".
    "<pre style = \"$fontstyle\">\n". $file->analyse (). "\n</pre>\n\n".
    "</body>\n\n</html>";
}

function browse () {
    global $cwd; global $file; global $global_input;
    global $SRV_PROCESS;
    $file_ = $SRV_PROCESS . "/" .
    $global_input ["process"] . "/" . $global_input ["file"];
    if (!file_exists ($file_)) {$file_ = $global_input ["file"];} ##
    if (file_exists ($file_)) {
        if (strtoupper (substr (PHP_OS, 0, 3)) != "WIN") {
            $file_ = preg_replace ("|\"$cwd|", "", $file_);
            return "<html>\n".
            " <meta http-equiv = \"refresh\" \n".
            " content = \"0; URL = \" . $file_ . \"\" />\n\n</html>";
        } else {
            return "GMS error: No file $file_";
        }
    }
}

```



```

function create () {
    global $header; global $language; global $page;
    global $GMS_BINARIES; global $GMS_SCRIPT;
    global $GMS_SETTING; global $GMS_TEMPLATE;
    srand ((double) microtime () * 1000000);
    $secret = rand (10000000, 99999999);
    $tp = "template_"; $op = "option";
    $mozilla_prop = " height = \"94%\"";
    $return = $header->header ("HTML 4.01 Transitional", "", "nometa", "") .
    "\n<body class = \"left_first\" text = \"#00008B\">\n
    <form method = \"post\" target = \"_parent\" action = \"\$GMS_SCRIPT\">\n" .
    $page->fields () .
    "\n <table style = \"height: 94%;\" width = \"100%\"
    summary = \"Select a template to create a new file\"
    cellpadding = \"0\" cellspacing = \"0\">\n\n <tr>
    <td width = \"100%\" height = \"100%\"
    align = \"center\" valign = \"middle\">
    <p>&nbsp;</p>
    <p><b>\" . $language->language ("button_create") . "</b></p>
    <p>\" . $language->language ("template_myfile") . " :</p>
    <p>\n <input type = \"input\" name = \"file_no_1\"
    value = \"\$secret.htm\" size = \"30\" maxlength= \"30\" />
    </p>\n <p>\" . $language->language ("template") . " :</p>
    <p>
    <select name = \"file_create\" size = \"12\"
    title = \"\" . $language->language ($tp) . "\">;
# Get templates:
$i = "0";
$dir_template = $GMS_TEMPLATE;
if (!is_dir ($dir_template)) {x_mkdir ($dir_template);}
$handle = opendir ($dir_template);
$templates = array ("");
while ($dirname = readdir ($handle)) {
    if (is_dir ($dir_template . "/" . $dirname) and
    $dirname != "." and $dirname != "..") {
        $templates [$dirname] = $dirname;}
closedir ($handle);
ksort ($templates);
foreach ($templates as $name) {# Fixme: use array
# if ($name == "default") {
# $op = "option selected = \"selected\" value = \"default\"";
# if ($name == "lclef") {
# $op = "option selected = \"selected\" value = \"lclef\"";
# elseif ($name == "adobe") {$op = "0";}
# elseif ($name == "amaya") {$op = "0";}
# elseif ($name == "apache") {$op = "0";}
# elseif ($name == "arabic") {$op = "0";}
# elseif ($name == "book") {$op = "0";}
# elseif ($name == "booklet") {$op = "0";}
# elseif ($name == "borowitz") {$op = "0";}
# elseif ($name == "context") {$op = "0";}
# elseif ($name == "cpan") {$op = "0";}
# elseif ($name == "ctan") {$op = "0";}
# elseif ($name == "css1") {$op = "0";}
# elseif ($name == "css2") {$op = "0";}
# elseif ($name == "cyrillic") {$op = "0";}
# elseif ($name == "dante") {$op = "0";}
# elseif ($name == "gentle") {$op = "0";}
# elseif ($name == "gnu") {$op = "0";}
# elseif ($name == "goldhtml") {$op = "0";}
# elseif ($name == "greek") {$op = "0";}
# elseif ($name == "guide") {$op = "0";}
# elseif ($name == "handbook") {$op = "0";}
# elseif ($name == "hebrew") {$op = "0";}
# elseif ($name == "hixie") {$op = "0";}
# elseif ($name == "html401") {$op = "0";}
# elseif ($name == "htmltidy") {$op = "0";}
# elseif ($name == "latex") {$op = "0";}
# elseif ($name == "latin") {$op = "0";}
# elseif ($name == "indian") {$op = "0";}
elseif ($name == "ms-typo") {$op = "0";}
elseif ($name == "pdftex") {$op = "0";}
elseif ($name == "primer") {$op = "0";}
elseif ($name == "rotation") {$op = "0";}
elseif ($name == "script") {$op = "0";}
elseif ($name == "tds") {$op = "0";}
# elseif ($name == "thai") {$op = "0";}
elseif ($name == "tug") {$op = "0";}
elseif ($name == "universe") {$op = "0";}
elseif ($name == "unicode") {$op = "0";}
elseif ($name == "unichart") {$op = "0";}
# elseif ($name == "vietnam") {$op = "0";}
elseif ($name == "w3c") {$op = "0";}
elseif ($name == "w3c_css") {$op = "0";}
elseif ($name == "w3c_html") {$op = "0";}
elseif ($name == "xhtml10") {$op = "0";}
elseif ($name == "xhtmlbas") {$op = "0";}
else {$op = "option";}
if ($name == "") {$op = "0";}
if ($op != "0") {$return = $return .
    "\n <$op>\" . $name . "</option>";}
$return = $return . "\n </select>\n </p>
    <p>\n <input type = \"submit\" name = \"accept\" value = \"\" .
    $language->language ("button_accept") . "\"
    title = \"\" . $language->language ("button_accept") . "\" />
    </p>\n </td>
    </tr>\n\n </table>\n\n</form>\n\n</body>\n\n</html>;
if (!is_file ("\$GMS_SETTING/font.map")) {###
$return = $language->language ("setup_write");}
elseif (!is_file ("\$GMS_BINARIES/gerolf.efmt")) {
$return = $language->language ("setup_init");}
return $return;}

function download () {
    global $header;
    return $header->header ("HTML 4.01 Transitional", "", "", "") . "\n" .
    "<body text = \"#00008B\">\n\n" . $this->download_no_header () .
    "</body>\n\n</html>;"}

function download_no_header () {
    global $header; global $global_input;
    global $GMS_ROOT; global $SRV_HANDBOOK;
    $return = $header->header ("HTML 4.01 Transitional", "", "", "") . "\n" .
    "#<body text = \"#00008B\">\n\n" .
    " <table summary = \"\" style = \"height: 90%\" width = \"95%\" \n" .
    " cellpadding = \"0\" cellspacing = \"0\" \n" .
    " border = \"0\" align = \"center\">\n" .
    " <tr>\n" .
    " <td height = \"70%\" align = \"center\" valign = \"middle\">\n" .
    " <a target = \"_blank\" href = \n" .
    " \"http://www.GDBrettschneider.de\";
# \"
    \"/doc/handbook/license.htm\" \n";
if (is_file ("\$GMS_ROOT/doc/rotation/gerolf.gif")) {$return = $return .
    " ><img src = \"/doc/rotation/gerolf.gif\" \n" .
    " alt = \"G. D. Brettschneider\" title = \n" .
    " \"Please support the author of Markup Shredder\"</a><br />\n";}
$return = $return .
    " <br />\n" .
    " <big><big><big><a target = \"_blank\" \n" .
    " style = \"text-decoration: none\" \n" .
    " href = \"\" . $SRV_HANDBOOK . "/download.htm\" \n" .
    " title = \"GMS download and installation instruction\"><b \n" .
    " class = \"blue\">Download</b></a>\n" .
    " <a target = \"_blank\" href = \"/gerolf.exe\" \n" .
    " style = \"text-decoration: none\" title = \n" .
    " \"GMS archive, 12MB, for Windows-32, with Linux binaries\"><b \n" .
    " class = \"gold\"><i>Gerolf</i></b><span \n" .
    " class = \"blue\">.exe</span></a><br />\n" .
    " and run <b class = \"green\"><i>Markup</i></b>\n" .

```



```

$tidy_p = $SRV_HANDBOOK . "/./htmltidy/htmltidy.pdf";
$prim_h = $SRV_HANDBOOK . "/./primer/primer.htm";
$prim_p = $SRV_HANDBOOK . "/./primer/primer.pdf";
$html_h = $SRV_HANDBOOK . "/./html401/html401.htm";
$html_p = $SRV_HANDBOOK . "/./html401/html401.pdf";
$xht1_h = $SRV_HANDBOOK . "/./xhtml10/xhtml10.htm";
$xht1_p = $SRV_HANDBOOK . "/./xhtml10/xhtml10.pdf";
$css1_h = $SRV_HANDBOOK . "/./css1/css1.htm";
$css1_p = $SRV_HANDBOOK . "/./css1/css1.pdf";
$css2_h = $SRV_HANDBOOK . "/./css2/css2.htm";
$css2_p = $SRV_HANDBOOK . "/./css2/css2.pdf";
else {
$hdbk_h = "http://www.Gerolf.org/doc/handbook/handbook.htm";
$hdbk_p = "http://www.Gerolf.org/doc/handbook/handbook.pdf";
$guid_h = "http://www.w3.org/MarkUp/Guide";
$guid_p = $SRV_HANDBOOK . "/./guide/guide.pdf";
$tidy_h = "http://www.w3.org/People/Raggett/tidy";
$tidy_p = $SRV_HANDBOOK . "/./htmltidy/htmltidy.pdf";
$prim_h = "http://archive.ncsa.uiuc.edu/General/Internet/WWW";
$prim_p = $SRV_HANDBOOK . "/./primer/primer.pdf";
$html_h = "http://www.w3.org/TR/html401";
$html_p = $SRV_HANDBOOK . "/./html401/html401.pdf";
$xht1_h = "http://www.w3.org/TR/xhtml1";
$xht1_p = $SRV_HANDBOOK . "/./xhtml10/xhtml10.pdf";
$css1_h = "http://www.w3.org/TR/REC-CSS1";
$css1_p = $SRV_HANDBOOK . "/./css1/css1.pdf";
$css2_h = "http://www.w3.org/TR/REC-CSS2";
$css2_p = $SRV_HANDBOOK . "/./css2/css2.pdf";
$return = $header->header ("HTML 4.01 Transitional", "", "", "").
"\n<body text = \"#00008B\">\n\n".
" <table summary = \"\" style = \"height: 100%\" width = \"95%\">\n".
" cellpadding = \"0\" cellspacing = \"0\">\n".
" border = \"0\" align = \"center\">\n <tr>\n".
" <td width = \"95%\" height = \"10%\">\n".
" align = \"center\" valign = \"middle\">\n".
" <b> . $language->language ("button_learn_") . "</b></td>\n".
" </tr>\n <tr>\n".
" <td height = \"10%\" align = \"center\" valign = \"middle\">\n".
" <i>G. D. Brettschneider:</i><br />\n".
" <a href = \"$hdbk_h\">\n".
" target = \"_blank\">handbook.htm</a>&nbsp;&nbsp;&nbsp;|\n".
" <a href = \"$hdbk_p\">\n".
" target = \"_blank\">handbook.pdf</a>&nbsp;&nbsp;&nbsp;#8211;\n".
" The Gerolf Markup Shredder Handbook<br /><br />\n".
" <i>Dave Raggett:</i><br />\n".
" <a href = \"$guid_h\">\n".
" target = \"_blank\">guide.htm</a>&nbsp;&nbsp;&nbsp;|\n".
" <a href = \"$guid_p\">\n".
" target = \"_blank\">guide.pdf</a>&nbsp;&nbsp;&nbsp;#8211;\n".
" Getting started with HTML<br />\n".
" <a href = \"$tidy_h\">\n".
" target = \"_blank\">htmltidy.htm</a>&nbsp;&nbsp;&nbsp;|\n".
" <a href = \"$tidy_p\">\n".
" target = \"_blank\">htmltidy.pdf</a>&nbsp;&nbsp;&nbsp;#8211;\n".
" Clean up your Web Pages with HTML Tidy<br /><br />\n".
" <i>National Center for Supercomputing Applications:</i><br />\n".
" <a href = \"$prim_h\">\n".
" target = \"_blank\">primer.htm</a>&nbsp;&nbsp;&nbsp;|\n".
" <a href = \"$prim_p\">\n".
" target = \"_blank\">primer.pdf</a>&nbsp;&nbsp;&nbsp;#8211;\n".
" A Beginner's Guide to HTML<br /><br />\n".
" <i>World Wide Web Consortium:</i><br />\n".
" <a href = \"$css1_h\">\n".
" target = \"_blank\">css1.htm</a>&nbsp;&nbsp;&nbsp;|\n".
" <a href = \"$css1_p\">\n".
" target = \"_blank\">css1.pdf</a>&nbsp;&nbsp;&nbsp;#8211;\n".
" Cascading Style Sheets Level 1 Specification<br />\n".
" <a href = \"$css2_h\">\n".
" target = \"_blank\">css2.htm</a>&nbsp;&nbsp;&nbsp;|\n".
" <a href = \"$css2_p\">\n".
" target = \"_blank\">css2.pdf</a>&nbsp;&nbsp;&nbsp;|\n".
" </tr>\n <tr>\n".
" <td height = \"10%\" align = \"center\" valign = \"middle\"></td>\n".
" </tr>\n </table>\n\n</body>\n\n</html>";
return $return;
}

function one_left () {
global $GMS_DATE;
global $GMS_VERSION;
global $header;
global $language;
global $meta;
$return = $header->header ("HTML 4.01 Transitional", "", "", "").
"\n<body text = \"#00008B\" class = \"left_first\">\n\n";
$fontsize = "3"; $tableheight = "100%";
$i = 0; $welcome_ = $language->language ("welcome_welcome");
$to_ = $language->language ("welcome_to");
$version_ = $language->language ("welcome_version");
$enjoy_ = $language->language ("welcome_enjoy");
$hint_ = $language->language ("welcome_hint");
$greeting = array ("", "<b>&lt;\" . $welcome_ . \"&gt;</b>", $to_,
"<b><i>&lt;Gerolf&gt;</i></b>", "<b><i>Markup</i></b>",
"<b><i>&lt;Shredder&gt;</i></b>", $version_ . " " . $GMS_VERSION ,
"&lt;/Shredder&gt;", $GMS_DATE, "&lt;/Gerolf&gt;",
"<b> . $enjoy_ . \"</b>", "&lt;/\" . $welcome_ . \"&gt;";
$height = array ( 4, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 3);
$colspanleft = array ( 1, 1, 2, 2, 3, 3, 4, 3, 3, 2, 2, 1, 1);
$widthleft = array (15, 15, 29, 29, 43, 43, 57, 43, 43, 29, 15, 15);
$colspanmiddle = array ( 6, 6, 5, 5, 4, 4, 3, 4, 4, 5, 5, 6, 6);
$widthmiddle = array (85, 85, 71, 71, 57, 57, 43, 57, 57, 71, 71, 85, 85);
$bgcolor = array ("#FFFFFF", "#87CEFA", "#AFDFC", "#AFDFC", "#D7EEFE",
"#D7EEFE", "#FFFFFF", "#D7EEFE", "#D7EEFE", "#AFDFC", "#AFDFC",
"#87CEFA", "#FFFFFF");
$return = $return . "\n<table \" . # $browser->prop_height ($tableheight) .
" style = \"height: $tableheight;\" width = \"100%\"
border = \"0\" align = \"right\" valign = \"middle\"
summary = \"Welcome to Gerolf Markup Shredder\"
title = \"$hint_\">\n\n";
for ($i = 0; $i < 13; $i++) {$return = $return . "
<tr style = \"height: $height[$i];\">
<td colspan = \"$colspanleft[$i]\" .
" width = \"$widthleft[$i]\" height = \"$height[$i]\"></td>
<td colspan = \"$colspanmiddle[$i]\" width = \"$widthmiddle[$i]\" .
" bgcolor = \"$bgcolor[$i]\"
class = \"$greeting\">$greeting[$i]</td>
</tr>\n\n";}
$return = $return . "\n</table>\n\n</body>\n\n</html>";
return $return;
}

function one_right () {
global $header; global $language;
global $GMS_LINKS; global $GMS_ROOT; global $GMS_VERSION;

```

```

$return = $header->header ("HTML 4.01 Transitional", "", "", "").
"\n<body text = \"#00008B\" class = \"left_first\">\n\n".
"<table height = \"100%\">\n".
" summary = \"Operating system and user agent\">\n".
" width = \"100%\" style = \"height: 100%;\">\n";
if (is_file ("{$GMS_ROOT}/doc/rotation/gerolf.gif")) {
$return = $return .
" <tr class = \"top\">\n".
" <td align = \"left\" class = \"left\">\n";}
else {$return = $return .
" <tr class = \"top\">\n".
" <td align = \"left\" class = \"left\"><br /><br />\n".
" <code>GMS . $language->language ("setup_unzip_en") . "</code>\n".
" <br /><br />\n".
" <code>GMS . $language->language ("setup_unzip_de") . "</code>\n".
" <br /><br />\n".
" <code>GMS . $GMS_VERSION . " / PHP " . PHP_VERSION .
" / " . PHP_OS . "</code>\n";}
$return = $return . " <br /><br />\n".
" </td>\n </tr>\n</table>\n</body>\n</html>";
return $return;}

```

```

function open () {
global $global_input;
global $header;
global $language;
global $GMS_MAXSIZE;
global $GMS_SCRIPT;
global $page;
$filenumber = 1; $field_size = "64";
$mozilla_prop = " height = \"94%\"";
$picture = "<tt>\n <span
class = \"red\"><b>&lt;/b></span><span
class = \"black\"><b>img </b></span><span
class = \"green\">src </span><span
class = \"red\"><b>= \"</b></span><span
class = \"gold\">picture.jpg</span><span
class = \"red\"><b>\&gt;</b></span>\n </tt>";
$linknext = "\n <tt>\n <span
class = \"red\"><b>&lt;/b></span><span
class = \"black\"><b>link </b></span><span
class = \"green\">rel </span><span
class = \"red\"><b>= \"</b></span><span
class = \"gold\">next</span><span
class = \"red\"><b>\&lt;/b></span><span
class = \"green\">href </span><span
class = \"red\"><b>= \"</b></span><span
class = \"gold\">preface.htm</span><span
class = \"red\"><b>\&gt;</b></span>\n </tt>";
$linkcss = "\n <tt>\n <span
class = \"red\"><b>&lt;/b></span><span
class = \"black\"><b>link </b></span><span
class = \"green\">rel </span><span
class = \"red\"><b>= \"</b></span><span
class = \"gold\">stylesheet</span><span
class = \"red\"><b>\&lt;/b></span><span
class = \"green\">href </span><span
class = \"red\"><b>= \"</b></span><span
class = \"gold\">print.css</span><span
class = \"red\"><b>\&lt;/b></span><br /><span
class = \"green\">type </span><span
class = \"red\"><b>= \"</b></span><span
class = \"gold\">text/css</span><span
class = \"red\"><b>\&lt;/b></span><span
class = \"green\">media </span><span
class = \"red\"><b>= \"</b></span><span
class = \"gold\">print</span><span
class = \"red\"><b>\&gt;</b></span>\n </tt>";
$return = $header->header ("HTML 4.01 Transitional", "", "nometa", "").

```

```

"body text = \"#00008B\" class = \"left_first\">\n\n".
# Open via file upload:
"<form method = \"post\" target = \"_parent\">\n".
" action = \"{$GMS_SCRIPT}\">\n".
" enctype = \"multipart/form-data\">\n". $page->fields (). "\n".
"<table summary = \"Open an existing markup file and associated images\">\n".
" style = \"height: 94%;\" width = \"100%\">\n<tr>\n".
" <td width = \"100%\" height = \"100%\">\n".
" align = \"center\" valign = \"middle\">\n".
"<p>&nbsp;</p>\n".
"<p><b> . $language->language ("button_open_right") . "</b></p>\n".
# "<p><input type = \"hidden\" name = \"MAX_FILE_SIZE\">\n".
# " value = \"{$GMS_MAXSIZE}\" />\n".
"<p><input type = \"file\" name = \"file_no[]\">\n".
" accept = \"text/*\" size = \"{$field_size}\" /></p>\n".
"<p>\n".
" <input type = \"input\" name = \"files\">\n".
" value = \"\" . $global_input ["files"] . "\" .
" size = \"3\" maxlength = \"3\" />\n".
" <b> . $language->language ("button_open_picture") . "<br />\n".
" (&nbsp;<span class = \"green\">*.css, *.htm, *.jpg,\n".
" *.pdf, *.png</span>&nbsp;</span>|&nbsp;\n".
" <span class = \"red\"><del>*.bmp,\n".
" *.gif, *.tif</del></span>&nbsp;</span>:</b><br />\n".
" $picture<br />{$linknext<br />{$linkcss<br />max. {$GMS_MAXSIZE} bytes\n".
" </p>";
$untaint = $global_input ["files"];
$result = preg_replace ("/^(.+)$/", "$1", $global_input ["files"]);
while ($filenumber <= $result) {$filenumber++; $return = $return .
"<p><input type = \"file\" name = \"file_no[]\">\n".
" accept = \"text/*\" size = \"{$field_size}\" /></p>";}
$return = $return .
"<p><input type = \"submit\" name = \"file_open\">\n".
" value = \"\" . $language->language ("button_accept") . "\">\n".
" title = \"\" . $language->language ("button_accept_") . "\" /></p>\n".
"<p>&nbsp;</p>\n".
"</td>\n</tr>\n</table>\n</form>\n\n".
"</body>\n</html>";
return $return;}

```

```

function open_internal () {
global $file; global $global_input; global $header;
global $language; global $master; global $page;
global $GMS_BINARIES; global $GMS_ROOT; global $GMS_SCRIPT;
global $GMS_SETTING; global $GMS_TEMPLATE;
$drives = array ("A:", "B:", "C:", "D:", "E:", "F:", "G:", "H:", "I:", "J:",
"K:", "L:", "M:", "N:", "O:", "P:", "Q:", "R:", "S:", "T:", "U:", "V:",
"W:", "X:", "Y:", "Z:");
$i = "0";
$file_ = $global_input ["file"];
if (($file_ == "") or ((dirname ($file_)) == ".")) {
chdir ("{$GMS_ROOT}/doc");}
else {chdir (dirname ($file_));}
clearstatcache ();
$workdir = preg_replace ("/\\\\\\/", "/", getcwd ());
$callback = $global_input ["file_intern"];
$callback = preg_replace ("/\\\\\\/", "/", $callback);
if (($callback != "") and is_dir ($callback)) {$workdir = $callback;}
if (is_dir ($workdir) and is_readable ($workdir) and !is_file ($callback)) {
chdir ($workdir);
$workdir = preg_replace ("/\\\\\\/", "/", getcwd ());
$handle = opendir ($workdir);
while ($dirname = readdir ($handle)) {
if (is_dir ($dirname) and ($dirname != ".")) {
$dirs [$dirname] = $dirname;}
if (is_file ($dirname) and (preg_match ("/\\.htm/", $dirname))) {
$files [$dirname] = $dirname;}}
closedir ($handle);
if (is_array ($dirs)) {ksort ($dirs);}
if (is_array ($files)) {ksort ($files);}

```



```

" \GLYPHNamesload\n" .
" \fillmessage 6-{\glyph}4\n" .
# Codepages:
" \fillmessage 6g{codepage}2\n";
if (is_dir ("${GMS_ROOT}/data/cp") {
  foreach (x_scandir ("${GMS_ROOT}/data/cp") as $entry) {
    if (preg_match ("\\.txt/i", $entry)) {
      $plug = $plug . " \CODEPAGEadd $entry\n";}}
$plug = $plug .
" \CODEPAGEloadthem\n" .
" \CODEPAGEencwrite\n" .
" \expandafter \CODEPAGEenable \CODEPAGE \relax\n" .
" \fillmessage 6-{\codepage}1\n" .
" \fillmessage 4-{\modules}3\n" .
" \endinput";
if (is_dir ("${GMS_ROOT}/data") {
  x_file_put_contents ("${GMS_SETTING}/plugin.cfg", $plug);}
}

class process {#=====
function del_process ($prcss) {# (oldest) process directory
  global $GMS_PROCESS;
  $dir_prcss = $GMS_PROCESS . "/" . $prcss;
  $prcss_files = array ("");
  if (is_dir ($dir_prcss)) {
    # Get all process files:
    $handle = opendir ($dir_prcss);
    while ($filename = readdir ($handle)) {
      if (is_file ($dir_prcss . "/" . $filename)) {
        array_push ($prcss_files, $filename);}}
    closedir ($handle);
    # Delete process files:
    foreach ($prcss_files as $filename) {
      if ($filename != "") {unlink (preg_replace (
        "/^([_\\-\\w\\.\\:]+)$/", "$1", "$dir_prcss/$filename");}}
    # Remove process subdirectory:
    $untaint = preg_replace ("/^([_\\-\\w\\.\\:]+)$/", "$1", $dir_prcss);
    if ($untaint != "${GMS_PROCESS}") {rmdir ($untaint);}}

function limit_processes ($max) {
  global $file;
  global $global_input;
  global $GMS_PROCESS;
  global $GMS_ROOT;
  $processes = array ("");
  if (!is_dir ("${GMS_ROOT}/tmp") {x_mkdir ("${GMS_ROOT}/tmp");}
  if (!is_dir ($GMS_PROCESS)) {x_mkdir ($GMS_PROCESS);}
  clearstatcache ();
  if (!is_dir ($GMS_PROCESS)) {
    echo ("GMS: No writing permission in ${GMS_ROOT}");}
  else {
    $handle = opendir ($GMS_PROCESS);
    # Get all process subdirectories and their ages:
    while ($dirname = readdir ($handle)) {
      if (is_dir ($GMS_PROCESS . "/" . $dirname) and
        $dirname != "." and $dirname != "..") {
        $stat = stat ($GMS_PROCESS . "/" . $dirname);
        $processes_num = array_push ($processes,
          $stat [9] . "-" . $dirname );}# file statistics [9]: age
    closedir ($handle);
    ksort ($processes);
    # Delete old process subdirectories:
    while ($processes_num > $max) {
      $test = array_shift ($processes);
      $search [0] = "%.*(.*?)%";
      $replace [0] = "$1";
      $return = preg_replace ($search, $replace, $test);
      if ($return != "") {$this->del_process ($return);}
      $processes_num --;}}

```

```

function run () {
  global $global_input;
  if ($global_input ["frame"] == "middle") {
    $this->limit_processes ("25");}
}

class write {#=====
# TDS: ../fonts/fonttype/supplier/typeface
# where fonttype = afm, tfm, type1 (for pfa and pfb), ttf, vf.

# According to the TeX directory structure, afm and pfa / pfb as well as ttf
# files are stored in different trees. - Copies of them must be collected at
# a common place, the tfm directory. The ttf fonts need computing of afm fi-
# les first, before tfm and vf files can be created. After processing of the
# metrics, all vf files are moved to the vf directory, while the pfa/pfb and
# afm copies are deleted. The "core" font metrics get processed during this
# procedure, too.
# The tfm and vf metrics that have been processed algorithmically are *not*
# stored in supplier/typeface subfolders, but in the top level of the tfm or
# vf tree. At the beginning of the metric building process, the top levels
# of the tfm and vf trees are cleared, but files in subfolders remain.

function build ($glyph, $encoding, $mark,
  $extension, $slanting, $newname, $suffix) {
  # $2: glyph file name
  # $3: encoding
  # $4: core/eroc/embed/corefamily/embedfamily mark
  # $5: extension factor
  # $6: slanting factor
  # $7: new name (or 'none')
  # $8: new suffix (optional)
  global $cwd; global $GMS_BINARIES; global $GMS_CODEPAGE;
  global $GMS_FONTS; global $GMS_ROOT; global $GMS_SETTING;
  $sc="";
  set_time_limit ("300");
  if ($glyph != "*.mark") {
    # Get font face file base name from glyph file name:
    $GMS_BASE = preg_replace ("/\\.*/", "", $glyph);
    # Modify name:
    $esab = $GMS_BASE;
    if ($newname == "none") {$esab = $esab . $suffix;}
    elseif ($newname != "") {$esab = $newname;}
    # Change encoding for non-latin fonts:
    if (!file_exists ("${GMS_SETTING}/enc_cfg.php")) {
      if (file_exists ("${GMS_SETTING}/encoding.cfg") and (
        $data = fopen ("${GMS_SETTING}/encoding.cfg", "r")) {
        while (!feof ($data)) {
          if ($line = fgets ($data)) {
            $line = preg_replace ("/\r/", "", $line);
            $line = preg_replace ("/\n/", "", $line);
            $line = $line . "\n";
            $line = preg_replace ("/ *$/", "", $line);
            $line = preg_replace ("/\.#.*$/", "", $line);
            if (!preg_match ("/^ *n*$/", $line)) {
              $line = preg_replace ("/ +/", " ", $line);
              $line = preg_replace ("/^ /", "", $line);
              $line = preg_replace ("/^(.+) : *(.+)$/",
                "if (\$GMS_BASE == \"\$1\") {\$encoding = \"\$2\"};", $line);
              $lines = $lines . $line;}}
            $lines = "<?php\n\n . $lines . \"\n?>";
            x_file_put_contents ("${GMS_SETTING}/enc_cfg.php", $lines);}
    include ("${GMS_SETTING}/enc_cfg.php");
    $encoding = strtolower ($encoding);
    # Run ttf2afm:
    $arga = "${GMS_BINARIES}/ttf2afm";
    $argb = "${GMS_ROOT}/data/enc/$encoding.enc";
    $argc = "${GMS_BASE}.afm";
    if (is_file ("${GMS_FONTS}/tfm/${GMS_BASE}.ttf") and

```



```

($glyph == "$GMS_BASE.ttf") {$argd = "./$GMS_BASE.ttf";
x_shell_exec (
"echo $arga -e $argb -o $argc $argd >> $GMS_SETTING/fontmap.log, """);
x_shell_exec ("$arga -e $argb -o $argc $argd", "$GMS_SETTING/font.log");}
elseif (is_file ("$GMS_FONTS/tfm/$GMS_BASE.TTF") and
($glyph == "$GMS_BASE.TTF")) {$argd = "./$GMS_BASE.TTF";
x_shell_exec (
"echo $arga -e $argb -o $argc $argd >> $GMS_SETTING/fontmap.log, """);
x_shell_exec ("$arga -e $argb -o $argc $argd", "$GMS_SETTING/font.log");}
# Define afm2tfm arguments:
if ((is_file ("$GMS_FONTS/tfm/$GMS_BASE.afm")) or (
is_file ("$GMS_FONTS/tfm/$GMS_BASE.AFM")) {
$arga = "$GMS_BINARIES/afm2tfm";
$argb = "./$GMS_BASE.afm";
$argc = "$GMS_ROOT/data/enc/$encoding.enc";
$argd = "$esab.vpl";
# Extend typeface:
if (($extension != "") and ($extension != "1") and ($extension != "1.0")
and ($extension != "1.00") and ($extension != "1.000") and
($extension != "1.0000") and ($extension != "1.00000") and
($extension != "hide")) {
$arge = "-e $extension";}
# Slant typeface:
if (($slanting != "") and ($slanting != "0") and ($slanting != "0.0")
and ($slanting != "0.00") and ($slanting != "0.000") and
($slanting != "0.0000") and ($slanting != "0.00000")) {
$argf = "-s $slanting";}
# Run afm2tfm, build map item, eventually add glyph:
x_shell_exec (
"echo $arga $argb -T $argc -v $argd $arge $argf $esab.tfm .
" > item_tmp.map >> $GMS_SETTING/fontmap.log, """);
x_shell_exec ($arga .
"$argb -T $argc -v $argd $arge $argf $esab.tfm .
" > item_tmp.map", "$GMS_SETTING/font.log");
$item = x_file_get_contents ("$GMS_FONTS/tfm/item_tmp.map");
if (($mark != "core") and ($mark != "eroc")) {
$item = preg_replace ("/<.*$/", "<$encoding.enc <$glyph", $item);}
else {$item = preg_replace ("/<.*$/", "<$encoding.enc", $item);}
$unsort = x_file_get_contents ("$GMS_SETTING/unsort.map") . $item;
x_file_put_contents ("$GMS_SETTING/unsort.map", $unsort);
# Run vptovf, move processed core afm fonts to _trans_:
$arga = "$GMS_BINARIES/vptovf";
x_shell_exec ("echo $arga $esab.vpl $esab.vf $esab.tfm .
" >> $GMS_SETTING/fontmap.log, """);
if (strtoupper (substr (PHP_OS, 0, 3)) == "WIN") {###
x_shell_exec ("echo. >> $GMS_SETTING/fontmap.log, """);}###
else {x_shell_exec ("echo >> $GMS_SETTING/fontmap.log, """);}###
x_shell_exec ($arga .
"$esab.vpl $esab.vf $esab.tfm", "$GMS_SETTING/font.log");
@unlink ("$GMS_FONTS/tfm/$esab.vpl");###
if ($mark == "eroc") {
@rename ("$GMS_FONTS/tfm/$GMS_BASE.afm",
"$GMS_FONTS/_trans_/$GMS_BASE.afm"); # Fixme: case
@rename ("$GMS_FONTS/tfm/$GMS_BASE.AFM",
"$GMS_FONTS/_trans_/$GMS_BASE.afm");}}}}

function build_all () {
global $cwd; global $global_input;
global $GMS_CODEPAGE; global $GMS_FONTS;
global $GMS_ROOT; global $GMS_SETTING;
global $GMS_VERSIONSTR; global $GMS_ZIP;
global $ZIP_BODY;
if (!is_dir ($GMS_SETTING)) {x_mkdir ($GMS_SETTING);}
x_file_put_contents ("$GMS_SETTING/font.log",
"% font.log - Warnings, errors, missing glyphs\n");
x_file_put_contents ("$GMS_SETTING/fontmap.log",
"% fontmap.log - Execution of font processor binaries\n\n");
# Clean up tfm folder if previous run was interrupted:
x_unlink ("\afm", "$GMS_FONTS/tfm");
x_unlink ("\pf[ab]", "$GMS_FONTS/tfm");

x_unlink ("\tff", "$GMS_FONTS/tfm");
x_unlink ("\vpl", "$GMS_FONTS/tfm");
x_unlink ("\vf", "$GMS_FONTS/tfm");
@unlink ("$GMS_FONTS/tfm/item_tmp.map");
# Delete old tfm and vf; get afm, ttf, pfa/pfb:
x_unlink ("\vf", "$GMS_FONTS/vf");
x_unlink ("\tfm", "$GMS_FONTS/tfm");
if (strtoupper (substr (PHP_OS, 0, 3)) == "WIN") {
x_mkdir ("$GMS_FONTS/ttf");
# Copy True Type Fonts from %windir%/Fonts:
$this->copy_fonts ("bh", "lucon", "lucon", "", "", "");
$this->copy_fonts ("bh", "lsans", "l10646", "", "", "");
$this->copy_fonts ("itc", "framd", "framd", "framdit", "", "");
$this->copy_fonts ("ms", "comic", "comic", "comibd", "", "");
$this->copy_fonts ("ms",
"georgia", "georgia", "georgiab", "georgiai", "georgiaz");
$this->copy_fonts ("ms", "sans", "micros", "", "", "");
$this->copy_fonts ("ms", "sylfaen", "sylfaen", "", "", "");
$this->copy_fonts ("ms", "tahoma", "tahoma", "tahomabd", "", "");
$this->copy_fonts ("ms",
"trebuc", "trebuc", "trebucbd", "trebuchi", "trebucit");
$this->copy_fonts ("ms",
"verdana", "verdana", "verdanab", "verdanai", "verdanaz");
$this->copy_fonts ("mt", "aharoni", "ahronbd", "", "", "");
$this->copy_fonts ("mt", "andalus", "andlso", "", "", "");
$this->copy_fonts ("mt", "arial", "arial", "arialbd", "arialbi", "ariali");
$this->copy_fonts ("mt", "artro", "artrobd", "artro", "", "");
$this->copy_fonts ("mt", "cour", "cour", "courbd", "courbi", "couri");
$this->copy_fonts ("mt", "david", "david", "davidbd", "davidtr", "");
$this->copy_fonts ("mt", "frank", "frank", "", "", "");
$this->copy_fonts ("mt", "garamond", "gara", "garabd", "garait", "");
$this->copy_fonts ("mt",
"gothic", "gothic", "gothicb", "gothicbi", "gothici");
$this->copy_fonts ("mt", "impact", "impact", "", "", "");
$this->copy_fonts ("mt", "levenim", "lvnm", "lvnmbd", "", "");
$this->copy_fonts ("mt",
"miriam", "mriam", "mriamc", "mriamfx", "mriamtr");
$this->copy_fonts ("mt", "narkisim", "nrkis", "", "", "");
$this->copy_fonts ("mt", "rod", "rod", "rodr", "", "");
$this->copy_fonts ("mt", "simpo", "simpbd", "simpfx", "simpo", "");
$this->copy_fonts ("mt", "times", "times", "timesbd", "timesbi", "timesi");
$this->copy_fonts ("mt", "trado", "tradbd", "trado", "", "");
# if exist "$GMS_FONTS\data$tagly.ttf" del "$GMS_FONTS\data$tagly.ttf">> $Z$
# if exist "$GMS_FONTS\outlook.ttf" del "$GMS_FONTS\outlook.ttf">> $Z$
}
if ($global_input ["extrafonts"] != "") {$this->copy_all_fonts ();}
$this->copy_to_tfm ("ttf");
$this->copy_to_tfm ("typ1");
$this->copy_to_tfm ("afm");
# Get new tfm and vf:
if (is_dir ("$GMS_FONTS/tfm")) {chdir ("$GMS_FONTS/tfm");}
@unlink ("$GMS_SETTING/font.map");
@unlink ("$GMS_SETTING/sort.map");
@unlink ("$GMS_SETTING/unsort.map");
@unlink ("$GMS_FONTS/_trans_");
x_mkdir ("$GMS_FONTS/_trans_");
# Process font files by encoding:
if (is_dir ("$GMS_ROOT/data/enc")) {
foreach (x_scandir ("$GMS_ROOT/data/enc") as $entry) {
if (($entry != ".") and ($entry != "..") and ($entry != "_ignore_")) {
if (preg_match ("/\.enc/i", $entry)) {
$base = preg_replace ("/\.*/", "", $entry);
$this->build_encode ($base);}}}}
# Process transformed files:
x_copy ("\afm", "$GMS_FONTS/_trans_", "$GMS_FONTS/tfm");
x_unlink ("\afm", "$GMS_FONTS/_trans_");
if (file_exists ("$GMS_SETTING/font.cfg") and (
$data = fopen ("$GMS_SETTING/font.cfg", "r")) {
while (!feof ($data)) {
## if ($line = fgets ($data, 1048576)) {

```

```

if ($line = fgets ($data, 1023)) {
    $line = preg_replace ("%GMS_CODEPAGE%/ ", $GMS_CODEPAGE, $line);
    $line = preg_replace ("/r/", "", $line);
    $line = preg_replace ("/ *$/", "", $line);
    $line = preg_replace ("/\.\.*/", "", $line);
    if (!preg_match ("/^ *n*$/", $line)) {
        $line = preg_replace ("/ +/", " ", $line);
        $line = preg_replace ("/^ /", "", $line);
        $line = preg_replace ("/ /", "\\", $line);
        $line = preg_replace ("/^(.*)$/",
            "\$this->build_trans (\\"$1", "\\"; ", $line);
        $lines = $lines . $line;}}
$lines = "<?php\n\n" . $lines . "\n?>";
x_file_put_contents ("GMS_SETTING/font_cfg.php", $lines);
include ("GMS_SETTING/font_cfg.php");
# Process font files by type:
if (is_dir ("GMS_FONTS/tfm")) {
    foreach (x_scandir ("GMS_FONTS/tfm") as $entry) {
        if (($entry != ".") and ($entry != "..") and ($entry != "_ignore_")
            and (is_file ("GMS_FONTS/tfm/$entry"))) {
            if (preg_match ("/pcr.*.afm/i", $entry)) {
                $this->build ($entry, $GMS_CODEPAGE, "eroc", "", "", "", "");}
            if (preg_match ("/phv.*.afm/i", $entry)) {
                $this->build ($entry, $GMS_CODEPAGE, "eroc", "", "", "", "");}
            if (preg_match ("/ptm.*.afm/i", $entry)) {
                $this->build ($entry, $GMS_CODEPAGE, "eroc", "", "", "", "");}
            if (preg_match ("/\.ttf/i", $entry)) {
                $this->build ($entry, $GMS_CODEPAGE, "embed", "", "", "", "");}
            @unlink ("GMS_FONTS/tfm/$entry");} ## save space
            if (preg_match ("/.pfa/i", $entry)) {
                $this->build ($entry, $GMS_CODEPAGE, "embed", "", "", "", "");}
            @unlink ("GMS_FONTS/tfm/$entry");}
            if (preg_match ("/.pfb/i", $entry)) {
                $this->build ($entry, $GMS_CODEPAGE, "embed", "", "", "", "");}
            @unlink ("GMS_FONTS/tfm/$entry");}}}}
# Install virtual fonts (and clean up tfm folder):
#@unlink ("GMS_SETTING/font.log"); ## save space
x_unlink ("\afm", "GMS_FONTS/tfm");
x_unlink ("\pf[ab]", "GMS_FONTS/tfm");
x_unlink ("\ttf", "GMS_FONTS/tfm");
x_unlink ("\vpl", "GMS_FONTS/tfm");
x_copy ("\vf", "GMS_FONTS/tfm", "GMS_FONTS/vf");
x_unlink ("\vf", "GMS_FONTS/tfm");
@unlink ("GMS_FONTS/tfm/item_tmp.map");
# Build font map:
$head = "% font.map - GMS_FONTS/[font-type]/[supplier]/[font-family]\n\n" .
    "% Generated by Gerolf Markup Shredder (www.Gerolf.org)\n" .
    "% on " . date ("D, Y/m/d, G:i:s T") . "\n" .
    "% Default codepage: $GMS_CODEPAGE\n\n" .
    "% _[1] Base name of font files\n" .
    "% [2] PostScript font face name\n" .
    "% \" [3] ReEncodeFont \" PostScript encoding name\n" .
    "% <[4] Encoding file [*].enc\n" .
    "% <[5] Glyph file [*].pfa, *.ttf\n\n";
$lines = array ();
if (file_exists ("GMS_SETTING/unsort.map") and (
    $data = fopen ("GMS_SETTING/unsort.map", "r"))) {
    while (!feof ($data)) {
        if ($line = fgets ($data, 1023)) {
            array_push ($lines, $line);}}
asort ($lines);
foreach ($lines as $line) {$sort = $sort . $line;}
x_file_put_contents ("GMS_SETTING/font.map", $head . $sort);
@unlink ("GMS_SETTING/enc_cfg.php");
@unlink ("GMS_SETTING/font_cfg.php");
@unlink ("GMS_SETTING/unsort.map");
if (!is_dir (GMS_FONTS)) {@unlink ("GMS_SETTING/font.map");}

function build_encode ($encoding) {
    global $GMS_FONTS;
    foreach (x_scandir ("GMS_FONTS/tfm") as $entry) {
        if (($entry != ".") and ($entry != "..") and ($entry != "_ignore_")) {
            if (preg_match ("/$encoding.*.afm/i", $entry)) {
                $this->build ($entry, $encoding, "eroc", "", "", "", "");}
            elseif (preg_match ("/$encoding.*.pfa/i", $entry)) {
                $this->build ($entry, $encoding, "embed", "", "", "", "");}
            elseif (preg_match ("/$encoding.*.ttf/i", $entry)) {
                $this->build ($entry, $encoding, "embed", "", "", "", "");}}}}

function build_trans ($basename, $encoding, $mark,
    $extension, $slanting, $newname, $suffix) {
    global $GMS_FONTS;
    if ($mark != "") {
        # Build transformed fonts:
        if ($mark == "core") {
            # This is only possible if a type1 glyph file is present:
            if (is_file ("GMS_FONTS/tfm/$basename.afm")) {
                $this->build ("$basename.pfa", $encoding, $mark,
                    $extension, $slanting, $newname, $suffix);}
            if ($mark == "corefamily") {
                foreach (x_scandir ("GMS_FONTS/tfm") as $entry) {
                    if (preg_match ("/$basename.*.afm/i", $entry)) {
                        $this->build ($entry, $encoding, "embed",
                            $extension, $slanting, $newname, $suffix);}}
            if ($mark == "embed") {
                foreach (x_scandir ("GMS_FONTS/tfm") as $entry) {
                    if (preg_match ("/$basename.pf[ab]/i", $entry)) {
                        $this->build ($entry, $encoding, $mark,
                            $extension, $slanting, $newname, $suffix);}
                    elseif (preg_match ("/$basename.ttf/i", $entry)) {
                        $this->build ($entry, $encoding, $mark,
                            "1.0", "0.0", $newname, $suffix);}}
            if ($mark == "embedfamily") {
                foreach (x_scandir ("GMS_FONTS/tfm") as $entry) {
                    if (preg_match ("/$basename.pf[ab]/i", $entry)) {
                        $this->build ($entry, $encoding, "embed",
                            $extension, $slanting, $newname, $suffix);}
                    elseif (preg_match ("/$basename.*.ttf/i", $entry)) {
                        $this->build ($entry, $encoding, "embed",
                            "1.0", "0.0", $newname, $suffix);}}}}

function copy_all_fonts () {
    global $global_input; global $GMS_FONTS; global $GMS_SETTING;
    $GMS_SRC = $global_input ["extrafonts"];
    if (is_dir ($GMS_SRC)) {
        foreach ((x_scandir ($GMS_SRC)) as $entry) {
            if (preg_match ("/.*.afm/i", $entry)) {
                @copy ("GMS_SRC/$entry", "GMS_FONTS/afm/$entry");}
            if (preg_match ("/.*.pfa/i", $entry)) {
                @copy ("GMS_SRC/$entry", "GMS_FONTS/type1/$entry");}
            if (preg_match ("/.*.ttf/i", $entry)) {
                @copy ("GMS_SRC/$entry", "GMS_FONTS/ttf/" .
                    strtolower ($entry));}}}}

function copy_fonts ($supplier, $family,
    $typefacea, $typefaceb, $typefacec, $typefaced) {
    global $GMS_FONTS; global $GMS_SETTING;
    # Copy Windows True Type fonts to GMS fonts folder:
    x_mkdir ("GMS_FONTS/ttf/$supplier");
    x_mkdir ("GMS_FONTS/ttf/$supplier/$family");
    $GMS_SRC = $_ENV ["windir"] . "/Fonts";
    $GMS_TRG = "GMS_FONTS/ttf/$supplier/$family";
    $typeface = array ($typefacea, $typefaceb, $typefacec, $typefaced);
    if ((is_dir ($GMS_SRC)) and (is_dir ($GMS_TRG))) {
        foreach ($typeface as $entry) {
            if (is_file ("GMS_SRC/$entry.ttf")) {
                @copy ("GMS_SRC/$entry.ttf", "GMS_TRG/$entry.ttf");}
            elseif (is_file ("GMS_SRC/$entry.TTF")) { ##
                @copy ("GMS_SRC/$entry.TTF", "GMS_TRG/$entry.TTF");} ##

```

```

        elseif ($entry != "") {x_shell_exec
            ("echo $GMS_SRC/$entry.ttf >> \"\$GMS_SETTING/fontmiss.log\", "");}}}}#
function copy_to_tfm ($fonttype) {
    global $GMS_FONTS;
    if ("fonttype" == "type1") {
        x_copy ("\pf[ab]", "$GMS_FONTS/$fonttype", "$GMS_FONTS/tfm");}
    elseif ("fonttype" == "ttf") {
        x_copy ("\tff", "$GMS_FONTS/$fonttype", "$GMS_FONTS/tfm");}
    elseif ("fonttype" == "afm") {
        x_copy ("\afm", "$GMS_FONTS/$fonttype", "$GMS_FONTS/tfm");}
    if (is_dir ("GMS_FONTS/$fonttype")) {
        foreach (x_scandir ("GMS_FONTS/$fonttype") as $entry) {
            if (($entry != ".") and ($entry != "..") and ($entry != "_ignore_")) {
                $this->handle_supplier ($fonttype, $entry);}}}
function handle_supplier ($fonttype, $supplier) {
    global $GMS_FONTS;
    if (($supplier != "_ignore_") and (
        is_dir ("GMS_FONTS/$fonttype/$supplier"))) {# Fixme: case
        if ("fonttype" == "type1") {
            x_copy ("\pf[ab]",
                "GMS_FONTS/$fonttype/$supplier", "$GMS_FONTS/tfm");}
        elseif ("fonttype" == "ttf") {
            x_copy ("\tff",
                "GMS_FONTS/$fonttype/$supplier", "$GMS_FONTS/tfm");}
        elseif ("fonttype" == "afm") {
            x_copy ("\afm",
                "GMS_FONTS/$fonttype/$supplier", "$GMS_FONTS/tfm");}
        foreach (x_scandir ("GMS_FONTS/$fonttype/$supplier") as $entry) {
            if (($entry != ".") and ($entry != "..") and ($entry != "_ignore_")) {
                $this->handle_typeface ($fonttype, $supplier, $entry);}}}
function handle_typeface ($fonttype, $supplier, $typeface) {
    global $GMS_FONTS;
    if (($typeface != "_ignore_") and (
        is_dir ("GMS_FONTS/$fonttype/$supplier/$typeface"))) {# Fixme: case
        if ("fonttype" == "type1") {
            x_copy ("\pf[ab]",
                "GMS_FONTS/$fonttype/$supplier/$typeface", "$GMS_FONTS/tfm");}
        elseif ("fonttype" == "ttf") {
            x_copy ("\tff",
                "GMS_FONTS/$fonttype/$supplier/$typeface", "$GMS_FONTS/tfm");}
        elseif ("fonttype" == "afm") {
            x_copy ("\afm",
                "GMS_FONTS/$fonttype/$supplier/$typeface", "$GMS_FONTS/tfm");}}}
function run () {
    global $header; global $language;
    global $GMS_SETTING; global $SRV_SETTING;
    $before = time ();
    $return = $header->header ("HTML 4.01 Transitional", "", "nometa", "") .
        "\n<body text = \"#00008B\">\n\n<br />\n\n" .
        "<pre style = \"fontstyle\">\n" . $language->language ("writing") . "\n";
    $this->build_all ();
    if (file_exists ("GMS_SETTING/font.map")) {
        $content = $language->language ("done_details") .
            "<a href = \"\$SRV_SETTING/font.map\">font.map</a>\n ";}
    $return = $return . $content . GMS_runtime ($before, time ());
    if (!is_file ("GMS_BINARIES/gerolf.efmt")) {
        $return = $return . $language->language ("setup_init");}
    return $return . "</pre>\n\n" . "</body>\n\n</html>";}
}??

```

gmsintro.txt

Gerolf Markup Shredder

The typesetting program that uses TeX
for document conversion from HTML to PDF

License

Copyright (c) 1999-2008 by G. D. Brettschneider, Luchtbergstr. 27,
D-28237 Bremen. All rights reserved. This GMS software comes
without ANY warranty. You may freely distribute and use it.
The author asks for a donation.

Installation

Run 'gmsunzip.bat' (Dos, Windows) or 'gmsunzip' (Linux)
in '[GMS_ROOT]', the Markup Shredder installation folder.
If the GMS archives are already extracted, run 'gmssetup.bat'
respectively 'gmssetup' in '[GMS_ROOT]/etc'.

Usage

After installation, Markup Shredder can be started from the
command line by saying 'gms' or 'gerolf' (text mode interface).
The complete GMS installation instruction can be found in
'[GMS_ROOT]/doc/handbook/handbook.htm'.

<http://www.Gerolf.org>

gms2exe.sed

```

[Version]
Class=EXPRESSION
SEDVersion=3
[Options]
PackagePurpose=ExtractOnly
ShowInstallProgramWindow=0
HideExtractAnimation=0
UseLongFileName=1
InsideCompressed=1
CAB_FixedSize=0
CAB_ResvCodeSigning=0
RebootMode=I
InstallPrompt=%InstallPrompt%
DisplayLicense=%DisplayLicense%
FinishMessage=%FinishMessage%
TargetName=%TargetName%
FriendlyName=%FriendlyName%
AppLaunched=%AppLaunched%
PostInstallCmd=%PostInstallCmd%
AdminQuietInstCmd=%AdminQuietInstCmd%
UserQuietInstCmd=%UserQuietInstCmd%
SourceFiles=SourceFiles
[Strings]
InstallPrompt=Do you wish to install GMS? Choose a top level directory
(like 'C:\gerolf', 'C:\gms008a') - or the 'document root' folder (for
example 'C:\wamp\www', 'C:\xampp\htdocs') of your PHP server (available at
www.wampserver.com, www.xampp.org).
DisplayLicense=gmsintro.txt
FinishMessage=To continue: Run 'gmsunzip.bat' (Dos, Windows) respectively
'gmsunzip' (Linux) in the installation folder that you have chosen (GMS textmode
interface) - or 'http://localhost/gerolf.php', if you have installed a PHP server
(GMS web browser interface).
TargetName=gms008a.exe
FriendlyName=Gerolf Markup Shredder 0.08a - HTML to PDF converter using TeX
AppLaunched=
PostInstallCmd=
AdminQuietInstCmd=
UserQuietInstCmd=
FILE0="bindos1.zip"
FILE1="bindos2.zip"
FILE2="binlinux.zip"
FILE3="binwin.zip"
FILE4="docdemo.zip"
FILE5="docw3c.zip"
FILE6="docweb.zip"
FILE7="fontsin.zip"
FILE8="fontst1c.zip"
FILE9="fontst1d.zip"
FILE10="fontst1e.zip"
FILE11="fontst1p.zip"
FILE12="fontst1u.zip"
FILE13="fontsth.zip"
FILE14="fontsttf.zip"
FILE15="gerolf.css"
FILE16="gerolf.php"
FILE17="gms008a.zip"
FILE18="gms2exe.sed"
FILE19="gmsintro.txt"
FILE20="gmssetup"
FILE21="gmsunzip"
FILE22="gmsunzip.bat"
FILE23="texhyph.zip"
FILE24="unzipdos.exe"
FILE25="unzipwin.exe"
[SourceFiles]
SourceFiles0=
[SourceFiles0]
%FILE0%=
%FILE1%=
%FILE2%=
%FILE3%=
%FILE4%=
%FILE5%=
%FILE6%=
%FILE7%=
%FILE8%=
%FILE9%=
%FILE10%=
%FILE11%=
%FILE12%=
%FILE13%=
%FILE14%=
%FILE15%=
%FILE16%=
%FILE17%=
%FILE18%=
%FILE19%=
%FILE20%=
%FILE21%=
%FILE22%=
%FILE23%=
%FILE24%=
%FILE25%=

```

gmsunzip

```

#!/bin/sh
# gmsunzip
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2008). All rights reserved.
# Send corrections to: MarkupShredder(at)Gerolf.org (www.Gerolf.org)
# Subject: GMS shell scripts

export GMSdateGMSUNZIP=20080107
GMS_VERSION="0.08a"
GMS_VERS="008a"

# =====

if [ "$1" = "" ]; then
    xterm -fg black -bg white -e "$0" -shell
    exit
fi

left="Running:      GMS unzip ..."
right="Gerolf Markup Shredder $GMS_VERSION"
arg="/////////////////////////////////////"
quit="[Ctrl+C] to quit ..."

if [ "$0" != "./gmsunzip" ]; then
    test=$(dirname "$0")
    export GMS_ROOT=$test
    if [ -d "$GMS_ROOT" ]; then cd "$GMS_ROOT"; fi
    test=
fi

# Any OS messages: -----

setterm -clear
echo " $left                               $right"

```

```

echo "$arg$arg"
echo
echo "      Welcome to Gerolf Markup Shredder,"
echo "      the typesetting program that uses TeX"
echo "      for document conversion from HTML to PDF."
echo
echo "      The latest version of GMS can always be found at:"
echo "      http://www.Gerolf.org"
echo
echo "      You should have installed a browser for hypertext"
echo "      markup files and a reader for portable document files."
echo
echo "      HTML Browsers are available at:"
echo "      http://www.konqueror.org"
echo "      http://www.mozilla.org"
echo "      http://wp.netscape.com/download"
echo "      http://www.opera.com"
echo
echo "      Adobe Acrobat Reader for PDF can be downloaded from:"
echo "      http://www.adobe.com/products/acrobat/alternate.html"
echo "$arg$arg"
echo -n "      Press [Enter] to continue or $quit"
read

setterm -clear
echo "$left"                $right"
echo "$arg$arg"
echo
echo "      GMS unzip will create the following directory structure:"
echo
echo "      [GMS_ROOT] ..... '../gmsNNNx', '../gerolf'"
echo "      +- batch ... DOS/Windows scripts"
echo "      +- bin .... DOS/Windows/Linux executables"
echo "      +- data .... Codepages and hyphenation patterns"
echo "      +- doc .... HTML template files, tested with GMS"
echo "      +- etc .... Configuration, setup and start files"
echo "      +- fonts ... PostScript, TrueType and TeX fonts"
echo "      +- shell ... Linux scripts for Bash"
echo "      +- tex .... TeX macro scripts"
echo "      +- tmp .... Temporary files"
echo
echo
echo
echo
echo
echo "$arg$arg"
echo -n "      Press [Enter] to continue or $quit"
read

# File extraction: -----
if [ ! -f /usr/bin/unzip ]; then
  setterm -clear
  echo "$left"                $right"
  echo "$arg$arg"
  echo
  echo "      Missing '/usr/bin/unzip' binary."
  echo
  echo "      Please install 'unzip' from your system disk."
  echo
  echo -n "      Press [Enter] to continue or $quit"
  read
fi

setterm -clear
echo "$left"                $right"
echo "$arg$arg"
echo

```

```

# unzip option:
# -u: update existing files
# -o: overwrite existing files without prompting

e="Extracting: "
m="Missing: "
GMS_UNZIP="unzip -u -o"

arg1="gms$GMS_VERS.zip"
if [ ! -f $arg1 ]; then echo "$m $arg1 (required)"; fi
if [ -f $arg1 ]; then echo "$e $arg1 ..."; fi
if [ -f $arg1 ]; then $GMS_UNZIP $arg1 > gmsunzip.log; fi
arg1=

if [ ! -f texhyph.zip ]; then echo "$m texhyph.zip (required)"; fi
if [ -f texhyph.zip ]; then echo "$e texhyph.zip ..."; fi
if [ -f texhyph.zip ]; then $GMS_UNZIP texhyph.zip >> gmsunzip.log; fi

if [ ! -f bindos1.zip ]; then echo "$m bindos1.zip (required on Dos)"; fi
if [ -f bindos1.zip ]; then echo "$e bindos1.zip ..."; fi
if [ -f bindos1.zip ]; then $GMS_UNZIP bindos1.zip >> gmsunzip.log; fi

if [ ! -f bindos2.zip ]; then echo "$m bindos2.zip (required on Dos)"; fi
if [ -f bindos2.zip ]; then echo "$e bindos2.zip ..."; fi
if [ -f bindos2.zip ]; then $GMS_UNZIP bindos2.zip >> gmsunzip.log; fi

if [ ! -f docdemo.zip ]; then echo "$m docdemo.zip (recommended)"; fi
if [ -f docdemo.zip ]; then echo "$e docdemo.zip ..."; fi
if [ -f docdemo.zip ]; then $GMS_UNZIP docdemo.zip >> gmsunzip.log; fi

if [ ! -f docw3c.zip ]; then echo "$m docw3c.zip (recommended)"; fi
if [ -f docw3c.zip ]; then echo "$e docw3c.zip ..."; fi
if [ -f docw3c.zip ]; then $GMS_UNZIP docw3c.zip >> gmsunzip.log; fi

if [ ! -f docweb.zip ]; then echo "$m docweb.zip (recommended)"; fi
if [ -f docweb.zip ]; then echo "$e docweb.zip ..."; fi
if [ -f docweb.zip ]; then $GMS_UNZIP docweb.zip >> gmsunzip.log; fi

if [ ! -f fontst1p.zip ]; then echo "$m fontst1p.zip (recommended)"; fi
if [ -f fontst1p.zip ]; then echo "$e fontst1p.zip ..."; fi
if [ -f fontst1p.zip ]; then $GMS_UNZIP fontst1p.zip >> gmsunzip.log; fi

if [ ! -f fontstlu.zip ]; then echo "$m fontstlu.zip (recommended)"; fi
if [ -f fontstlu.zip ]; then echo "$e fontstlu.zip ..."; fi
if [ -f fontstlu.zip ]; then $GMS_UNZIP fontstlu.zip >> gmsunzip.log; fi

if [ ! -f fontsttf.zip ]; then echo "$m fontsttf.zip (recommended)"; fi
if [ -f fontsttf.zip ]; then echo "$e fontsttf.zip ..."; fi
if [ -f fontsttf.zip ]; then $GMS_UNZIP fontsttf.zip >> gmsunzip.log; fi

if [ ! -f fontstlc.zip ]; then echo "$m fontstlc.zip (optional)"; fi
if [ -f fontstlc.zip ]; then echo "$e fontstlc.zip ..."; fi
if [ -f fontstlc.zip ]; then $GMS_UNZIP fontstlc.zip >> gmsunzip.log; fi

if [ ! -f fontstld.zip ]; then echo "$m fontstld.zip (optional)"; fi
if [ -f fontstld.zip ]; then echo "$e fontstld.zip ..."; fi
if [ -f fontstld.zip ]; then $GMS_UNZIP fontstld.zip >> gmsunzip.log; fi

if [ ! -f fontstle.zip ]; then echo "$m fontstle.zip (optional)"; fi
if [ -f fontstle.zip ]; then echo "$e fontstle.zip ..."; fi
if [ -f fontstle.zip ]; then $GMS_UNZIP fontstle.zip >> gmsunzip.log; fi

if [ ! -f binwin.zip ]; then echo "$m binwin.zip (required on Windows)"; fi
if [ -f binwin.zip ]; then echo "$e binwin.zip ..."; fi
if [ -f binwin.zip ]; then $GMS_UNZIP binwin.zip >> gmsunzip.log; fi

if [ ! -f binlinux.zip ]; then echo "$m binlinux.zip (required on Linux)"; fi
if [ -f binlinux.zip ]; then echo "$e binlinux.zip ..."; fi
if [ -f binlinux.zip ]; then $GMS_UNZIP binlinux.zip >> gmsunzip.log; fi

```

```

# Additional archives:

if [ -f fontsin.zip ]; then $GMS_UNZIP fontsin.zip >> gmsunzip.log; fi
if [ -f fontsth.zip ]; then $GMS_UNZIP fontsth.zip >> gmsunzip.log; fi

unset e
unset m
unset GMS_UNZIP

if [ ! -f "$GMS_ROOT/etc/gmssetup" ]; then
  if [ -f "$PWD/etc/gmssetup" ]; then export GMS_ROOT="$PWD"; fi; fi

if [ -f "$GMS_ROOT/etc/gmssetup" ]; then
  chmod 755 "$GMS_ROOT/etc/gmssetup"
  chmod 755 "$GMS_ROOT/etc/folder"

# Create [GMS_ROOT] level link to gmssetup:
GMS_LINK="$GMS_ROOT/gmssetup"
echo "#!/bin/sh" > "$GMS_LINK"
echo " cd \"\$GMS_ROOT/etc\" >> \"$GMS_LINK"
echo " \"\$GMS_ROOT/etc/gmssetup\" \"\$1\" >> \"$GMS_LINK"
chmod 755 "$GMS_LINK"

GMS_LINK="$HOME/Desktop/gmssetup"
echo "#!/bin/sh" > "$GMS_LINK"
echo " cd \"\$GMS_ROOT/etc\" >> \"$GMS_LINK"
echo " \"\$GMS_ROOT/etc/gmssetup\" \"\$1\" >> \"$GMS_LINK"
chmod 755 "$GMS_LINK"

# Create desktop link to gmssetup:
# if [ ! -d "$HOME/Desktop" ]; then mkdir "$HOME/Desktop"; fi
# GMS_LINK="$HOME/Desktop/gmssetup.desktop"
# echo "[Desktop Entry]" > "$GMS_LINK"
# echo "Exec=\"\$GMS_ROOT/gmssetup\" >> \"$GMS_LINK"
# echo "Icon=run" >> "$GMS_LINK"

```

```

# echo "Name=gmssetup" >> "$GMS_LINK"
# echo "Terminal=true" >> "$GMS_LINK"
# echo "Type=Application" >> "$GMS_LINK"
# chmod 755 "$GMS_LINK"

# Done GMS unzip message:
echo
echo
echo " Done:      GMS unzip."
echo
echo " $arg$arg"
echo -n "          Press [Enter] to run GMS setup or $quit"
read

# Run GMS setup:
cd "$GMS_ROOT/etc"
setterm -clear
source "$GMS_ROOT/etc/gmssetup" -shell

export GMS_LINK=
else
setterm -reset
echo " $left          $right"
echo " $arg$arg"
echo
echo "          Did not find GMS setup script in directory"
echo "          '$PWD/etc'."
echo
echo "          Repeat installation procedure in terminal."
echo "          Download an updated version of GMS from"
echo "          'http://www.Gerolf.org'."
echo
read
fi

```

gmsunzip.bat

```

echo off
cls
echo.

REM gmsunzip.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2008). All rights reserved.
REM Send corrections to: MarkupShredder(at)Gerolf.org (www.Gerolf.org)

set GMSdateGMSUNZIP=20080107
set GMS_VERSION=0.08a
set GMS_VERS=008a

REM =====

REM Start shell:

if "%comspec%" == "" echo Please set COMSPEC variable. Press any key ...
if "%comspec%" == "" pause > nul
if "%comspec%" == "" goto fi_sh
if "%1" == "" %comspec% /E:4096 /C %0 -shell
if "%1" == "" goto exit
:fi_sh

REM Check operating system to get OS and LINKDIR variables:

REM Assume support for long filenames if environment variable OS is defined:

set any=*. *

```

```

if "%OS%" == "" set any=nul

REM FreeDOS:
if not "%OS%" == "" goto fi_fd
if exist C:\kernel.sys if not exist C:\msdos.sys set GMS_FreeDOS=FreeDOS
:fi_fd

REM DRDOS:
if "%OS%" == "DRDOS" set GMS_DRDOS=6
if not "%DRDOSCFG%" == "" set GMS_DRDOS=7
if "%OS%" == "DRDOS" set OS=
if not "%GMS_DRDOS%" == "" goto fi_os

REM ReactOS:
if "%OS%" == "ReactOS" goto fi_noros
if exist "%windir%\startm-1\any%" set GMS_LINKDIR=%windir%\startm-1
:fi_noros

REM Windows 9x:
if "%OS%" == "" if not "%winbootdir%" == "" set OS=Windows_9x
if "%OS%" == "Windows_9x" if "%winbootdir%" == "." set OS=
if "%OS%" == "Windows_9x" set any=nul
if "%windir%" == "" if not "%winbootdir%" == "" set GMS_DOSMODE=1
if "%windir%" == "" if "%winbootdir%" == "." set GMS_DOSMODE=
if not "%GMS_DOSMODE%" == "" set windir=%winbootdir%
if exist "%windir%\Desktop\any%" set GMS_LINKDIR=%windir%\Desktop

REM Windows NT:
if "%USERPROFILE%" == "" goto fi_nont
if not exist "%USERPROFILE%\Desktop\any%" goto fi_nont
set GMS_LINKDIR=%USERPROFILE%\Desktop

```

```

:fi_nont
echo
goto fi_dn
GMS requires DOS 3x and an i386-compatible machine.
:else_dn
echo
Adobe Acrobat Reader for PDF can be downloaded from:
echo
http://www.adobe.com/products/acrobat/alternate.html
echo
HTML-Kit editor for markup files is available at:
echo
http://www.chami.com/html-kit/download
echo.
The easiest way to open a markup file with GMS is
echo
clicking on it with the right mouse button in Windows
echo
Explorer. Select 'Send to gerolf.bat' in context menu.
:fi_dn
echo.
echo %arg%%arg%
echo
Press any key to continue or [Ctrl+C] to quit ...
pause > nul

cls
echo %left% %right%
echo %arg%%arg%
echo.
GMS unzip will create the following directory structure:
echo.
[GMS_ROOT] ..... e.g. 'c:\gms\NWX', 'c:\gerolf'
echo
+ batch ... DOS/Windows scripts
echo
+ bin .... DOS/Windows/Linux executables
echo
+ data .... Codepages and hyphenation patterns
echo
+ doc .... HTML template files, tested with GMS
echo
+ etc .... Configuration, setup and start files
echo
+ fonts ... PostScript, TrueType and TeX fonts
echo
+ shell ... Linux scripts for Bash
echo
+ tex .... TeX macro scripts
echo
+ tmp .... Temporary files
for %i in ( 1 2 3 4 5 6 7 ) do echo.
echo %arg%%arg%
echo
Press any key to continue or [Ctrl+C] to quit ...
pause > nul

REM Welcome message: -----
cls
echo %left% %right%
echo %arg%%arg%
echo.
GMS does not run on %OS%,
echo
sorry.
for %i in ( 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 ) do echo.
set i=
echo %arg%%arg%
echo
Press any key to continue or [Ctrl+C] to quit ...
pause > nul
:fi_ros

REM Any OS messages: -----
echo %left% %right%
echo %arg%%arg%
echo.
Welcome to Gerolf Markup Shredder,
echo
the typesetting program that uses TeX
echo
for document conversion from HTML to PDF.
echo.
The latest version of GMS can always be found at:
echo
http://www.Gerolf.org
echo.
You should have installed a browser for hypertext
echo
markup files and a reader for portable document files.
echo.
if not "%OS%" == "" goto else_dn
echo
Browsers for DOS/Windows 3x are available at:
echo
http://browsers.evolt.org
echo
http://arachne.browser.org
echo.
Acrobat Reader for Windows 3x can be downloaded from:
echo
http://vetusware.com
echo.

```

```

if not exist bindos2.zip echo Missing:    bindos2.zip (required on Dos)
if exist bindos2.zip echo Extracting:    bindos2.zip ...
if exist bindos2.zip %GMS_UNZIP% bindos2.zip >> gmsunzip.log

if not exist docdemo.zip echo Missing:    docdemo.zip (recommended)
if exist docdemo.zip echo Extracting:    docdemo.zip ...
if exist docdemo.zip %GMS_UNZIP% docdemo.zip >> gmsunzip.log

if not exist docw3c.zip echo Missing:    docw3c.zip (recommended)
if exist docw3c.zip echo Extracting:    docw3c.zip ...
if exist docw3c.zip %GMS_UNZIP% docw3c.zip >> gmsunzip.log

if not exist docweb.zip echo Missing:    docweb.zip (recommended)
if exist docweb.zip echo Extracting:    docweb.zip ...
if exist docweb.zip %GMS_UNZIP% docweb.zip >> gmsunzip.log

if not exist fontst1p.zip echo Missing:    fontst1p.zip (recommended)
if exist fontst1p.zip echo Extracting:    fontst1p.zip ...
if exist fontst1p.zip %GMS_UNZIP% fontst1p.zip >> gmsunzip.log

if not exist fontst1u.zip echo Missing:    fontst1u.zip (recommended)
if exist fontst1u.zip echo Extracting:    fontst1u.zip ...
if exist fontst1u.zip %GMS_UNZIP% fontst1u.zip >> gmsunzip.log

if not exist fontsttf.zip echo Missing:    fontsttf.zip (recommended)
if exist fontsttf.zip echo Extracting:    fontsttf.zip ...
if exist fontsttf.zip %GMS_UNZIP% fontsttf.zip >> gmsunzip.log

if not exist fontst1c.zip echo Missing:    fontst1c.zip (optional)
if exist fontst1c.zip echo Extracting:    fontst1c.zip ...
if exist fontst1c.zip %GMS_UNZIP% fontst1c.zip >> gmsunzip.log

if not exist fontst1d.zip echo Missing:    fontst1d.zip (optional)
if exist fontst1d.zip echo Extracting:    fontst1d.zip ...
if exist fontst1d.zip %GMS_UNZIP% fontst1d.zip >> gmsunzip.log

if not exist fontst1e.zip echo Missing:    fontst1e.zip (optional)
if exist fontst1e.zip echo Extracting:    fontst1e.zip ...
if exist fontst1e.zip %GMS_UNZIP% fontst1e.zip >> gmsunzip.log

if not exist binwin.zip echo Missing:    binwin.zip (required on Windows)
if exist binwin.zip echo Extracting:    binwin.zip ...
if exist binwin.zip %GMS_UNZIP% binwin.zip >> gmsunzip.log

if not exist binlinux.zip echo Missing:    binlinux.zip (required on Linux)
if exist binlinux.zip echo Extracting:    binlinux.zip ...
if exist binlinux.zip %GMS_UNZIP% binlinux.zip >> gmsunzip.log

REM Additional archives:

if exist fontsin.zip %GMS_UNZIP% fontsin.zip >> gmsunzip.log
if exist fontsth.zip %GMS_UNZIP% fontsth.zip >> gmsunzip.log

echo.
echo.
echo Done:          GMS unzip.

REM Check for 'etc' subfolder: .....

if not exist etc\%any% goto fi_noetc

REM Move logfile to 'etc' subfolder: .....

if not exist gmsunzip.log goto fi_mvlog
copy gmsunzip.log etc > nul
del gmsunzip.log
:fi_mvlog

REM Get GMS_SETTING (current directory): .....

cd etc

```

```

REM pwd.sys just contains the command "set PWD=":
copy pwd.sys pwd_tmp.bat > nul
cd>> pwd_tmp.bat
if exist pwd_tmp.bat call pwd_tmp
if exist pwd_tmp.bat del pwd_tmp.bat
set GMS_SETTING=%PWD%

REM Create desktop links (Windows 9x/NT/XP): .....

if "%OS%" == "" goto fi_nowin

if "%GMS_LINKDIR%" == "" goto fi_nowin

REM Writing test:
if not "%OS%" == "Windows_9x" goto else_x1
echo GMS writing test (1).> "%GMS_LINKDIR%\gms.001"
echo This file can be removed.>> "%GMS_LINKDIR%\gms.001"
goto fi_x1
:else_x1
echo GMS writing test (1).> "%GMS_LINKDIR%\gms.001" 2> nul
echo This file can be removed.>> "%GMS_LINKDIR%\gms.001" 2> nul
:fi_x1
if exist "%GMS_LINKDIR%\gms.001" goto fi_notxt
echo          Cannot write start links to '%GMS_LINKDIR%'.
echo.
if not "%OS%" == "Windows_NT" if not "%OS%" == "Windows_XP" goto fi_x2
echo          Please log in as administrator to install GMS.
goto exit
:fi_x2
echo          Press any key to continue or [Ctrl+C] to break ...
pause > nul
goto fi_nowin
:fi_notxt
if exist "%GMS_LINKDIR%\gms.001" del "%GMS_LINKDIR%\gms.001"

REM Create [GMS_ROOT] level link to gmssetup: .....
set GMS_LINK=..\gmssetup.bat
echo @echo off> %GMS_LINK%
echo.>> %GMS_LINK%
echo REM gmssetup.bat Link script to:>> %GMS_LINK%
echo REM ===== Gerolf Markup Shredder setup>> %GMS_LINK%
echo.>> %GMS_LINK%
if "%OS%" == "Windows_9x" goto th_w98_1
echo cd /d "%GMS_SETTING%">> %GMS_LINK%
goto fi_w98_1
:th_w98_1
echo echo You may have to change the drive before>> %GMS_LINK%
echo echo calling '%GMS_SETTING%\gmssetup.bat'.>> %GMS_LINK%
echo echo.>> %GMS_LINK%
echo pause>> %GMS_LINK%
echo cd "%GMS_SETTING%">> %GMS_LINK%
:fi_w98_1
echo "%GMS_SETTING%\gmssetup.bat" %1>> %GMS_LINK%
copy "%GMS_LINK%" "%GMS_LINKDIR%" > nul
set GMS_LINK=%GMS_LINKDIR%\gmssetup.bat
goto fi_etc
:fi_nowin

REM Create [GMS_ROOT] level link to gmssetup.bat (DOS): .....

if not "%OS%" == "" goto fi_nodos
set GMS_LINK=..\gmssetup.bat
if not "%GMS_DRDOS%" == "" echo echo off > %GMS_LINK%
if not "%GMS_DRDOS%" == "" echo cls >> %GMS_LINK%
if not "%GMS_DRDOS%" == "" echo echo Running ...>> %GMS_LINK%
if "%GMS_DRDOS%" == "" echo @echo off > %GMS_LINK%
echo cd %GMS_SETTING% >> %GMS_LINK%
echo gmssetup.bat %1>> %GMS_LINK%
goto fi_etc
:fi_nodos

```



```

if exist %dlist% del %dlist% > nul
if exist %batch% del %batch% > nul
set dlist=
set batch=

:fi_noetc

cls
echo %left%                %right%
echo %arg%%arg%
echo.
echo          Did not find GMS setup in directory
echo          '[GMS_ROOT]\etc'.
echo.
echo          Repeat installation procedure. Download a new
echo          version of GMS from 'http://www.Gerolf.org'.
echo.
echo.
goto exit

:fi_etc

echo.
echo %arg%%arg%

REM -----

set any=
set arg=
set left=

set right=
set GMSdateGMSUNZIP=
set GMS_DRIVE=
set GMS_FreeDOS=
set GMS_LINK=
set GMS_LINKDIR=
set GMS_SETTING=
set GMS_UNZIP=
set GMS_VERSION=
set GMS_VERS=
set PWD=

echo          Press any key to run GMS setup or [Ctrl+C] to quit ...
pause > nul

if not "%GMS_DRDOS%" == "" set OS=DRDOS
if "%OS%" == "Windows_9x" set OS=
if "%OS%" == "Windows_XP" set OS=Windows_NT
if not "%GMS_DOSMODE%" == "" set windir=
if exist gmssetup.bat gmssetup.bat -shell

:exit

set any=
set GMSdateGMSUNZIP=
set GMS_DOSMODE=
set GMS_DRDOS=
set GMS_VERSION=
set GMS_VERS=

```

index.htm

```

<!-- IMPORTANT: Users may have to change the script address in line 87! -->
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <title>Gerolf Markup Shredder</title>

  <meta name = "description"
    content = "A desktop publishing system that runs in your web browser!" />
  <meta name = "keywords"
    content = "typography, typesetting, layout, fonts, HTML parser,
hypertext, document portability, Schriftsatz, Buchsatz, Buchgestaltung,
Typographie, desktop publishing, PDF, tex, etex, pdftex, pdfetex, PostScript,
Acrobat Reader" />
  <meta name = "author"
    content = "Gerolf D. Brettschneider, Luchtbergstr. 27, D-28237 Bremen" />
  <meta name = "email"
    content = "MarkupShredder@Gerolf.org" />
  <meta name = "copyright"
    content = "Gerolf Markup Shredder - Copyright (c) 1999-2004 by
Gerolf Diethelm Brettschneider, Luchtbergstr. 27, D-28237 Bremen.
All rights reserved. The GMS software comes without ANY warranty.
You may freely distribute and use it. The author, however, is not a rich
man. If you like this program and want it to be developed any further,
please be kind and send him a little money: Brettschneider c/o Hr. Wulf,
Deutsche Bank AG Bremen, BIC: DEUTDE33HAN, IBAN: DE17 2907 0024 0244 9510 00,
BLZ: 290 700 24, KTO: 244 9510 00. Thank you very much. - GDB" />
  <meta name = "date"
    content = "2004-11-12" /><!-- created: 2003-08-13 -->
  <meta http-equiv = "content-language"
    content = "de, en, fr" />
  <meta http-equiv = "content-type"
    content = "text/html; charset=windows-1252" />
  <meta http-equiv = "content-style-type"
    content = "text/css" />
  <meta name = "generator"
    content = "Gerolf Markup Shredder (www.Gerolf.org)" />
  <meta name = "robots"
    content = "index, follow" />
  <meta name = "revisit-after"
    content = "30 days" />
  <meta name = "MSSmartTagsPreventParsing"
    content = "true" />
  <xmeta http-equiv = "refresh"
    content = "0; URL=./gerolf.php" />

  <link rel = "shortcut icon" type = "image/ico"
    href = "./doc/handbook/favicon.ico" />

  <style type = "text/css" media = "print">
  <!--
  @page {size: 29.7cm 21cm; margin: 2cm 0cm 0cm 0cm}
  h1 {line-height: 18px}
  h4 {line-height: 16px}
  .download {line-height: 16px}
  .gerolf {line-height: 60px}
  -->
</style>

```

```

<style type = "text/css">
<!--
a {text-decoration: none}
a:link {color: #000080}
a:visited {color: #000080}
a:hover {color: #87CEFA; background-color: #FFFFFF}
body {background: #FFFFFF; color: #000080;
  font-family: Verdana, Helvetica, Arial, sans-serif}
.download {margin-left: 123px}
.version {margin-left: 50px}
td {font-family: Verdana, Helvetica, Arial, sans-serif}
-->
</style>
</head>
<body text = "white" link = "white" alink = "white" vlink = "white">
  <table summary = "" width = "100%">
  <tr>
    <td width = "15%" height = "400">&nbsp;&nbsp;&nbsp;</td>
    <td width = "10%"><h1><a href = "http://www.GDBrettschneider.de"
      class = "gerolf" target = "_blank"
      title = "www.GDBrettschneider.de"><img
      alt = "www.GDBrettschneider.de" src = "./doc/rotation/gerolf.gif"
      width = "60" height = "60" border = "0" /></a></h1></td>
    <td width = "65%"><h2 class = "download">&nbsp;&nbsp;&nbsp;</h2>
    <h1><a href = "gerolf.php"
      title = "Browser Interface"><i>Gerolf Markup
      Shredder</i></a></h1></td>
  </tr>
  <tr>
    <td width = "15%" height = "100">&nbsp;&nbsp;&nbsp;</td>
    <td width = "10%">&nbsp;&nbsp;&nbsp;</td>
    <td width = "65%">
    <h4>The <a target = "_blank" href = "./doc/handbook/handbook.htm"
      title = "The Gerolf Markup Shredder Handbook (HTML)"
      ><big>typesetting</big></a>
    <a target = "_blank" href = "./doc/handbook/handbook.pdf" title =
      "The Gerolf Markup Shredder Handbook (PDF, 1.7MB)"
      ><big>program</big></a> that uses
    <a href = "http://www.ctan.org" target = "_blank" title =
      "Typesetting engine; say 'Tech', like German 'Blech' (www.ctan.org)"
      ><big>T</big><sub><big>E</big></sub></big></a></sub></big></a><br /> for
    <a href = "./doc/index.htm" target = "_blank"
      title = "Examples"><big>document</big></a>
    <a href = "./doc/index-de.htm" target = "_blank"
      title = "Beispiele"><big>conversion</big></a> from
    <a href = "http://www.w3.org"
      title = "Hypertext Markup Language (www.w3.org)"
      target = "_blank"><big>HTML</big></a> to
    <a href =
      "https://partners.adobe.com/asn/developer/technotes/main.html"
      title = "Portable Document Format (www.adobe.com)"
      target = "_blank"><big>PDF</big></a></h4></td>
  </tr>
</table>
</body>
</html>

```

[GMS_ROOT]/data/cp

[GMS_ROOT]/batch

Batch Files (Dos, Windows)

browser.bat

```

REM browser.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateBROWSER=20060927

REM =====

REM Debug:
if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> "%Z%"
:fi_db

REM Error message on Dos:

if not "%OS%" == "" goto fi_dos
set GMS_BRWSR=
REM Windows 3x check:
set | find "windir" > }{.bat
echo set windir=%%1> windir.bat
call }{
del }{.bat
del windir.bat
REM Search Netscape Navigator and Internet Explorer:
set GMS_TEST=%windir%\..\netscape\comm\program\netscape.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=\netscape\comm\program\netscape.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=C:\netscape\comm\program\netscape.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%

set GMS_TEST=%windir%\..\iexplore\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=\iexplore\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=C:\iexplore\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%

set GMS_TEST=%windir%\..\msie\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=\msie\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=C:\msie\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%

set GMS_TEST=%windir%\..\msie20\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=\msie20\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=C:\msie20\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%

set GMS_TEST=%windir%\..\msie30\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=\msie30\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=C:\msie30\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%

set GMS_TEST=%windir%\..\msie40\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=\msie40\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=C:\msie40\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%

set GMS_TEST=%windir%\..\msie50\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=\msie50\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=C:\msie50\iexplore.exe
if exist %GMS_TEST% set GMS_BRWSR=%GMS_TEST%

if "%GMS_BRWSR%" == "" goto then_dos
if "%windir%" == "" goto fi_sw
cls
echo. %GMS_FILE%
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%%arg%
echo.
echo. You may have to quit Windows 3x
echo. before starting Markup Shredder.
echo.
echo. Press any key ...
pause > nul
set arg=
goto exit
:fi_sw
win %GMS_BRWSR% %GMS_FOLDER%\%GMS_FILE%
goto fi_dos
:then_dos
set GMS_TEST=
set GMS_BRWSR=
cls
echo. %GMS_FILE%
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%%arg%
echo.
echo. Could not find HTML browser.
echo.
echo. A browser for Dos or Windows 3x can be downloaded from:
echo. http://browsers.evolt.org
echo. http://arache.browser.org
echo. Install it at 'C:\netscape' or 'C:\iexplore'.
echo. Otherwise add the path to the browser binary to the
echo. PATH environment variable in 'C:\autoexec.bat'
echo. or add a startup batch file to:
echo. %GMS_BINARIES%
echo.
echo. Then change the 'Browser' entry in the 'Select Programs'
echo. dialog box or the GMS_BROWSER environment variable in:
echo. %GMS_SETTING%\gerolf.bat
echo.
echo. You may have to quit Windows 3x before starting GMS.
echo.

```

```

echo          Press any key ...
echo.
echo %arg%%arg%
echo. %GMS_FOLDER%
set arg=
pause > nul
goto exit
:fi_dos

REM Error message on Windows:

if "%OS%" == "" goto fi_win
set GMS_BRWSR=
REM Search Internet Explorer:
REM Windows 9x:
set GMS_TEST=%windir%\..\Programs\Intern-1\iexplore.exe
if exist "%GMS_TEST%" set GMS_BRWSR=%GMS_TEST%
set GMS_TEST=%windir%\..\Progra-1\Intern-1\iexplore.exe
if exist "%GMS_TEST%" set GMS_BRWSR=%GMS_TEST%
REM Windows XP:
set GMS_TEST=%ProgramFiles%\Internet Explorer\iexplore.exe
if exist "%GMS_TEST%" set GMS_BRWSR=%GMS_TEST%
REM Browser call:
if not "%GMS_BRWSR%" == "" "%GMS_BRWSR%" "%GMS_FOLDER%\%GMS_FILE%"
if not "%GMS_BRWSR%" == "" goto fi_win
set GMS_TEST=
set GMS_BRWSR=
cls
echo. %GMS_FILE%
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%%arg%
echo.

```

```

echo Problem:    Could not find HTML browser.
echo.
echo Advice:    Try Internet Explorer, Mozilla, Opera, or
echo            Lynx (text mode).
echo            Add the path to the browser binary to the
echo            PATH environment variable in:
if "%OS%" == "Windows_9x" set arg1=C:\autoexec.bat
if not "%OS%" == "Windows_9x" set arg1=%SystemRoot%\System32\autoexec.nt
echo            %arg1%
set arg1=
echo            or add a startup .BAT or .LNK file to:
echo.            %GMS_BINARIES%
if not "%OS%" == "Windows_9x" echo.            or to: %SystemRoot%
echo.
echo            Then change the 'Browser' entry in the 'Select Programs'
echo            dialog box or the GMS_BROWSER environment variable in:
echo            %GMS_SETTING%\gerolf.bat
echo.
echo            Press any key ...
if "%OS%" == "Windows_9x" echo.
echo.
echo.
echo.
echo %arg%%arg%
echo. %GMS_FOLDER%
set arg=
pause > nul
goto exit
:fi_win

:exit

set GMS_BRWSR=
set GMS_TEST=

```

compiler.bat

```
echo off

REM compiler.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM GMSdatecompiler=20060927

REM =====

REM If necessary, read launcher script to load environment:

if not "%GMS_BATCH%" == "" goto fi_ll
call gerolf -passive
if not "%GMS_BATCH%" == "" goto fi_lll
echo GMS error: Could not set environment variables.
echo Run 'gmssetup' to create GMS launcher script 'gerolf!'
pause > nul
echo.
goto exit
:fi_lll
:fi_ll

REM Debug: -----

if "%OS%" == "" goto fi_db
if %GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
:fi_db

REM Check if bat2exec compiler is present: -----

REM set COMPILER=%GMS_BINARIES%\bat2exec.com

REM if exist %GMS_BINARIES%\bat2exec.com goto fi_1
if "%OS%" == "" if exist %GMS_BINARIES%\bat2exec.com goto fi_1
if not "%OS%" == "" if exist %GMS_BINARIES%\bat2exec.com" goto fi_1
echo GMS error: Missing '%GMS_BINARIES%\bat2exec.com'.
echo.
goto exit
:fi_1

REM Convert batch files into binaries: -----

REM cd %GMS_BATCH% > nul
if "%OS%" == "" cd %GMS_BATCH% > nul
if not "%OS%" == "" cd "%GMS_BATCH%" > nul

if not "%1" == "" goto fi_2
cls
echo compiler.log - GMS: Running batch compiler and syntax checker ...
REM type %GMS_SETTING%\desktop.scn
if "%OS%" == "" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" type "%GMS_SETTING%\desktop.scn"
echo %GMS_BATCH%
:fi_2

set GMS_PROGRAM=batch compiler

REM Compilation is only possible for command scripts that do neither assign-
REM ments to "globally" used environment variables nor output redirection to
REM a file. No whitespace is allowed behind file names.

echo GMS: browser.bat --->> compiler.txt
bat2exec browser.bat>> compiler.txt
if exist browser.com del browser.com

echo GMS: compiler.bat --->> compiler.txt
bat2exec compiler.bat>> compiler.txt
if exist compiler.com del compiler.com

echo GMS: editor.bat --->> compiler.txt
bat2exec editor.bat>> compiler.txt
if exist editor.com del editor.com

echo GMS: folder.bat --->> compiler.txt
if "%OS%" == "" copy %GMS_SETTING%\folder.bat> nul
if not "%OS%" == "" copy "%GMS_SETTING%\folder.bat"> nul
bat2exec folder.bat>> compiler.txt
if exist folder.com del folder.com
if exist folder.bat del folder.bat

echo GMS: g_code.bat --->> compiler.txt
bat2exec g_code.bat>> compiler.txt
if exist g_code.com del g_code.com
echo GMS: g_color.bat --->> compiler.txt
bat2exec g_color.bat>> compiler.txt
if exist g_color.com del g_color.com
echo GMS: g_dos.bat --->> compiler.txt
bat2exec g_dos.bat>> compiler.txt
if exist g_dos.com del g_dos.com
echo GMS: g_file.bat --->> compiler.txt
bat2exec g_file.bat>> compiler.txt
if exist g_file.com del g_file.com
echo GMS: g_font.bat --->> compiler.txt
bat2exec g_font.bat>> compiler.txt
if exist g_font.com del g_font.com
echo GMS: g_good.bat --->> compiler.txt
bat2exec g_good.bat>> compiler.txt
if exist g_good.com del g_good.com
echo GMS: g_launch.bat --->> compiler.txt
bat2exec g_launch.bat>> compiler.txt
if exist g_launch.com del g_launch.com
echo GMS: g_list.bat --->> compiler.txt
bat2exec g_list.bat>> compiler.txt
if exist g_list.com del g_list.com
echo GMS: g_menu.bat --->> compiler.txt
bat2exec g_menu.bat>> compiler.txt
if exist g_menu.com del g_menu.com
echo GMS: g_palet.bat --->> compiler.txt
bat2exec g_palet.bat>> compiler.txt
if exist g_palet.com del g_palet.com
echo GMS: g_plug.bat --->> compiler.txt
bat2exec g_plug.bat>> compiler.txt
if exist g_plug.com del g_plug.com
echo GMS: g_prog.bat --->> compiler.txt
bat2exec g_prog.bat>> compiler.txt
if exist g_prog.com del g_prog.com
echo GMS: g_rain.bat --->> compiler.txt
bat2exec g_rain.bat>> compiler.txt
if exist g_rain.com del g_rain.com
echo GMS: g_save.bat --->> compiler.txt
bat2exec g_save.bat>> compiler.txt
if exist g_save.com del g_save.com
echo GMS: g_select.bat --->> compiler.txt
bat2exec g_select.bat>> compiler.txt
if exist g_select.com del g_select.com
echo GMS: g_vars.bat --->> compiler.txt
bat2exec g_vars.bat>> compiler.txt
if exist g_vars.com del g_vars.com
echo GMS: g_wel.bat --->> compiler.txt
bat2exec g_wel.bat>> compiler.txt
if exist g_wel.com del g_wel.com
```

```

echo GMS: gerolf.bat --->> compiler.txt
if "%OS%" == "" copy %GMS_SETTING%\gerolf.bat> nul
if not "%OS%" == "" copy "%GMS_SETTING%\gerolf.bat"> nul
bat2exec gerolf.bat>> compiler.txt
if exist gerolf.com del gerolf.com
if exist gerolf.bat del gerolf.bat

```

```

echo GMS: gms.bat --->> compiler.txt
bat2exec gms.bat>> compiler.txt
if exist gms.com del gms.com

```

```

echo GMS: gmssetup.bat --->> compiler.txt
if "%OS%" == "" copy %GMS_SETTING%\gmssetup.bat> nul
if not "%OS%" == "" copy "%GMS_SETTING%\gmssetup.bat"> nul
bat2exec gmssetup.bat>> compiler.txt
if exist gmssetup.com del gmssetup.com
if exist gmssetup.bat del gmssetup.bat

```

```

echo GMS: gmsunzip.bat --->> compiler.txt
if "%OS%" == "" copy ..\gmsunzip.bat> nul
if not "%OS%" == "" copy "..\gmsunzip.bat"> nul
bat2exec gmsunzip.bat>> compiler.txt
if exist gmsunzip.com del gmsunzip.com
if exist gmsunzip.bat del gmsunzip.bat

```

```

echo GMS: l_banner.bat --->> compiler.txt
bat2exec l_banner.bat>> compiler.txt
echo GMS: l_box.bat --->> compiler.txt
bat2exec l_box.bat>> compiler.txt
echo GMS: l_code.bat --->> compiler.txt
bat2exec l_code.bat>> compiler.txt
echo GMS: l_color.bat --->> compiler.txt
bat2exec l_color.bat>> compiler.txt
echo GMS: l_desk.bat --->> compiler.txt
bat2exec l_desk.bat>> compiler.txt
echo GMS: l_file.bat --->> compiler.txt
bat2exec l_file.bat>> compiler.txt
echo GMS: l_gms.bat --->> compiler.txt
bat2exec l_gms.bat>> compiler.txt
echo GMS: l_good.bat --->> compiler.txt
bat2exec l_good.bat>> compiler.txt
echo GMS: l_list.bat --->> compiler.txt
bat2exec l_list.bat>> compiler.txt
echo GMS: l_menu.bat --->> compiler.txt
bat2exec l_menu.bat>> compiler.txt
echo GMS: l_prog.bat --->> compiler.txt
bat2exec l_prog.bat>> compiler.txt
echo GMS: l_rain.bat --->> compiler.txt
bat2exec l_rain.bat>> compiler.txt
echo GMS: l_save.bat --->> compiler.txt
bat2exec l_save.bat>> compiler.txt
echo GMS: l_select.bat --->> compiler.txt
bat2exec l_select.bat>> compiler.txt
echo GMS: l_wel.bat --->> compiler.txt
bat2exec l_wel.bat>> compiler.txt

```

```

echo GMS: reader.bat --->> compiler.txt
bat2exec reader.bat>> compiler.txt
if exist reader.com del reader.com

```

```

echo GMS: shredder.bat --->> compiler.txt
bat2exec shredder.bat>> compiler.txt
if exist shredder.com del shredder.com

```

```

echo GMS: viewer.bat --->> compiler.txt
bat2exec viewer.bat>> compiler.txt
if exist viewer.com del viewer.com

```

REM Shorten and view log file -----

```

sed "s/BAT2EXEC.*$/;s/PC Magazine.*$/ " < compiler.txt > compiler.tmp
sed "s/^\.*Douglas Boling//;s/\r//;\d;" < compiler.tmp > compiler.txt
sed "s/GMS: /\n- /" < compiler.txt > compiler.tmp
set arg=compiler.log - GMS: Running batch compiler and syntax checker ...
echo %arg%> compiler.pmt
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%>> compiler.pmt
echo. >> compiler.pmt
echo This is BAT2EXEC 1.5, written by Douglas Boling>> compiler.pmt
echo PC Magazine (c) 1990, 1991 Ziff Communications Co.>> compiler.pmt
copy compiler.pmt + compiler.tmp compiler.log > nul
echo. >> compiler.log
echo %arg%>> compiler.log
echo %GMS_BATCH%>> compiler.log
set arg=
if exist compiler.pmt del compiler.pmt
if exist compiler.tmp del compiler.tmp
if exist compiler.txt del compiler.txt
if "%1" == "-quiet" goto fi_v
set backup_folder=%GMS_FOLDER%
set backup_file=%GMS_FILE%
set backup_base=%GMS_BASE%
set backup_short=%GMS_SHORT%
set GMS_FOLDER=%GMS_BATCH%
set GMS_FILE=compiler.log
set GMS_PROGRAM=batch compiler and syntax checker
set GMS_COMMAND=%GMS_VIEWER%
%GMS_DRIVE%
call g_dos -chdir
call gms -execute
cls
:fi_v
if "%OS%" == "" cd %GMS_BATCH%
if not "%OS%" == "" cd "%GMS_BATCH%"
if "%OS%" == "" copy compiler.log %GMS_SETTING%> nul
if not "%OS%" == "" copy compiler.log "%GMS_SETTING%"> nul
del compiler.log
:exit

```

editor.bat

```
REM editor.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateEDITOR=20060927

REM =====

REM Debug:
if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == . goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8) (%9)>>%Z%

:fi_db

REM Select number of console text lines (25, 43 or 50):
if "%GMS_FreeDOS%" == "" mode con lines=50 > nul
if not "%GMS_FreeDOS%" == "" mode co80,50

REM Execute external program (filename as %1 or %GMS_SHORT%, in 8.3 form):
edit %GMS_FOLDER%\%GMS_SHORT%

REM Reset console text lines:
if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%GMS_FreeDOS%" == "" mode co80,25
```


g_code.bat

```
REM g_code.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_CODE=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_code) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8) >> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_code %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_code
goto exit
:fi_nd

REM Resize: -----

if %1 == -resize goto then_rs
if %1 == -cp_up_build goto then_b
if %1 == -cp_lo_build goto then_b
if not %1 == -remove goto fi_rs
:then_rs
set REPLY_OFFSET=3
set REPLY_SIZE=72
set REPLY_ITEMS=1
if %1 == -build goto then_b
if %1 == -remove goto then_r
goto exit
:fi_rs

REM Remove: -----

if not %1 == -remove goto fi_r
:then_r
call l_rain -remove called_by g_code %1
call l_banner -lower -folder called_by g_code %1
goto exit
:fi_r

REM Build: -----

if %1 == -cp_up_build goto then_b
if not %1 == -cp_lo_build goto fi_b
:then_b

set BACKUP=%1
REM Allocate memory:
set BACKUP_OFFSET=123
set BACKUP_SIZE=123
set i=123
set OFFSET_A=123
set OFFSET_B=123
set OFFSET_C=123
set PAGE=12345
set REPLY_SIZE=123
set REPLY_OFFSET=1234567890
call g_select -remove called_by g_code %1
call g_menu -remove called_by g_code %1
if "%1" == "-cp_lo_build" goto else_bb
if "%OS%" == "" goto else_b1
if "%OS%" == "Windows_9x" goto else_b1
REM Remove desktop and switch codepage on Windows:
call l_desk -remove -nobanners called_by g_code %1
if not "%GMS_CHCP%" == "" chcp %GMS_CHCP% > nul 2> nul
cls
call l_gms -desktop -draw -nofolder called_by g_code %1
REM Get current codepage, show banner, rebuild desktop:
call g_dos -current_cp called_by g_code %1
call l_banner -upper -cp_up called_by g_code %1
call l_banner -lower -terminal called_by g_code %1
call g_code -build_cp_desk called_by g_code %1
goto fi_b1
:else_b1
call l_banner -upper -cp_up_dos called_by g_code %1
call l_banner -lower -codepage called_by g_code %1
:fi_b1
REM Build codepage (upper half) and wait:
call l_code -cp_up called_by g_code %1
pause > nul
REM Remove codepage:
call g_code -remove called_by g_code %1
if "%OS%" == "" goto else_b2
if "%OS%" == "Windows_9x" goto else_b2
set REPLY_OFFSET=0
set REPLY_SIZE=78
call l_desk -remove called_by g_code %1
REM Switch codepage:
cls
if not "%GMS_CHCP%" == "" chcp %GMS_INICP% > nul 2> nul
cls
REM Show banner, rebuild desktop:
set GMS_CURCP=%GMS_INICP%
set REPLY_SIZE=82
set REPLY_OFFSET=0
reply -banner 1 "%GMS_FILE%"
title %GMS_FILE% - GMS
call l_banner -lower -folder called_by g_code %1
call g_code -build_cp_desk called_by g_code %1
goto fi_b2
:else_b2
call l_banner -upper -file called_by g_code %1
call l_banner -lower -folder called_by g_code %1
:fi_b2
goto fi_bb
:else_bb
REM Show banner, build codepage (lower half), wait, remove codepage:
call l_banner -upper -cp_lo called_by g_code %1
call l_code -cp_lo called_by g_code %1
pause > nul
call g_code -remove called_by g_code %1
call l_banner -upper -file called_by g_code %1
:fi_bb
REM Build menu:
```

```
set GMS_COLD=S
set GMS_HOT=S
call g_menu -build called_by g_code %1
REM Free memory:
set BACKUP=
set OFFSET_A=
set OFFSET_B=
set OFFSET_C=
set PAGE=
goto exit
:fi_b
```

```
REM Build codepage desktop: .....
```

```
if not %1 == -build_cp_desk goto fi_cpd
:then_cpd
set GMS_RECEIVE=1
```

```
set GMS_RETURN=1
set REPLY_OFFSET=0
set REPLY_SIZE=78
call l_desk -build called_by g_code %1
set REPLY_OFFSET=0
goto exit
:fi_cpd
```

```
REM Not found: .....
```

```
:else
call l_banner -no_action g_code %1
goto exit
:fi
:exit
```

g_color.bat

```
REM g_color.bat
REM =====
```

```
REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).
```

```
REM set GMSdateG_COLOR=20060927
```

```
REM Prologue: =====
```

```
REM Not running:
```

```
if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_color) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr
```

```
REM Debug:
```

```
if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> "%Z%"
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_color %1 %2 %3 %4
:fi_db
```

```
REM Chapters: =====
```

```
REM Not defined:
```

```
if not "%1" == "" goto fi_nd
call l_banner -no_action g_color
goto exit
:fi_nd
```

```
REM Resize: -----
```

```
if %1 == -resize goto then_rs
if %1 == -build goto then_rs
if not %1 == -remove goto fi_rs
:then_rs
set REPLY_OFFSET=26
set REPLY_SIZE=20
set REPLY_ITEMS=8
if %1 == -build goto then_b
if %1 == -remove goto then_r
goto exit
:fi_rs
```

```
REM Build: -----
```

```
if not %1 == -build goto fi_b
:then_b
call l_box -build called_by g_color %1
call l_banner -lower -color
call l_color -build
set GMS_HOT=Q
call l_color -update Q Q called_by g_color %1
set REPLY_MODULE=g_color
set REPLY_ACTION=-update
set GMS_RECEIVE=1
goto exit
:fi_b
```

```
REM Remove: -----
```

```
if not %1 == -remove goto fi_r
:then_r
call l_box -remove called_by g_color %1
call g_menu -remove called_by g_color %1
call g_rain -remove called_by g_color %1
call l_desk -remove called_by g_color %1
set REPLY_MODULE=g_palet
set REPLY_ACTION=-build
set REPLY_OFFSET=0
set REPLY_SIZE=0
set REPLY_ITEMS=0
reply -random 2
set GMS_RECEIVE=1
set GMS_RETURN=1
goto exit
:fi_r
```

```
REM Update: -----
```

```
if not %1 == -update goto fi_u
REM Cold:
if not "%3" == "" goto else_u1
set GMS_COLD=%GMS_HOT%
goto fi_u1
:else_u1
set GMS_COLD=%3
:fi_u1
```

```
REM Hot:
```

```
if not "%2" == "" goto else_u2
call l_banner -no_hot g_color
goto fi_u2
:else_u2
set GMS_HOT=Q
if %2 == T set GMS_HOT=T
if %2 == t set GMS_HOT=T
if %2 == 1 set GMS_HOT=T
if %2 == H set GMS_HOT=H
if %2 == h set GMS_HOT=H
if %2 == 2 set GMS_HOT=H
if %2 == P set GMS_HOT=P
if %2 == p set GMS_HOT=P
if %2 == 3 set GMS_HOT=P
if %2 == B set GMS_HOT=B
if %2 == b set GMS_HOT=B
if %2 == 4 set GMS_HOT=B
if %2 == S set GMS_HOT=S
if %2 == s set GMS_HOT=S
if %2 == 5 set GMS_HOT=S
if %2 == D set GMS_HOT=D
if %2 == d set GMS_HOT=D
if %2 == 6 set GMS_HOT=D
if %2 == L set GMS_HOT=L
if %2 == l set GMS_HOT=L
if %2 == 7 set GMS_HOT=L
if %2 == R set GMS_HOT=T
if %2 == r set GMS_HOT=T
:fi_u2
```

```
REM Handle old, update new:
```

```
if not %GMS_HOT% == %GMS_COLD% goto else_u3
call g_color -handle %GMS_HOT% %GMS_COLD%
goto fi_u3
:else_u3
```

```
REM Update colors:
```

```
if "%4" == "" goto fi_u33
if %GMS_COLD% == T set GMS_TEXT=%4
if %GMS_COLD% == H set GMS_HOTKEY=%4
```

```

    if %GMS_COLD% == P set GMS_PATTERN=%4
    if %GMS_COLD% == B set GMS_BANNER=%4
    if %GMS_COLD% == S set GMS_SHADE=%4
    if %GMS_COLD% == D set GMS_DESKTOP=%4
    if %GMS_COLD% == L set GMS_LETTER=%4
:fi_u33
REM Update color box:
    call l_color -update %GMS_HOT% %GMS_COLD%
:fi_u3
set GMS_RECEIVE=1
goto exit
:fi_u

```

REM Handle:

```

if not %1 == -handle goto fi_h
:then_h

```

REM Quit:

```

if not %GMS_HOT% == Q goto fi_hq
set arga=%REPLY_DESKTOP% %REPLY_PATTERN%
set argb=%REPLY_BANNER% %REPLY_TEXT%
set argc=%REPLY_SHADE% %REPLY_HOTKEY%
set args=%arga% %argb% %argc% %REPLY_LETTER%
call g_palet -handle %args% called_by_(g_color)_(%1
call g_launch -build called_by g_color %1
call g_color -remove called_by g_color %1
set arga=
set argb=
set argc=
set args=
goto exit
:fi_hq

```

REM Text:

```

if not %GMS_HOT% == T goto fi_ht
call l_color -update - T
reply -question 8 %GMS_TEXT% TT 1
goto exit
:fi_ht

```

REM Hotkey:

```

if not %GMS_HOT% == H goto fi_hh
call l_color -update - H
reply -question 10 %GMS_HOTKEY% HH 2
goto exit
:fi_hh

```

REM Pattern:

```

if not %GMS_HOT% == P goto fi_hp
call l_color -update - P
reply -question 12 %GMS_PATTERN% PP 3
goto exit
:fi_hp

```

REM Banner:

```

if not %GMS_HOT% == B goto fi_hb
call l_color -update - B
reply -question 14 %GMS_BANNER% BB 4
goto exit
:fi_hb

```

REM Shadow:

```

if not %GMS_HOT% == S goto fi_hs
call l_color -update - S
reply -question 16 %GMS_SHADE% SS 5
goto exit
:fi_hs

```

REM Desktop:

```

if not %GMS_HOT% == D goto fi_hd
call l_color -update - D
reply -question 18 %GMS_DESKTOP% DD 6
goto exit
:fi_hd

```

REM Character:

```

if not %GMS_HOT% == L goto fi_hl
call l_color -update - L
reply -question 20 %GMS_LETTER% LL 7
goto exit
:fi_hl

```

REM Not found:

```

:else_h
call l_banner -no_hotkey g_color %2
:fi_h

```

REM Not found:

```

:else
call l_banner -no_action g_color %1
:fi
:exit

```

g_dos.bat

```
REM g_dos.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_DOS=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_dos) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Get short name of a file (8 plus 3 characters) on Windows: -----

if not "%1" == "-short" goto fi_s
set GMS_SHORT=%5
if "%OS%" == "" goto exit
if %GMS_DEBUG% == Z echo GMS_SHORT=%GMS_SHORT%>> %Z%
goto exit
:fi_s

REM Check for spaces within filename: -----

if not "%1" == "-check_spacy" goto fi_cs
if not "%2" == "" if not "%3" == "" set GMS_FILE=%GMS_SHORT%
goto exit
:fi_cs

REM Debug: -----

if "%OS%" == "" goto fi_db
if %GMS_DEBUG% == . goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
if "%DEBUG%" == "" goto fi_db1
call l_banner -debug g_dos %1 %2 %3 %4
:fi_db1
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_dos
goto exit
:fi_nd

REM Change directory to GMS_FOLDER, if it exists: -----

REM The folder name may contain spaces on Windows, but quoted folder names can
REM not be recognized on Dos. %GMS_FOLDER% may be set to a file name. In
REM order to find out whether or not a directory really exists, one must try
REM to change into the new directory and look if the current folder has been
REM altered.

REM Fixme: split up into chained parts

if %1 == -pwd goto then_pwd
if %1 == -chdir goto then_cd
if not %1 == -chdir goto fi_cd
REM Get current folder (PWD): .....
:then_pwd
set txt=%GMS_TEMP%\pwd.txt
cd> %txt%
set bat=%GMS_TEMP%\pwd.bat
sed "s/\^/set PWD=/" < %txt% > %bat%
if "%OS%" == "" if not exist %bat% goto fi_pwd1
if not "%OS%" == "" if not exist "%bat%" goto fi_pwd1
call %bat%
if "%OS%" == "" goto elspwd2
if not "%OS%" == "Windows_9x" del %bat% > nul 2> nul
if not "%OS%" == "Windows_9x" del %txt% > nul 2> nul
if "%OS%" == "Windows_9x" del %bat% > nul
if "%OS%" == "Windows_9x" del %txt% > nul
if %GMS_DEBUG% == Z echo PWD=%PWD%>> %Z%
goto fi_pwd2
:elspwd2
del %bat% > nul
del %txt% > nul
:fi_pwd2
:fi_pwd1
set bat=
set txt=
REM Directory name shall not end with a slash:
if "%PWD%" == "A:\^" set PWD=A:
if "%PWD%" == "B:\^" set PWD=B:
if "%PWD%" == "C:\^" set PWD=C:
if "%PWD%" == "D:\^" set PWD=D:
if "%PWD%" == "E:\^" set PWD=E:
if "%PWD%" == "F:\^" set PWD=F:
if "%PWD%" == "G:\^" set PWD=G:
if "%PWD%" == "H:\^" set PWD=H:
if "%PWD%" == "I:\^" set PWD=I:
if "%PWD%" == "J:\^" set PWD=J:
if "%PWD%" == "K:\^" set PWD=K:
if "%PWD%" == "L:\^" set PWD=L:
if "%PWD%" == "M:\^" set PWD=M:
if "%PWD%" == "N:\^" set PWD=N:
if "%PWD%" == "O:\^" set PWD=O:
if "%PWD%" == "P:\^" set PWD=P:
if "%PWD%" == "Q:\^" set PWD=Q:
if "%PWD%" == "R:\^" set PWD=R:
if "%PWD%" == "S:\^" set PWD=S:
if "%PWD%" == "T:\^" set PWD=T:
if "%PWD%" == "U:\^" set PWD=U:
if "%PWD%" == "V:\^" set PWD=V:
if "%PWD%" == "W:\^" set PWD=W:
if "%PWD%" == "X:\^" set PWD=X:
if "%PWD%" == "Y:\^" set PWD=Y:
if "%PWD%" == "Z:\^" set PWD=Z:
if %GMS_STARTUP% == "" set GMS_STARTUP=%PWD%
if %1 == -pwd goto exit
:fi_pwd
REM Change directory: .....
:then_cd
if not "%OS%" == "" if exist "%GMS_FOLDER%\^any%" goto thencd1
if "%OS%" == "" if exist %GMS_FOLDER%\^any% goto elscd1
goto fi_cd1
:thencd1
if %GMS_DEBUG% == Z echo GMS_FOLDER=%GMS_FOLDER%>>%Z%
if exist "%GMS_FOLDER%\^any%" cd "%GMS_FOLDER%"> nul
set PWD=%GMS_FOLDER%
goto fi_cd1
:elscd1
if exist %GMS_FOLDER%\^any% cd %GMS_FOLDER%> nul
```

```

set PWD=%GMS_FOLDER%
:fi_cd1
if "%OS%" == "" goto fi_cd2
if %GMS_DEBUG% == Z echo PWD=%PWD%>> %Z%
:fi_cd2
REM The rest should only be done if a file is given:
if "%OS%" == "" goto fi_cd3
if %GMS_DEBUG% == Z echo GMS_FILE=%GMS_FILE%>> %Z%
:fi_cd3
if "%GMS_FILE%" == "" goto exit
REM Get short name of a file (8 plus 3 characters) on Windows 32: .....
set GMS_ERRSHORT=
set GMS_LOGSHORT=
if "%OS%" == "" set GMS_SHORT=%GMS_FILE%
if "%OS%" == "" goto fi_sn
REM: Fixme: No long file names supported on Windows 9x (no "dir /x"):
if "%OS%" == "Windows 9x" set GMS_SHORT=%GMS_FILE%
if "%OS%" == "Windows 9x" goto fi_sn
REM: Fixme: better get 'find' from Windows subfolder (file.pm analyse)
if %GMS_SKIP% == "1" set GMS_SHORT=%GMS_FILE%
if %GMS_SKIP% == "1" goto fi_sn
if "%OS%" == "" if not exist %GMS_FOLDER%\%GMS_FILE% goto fi_sn
if not "%OS%" == "" if not exist "%GMS_FOLDER%\%GMS_FILE%" goto fi_sn
set trohs=%GMS_TEMP%\trohs.txt
dir /x "%GMS_FOLDER%\%GMS_FILE%" | find /i "%GMS_FILE%" > %trohs%
set short=%GMS_TEMP%\short.bat
sed "s/^/call g_dos -short /" < %trohs% > %short%
if exist "%short%" call %short%
if exist "%short%" del %short%
REM Fallback:
if "%GMS_SHORT%" == "" set GMS_SHORT=%GMS_FILE%
if "%OS%" == "" goto fi_sn1
if %GMS_DEBUG% == Z echo GMS_SHORT=%GMS_SHORT%>> %Z%
:fi_sn1
set short=
set trohs=
REM Replace GMS_FILE by GMS_SHORT, if there are spaces within file name:
call g_dos -check_spacy %GMS_FILE%
:fi_sn
REM Get base name of %GMS_FILE% (without extension): .....
set GMS_ERRBASE=
set GMS_LOGBASE=
set esab=%GMS_TEMP%\esab.txt
echo %GMS_FILE%> %esab%
set base=%GMS_TEMP%\base.bat
if not exist %esab% goto fi_bn1
sed "s/\\/\\\\/g;s/\\.*/\\\\/g;s/\/set GMS_BASE=/" < %esab% > %base%
if "%OS%" == "" if not exist %base% goto fi_bn1
if not "%OS%" == "" if not exist "%base%" goto fi_bn1
call %base%
del %base%
if exist %esab% del %esab%
if "%OS%" == "" goto fi_bn1
if %GMS_DEBUG% == Z echo GMS_BASE=%GMS_BASE%>> %Z%
:fi_bn1
set base=
set esab=
goto exit
:fi_cd

REM Get file name if %GMS_FILE% contains a path: .....

if not %1 == -filename goto fi_fn
set elif_=%GMS_TEMP%\elif.txt
echo %GMS_FILE%> %elif_%
set file_=%GMS_TEMP%\file.bat
if not "%GMS_DRDOS%" == "" goto elsfnl1
sed "s/\\/\\\\/g;s/\.*/\\\\/g;s/\/set file=/" < %elif_% > %file_%
goto fi_fn1a
:elsfnl1
sed "s/\\/\\\\/g;s/\.*/\\\\/g;s/\/set file=/" < %elif_% > %file_%

:fi_fn1a
if "%OS%" == "" if not exist %file_% goto fi_fn1
if not "%OS%" == "" if not exist "%file_" goto fi_fn1
call %file_%
del %file_%
del %elif_%
:fi_fn1
if "%OS%" == "" goto fi_fn2
if %GMS_DEBUG% == Z echo file=%file%>> %Z%
:fi_fn2
set file_=
set elif_=
goto exit
:fi_fn

REM Get path name if %GMS_FOLDER% contains %GMS_FILE%: .....

if not %1 == -pathname goto fi_pn
set htap_=%GMS_TEMP%\htap.txt
echo %GMS_FOLDER%> %htap_%
set path2_=%GMS_TEMP%\path2.txt
if not "%GMS_DRDOS%" == "" goto elspn1a
sed "s/\\/\\\\/g;s/\/set path2_/" < %htap_% > %path2_%
goto fi_pn1a
:elspn1a
sed "s/\\/\\\\/g" < %htap_% > %path2_%
:fi_pn1a
set path_=%GMS_TEMP%\path.bat
sed "s/%GMS_FILE%$/;/s/\/set path_/" < %path2_% > %path_%
if "%OS%" == "" if not exist %path_% goto fi_pn1
if not "%OS%" == "" if not exist "%path_" goto fi_pn1
call %path_%
del %path_%
del %path2_%
del %htap_%
:fi_pn1
if "%OS%" == "" goto fi_pn2
if %GMS_DEBUG% == Z echo path_=%path_%>> %Z%
:fi_pn2
set path_=
set path2_=
set htap_=
goto exit
:fi_pn

REM For each subfolder %i do %2 %3 %4 %5 %6 %7 %8 %9 %1: .....

REM This would bring just the files, not the folders:
REM for %i in (*) do %2 %3 %4 %5 %6 %7 %8 %9 %1
REM So the "dir" function must be used instead (compare g_list -handle).

if not %1 == -for_folder goto fi_gf
if not "%GMS_DRDOS%" == "" goto else_dr
REM 'dir /b' fails on MS-DOS 3x, so use 'mdir':
if "%GMS_DRDOS%" == "" if "%OS%" == "" set argdr=mdir /b
if "%GMS_DRDOS%" == "" if not "%OS%" == "" set argdr=dir /b
%argdr% /-p /ogen /ad > each.lst
set argdr=
sed "s/^\%2 %3 %4 %5 %6 %7 %8 %9 /" < each.lst >> each.bat
goto fi_dr
:else_dr
xdir +d /b /x > each.lst
sed "s/^\%2 %3 %4 %5 %6 %7 %8 %9 /" < each.lst >> each.bat
:fi_dr
if exist each.lst del each.lst
goto exit
:fi_gf

REM Get current codepage number: .....

if not %1 == -current_cp goto fi_cur

```

```

set pchc=%TEMP%\pchc_.txt
chcp > %pchc_%
set chcp=%TEMP%\chcp_.bat
if "%GMS_DRDOS%" == "" goto elscur1
    sed "s/...$/:/s/^\./: +/set GMS_CURCP=/;s/\.// " < %pchc_% > %chcp_%
    goto fi_curl
:elscur1
    sed "s/^\./: //;s/^\./set GMS_CURCP=/;s/\.// " < %pchc_% > %chcp_%
:fi_curl
call %chcp_%
if exist %chcp_% del %chcp_%
set chcp=
if exist %pchc_% del %pchc_%
set pchc=
goto exit
:fi_cur

REM Get integer part of %GMS_CODEPAGE%: -----
if not %1 == -get_chcp goto fi_chcp
set cptmp=%TEMP%\cp.tmp
set cpbat=%TEMP%\cp.bat
echo %GMS_CODEPAGE% > %cptmp%
sed "s/^\./:/s/[A-Z]*/:/s/[a-z]*/:/s/^\./set GMS_CHCP=/ " <%cptmp% >%cpbat%
if exist %cpbat% call %cpbat%
if exist %cpbat% del %cpbat%
if exist %cptmp% del %cptmp%
set cpbat=
set cptmp=
goto exit
:fi_chcp

REM Get runtime: -----
if not %1 == -begin_runtime goto fi_brt
set REPLY_AFTER=0
set REPLY_BEFORE=0
if "%OS%" == "" set backup_REPLY_TEMP=%REPLY_TEMP%
set REPLY_TEMP=runtime.bat

if "%OS%" == "" if exist %REPLY_TEMP% del %REPLY_TEMP%
if not "%OS%" == "" if exist "%REPLY_TEMP%" del %REPLY_TEMP%
reply -time
if "%OS%" == "" if exist %REPLY_TEMP% call %REPLY_TEMP%
if not "%OS%" == "" if exist "%REPLY_TEMP%" call %REPLY_TEMP%
if "%OS%" == "" if exist %REPLY_TEMP% del %REPLY_TEMP%
if not "%OS%" == "" if exist "%REPLY_TEMP%" del %REPLY_TEMP%
if "%OS%" == "" set REPLY_TEMP=backup_REPLY_TEMP%
goto exit
:fi_brt

if not %1 == -end_runtime goto fi_ert
set REPLY_BEFORE=%REPLY_AFTER%
if "%OS%" == "" set backup_REPLY_TEMP=%REPLY_TEMP%
set REPLY_TEMP=runtime.bat
if "%OS%" == "" if exist %REPLY_TEMP% del %REPLY_TEMP%
if not "%OS%" == "" if exist "%REPLY_TEMP%" del %REPLY_TEMP%
reply -time
if "%OS%" == "" if exist %REPLY_TEMP% call %REPLY_TEMP%
if not "%OS%" == "" if exist "%REPLY_TEMP%" call %REPLY_TEMP%
if "%OS%" == "" if exist %REPLY_TEMP% del %REPLY_TEMP%
if not "%OS%" == "" if exist "%REPLY_TEMP%" del %REPLY_TEMP%
if "%REPLY_AFTER%" == "0" set REPLY_AFTER=less than 1
if "%OS%" == "" goto fi_ert1
if "%OS%" == "Windows_9x" goto fi_ert1
title %GMS_FILE% - %REPLY_AFTER%
:fi_ert1
if "%OS%" == "" set REPLY_TEMP=backup_REPLY_TEMP%
goto exit
:fi_ert

REM Not found: -----
:else_c
call l_banner -no_action g_dos %1
:fi_c

:exit

```

g_file.bat

```
REM g_file.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_file=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
  echo Gerolf Markup Shredder (g_file) . . .
  pause > nul
  gerolf 2> nul
  goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
  call l_banner -debug g_file %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
  call l_banner -no_action g_file
  goto exit
:fi_nd

REM Resize: -----

if %1 == -update goto then_rs
if %1 == -resize goto then_rs
if %1 == -build goto then_rs
if not %1 == -remove goto fi_rs
:then_rs
  set REPLY_OFFSET=3
  set REPLY_SIZE=44
  set REPLY_ITEMS=13
  if %1 == -update goto then_u
  if %1 == -build goto then_b
  if %1 == -remove goto then_r
  goto exit
:fi_rs

REM Build: -----

if not %1 == -build goto fi_b
:then_b
  set GMS_STATE=%2
  set GMS_TOPLEVEL=
  call l_box -build called_by g_file %1
  call l_file -build called_by g_file %1
  call g_dos -pwd called_by g_file %1
  set GMS_FOLDER=%PWD%
  call l_banner -lower -folder called_by g_file %1
  set REPLY_LIST=
  call g_list -handle called_by g_file %1

  call l_list -build called_by g_file %1
  set GMS_HOT=N
  if "%GMS_STATE%" == "-folder" set GMS_HOT=H
  call l_file -update %GMS_HOT% %GMS_HOT% called_by g_file %1
  set REPLY_MODULE=g_file
  set REPLY_ACTION=-update
  set GMS_RECEIVE=1
  goto exit
:fi_b

REM Remove:-----

if not %1 == -remove goto fi_r
:then_r
  call l_box -remove
  if not %GMS_STATE% == -folder goto fi_r1
  REM Selected a folder to save the new file:
  call g_save -build called_by g_file %1
  goto fi_r2
:fi_r1
  if not %GMS_STATE% == -template goto else_r2
  REM Selected a template for the new file:
  call g_menu -resize called_by g_file %1
  call l_menu -update C Q called_by g_file %1
  set REPLY_MODULE=g_menu
  set REPLY_ACTION=-update
  set GMS_RECEIVE=1
  goto fi_r2
:else_r2
  REM Changed folder and opened markup file:
  call g_menu -resize called_by g_file %1
  call l_menu -update O Q called_by g_file %1
  set REPLY_MODULE=g_menu
  set REPLY_ACTION=-update
  set GMS_RECEIVE=1
:fi_r2
  set GMS_TOPLEVEL=
  goto exit
:fi_r

REM Update: -----

if not %1 == -update goto fi_u
:then_u
  REM Cold:
  if not "%3" == "" goto else_u1
  set GMS_COLD=%GMS_HOT%
  goto fi_u1
:else_u1
  set GMS_COLD=%3
:fi_u1
  REM Hot:
  if not "%2" == "" goto else_u2
  call l_banner -no_hot called_by g_file %1
  goto fi_u2
:else_u2
  set GMS_HOT=Q
  if %2 == 1 set GMS_HOT=1
  if %2 == 2 set GMS_HOT=2
  if %2 == 3 set GMS_HOT=3
  if %2 == 4 set GMS_HOT=4
  if %2 == 5 set GMS_HOT=5
  if %2 == 6 set GMS_HOT=6
  if %2 == 7 set GMS_HOT=7
  if %2 == 8 set GMS_HOT=8
  if %2 == 9 set GMS_HOT=9
  if %2 == 11 set GMS_HOT=P
  if %2 == P set GMS_HOT=P
```



```

if %2 == p set GMS_HOT=P
if %2 == l2 set GMS_HOT=N
if %2 == N set GMS_HOT=N
if %2 == n set GMS_HOT=N
if %2 == 13 set GMS_HOT=H
if %2 == H set GMS_HOT=H
if %2 == h set GMS_HOT=H
:fi_u2
REM Handle old, update new:
if not %GMS_HOT% == %GMS_COLD% goto else_u3
call g_file -handle %GMS_HOT%
goto fi_u3
REM goto then_h
:else_u3
call l_file -update %GMS_HOT% %GMS_COLD% called_by g_file %1
:fi_u3
set GMS_FILETYPE=
set GMS_NEWFILE=
set GMS_RECEIVE=1
goto exit
:fi_u

REM Handle: -----
if not %1 == -handle goto fi_h
:then_h

REM Quit:
if not %GMS_HOT% == Q goto fi_hq
call g_file -remove called_by g_file %1
goto exit
:fi_hq

REM Higher:
if not %GMS_HOT% == H goto fi_hh
set REPLY_LIST=
set GMS_FILE=
call l_banner -upper -first called_by g_file %1
if not "%OS%" == "" if not "%OS%" == "Windows_9x" title GMS %GMS_VERSION%
REM Check if it is the top level (drives):
call g_dos -pwd called_by g_file %1
set PWD_COLD=%PWD%
if "%GMS_DRDOS%" == "" if exist ..\%any% cd ..
if "%GMS_DRDOS%" == "" goto fi_hhdr
if "%PWD%" == "A:\ " goto fi_hhdr
if "%PWD%" == "B:\ " goto fi_hhdr
if "%PWD%" == "C:\ " goto fi_hhdr
if "%PWD%" == "D:\ " goto fi_hhdr
if "%PWD%" == "E:\ " goto fi_hhdr
if "%PWD%" == "F:\ " goto fi_hhdr
if "%PWD%" == "G:\ " goto fi_hhdr
if "%PWD%" == "H:\ " goto fi_hhdr
if "%PWD%" == "I:\ " goto fi_hhdr
if "%PWD%" == "J:\ " goto fi_hhdr
if "%PWD%" == "K:\ " goto fi_hhdr
if "%PWD%" == "L:\ " goto fi_hhdr
if "%PWD%" == "M:\ " goto fi_hhdr
if "%PWD%" == "N:\ " goto fi_hhdr
if "%PWD%" == "O:\ " goto fi_hhdr
if "%PWD%" == "P:\ " goto fi_hhdr
if "%PWD%" == "Q:\ " goto fi_hhdr
if "%PWD%" == "R:\ " goto fi_hhdr
if "%PWD%" == "S:\ " goto fi_hhdr
if "%PWD%" == "T:\ " goto fi_hhdr
if "%PWD%" == "U:\ " goto fi_hhdr
if "%PWD%" == "V:\ " goto fi_hhdr
if "%PWD%" == "W:\ " goto fi_hhdr
if "%PWD%" == "X:\ " goto fi_hhdr
if "%PWD%" == "Y:\ " goto fi_hhdr
if "%PWD%" == "Z:\ " goto fi_hhdr

```

```

cd ..
:fi_hhdr
call g_dos -pwd called_by g_file %1
set PWD_HOT=%PWD%
set GMS_FOLDER=%PWD%
REM Handle drive or file/folder/template:
if not "%OS%" == "Windows_9x" goto fi_w9x
if "%PWD_COLD%" == "A:" goto then_hhl
if "%PWD_COLD%" == "B:" goto then_hhl
if "%PWD_COLD%" == "C:" goto then_hhl
if "%PWD_COLD%" == "D:" goto then_hhl
if "%PWD_COLD%" == "E:" goto then_hhl
if "%PWD_COLD%" == "F:" goto then_hhl
if "%PWD_COLD%" == "G:" goto then_hhl
if "%PWD_COLD%" == "H:" goto then_hhl
if "%PWD_COLD%" == "I:" goto then_hhl
if "%PWD_COLD%" == "J:" goto then_hhl
if "%PWD_COLD%" == "K:" goto then_hhl
if "%PWD_COLD%" == "L:" goto then_hhl
if "%PWD_COLD%" == "M:" goto then_hhl
if "%PWD_COLD%" == "N:" goto then_hhl
if "%PWD_COLD%" == "O:" goto then_hhl
if "%PWD_COLD%" == "P:" goto then_hhl
if "%PWD_COLD%" == "Q:" goto then_hhl
if "%PWD_COLD%" == "R:" goto then_hhl
if "%PWD_COLD%" == "S:" goto then_hhl
if "%PWD_COLD%" == "T:" goto then_hhl
if "%PWD_COLD%" == "U:" goto then_hhl
if "%PWD_COLD%" == "V:" goto then_hhl
if "%PWD_COLD%" == "W:" goto then_hhl
if "%PWD_COLD%" == "X:" goto then_hhl
if "%PWD_COLD%" == "Y:" goto then_hhl
if "%PWD_COLD%" == "Z:" goto then_hhl
goto else_hhl
:fi_w9x
if not "%PWD_COLD%" == "%PWD_HOT%" goto else_hhl
:then_hhl
set GMS_TOPLEVEL=1
call g_list -drives called_by g_file %1
goto fi_hhl
:else_hhl
set GMS_TOPLEVEL=
call g_list -handle called_by g_file %1
:fi_hhl
set PWD_COLD=
set PWD_HOT=
REM Rebuild list:
call l_list -build called_by g_file %1
call l_banner -lower -folder called_by g_file %1
set GMS_RECEIVE=1
call l_file -update H - called_by g_file %1
goto exit
:fi_hh

REM Previous:
if not %GMS_HOT% == P goto fi_hp
if "%REPLY_LIST%" == "" goto fi_hpl
if not %REPLY_LIST% == * goto else_hpl
:then_hpl
set REPLY_LIST=
goto fi_hpl
:else_hpl
REM Diminish list offset by 1:
set INDEX=
set d=
set i=*
:loop
set i=%i%*
set d=%d%*
if %i%* == %REPLY_LIST%* goto pool

```

```

        goto loop
:pool
    set REPLY_LIST=%d%
    set d=
    set i=
:fi_hp1
REM Rebuild list:
    set GMS_RECEIVE=1
    if not "%GMS_TOPLEVEL%" == "" goto else_hp2
        call g_list -handle called_by g_file %1
        goto fi_hp2
    :else_hp2
        call g_list -drives called_by g_file %1
    :fi_hp2
    call l_list -build called_by g_file %1
    call l_file -update P - called_by g_file %1
    goto exit
:fi_hp

REM Next:
if not %GMS_HOT% == N goto fi_hn
if "%REPLY9%" == "" goto fi_hnn
    set REPLY_LIST=%REPLY_LIST%*
:fi_hnn
set GMS_RECEIVE=1
if not "%GMS_TOPLEVEL%" == "" goto else_hnm
    call g_list -handle called_by g_file %1
    goto fi_hnm
:else_hnm
    call g_list -drives called_by g_file %1
:fi_hnm
call l_list -build called_by g_file %1
call l_file -update N - called_by g_file %1
goto exit
:fi_hn

REM Selected file no. 1 - 9:
if not %GMS_HOT% == 1 goto fi_h1
    set GMS_NEWFILE=%REPLY1%
    set GMS_FILETYPE=%GMS1%
    goto then_hf
:fi_h1
if not %GMS_HOT% == 2 goto fi_h2
    set GMS_NEWFILE=%REPLY2%
    set GMS_FILETYPE=%GMS2%
    goto then_hf
:fi_h2
if not %GMS_HOT% == 3 goto fi_h3
    set GMS_NEWFILE=%REPLY3%
    set GMS_FILETYPE=%GMS3%
    goto then_hf
:fi_h3
if not %GMS_HOT% == 4 goto fi_h4
    set GMS_NEWFILE=%REPLY4%
    set GMS_FILETYPE=%GMS4%
    goto then_hf
:fi_h4
if not %GMS_HOT% == 5 goto fi_h5
    set GMS_NEWFILE=%REPLY5%
    set GMS_FILETYPE=%GMS5%
    goto then_hf
:fi_h5
if not %GMS_HOT% == 6 goto fi_h6
    set GMS_NEWFILE=%REPLY6%
    set GMS_FILETYPE=%GMS6%
    goto then_hf
:fi_h6
if not %GMS_HOT% == 7 goto fi_h7
    set GMS_NEWFILE=%REPLY7%
    set GMS_FILETYPE=%GMS7%
    goto then_hf

:fi_h7
if not %GMS_HOT% == 8 goto fi_h8
    set GMS_NEWFILE=%REPLY8%
    set GMS_FILETYPE=%GMS8%
    goto then_hf
:fi_h8
if not %GMS_HOT% == 9 goto fi_h9
    set GMS_NEWFILE=%REPLY9%
    set GMS_FILETYPE=%GMS9%
    goto then_hf
:fi_h9

REM Handle file or folder:

if not %GMS_HOT% == -entry goto fi_hf
:then_hf
    REM Clear all file/folder information:
    set backup_file=
    set GMS_FILE=
    set backup_folder=
    set GMS_FOLDER=
    set backup_short=
    set GMS_SHORT=
    set GMS_ERRSHORT=
    set GMS_LOGSHORT=
    set backup_base=
    set GMS_BASE=
    set GMS_ERRBASE=
    set GMS_LOGBASE=
    REM Check if it is a drive:
    if not %GMS_FILETYPE% == -[()] goto fi_hf1
        REM Change drive, update GMS_FOLDER:
        REM Fixme: check for spacy names
        %GMS_NEWFILE%
        set GMS_REMODRV=%GMS_NEWFILE%
        %GMS_REMODRV%
        call g_dos -pwd called_by g_file %1
        set GMS_FOLDER=%PWD%
        set GMS_FILE=
        goto then_hfo
    :fi_hf1
    REM Check if it is a folder:
    if not %GMS_FILETYPE% == -[\\] goto fi_hf2
        set GMS_FOLDER=%GMS_NEWFILE%
        if "%OS%" == "" cd %GMS_FOLDER% > nul
        if not "%OS%" == "" cd "%GMS_FOLDER%" > nul
        call g_dos -pwd called_by g_file %1
        set GMS_FOLDER=%PWD%
        set GMS_FILE=
        goto then_hfo
    :fi_hf2
    REM Then it will be a file:
    call g_dos -pwd called_by g_file %1
    set GMS_FOLDER=%PWD%
    set GMS_FILE=%GMS_NEWFILE%
    call g_dos -chdir called_by g_file %1
    if "%GMS_STATE%" == "" goto fi_hf3
    if %GMS_STATE% == -open goto then_hfi
    if %GMS_STATE% == -template goto then_hft
    :fi_hf3
    goto then_hfi
:fi_hf

REM Handle folder:
if not %GMS_HOT% == -folder goto fi_hfo
:then_hfo
    call l_banner -upper -first called_by g_file %1
    call l_banner -lower -folder called_by g_file %1
    set REPLY_LIST=
    set GMS_TOPLEVEL=
    call g_list -handle called_by g_file %1

```

```

call l_list -build called_by g_file %1
call l_file -update H 1 called_by g_file %1
set REPLY_MODULE=g_file
set REPLY_ACTION=-update
set GMS_RECEIVE=1
goto exit
:fi_hfo

REM Handle file:
if not %GMS_HOT% == -file goto fi_hfi
:then_hfi
if not "%GMS_FILE%" == "" goto else_hfj
call l_banner -upper -first called_by g_file %1
goto fi_hfj
:else_hfj
call g_vars -ini_file called_by g_file %1
REM call l_banner -upper -file called_by g_save %1
set REPLY_SIZE=50
set REPLY_OFFSET=0
reply -banner 1 "%GMS_FILE%"
if not "%OS%" == "" if not "%OS%" == "Windows_9x" title %GMS_FILE%
:fi_hfj
call g_file -resize called_by g_file %1
call l_box -remove called_by g_file %1
call g_menu -resize called_by g_file %1
call l_menu -update O Q called_by g_file %1
call g_vars -write_memo called_by g_file %1
set REPLY_MODULE=g_menu
set REPLY_ACTION=-update
set GMS_RECEIVE=1
set GMS_TOPELVEL=
goto exit
:fi_hfi

REM Handle template:
if not %GMS_HOT% == -template goto fi_ht
:then_hft
set GMS_TEMPLATEFILE=%GMS_NEWFILE%

call g_dos -pwd called_by g_file %1
set GMS_TFOLDER=%PWD%
call l_banner -upper -template called_by g_file %1
set GMS_FOLDER=%GMS_LASTDIR%
set GMS_LASTDIR=
call g_dos -chdir called_by g_file %1
REM Select folder:
set BACKUP_ANIMATE=%GMS_ANIMATE%
set GMS_ANIMATE=off
set GMS_FILEBOX=2. Select a folder to save this new copy
set GMS_FOLDER=%GMS_TEMPLATE%
if "%HOMEPATH%" == "" goto fi_ht1
%HOMEDRIVE%
set GMS_FOLDER=%HOMEDRIVE%\%HOMEPATH%
:fi_ht1
call g_dos -chdir called_by g_file %1
call g_file -build -folder called_by g_file %1
set GMS_FILEBOX=
set GMS_ANIMATE=%BACKUP_ANIMATE%
set BACKUP_ANIMATE=
set GMS_TOPELVEL=
goto exit
:fi_ht

REM Not found:
:else_h
call l_banner -no_hot g_file %GMS_HOT%
goto exit
:fi_h

REM Not found: .....

call l_banner -no_action g_file %1

:exit

set GMS_NEWFILE=

```

g_font.bat

```
REM g_font.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_FONT=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_font) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_font %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_font
goto exit
:fi_nd

REM Copy Windows True Type fonts to GMS fonts folder: -----

if not %1 == -copy_fonts goto fi_cf
if "%OS%" == "" goto exit
REM %2 supplier
REM %3 family
REM %4 typeface1
REM %5 typeface2
REM %6 typeface3
REM %7 typeface4
if not exist "%GMS_FONTS%\ttf\%any%" md "%GMS_FONTS%\ttf"
if not exist "%GMS_FONTS%\ttf\%2\%any%" md "%GMS_FONTS%\ttf\%2"
if not exist "%GMS_FONTS%\ttf\%2\%3\%any%" md "%GMS_FONTS%\ttf\%2\%3"
set GMS_SRC=%windir%\Fonts
set GMS_MISS=%GMS_SETTING%\fontmiss.log
if "%4" == "" goto fi_cf4
if not exist "%GMS_SRC%\%4.ttf" echo Missing: %GMS_SRC%\%4.ttf>> %Z%
if not exist "%GMS_SRC%\%4.ttf" echo %GMS_SRC%\%4.ttf>> "%GMS_MISS%"
if not exist "%GMS_SRC%\%4.ttf" goto fi_cf4
copy "%GMS_SRC%\%4.ttf" "%GMS_FONTS%\ttf\%2\%3" > nul
:fi_cf4
if "%5" == "" goto fi_cf5
if not exist "%GMS_SRC%\%5.ttf" echo Missing: %GMS_SRC%\%5.ttf>> %Z%
if not exist "%GMS_SRC%\%5.ttf" echo %GMS_SRC%\%5.ttf>> "%GMS_MISS%"
if not exist "%GMS_SRC%\%5.ttf" goto fi_cf5
copy "%GMS_SRC%\%5.ttf" "%GMS_FONTS%\ttf\%2\%3" > nul
:fi_cf5
if "%6" == "" goto fi_cf6
if not exist "%GMS_SRC%\%6.ttf" echo Missing: %GMS_SRC%\%6.ttf>> %Z%
if not exist "%GMS_SRC%\%6.ttf" echo %GMS_SRC%\%6.ttf>> "%GMS_MISS%"
if not exist "%GMS_SRC%\%6.ttf" goto fi_cf6
copy "%GMS_SRC%\%6.ttf" "%GMS_FONTS%\ttf\%2\%3" > nul
:fi_cf6
if "%7" == "" goto fi_cf7
if not exist "%GMS_SRC%\%7.ttf" echo Missing: %GMS_SRC%\%7.ttf>> %Z%
if not exist "%GMS_SRC%\%7.ttf" echo %GMS_SRC%\%7.ttf>> "%GMS_MISS%"
if not exist "%GMS_SRC%\%7.ttf" goto fi_cf7
copy "%GMS_SRC%\%7.ttf" "%GMS_FONTS%\ttf\%2\%3" > nul
:fi_cf7

REM Install GMS fonts for Windows: -----

if not %1 == -install_win goto fi_iw
if "%OS%" == "" goto exit
REM %2 supplier
REM %3 family
REM %4 typeface1
REM %5 typeface2
REM %6 typeface3
REM %7 typeface4
set GMS_TRG=%windir%\Fonts
set GMS_INST=%GMS_SETTING%\fontinst.log
if "%4" == "" goto fi_iw4
if exist "%GMS_TRG%\%4.ttf" goto fi_iw4
if not exist "%GMS_FONTS%\ttf\%2\%3\%4.ttf" goto fi_iw4
echo %4.ttf>> "%GMS_INST%"
copy "%GMS_FONTS%\ttf\%2\%3\%4.ttf" "%GMS_TRG%"> nul
:fi_iw4
if "%5" == "" goto fi_iw5
if exist "%GMS_TRG%\%5.ttf" goto fi_iw5
if not exist "%GMS_FONTS%\ttf\%2\%3\%5.ttf" goto fi_iw5
echo %5.ttf>> "%GMS_INST%"
copy "%GMS_FONTS%\ttf\%2\%3\%5.ttf" "%GMS_TRG%"> nul
:fi_iw5
if "%6" == "" goto fi_iw6
if exist "%GMS_TRG%\%6.ttf" goto fi_iw6
if not exist "%GMS_FONTS%\ttf\%2\%3\%6.ttf" goto fi_iw6
echo %6.ttf>> "%GMS_INST%"
copy "%GMS_FONTS%\ttf\%2\%3\%6.ttf" "%GMS_TRG%"> nul
:fi_iw6
if "%7" == "" goto fi_iw7
if exist "%GMS_TRG%\%7.ttf" goto fi_iw7
if not exist "%GMS_FONTS%\ttf\%2\%3\%7.ttf" goto fi_iw7
echo %7.ttf>> "%GMS_INST%"
copy "%GMS_FONTS%\ttf\%2\%3\%7.ttf" "%GMS_TRG%"> nul
:fi_iw7
set GMS_TRG=
set GMS_INST=
goto exit
:fi_iw

REM Copy Windows true type fonts: -----

if not %1 == -exchange_fonts goto fi_xf
if "%OS%" == "" goto exit
REM echo.
REM echo Copying: * .ttf - True Type Fonts from %windir%\Fonts ...
REM Log file for missing fonts: .....
set GMS_MISS=%GMS_SETTING%\fontmiss.log
echo fontmiss.log - Missing fonts in %windir%\Fonts>> "%GMS_MISS%"
```

```

set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%>> "%GMS_MISS%"
echo.>> "%GMS_MISS%"
REM Bigelow & Holmes fonts: .....
call g_font -copy_fonts bh lucon lucon
call g_font -copy_fonts bh lsans l_10646
REM ITC fonts: .....
call g_font -copy_fonts itc framd framd framdit
REM Microsoft fonts: .....
call g_font -copy_fonts ms comic comic comicbd
call g_font -copy_fonts ms georgia georgia georgiab georgial georgiaz
call g_font -copy_fonts ms sans micross
call g_font -copy_fonts ms sylfaen sylfaen
call g_font -copy_fonts ms tahoma tahoma tahomabd
call g_font -copy_fonts ms trebuc trebuc trebucbd trebucbi trebucit
call g_font -copy_fonts ms verdana verdana verdanab verdanai verdanz
REM Monotype fonts: .....
call g_font -copy_fonts mt aharoni ahronbd
call g_font -copy_fonts mt andalus andlso
call g_font -copy_fonts mt arial arial arialbd arialbi ariali
call g_font -copy_fonts mt artro arttrbdo artro
call g_font -copy_fonts mt cour cour courbd couri
call g_font -copy_fonts mt david david davidbd davidtr
call g_font -copy_fonts mt frank frank
call g_font -copy_fonts mt garamond gara garabd garait
call g_font -copy_fonts mt gothic gothic gothicb gothicbi gothici
call g_font -copy_fonts mt impact impact
call g_font -copy_fonts mt levenim lvnm lvnmbd
call g_font -copy_fonts mt miriam mriam mriamc mriamfx mriamtr
call g_font -copy_fonts mt narkisim nrkis
call g_font -copy_fonts mt rod rod rodtr
call g_font -copy_fonts mt simpo simpbdo simpfxo simpo
call g_font -copy_fonts mt times times timesbd timesbi timesi
call g_font -copy_fonts mt trado tradbdo trado
REM .....
set GMS_MISS=%GMS_SETTING%\fontmiss.log
echo.>> "%GMS_MISS%"
echo %arg%>> "%GMS_MISS%"
echo. %GMS_SETTING%>> "%GMS_MISS%"
set arg=
REM Now copying *.ttf from %GMS_FONTS% to %windir%\Fonts:
REM Log file for installed fonts: .....
set GMS_INST=%GMS_SETTING%\fontinst.log
echo fontinst.log - Installed fonts in %windir%\Fonts>> "%GMS_INST%"
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%>> "%GMS_INST%"
echo.>> "%GMS_INST%"
REM Bigelow & Holmes fonts: .....
call g_font -install_win bh lhandw lhandw
call g_font -install_win bh lsans lsansi
REM Bitstream fonts: .....
call g_font -install_win bt parkave parkave
call g_font -install_win bt zapfchan bzcb bzcm
REM Monotype fonts: .....
call g_font -install_win mt courthai crmthai crpthai
REM Public domain fonts: .....
call g_font -install_win pub ayummy ayummy
call g_font -install_win pub beauti beauti
call g_font -install_win pub cardo cardo
call g_font -install_win pub champ champ
call g_font -install_win pub garamond fgmr fgMRI fgmb fgmbi
call g_font -install_win pub gentium genr geni
call g_font -install_win pub shree ass1 ass2 ban1 ban2
call g_font -install_win pub shree dev1 dev2 guj1 guj2
call g_font -install_win pub shree kan1 kan2 mal1 mal2
call g_font -install_win pub shree ori1 ori2 pun1 pun2
call g_font -install_win pub shree tam1 tam2 tel1 tel2

```

```

call g_font -install_win pub shusha shusha
REM .....
set GMS_INST=%GMS_SETTING%\fontinst.log
echo.>> "%GMS_INST%"
echo %arg%>> "%GMS_INST%"
echo. %GMS_SETTING%>> "%GMS_INST%"
set arg=
goto exit
:fi_xf

REM Install fonts: .....

REM TDS: ../fonts/fonttype/supplier/typeface
REM where fonttype = afm, tfm, type1 (for pfa and pfb), ttf, vf.

REM According to the TeX directory structure, afm and pfa / pfb as well as ttf
REM files are stored in different trees. - Copies of them must be collected at
REM a common place, the tfm directory. The ttf fonts need computing of afm fi-
REM les first, before tfm and vf files can be created. After processing of the
REM metrics, all vf files are moved to the vf directory, while the pfa/pfb and
REM afm copies are deleted. The "core" font metrics get processed during this
REM procedure, too.
REM The tfm and vf metrics that have been processed algorithmically are *not*
REM stored in supplier/typeface subfolders, but in the top level of the tfm or
REM vf tree. At the beginning of the metric building process, the top levels
REM of the tfm and vf trees are cleared, but files in subfolders remain.

REM Handle typeface: .....
if not %1 == -handle_typeface goto fi_ht
REM %2 fonttype
REM %3 supplier
REM %4 typeface
if "%4" == "_ignore_" goto exit
if "%4" == "_IGNORE_" goto exit
if "%4" == "cm" goto exit
if "%4" == "Cm" goto exit
if "%4" == "CM" goto exit
if not "%OS%" == "" goto else_ht
if "%GMS_DRDOS%" == "" if not exist %GMS_FONTS%\%2\%3\%4\%any% goto exit
cd %GMS_FONTS%\%2\%3\%4
if exist *.%2 copy *.%2 %GMS_FONTS%\tfm > nul
if not %2 == type1 goto fi_ht1
if exist *.pfa copy *.pfa %GMS_FONTS%\tfm > nul
if exist *.pfb copy *.pfb %GMS_FONTS%\tfm > nul
:fi_ht1
goto exit
:else_ht
if not exist "%GMS_FONTS%\%2\%3\%4\%any%" goto exit
cd "%GMS_FONTS%\%2\%3\%4"
if exist *.%2 copy *.%2 "%GMS_FONTS%\tfm" > nul
if not %2 == type1 goto fi_ht2
if exist *.pfa copy *.pfa "%GMS_FONTS%\tfm" > nul
if exist *.pfb copy *.pfb "%GMS_FONTS%\tfm" > nul
:fi_ht2
goto exit
:fi_ht

REM Handle supplier: .....
if not %1 == -handle_supplier goto fi_hs
REM %2 fonttype
REM %3 supplier
if "%3" == "_ignore_" goto exit
if "%3" == "_IGNORE_" goto exit
if not "%OS%" == "" goto elshs1
if "%GMS_DRDOS%" == "" if not exist %GMS_FONTS%\%2\%3\%any% goto exit
cd %GMS_FONTS%\%2\%3
if exist *.%2 copy *.%2 %GMS_FONTS%\tfm > nul
goto fi_hs1
:elshs1

```

```

if not exist "%GMS_FONTS%\%2\%3\%any%" goto exit
cd "%GMS_FONTS%\%2\%3"
if exist *.%2 copy *.%2 "%GMS_FONTS%\tfm" > nul
:fi_hs1
if not %2 == type1 goto fi_hs2
if not "%OS%" == "" goto elsehs2
if exist *.pfa copy *.pfa %GMS_FONTS%\tfm > nul
if exist *.pfb copy *.pfb %GMS_FONTS%\tfm > nul
goto fi_hs2
:elsehs2
if exist *.pfa copy *.pfa "%GMS_FONTS%\tfm" > nul
if exist *.pfb copy *.pfb "%GMS_FONTS%\tfm" > nul
:fi_hs2
call g_dos -for_folder call g_font -handle_typeface %2 %3
if exist each.bat call each
if not "%OS%" == "" goto elsehs3
cd %GMS_FONTS%\%2\%3
goto fi_hs3
:elsehs3
cd "%GMS_FONTS%\%2\%3"
:fi_hs3
if exist each.bat del each.bat
goto exit
:fi_hs

REM Copy font files to tfm top level: -----
if not %1 == -copy_to_tfm goto fi_ctt
REM %2 fonttype
if "%2" == "_ignore_" goto exit
if "%2" == "_IGNORE_" goto exit
if not "%OS%" == "" goto elsect1
if "%GMS_DRDOS%" == "" if not exist %GMS_FONTS%\%2\%any% goto exit
cd %GMS_FONTS%\%2
if exist *.%2 copy *.%2 %GMS_FONTS%\tfm > nul
goto fi_ct1
:elsect1
if not exist "%GMS_FONTS%\%2\%any%" goto exit
cd "%GMS_FONTS%\%2"
if exist *.%2 copy *.%2 "%GMS_FONTS%\tfm" > nul
:fi_ct1
if not %2 == type1 goto fi_ct2
if not "%OS%" == "" goto elsect2
if exist *.pfa copy *.pfa %GMS_FONTS%\tfm > nul
if exist *.pfb copy *.pfb %GMS_FONTS%\tfm > nul
goto fi_ct2
:elsect2
if exist *.pfa copy *.pfa "%GMS_FONTS%\tfm" > nul
if exist *.pfb copy *.pfb "%GMS_FONTS%\tfm" > nul
:fi_ct2
call g_dos -for_folder call g_font -handle_supplier %2
if exist each.bat call each
if not "%OS%" == "" goto elsect3
cd %GMS_FONTS%\%2
goto fi_ct3
:elsect3
cd "%GMS_FONTS%\%2"
:fi_ct3
if exist each.bat del each.bat
goto exit
:fi_ctt

REM Build any font: -----
if not %1 == -build goto fi_bf
REM %2: glyph file name
REM %3: encoding
REM %4: core/eroc/embed/corefamily/embedfamily mark
REM %5: extension factor
REM %6: slanting factor
REM %7: new name (or 'none')
REM %8: new suffix (optional)
set enc=%3
if "%2" == ".*%4" goto fi_bf1
REM .....
set glyph=%2
REM Get font face file base name from glyph file name:
echo %glyph% > font_tmp.txt
sed "s/\. *$//;s/^\^/set GMS_BASE=/ < font_tmp.txt > map_tmp.bat
if exist map_tmp.bat call map_tmp
REM Modify name:
set esab=%GMS_BASE%
if "%7" == "" goto fi_at0
if %7 == none set esab=%esab%%8
if not %7 == none set esab=%7
:fi_at0
REM Change encoding for non-latin fonts: .....
if "%OS%" == "" set fo=%GMS_TEMP%\fo.bat
if not "%OS%" == "" set fo="%GMS_TEMP%\fo.bat"
REM if exist %fo% goto fiat00
if "%OS%" == "" set fi=%GMS_SETTING%\encoding.cfg
if not "%OS%" == "" set fi="%GMS_SETTING%\encoding.cfg"
if "%OS%" == "" set fq=%GMS_TEMP%\fq.bat
if not "%OS%" == "" set fq="%GMS_TEMP%\fq.bat"
set myif=if
sed "s/#.*$/;/^\^ *$/d;s/^\^ \[t]*$/;/s:/ set enc=/ <%fi% >%fq%"
REM sed "s/= \ */=/;s/^\^/myif% %GMS_BASE% == /" <%fq% >%fo%"
sed "s/= \ */=/;s/^\^/myif% %GMS_BASE% == /" <%fq% >%fo%"
if "%OS%" == "" if exist %fq% del %fq%
if not "%OS%" == "" if exist "%fq%" del "%fq%"
if not "%OS%" == "" goto fiat01
REM Lowercase conversion: .....
if "%OS%" == "" set fl=%GMS_TEMP%\fl.bat
if not "%OS%" == "" set fl="%GMS_TEMP%\fl.bat"
copy %fo% %fl% > nul
set fa=ABCDEFGHIJKLMNOPQRSTUVWXYZ
set fb=abcdefghijklmnopqrstuvwxyz
sed "y/%fa%/%fb%/" <%fl% >%fo%
if exist %fl% del %fl%
set fu=
set fa=
set fb=
set fl=
:fiat01
set myif=
set fq=
set fi=
:fiat00
if exist %fo% call %fo%
set fo=
REM Run ttf2afm: .....
if "%OS%" == "" if not exist %GMS_BASE%.ttf goto fi_bf2
if not "%OS%" == "" if not exist "%GMS_BASE%.ttf" goto fi_bf2
if "%OS%" == "" set ttfarg1=%GMS_ROOT%\data\enc\%enc%.enc
if not "%OS%" == "" set ttfarg1="%GMS_ROOT%\data\enc\%enc%.enc"
if "%OS%" == "" set ttfarg2=%GMS_BASE%.afm
if not "%OS%" == "" set ttfarg2="%GMS_BASE%.afm"
echo ttf2afm -e %ttfarg1% -o %ttfarg2% %glyph% >>1
ttf2afm -e %ttfarg1% -o %ttfarg2% %glyph% >nul
set ttfarg1=
set ttfarg2=
if "%OS%" == "" goto thenbf2
if "%OS%" == "Windows_9x" goto thenbf2
goto fi_bf2
:thenbf2
REM Remove warning messages:
cls
echo Running: GMS setup ...
if "%OS%" == "" type %GMS_SETTING%\writing.scn
if not "%OS%" == "" type "%GMS_SETTING%\writing.scn"
echo %GMS_FOLDER%

```

```

:fi_bf2
REM Define afm2tfm arguments: .....
if "%OS%" == "" if not exist %GMS_BASE%.afm goto fi_bf4
if not "%OS%" == "" if not exist "%GMS_BASE%.afm" goto fi_bf4
REM A redirection symbol (" < CP1252.enc ") is included in afm2tfm's
REM output, so afm2tfm must be put into its argument variable:
if not "%OS%" == "" goto fiat0w
set arguments=afm2tfm .\%GMS_BASE%.afm
set arguments=%arguments% -T %GMS_ROOT%\data\enc\%enc%.enc
set arguments=%arguments% -v %esab%.vpl
goto fiat0
:fiat0w
set arguments=afm2tfm .\%GMS_BASE%.afm
set arguments=%arguments% -T "%GMS_ROOT%\data\enc\%enc%.enc"
set arguments=%arguments% -v %esab%.vpl
:fiat0
REM Extend typeface: .....
if "%5%" == "" goto fi_at1
if %5 == 1 goto fi_at1
if %5 == 1.0 goto fi_at1
if %5 == 1.00 goto fi_at1
if %5 == 1.000 goto fi_at1
if %5 == 1.0000 goto fi_at1
if %5 == 1.00000 goto fi_at1
if %5 == hide goto fi_at1
set arguments=%arguments% -e %5
:fi_at1
REM Slant typeface: .....
if "%6%" == "" goto fi_at2
if %6 == 0 goto fi_at2
if %6 == 0.0 goto fi_at2
if %6 == 0.00 goto fi_at2
if %6 == 0.000 goto fi_at2
if %6 == 0.0000 goto fi_at2
if %6 == 0.00000 goto fi_at2
set arguments=%arguments% -s %6
:fi_at2
REM Display font file base name: .....
set REPLY_OFFSET=0
set REPLY_SIZE=82
reply -banner 19 ""
reply -banner 20 ""
reply -banner 21 ""
set REPLY_OFFSET=43
set REPLY_SIZE=39
reply -banner 17 "(%enc%) %GMS_BASE%"
REM Run afm2tfm, build map item, eventually add glyph: .....
echo %arguments% _%esab%.tfm>>1
%arguments% _%esab%.tfm>i
if "%4%" == "core" goto else_at3
if "%4%" == "eroc" goto else_at3
sed "s/<.*$/<%enc%.enc <glyph%/ " <i >m
if "%OS%" == "" type m >> %GMS_SETTING%\unsort.map
if "%OS%" == "" goto fi_at3
type m >>"%GMS_SETTING%\unsort.map"
goto fi_at3
:else_at3
sed "s/<.*$/<%enc%.enc/" <i >m
if "%OS%" == "" type m >> %GMS_SETTING%\unsort.map
if "%OS%" == "" goto fi_at3
type m >>"%GMS_SETTING%\unsort.map"
:fi_at3
set arguments=
REM Run vptovf, move processed core afm fonts to _trans_: .....
echo.
echo vptovf %esab%.vpl %esab%.vf %esab%.tfm>>1
vptovf %esab%.vpl %esab%.vf %esab%.tfm
if not "%4%" == "eroc" goto fi_at4
if not "%OS%" == "" goto fiat4w
if exist %GMS_BASE%.afm copy %GMS_BASE%.afm ..\_trans_ > nul
if exist %GMS_BASE%.afm del %GMS_BASE%.afm

goto fi_at4
:fiat4w
if exist "%GMS_BASE%.afm" copy "%GMS_BASE%.afm" ..\_trans_ > nul
if exist "%GMS_BASE%.afm" del "%GMS_BASE%.afm"
:fi_at4
:fi_bf4
set GMS_BASE=
set esab=
set glyph=
REM .....
:fi_bf1
echo.>>1
set enc=
goto exit
:fi_bf

REM Build specifically encoded fonts: .....
if not %1 == -build_encode goto fi_enc
REM %2: encoding
if "%2%" == "" goto exit
for %i in (%2*.afm) do call g_font -build %i %2 eroc
for %i in (%2*.pfa) do call g_font -build %i %2 embed
for %i in (%2*.pfb) do call g_font -build %i %2 embed
for %i in (%2*.ttf) do call g_font -build %i %2 embed
goto exit
:fi_enc

REM Build transformed fonts: .....
if not %1 == -build_trans goto fi_trs
REM %2: base name
REM %3: encoding
REM %4: core/embed mark
REM %5: extension factor
REM %6: slanting factor
REM %7: new name (or 'none')
REM %8: new suffix (optional)
if "%4%" == "" goto exit
REM Display font file base name: .....
set REPLY_OFFSET=14
set REPLY_SIZE=68
reply -banner 17 "*.tfm - TeX Font Metrics ... (%3) %2"
REM These lines contain cleaning spaces up to column 79:
echo.
echo.
echo.
echo.
echo.
if not %4 == core goto fi_trs1
REM This is only possible if a type1 glyph file is present:
if exist %2.afm call g_font -build %2.pfa %3 %4 %5 %6 %7 %8
:fi_trs1
if not %4 == corefamily goto fi_trs2
for %i in (%2*.afm) do call g_font -build %i %3 embed %5 %6 %7 %8
:fi_trs2
if not %4 == embed goto fi_trs3
if exist %2.pfa call g_font -build %2.pfa %3 %4 %5 %6 %7 %8
if exist %2.pfb call g_font -build %2.pfb %3 %4 %5 %6 %7 %8
if exist %2.ttf call g_font -build %2.ttf %3 %4 1.0 0.0 %7 %8
:fi_trs3
if not %4 == embedfamily goto fi_trs4
for %i in (%2*.pfa) do call g_font -build %i %3 embed %5 %6 %7 %8
for %i in (%2*.pfb) do call g_font -build %i %3 embed %5 %6 %7 %8
for %i in (%2*.ttf) do call g_font -build %i %3 embed %5 %6 %7 %8
:fi_trs4
goto exit
:fi_trs

REM Build all fonts: .....

```

```

if not %1 == -build_all goto fi_f1
set REPLY_BANNER=0
set REPLY_TEXT=7
set REPLY_OFFSET=0
set REPLY_SIZE=82
if not %2 == "" goto fi_af0
cls
echo %GMS_FILE%
if "%OS%" == "" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" type "%GMS_SETTING%\desktop.scn"
echo GMS: Running font metric processor ...
reply -banner 14
reply -banner 15 "Please remember to re-initialize the format"
set REPLY_OFFSET=45
set REPLY_SIZE=37
reply -banner 15 "file after re-writing the font map"
:fi_af0
set REPLY_OFFSET=0
set REPLY_SIZE=82
reply -banner 22
reply -banner 16
reply -banner 21
reply -banner 17
reply -banner 20
reply -banner 18
reply -banner 19
REM Delete old tfm and vf; get afm, ttf, pfa/pfb: .....
set REPLY_OFFSET=0
set REPLY_SIZE=32
reply -banner 17 "Collecting: Font files ..."
echo.
echo.
if not "%OS%" == "" goto else_os
echo          These processes require several megabytes of free
echo          disk space, depending on the number of fonts.
goto fi_os
:else_os
echo          These processes may run faster if there is no file
echo          manager task to be updated in the background.
:fi_os
if "%OS%" == "" if exist %GMS_FONTS%\tfm\%any% cd %GMS_FONTS%\tfm
if not "%OS%" == "" if exist "%GMS_FONTS%\tfm\%any%" cd "%GMS_FONTS%\tfm"
if not "%GMS_DRDOS%" == "" cd %GMS_FONTS%\tfm
if exist *.tfm del *.tfm
if "%OS%" == "" if exist %GMS_FONTS%\vf\%any% cd %GMS_FONTS%\vf
if not "%OS%" == "" if exist "%GMS_FONTS%\vf\%any%" cd "%GMS_FONTS%\vf"
if not "%GMS_DRDOS%" == "" cd %GMS_FONTS%\vf
if exist *.vf del *.vf
set REPLY_OFFSET=14
set REPLY_SIZE=68
reply -banner 17 "*.ttf - True Type Fonts ..."
call g_font -exchange_fonts called_by gmssetup
call g_font -copy_to_tfm ttf
reply -banner 17 "*.*.pfa - Postscript Fonts ... Ascii/Binary"
call g_font -copy_to_tfm type1
reply -banner 17 "*.*.afm - Adobe Font Metrics ..."
call g_font -copy_to_tfm afm
REM Get new tfm and vf: .....
set REPLY_OFFSET=0
set REPLY_SIZE=14
reply -banner 17 Computing:
set REPLY_OFFSET=14
set REPLY_SIZE=68
reply -banner 17 "*.*.tfm - TeX Font Metrics ..."
if "%OS%" == "" if exist %GMS_SETTING%\%any% cd %GMS_SETTING%
if not "%OS%" == "" if exist "%GMS_SETTING%\%any%" cd "%GMS_SETTING%"
if not "%GMS_DRDOS%" == "" cd %GMS_SETTING%
if exist font.map del font.map
if exist unsort.map del unsort.map
if "%OS%" == "" if exist %GMS_FONTS%\%any% cd %GMS_FONTS%
if not "%OS%" == "" if exist "%GMS_FONTS%\%any%" cd "%GMS_FONTS%"

if not "%GMS_DRDOS%" == "" cd %GMS_FONTS%
if "%OS%" == "Windows 9x" goto th_98_1
REM %any% in next line for Dos 3x:
if exist _trans_\%any% echo j | del _trans_ > nul
goto fi_98_1
:th_98_1
if exist _trans_del_trans_ > nul
:fi_98_1
if not "%GMS_DRDOS%" == "" echo Delete me! > _trans_\deleteme.txt > nul
if not exist _trans_\%any% md _trans_
set REPLY_OFFSET=0
set REPLY_SIZE=80
reply -banner 18 ""
set REPLY_SIZE=68
set REPLY_OFFSET=14
REM Process font files by encoding: .....
if "%OS%" == "" if exist %GMS_ROOT%\data\enc\%any% cd %GMS_ROOT%\data\enc
if not "%OS%" == "" if exist "%GMS_ROOT%\%any%" cd "%GMS_ROOT%" > nul
if not "%OS%" == "" if exist "data\enc\%any%" cd "data\enc"
if not "%GMS_DRDOS%" == "" goto else_dr
REM 'dir /b' fails on MS-DOS 3x, so use 'mdir':
if "%GMS_DRDOS%" == "" if "%OS%" == "" set argdr=mdir /b
if "%GMS_DRDOS%" == "" if not "%OS%" == "" set argdr=dir /b
echo j | %argdr% *.enc > enc.txt
sed "s/^\s*call g_font -build_encode /;s/^\s*$/ /" < enc.txt > enc.bat
set argdr=
if "%OS%" == "" if exist enc.bat copy enc.bat %GMS_FONTS%\tfm > nul
if not "%OS%" == "" if exist enc.bat copy enc.bat "%GMS_FONTS%\tfm">nul
goto fi_dr
:else_dr
echo j | xdir *.enc /b > enc.txt
sed "s/^\s*call g_font -build_encode /;s/^\s*$/ /" < enc.txt > enc.bat
if exist enc.bat copy enc.bat %GMS_FONTS%\tfm > nul
:fi_dr
if exist enc.bat del enc.bat
if exist enc.txt del enc.txt
if "%OS%" == "" if exist %GMS_FONTS%\tfm\%any% cd %GMS_FONTS%\tfm
if not "%OS%" == "" if exist "%GMS_FONTS%\tfm\%any%" cd "%GMS_FONTS%\tfm"
if not "%GMS_DRDOS%" == "" cd %GMS_FONTS%\tfm
if exist enc.bat call enc
if exist enc.bat del enc.bat
REM Process font files by type: .....
if "%OS%" == "" if exist %GMS_FONTS%\tfm\%any% cd %GMS_FONTS%\tfm
if not "%OS%" == "" if exist "%GMS_FONTS%\tfm\%any%" cd "%GMS_FONTS%\tfm"
if not "%GMS_DRDOS%" == "" cd %GMS_FONTS%\tfm
for %i in (pca*.afm) do call g_font -build %i %GMS_CODEPAGE% eroc
for %i in (phv*.afm) do call g_font -build %i %GMS_CODEPAGE% eroc
for %i in (ptm*.afm) do call g_font -build %i %GMS_CODEPAGE% eroc
for %i in (*.pfa) do call g_font -build %i %GMS_CODEPAGE% embed
for %i in (*.pfb) do call g_font -build %i %GMS_CODEPAGE% embed
for %i in (*.ttf) do call g_font -build %i %GMS_CODEPAGE% embed
REM Process transformed files: .....
if "%OS%" == "" if exist %GMS_FONTS%\_trans_\%any% cd %GMS_FONTS%\_trans_
if not "%OS%" == "" if exist "%GMS_FONTS%\_trans_\%any%" cd "%GMS_FONTS%\_trans_"
if not "%OS%" == "" if exist "%GMS_FONTS%\_trans_\%any%" cd "%GMS_FONTS%\_trans_"
if not "%GMS_DRDOS%" == "" cd %GMS_FONTS%\_trans_
if exist *.afm copy *.afm ..\tfm > nul
if exist *.afm del *.afm
if "%OS%" == "" if exist %GMS_FONTS%\_trans_\%any% cd %GMS_FONTS%\_trans_
if not "%OS%" == "" if exist "%GMS_FONTS%\_trans_\%any%" cd "%GMS_FONTS%\_trans_"
if not "%GMS_DRDOS%" == "" cd %GMS_FONTS%\_trans_
rmdir _trans_
if "%OS%" == "Windows 9x" goto th_98_2
if exist _trans_echo j | del _trans_ > nul
goto fi_98_2
:th_98_2
if exist _trans_del_trans_ > nul
:fi_98_2
if "%OS%" == "" if exist %GMS_SETTING%\%any% cd %GMS_SETTING%
if not "%OS%" == "" if exist "%GMS_SETTING%\%any%" cd "%GMS_SETTING%"

```



```

if not %GMS_DRDOS% == "" cd %GMS_SETTING%
if not exist font.cfg goto fi_af
sed "s/^\.#.*$/;/^\ *$/d;" < font.cfg > snart.gfc
sed "s/^\call g_font -build_trans/" < snart.gfc > trans.bat
if exist snart.gfc del snart.gfc
if "%OS%" == "" if exist trans.bat copy trans.bat %GMS_FONTS%\tfm>nul
if not "%OS%" == "" if exist trans.bat copy trans.bat "%GMS_FONTS%\tfm">nul
if exist trans.bat del trans.bat
if "%OS%" == "" if exist %GMS_FONTS%\tfm\%any% cd %GMS_FONTS%\tfm
if not "%OS%" == "" if exist "%GMS_FONTS%\tfm\%any%" cd "%GMS_FONTS%\tfm"
if not %GMS_DRDOS% == "" cd %GMS_FONTS%\tfm
if exist trans.bat call trans
if exist trans.bat del trans.bat
:fi_af
if "%OS%" == "" cd %GMS_TEMP% > nul
if not "%OS%" == "" cd "%GMS_TEMP%" > nul
if "%OS%" == "" if exist %GMS_TEMP%\%any% cd %GMS_TEMP%
if not "%OS%" == "" if exist "%GMS_TEMP%\%any%" cd "%GMS_TEMP%"
if not %GMS_DRDOS% == "" cd %GMS_TEMP%
if exist fo.bat del fo.bat > nul
REM Install virtual fonts (and clean up the tfm folder): .....
set REPLY_OFFSET=0
set REPLY_SIZE=15
reply -banner 17 Installing:
set REPLY_OFFSET=14
set REPLY_SIZE=68
reply -banner 17 *.vf* - Virtual Fonts ... %GMS_CODEPAGE%
if "%OS%" == "" if exist %GMS_FONTS%\tfm\%any% cd %GMS_FONTS%\tfm
if not "%OS%" == "" if exist "%GMS_FONTS%\tfm\%any%" cd "%GMS_FONTS%\tfm"
if not %GMS_DRDOS% == "" cd %GMS_FONTS%\tfm
if exist *.afm del *.afm
if exist *.pfa del *.pfa
if exist *.pfb del *.pfb
if exist *.ttf del *.ttf
if exist *.vpl del *.vpl
if "%OS%" == "" if exist *.vf copy *.vf %GMS_FONTS%\vf > nul
if not "%OS%" == "" if exist *.vf copy *.vf "%GMS_FONTS%\vf" > nul
if exist *.vf del *.vf
if exist font_tmp.txt del font_tmp.txt
if exist map_tmp.bat del map_tmp.bat
if exist i del i
if exist m del m
if exist short.txt del short.txt
if exist short.bat del short.bat
REM Build font map (and log): .....
set slashline=////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
if "%OS%" == "Windows_9x" set slashline=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set slashline=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%OS%" == "" if exist %GMS_SETTING%\%any% cd %GMS_SETTING%
if not "%OS%" == "" if exist "%GMS_SETTING%\%any%" cd "%GMS_SETTING%"
if exist encoding.bat del encoding.bat
if "%OS%" == "" goto fi_f11
if "%OS%" == "Windows_9x" goto fi_f11
echo % font.log - Warnings, errors, missing glyphs> font.log
echo %% %slashline%%slashline%%>> font.log
:fi_f11
set arguments=%GMS_FONTS%\[font-type]\[supplier]\[font-family]
echo % font.map - %arguments%> head.map
set arguments=
if "%OS%" == "" goto fi_f13
if "%OS%" == "Windows_9x" goto fi_f12
echo %% ////////////////////////////////////////////////////////////////////>> head.map
:fi_f12
if not "%OS%" == "Windows_9x" goto fi_f13
echo %% XXXXXXXXXXXXslashline%%slashlineXXXXXXXXXXXX>> head.map
:fi_f13
echo.>> head.map
echo %% Generated by Gerolf Markup Shredder (www.Gerolf.org)>>head.map
echo.>> head.map
echo %% _[1] Base name of font files>> head.map
echo %% [2] PostScript font face name>> head.map
echo %% [3] ReEncodeFont " PostScript encoding name>> head.map
echo %% ([4] Encoding file [*enc]>> head.map
echo %% ([5] Glyph file [*pf*, *.ttf]>> head.map
echo.>> head.map
sed "s/(\[4])/<[4]/;s/(\[5])/<[5]/" < head.map > header.map
if exist unsort.map sort < unsort.map > sort.map
copy header.map + sort.map font.pam > nul
sed "s/\\//g" < font.pam > font.map
if not "%OS%" == "" goto fi_no89
if "%OS%" == "" sed "/^\.\{8\} /d" < font.map > font.pam
if "%OS%" == "" sed "/^\.\{9\} /d" < font.pam > font.map
:fi_no89
if exist font.pam del font.pam
if exist sort.map del sort.map
if exist header.map del header.map
if exist unsort.map del unsort.map
if exist head.map del head.map
if "%OS%" == "" if exist %GMS_FONTS%\tfm\%any% cd %GMS_FONTS%\tfm
if not "%OS%" == "" if exist "%GMS_FONTS%\tfm\%any%" cd "%GMS_FONTS%\tfm"
if not %GMS_DRDOS% == "" cd %GMS_FONTS%\tfm
if not exist l goto fi_nofm
if "%OS%" == "" copy l %GMS_SETTING%\fontmap.log > nul
if not "%OS%" == "" copy l "%GMS_SETTING%\fontmap.log" > nul
if exist l del l > nul
:fi_nofm
set slashline=
set REPLY_OFFSET=0
set REPLY_SIZE=15
reply -banner 17 Writing:
set REPLY_OFFSET=14
set REPLY_SIZE=68
reply -banner 17 " %GMS_SETTING%\font.map"
if "%OS%" == "" if exist %GMS_SETTING%\%any% cd %GMS_SETTING%
if not "%OS%" == "" if exist "%GMS_SETTING%\%any%" cd "%GMS_SETTING%"
if "%OS%" == "" if exist font.log del font.log > nul
set i=
set l=
set m=
goto exit
:fi_f1
REM Not found: .....
:else
call l_banner -no_action g_font %1
:fi
:exit

```

g_good.bat

```
REM g_good.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_GOOD=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_good) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if %GMS_DEBUG% == Y goto then_db
goto fi_db
:then_db
call l_banner -debug g_good %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_good
goto exit
:fi_nd

REM Resize: -----

if %1 == -resize goto then_rs
if %1 == -build goto then_rs
if not %1 == -remove goto fi_rs
:then_rs
set REPLY_OFFSET=3
set REPLY_SIZE=44
set REPLY_ITEMS=1
if %1 == -build goto then_b
if %1 == -remove goto then_r
goto exit
:fi_rs

REM Build: -----

if not %1 == -build goto fi_b
:then_b
if "%OS%" == "" goto fi_b1
if "%OS%" == "Windows_9x" goto fi_b1
title Please support the author of Markup Shredder - www.Gerolf.org> nul
:fi_b1
call l_banner -lower -first
call l_banner -upper -last
call l_box -build
call l_good -build
set GMS_HOT=Q
call l_good -update Q Q
set REPLY_MODULE=g_good
set REPLY_ACTION=-remove
set GMS_RECEIVE=1
goto exit
:fi_b

REM Remove: -----

if not %1 == -remove goto fi_r
:then_r
call l_good -update - Q
set backup_offset=%REPLY_OFFSET%
set backup_size=%REPLY_SIZE%
call g_menu -remove called_by g_good -remove
set REPLY_SIZE=%backup_size%
set REPLY_OFFSET=%backup_offset%
set backup_size=
set backup_offset=
call l_box -remove
call l_desk -remove
cls
if "%OS%" == "" type %GMS_SETTING%\launch_x.scn
if not "%OS%" == "" type "%GMS_SETTING%\launch_x.scn"
echo www.Gerolf.org
call gms -q called_by g_good -remove > nul
set GMS_TEXTMODE=
set GMS_BREAK=1
goto exit
:fi_r

REM Not found: -----

:else
call l_banner -no_action g_good %1
:fi

:exit
```

g_launch.bat

```
REM g_launch.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2005).

REM set GMSdateG_LAUNCH=20060927

REM Prologue: =====

set line#####
set dash-----

REM Not running: -----

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_launch) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_launch %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_launch
goto exit
:fi_nd

REM Build 'gerolf' launcher script: -----

if not %1 == -build goto fi_b
echo @echo off > g
if "%GMS_DRDOS%" == "" goto fi_nodr
echo echo off > g
echo cls >> g
:fi_nodr
echo. >> g
echo REM gerolf.bat>> g
echo REM =====>> g
echo. >> g
echo REM Launcher script for Gerolf Markup Shredder (Dos).>> g
echo.>> g
echo REM Web: www.Gerolf.org>> g
echo REM eMail: MarkupShredder@Gerolf.org>> g
echo.>> g
echo REM %line%>> g
echo.>> g

echo REM 0) Start shell: ----->> g
echo.>> g
echo if (%1) == (-passive) goto fi_sh>> g
echo if (%1) == (-shell) goto else_sh>> g
set args=%1 %2 %3 %4 %5 %6 %7 %8 %9
set quot=

if not "%OS%" == "" set quot="
if "%OS%" == "Windows 9x" set quot=
echo %%comspec% /E:4096 /C %quot%%0 -shell %args%%quot%>>g
set quot=
set args=
echo goto exit>> g
echo :else_sh>> g
echo shift>> g
echo :fi_sh>> g
echo.>> g

echo REM 1) General settings: ----->> g
echo.>> g
echo REM Setting directory:>>g
if not "%GMS_DRDOS%" == "" echo set GMS_DRDOS=%GMS_DRDOS%>> g
if not "%GMS_FreeDOS%" == "" echo set GMS_FreeDOS=%GMS_FreeDOS%>> g
echo set GMS_SETTING=%GMS_SETTING%>> g
if not "%GMS_FreeDOS%" == "" goto else_fd
echo if (%2) == (-noclearscreen) goto fi_fdq>> g
echo if (%GMS_MODE%) == (quiet) goto fi_fdq>> g
REM echo mode con lines=25>> g
echo cls>> g
echo :fi_fdq>> g
goto fi_fd
:else_fd
echo if not (%2) == (-noclearscreen) mode co80,25>> g
:fi_fd
if "%OS%" == "" set GMS_CMD=%GMS_SETTING%\launch_1.scn
if not "%OS%" == "" set GMS_CMD=%GMS_SETTING%\launch_1.scn"
echo if not (%1) == (-passive) type %GMS_CMD%>> g
echo REM Main directories (can be web server subfolders):>>g
echo set GMS_ROOT=%GMS_ROOT%>> g
echo set GMS_DRIVE=%GMS_DRIVE%>> g
echo REM Debugging mode (0, 1, 2, 3, ... or X, Y, Z. Not empty!):>> g
REM These lines end with one space:
if -%GMS_DEBUG% == - set GMS_DEBUG=0
if %GMS_DEBUG% == 0 set GMS_DEBUG=0
if %GMS_DEBUG% == 1 set GMS_DEBUG=1
if %GMS_DEBUG% == 2 set GMS_DEBUG=2
if %GMS_DEBUG% == 3 set GMS_DEBUG=3
if %GMS_DEBUG% == 4 set GMS_DEBUG=4
if %GMS_DEBUG% == 5 set GMS_DEBUG=5
if %GMS_DEBUG% == 6 set GMS_DEBUG=6
if %GMS_DEBUG% == 7 set GMS_DEBUG=7
if %GMS_DEBUG% == 8 set GMS_DEBUG=8
if %GMS_DEBUG% == 9 set GMS_DEBUG=9
echo set GMS_DEBUG=%GMS_DEBUG%>> g
if "%OS%" == "" goto fi_deb
echo REM Rewrite gmsdebug log file:>> g
echo set Z=%GMS_SETTING%\gmsdebug.log">> g
echo if "%4" == "gmssetup" goto fi_debug>> g
echo if exist %%Z%% del %%Z%%>> g
echo set gmshead=%GMS_SETTING%\_gmshead.txt">> g
echo if exist %%gmshead% ren %%gmshead% gmsdebug.log>> g
echo set gmshead=>> g
echo goto fi_debug>> g
echo :fi_debug>> g
:fi_deb
echo REM Main codepage name:>> g
echo set GMS_CODEPAGE=%GMS_CODEPAGE%>> g
echo REM Main codepage number (for chcp-command, if %OS% defined):>>g
call g_dos -get_chcp
echo set GMS_CHCP=%GMS_CHCP%>> g
echo REM Initial codepage number (before running gmssetup):>>g
if not "%GMS_CURCP%" == "" set GMS_INICP=%GMS_CURCP%
echo set GMS_INICP=%GMS_INICP%>> g
echo.>> g
echo REM 2) Programs in search path: ----->> g
```

```

echo.>> g
echo REM Plain text viewer and editor:>> g
echo set GMS_VIEWER=%GMS_VIEWER%>> g
echo set GMS_EDITOR=%GMS_EDITOR%>> g
echo REM HTML browser, syntax checker, typesetting engine:>> g
echo set GMS_BROWSER=%GMS_BROWSER%>> g
echo set GMS_ANALYST=%GMS_ANALYST%>> g
echo set GMS_TSETTER=%GMS_TSETTER%>> g
echo REM PDF reader:>> g
echo set GMS_READER=%GMS_READER%>> g
echo.>> g
echo REM 3) Textmode interface variables: ----->> g
echo.>> g
echo REM Menu animation ('on' or 'off'):>> g
echo set GMS_ANIMATE=%GMS_ANIMATE%>> g
echo REM Colors and pattern ('R'== 'Random'):>> g
echo REM Foreground colors (0 to 15):>> g
REM These lines end with one space:
if %GMS_TEXT% == 0 set GMS_TEXT=0
if %GMS_TEXT% == 1 set GMS_TEXT=1
if %GMS_TEXT% == 2 set GMS_TEXT=2
if %GMS_TEXT% == 3 set GMS_TEXT=3
if %GMS_TEXT% == 4 set GMS_TEXT=4
if %GMS_TEXT% == 5 set GMS_TEXT=5
if %GMS_TEXT% == 6 set GMS_TEXT=6
if %GMS_TEXT% == 7 set GMS_TEXT=7
if %GMS_TEXT% == 8 set GMS_TEXT=8
if %GMS_TEXT% == 9 set GMS_TEXT=9
echo set GMS_TEXT=%GMS_TEXT%>> g
if %GMS_HOTKEY% == 0 set GMS_HOTKEY=0
if %GMS_HOTKEY% == 1 set GMS_HOTKEY=1
if %GMS_HOTKEY% == 2 set GMS_HOTKEY=2
if %GMS_HOTKEY% == 3 set GMS_HOTKEY=3
if %GMS_HOTKEY% == 4 set GMS_HOTKEY=4
if %GMS_HOTKEY% == 5 set GMS_HOTKEY=5
if %GMS_HOTKEY% == 6 set GMS_HOTKEY=6
if %GMS_HOTKEY% == 7 set GMS_HOTKEY=7
if %GMS_HOTKEY% == 8 set GMS_HOTKEY=8
if %GMS_HOTKEY% == 9 set GMS_HOTKEY=9
echo set GMS_HOTKEY=%GMS_HOTKEY%>> g
if %GMS_PATTERN% == 0 set GMS_PATTERN=0
if %GMS_PATTERN% == 1 set GMS_PATTERN=1
if %GMS_PATTERN% == 2 set GMS_PATTERN=2
if %GMS_PATTERN% == 3 set GMS_PATTERN=3
if %GMS_PATTERN% == 4 set GMS_PATTERN=4
if %GMS_PATTERN% == 5 set GMS_PATTERN=5
if %GMS_PATTERN% == 6 set GMS_PATTERN=6
if %GMS_PATTERN% == 7 set GMS_PATTERN=7
if %GMS_PATTERN% == 8 set GMS_PATTERN=8
if %GMS_PATTERN% == 9 set GMS_PATTERN=9
echo set GMS_PATTERN=%GMS_PATTERN%>> g
echo REM Initial values:>>g
echo set REPLY_TEXT=15 >> g
echo set REPLY_HOTKEY=14 >> g
echo set REPLY_PATTERN=13 >> g
echo REM Background colors (0 to 7):>> g
if %GMS_BANNER% == 0 set GMS_BANNER=0
if %GMS_BANNER% == 1 set GMS_BANNER=1
if %GMS_BANNER% == 2 set GMS_BANNER=2
if %GMS_BANNER% == 3 set GMS_BANNER=3
if %GMS_BANNER% == 4 set GMS_BANNER=4
if %GMS_BANNER% == 5 set GMS_BANNER=5
if %GMS_BANNER% == 6 set GMS_BANNER=6
if %GMS_BANNER% == 7 set GMS_BANNER=7
if %GMS_BANNER% == 8 set GMS_BANNER=8
if %GMS_BANNER% == 9 set GMS_BANNER=9
echo set GMS_BANNER=%GMS_BANNER%>> g
if %GMS_SHADE% == 0 set GMS_SHADE=0
if %GMS_SHADE% == 1 set GMS_SHADE=1
if %GMS_SHADE% == 2 set GMS_SHADE=2
if %GMS_SHADE% == 3 set GMS_SHADE=3
if %GMS_SHADE% == 4 set GMS_SHADE=4
if %GMS_SHADE% == 5 set GMS_SHADE=5
if %GMS_SHADE% == 6 set GMS_SHADE=6
if %GMS_SHADE% == 7 set GMS_SHADE=7
if %GMS_SHADE% == 8 set GMS_SHADE=8
if %GMS_SHADE% == 9 set GMS_SHADE=9
echo set GMS_SHADE=%GMS_SHADE%>> g
if %GMS_DESKTOP% == 0 set GMS_DESKTOP=0
if %GMS_DESKTOP% == 1 set GMS_DESKTOP=1
if %GMS_DESKTOP% == 2 set GMS_DESKTOP=2
if %GMS_DESKTOP% == 3 set GMS_DESKTOP=3
if %GMS_DESKTOP% == 4 set GMS_DESKTOP=4
if %GMS_DESKTOP% == 5 set GMS_DESKTOP=5
if %GMS_DESKTOP% == 6 set GMS_DESKTOP=6
if %GMS_DESKTOP% == 7 set GMS_DESKTOP=7
if %GMS_DESKTOP% == 8 set GMS_DESKTOP=8
if %GMS_DESKTOP% == 9 set GMS_DESKTOP=9
echo set GMS_DESKTOP=%GMS_DESKTOP%>> g
echo REM Initial values:>>g
echo set REPLY_BANNER=1 >> g
echo set REPLY_SHADE=2 >> g
echo set REPLY_DESKTOP=3 >> g
echo REM Fill character (ASCII code 32 to 126):>> g
echo set GMS_LETTER=%GMS_LETTER% >> g
echo set REPLY_LETTER=123 >> g
echo.>> g
echo REM 4) Browser interface variables:----->> g
echo.>> g
echo REM Maximal size and number of files:>> g
echo set GMS_MAXSIZE=%GMS_MAXSIZE%>> g
echo set GMS_MAXFILES=%GMS_MAXFILES%>> g
echo REM Width of text area (join/split):>> g
echo set GMS_WIDTH_ONE=%GMS_WIDTH_ONE%>> g
echo set GMS_WIDTH_TWO=%GMS_WIDTH_TWO%>> g
echo REM Height of text area (join/split):>> g
echo set GMS_HEIGHT_ONE=%GMS_HEIGHT_ONE%>> g
echo set GMS_HEIGHT_TWO=%GMS_HEIGHT_TWO%>> g
echo REM Font size, in point (join/split):>> g
if %GMS_SIZE_ONE% == 0 set GMS_SIZE_ONE=0
if %GMS_SIZE_ONE% == 1 set GMS_SIZE_ONE=1
if %GMS_SIZE_ONE% == 2 set GMS_SIZE_ONE=2
if %GMS_SIZE_ONE% == 3 set GMS_SIZE_ONE=3
if %GMS_SIZE_ONE% == 4 set GMS_SIZE_ONE=4
if %GMS_SIZE_ONE% == 5 set GMS_SIZE_ONE=5
if %GMS_SIZE_ONE% == 6 set GMS_SIZE_ONE=6
if %GMS_SIZE_ONE% == 7 set GMS_SIZE_ONE=7
if %GMS_SIZE_ONE% == 8 set GMS_SIZE_ONE=8
if %GMS_SIZE_ONE% == 9 set GMS_SIZE_ONE=9
echo set GMS_SIZE_ONE=%GMS_SIZE_ONE%>> g
if %GMS_SIZE_TWO% == 0 set GMS_SIZE_TWO=0
if %GMS_SIZE_TWO% == 1 set GMS_SIZE_TWO=1
if %GMS_SIZE_TWO% == 2 set GMS_SIZE_TWO=2
if %GMS_SIZE_TWO% == 3 set GMS_SIZE_TWO=3
if %GMS_SIZE_TWO% == 4 set GMS_SIZE_TWO=4
if %GMS_SIZE_TWO% == 5 set GMS_SIZE_TWO=5
if %GMS_SIZE_TWO% == 6 set GMS_SIZE_TWO=6
if %GMS_SIZE_TWO% == 7 set GMS_SIZE_TWO=7
if %GMS_SIZE_TWO% == 8 set GMS_SIZE_TWO=8
if %GMS_SIZE_TWO% == 9 set GMS_SIZE_TWO=9
echo set GMS_SIZE_TWO=%GMS_SIZE_TWO%>> g
echo REM Linking to internal or external target:>> g
echo set GMS_LINKS=%GMS_LINKS%>> g
echo.>> g
echo REM 5) Start program: ----->>g
echo.>> g
echo REM Do not change the rest of this file:>> g
echo.>> g
echo REM Set version number and date:>>g
echo set GMS_VERSION=%GMS_VERSION%>> g
echo set GMS_DATE=%GMS_DATE%>> g
echo REM System variables:>>g

```

```

echo      set OS=%OS%>> g
echo      set any=%any%>> g
if "%OS%" == "" goto fi_b44
if "%OS%" == "Windows_9x" goto th_b44
echo      if not "%GMS_MODE%" == "quiet" title GMS %GMS_VERSION%>> g
goto fi_b44
:th_b44
REM Fixme, this doesn't help:
REM echo      set windir=%windir%>> g
:fi_b44
echo      REM Get folder structure:>>g
if "%OS%" == "" set GMS_CMD=%GMS_SETTING%\folder
if not "%OS%" == "" set GMS_CMD="%GMS_SETTING%\folder"
echo      call %GMS_CMD% called_by gerolf>> g
echo      REM If necessary, backup prompt and path:>> g
echo      if not "%GMS_PROMPT%" == "" goto fi_pmt>> g
echo      set GMS_PROMPT=%PROMPT%>> g
if "%GMS_DRDOS%" == "" echo      set PROMPT= $P$G>> g
if not "%GMS_DRDOS%" == "" echo      prompt $L$P$G>> g
echo      :fi_pmt>> g
echo      if not "%GMS_PATH%" == "" goto fi_pth>> g
echo      set GMS_PATH=%PATH%>> g
if "%GMS_DRDOS%" == "" goto fi_dr7
echo      path %GMS_BATCH%;%GMS_BINARIES%;%PATH%>> g
goto fi_9x
:fi_dr7
if "%OS%" == "Windows_9x" goto then_9x
echo      set PATH=%GMS_BATCH%;%GMS_BINARIES%;%PATH%>> g
goto fi_9x
:then_9x
echo      set PATH=%GMS_BINARIES%;%PATH%>> g
echo      set PATH=%GMS_BATCH%;%PATH%>> g
:fi_9x
echo      :fi_pth>> g
echo      cls>> g
if "%OS%" == "" set GMS_CMD=%GMS_SETTING%\launch_2.scn

if not "%OS%" == "" set GMS_CMD=%GMS_SETTING%\launch_2.scn"
echo      if not (%1) == (-passive) type %GMS_CMD%>> g
echo      REM Get work file:>>g
echo      if "%GMS_MODE%" == "quiet" goto fi_ini>> g
set arg=%1 %2 %3 %4 %5 %6 %7 %8 %9
echo      call g_vars -ini_file called_by gerolf %arg%>> g
set arg=
echo      :fi_ini>> g
echo      cls>> g
if "%OS%" == "" set GMS_CMD=%GMS_SETTING%\launch_3.scn
if not "%OS%" == "" set GMS_CMD="%GMS_SETTING%\launch_3.scn"
echo      if not (%1) == (-passive) type %GMS_CMD%>> g
echo      REM Launch textmode interface:>> g
echo      if not (%1) == (-passive) call shredder called_by gerolf>>g
echo      :exit>> g
echo.>> g
echo      REM End of GMS launcher script .....>> g
if "%OS%" == "" if exist g copy g %GMS_SETTING%\gerolf.bat >nul
if not "%OS%" == "" if exist g copy g "%GMS_SETTING%\gerolf.bat">nul
if exist g del g
set launch=
set GMS_CMD=
goto exit
:fi_b

REM Not found: .....

:else
call l_banner -no_action g_launch %1
:fi

:exit

set line=

```

g_list.bat

```

REM g_list.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_LIST=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_list) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_list %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_list
goto exit
:fi_nd

REM Clear/Handle:-----

if %1 == -clear goto then_cl
if not %1 == -handle goto fi_h

REM Clear: .....

:then_cl
set FULL=
set INDEX=
REM file names:
set REPLY1=
set REPLY2=
set REPLY3=
set REPLY4=
set REPLY5=
set REPLY6=
set REPLY7=
set REPLY8=
set REPLY9=
REM Folder marks \ :
set GMS1=
set GMS2=
set GMS3=
set GMS4=
set GMS5=
set GMS6=
set GMS7=
set GMS8=

set GMS9=
if %1 == -clear goto exit
:fi_cl

REM Handle: .....

REM Fixme: put tmp files in user directory to avoid interference
set tmp1=%GMS_TEMP%\tmp1.tmp
set tmp2=%GMS_TEMP%\tmp2.tmp
set tmp3=%GMS_TEMP%\tmp3.tmp
set tmp4=%GMS_TEMP%\tmp4.bat
REM There must be at least one fake file in a folder:
if not exist _fake_.tmp echo. > _fake_.tmp 2> nul
REM Then I can build a list of folders (/ad) and files (/a-d):
if not "%GMS_DRDOS%" == "" goto else_dr
REM 'dir /b' fails on MS-DOS 3x, so use 'mdir':
if "%GMS_DRDOS%" == "" if "%OS%" == "" set argdr=mdir /b
if "%GMS_DRDOS%" == "" if not "%OS%" == "" set argdr=dir /b
%argdr% /-p /ogen /ad | sed "s/^-d /" > %tmp1%
if not exist *.* goto fi_dr
%argdr% /-p /ogen /a-d | sed "s/^-f /" >> %tmp1%
goto fi_dr
:else_dr
xdir +d /x | sed "s/^.*/-d /;1,2d;$d" > %tmp1%
if not exist %any% goto fi_dr
xdir -d /x | sed "s/^.*/-f /;$d" >> %tmp1%
:fi_dr
set argdr=
REM Remove fake file from folder and list:
if exist _fake_.tmp del _fake_.tmp
sed "/-f _fake_.tmp/d" < %tmp1% > %tmp2%
if exist %tmp1% del %tmp1%
REM Shorten list by 8 lines (top) during each loop:
copy %tmp2% %tmp1% > nul
set _diff=
if "%REPLY_LIST%" == "" goto spool
:loops
sed "1,8d" < %tmp1% > %tmp2%
if exist %tmp1% del %tmp1%
copy %tmp2% %tmp1% > nul
if "%_diff%" == "%REPLY_LIST%" goto spool
set _diff=%_diff%*
goto loops
:spool
set _diff=
REM Shorten rest of list (bottom):
sed "1,9!d" < %tmp2% > %tmp1%
if exist %tmp2% del %tmp2%
REM Add line numbers and replace them with list variables:
sed = %tmp1% | sed "N;s/\n/ /;s/^.*/set REPLY&=/" > %tmp3%
if exist %tmp1% del %tmp1%
REM Remove old folder/file mark:
sed "s/^-f //;s/^-d //" < %tmp3% > %tmp4%
REM Set new file/folder/drive mark:
sed "s/REPLY/GMS;/s/^-f .*/ /;s/^-d .*/\[\]\]" < %tmp3% >> %tmp4%
call %tmp4%
if exist %tmp4% del %tmp4%
if exist %tmp3% del %tmp3%
set tmp1=
set tmp2=
set tmp3=
set tmp4=
goto exit
:fi_h

REM Drives: -----

if not %1 == -drives goto fi_d

```

```

set str=([)
set GMS1=%str%
set GMS2=%str%
set GMS3=%str%
set GMS4=%str%
set GMS5=%str%
set GMS6=%str%
set GMS7=%str%
set GMS8=%str%
set GMS9=%str%
if not "%REPLY_LIST%" == "" goto fi_d1
    set REPLY1=A:
    set REPLY2=B:
    set REPLY3=C:
    set REPLY4=D:
    set REPLY5=E:
    set REPLY6=F:
    set REPLY7=G:
    set REPLY8=H:
    set REPLY9=I:
    goto exit
:fi_d1
if not "%REPLY_LIST%" == "" goto fi_d2
    set REPLY1=I:
    set REPLY2=J:
    set REPLY3=K:
    set REPLY4=L:
    set REPLY5=M:
    set REPLY6=N:
    set REPLY7=O:
    set REPLY8=P:
    set REPLY9=Q:
    goto exit
:fi_d2
if not "%REPLY_LIST%" == "" goto fi_d3
    set REPLY1=Q:
    set REPLY2=R:
    set REPLY3=S:

```

```

set REPLY4=T:
set REPLY5=U:
set REPLY6=V:
set REPLY7=W:
set REPLY8=X:
set REPLY9=Y:
    goto exit
:fi_d3
if not "%REPLY_LIST%" == "" goto fi_d4
    set REPLY1=Y:
    set REPLY2=Z:
    set REPLY3=
    set REPLY4=
    set REPLY5=
    set REPLY6=
    set REPLY7=
    set REPLY8=
    set REPLY9=
    set GMS3=
    set GMS4=
    set GMS5=
    set GMS6=
    set GMS7=
    set GMS8=
    set GMS9=
    goto exit
:fi_d4
    goto exit
:fi_d
REM Not found: .....
:else
    call l_banner -no_action g_list %1
:fi
:exit

```

g_menu.bat

```
REM g_menu.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_MENU=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_menu) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_menu %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_menu
goto exit
:fi_nd

REM Resize: -----

if %1 == -update goto then_rs
if %1 == -resize goto then_rs
if %1 == -build goto then_rs
if not %1 == -remove goto fi_rs
:then_rs
set REPLY_OFFSET=52
set REPLY_SIZE=23
set REPLY_ITEMS=13
if %1 == -update goto then_u
if %1 == -build goto then_b
if %1 == -remove goto then_r
goto exit
:fi_rs

REM Build: -----

if not %1 == -build goto fi_b
:then_b
if not "%GMS_FILE%" == "" goto else_b1
call l_banner -upper -first g_menu -build
goto fi_b1
:else_b1
if not "%OS%" == "" if not "%OS%" == "Windows_9x" title %GMS_FILE% - GMS
:fi_b1
REM equals call g_menu -resize g_menu -build:
set REPLY_OFFSET=52
set REPLY_SIZE=23

set REPLY_ITEMS=13
call l_box -build g_menu -build
call l_menu -build g_menu -build
if "%GMS_HOT%" == "" set GMS_HOT=Q
call l_menu -update %GMS_HOT% %GMS_HOT% g_menu -build
set REPLY_MODULE=g_menu
set REPLY_ACTION=-update
set GMS_RECEIVE=1
goto exit
:fi_b

REM Remove: -----

if not %1 == -remove goto fi_r
:then_r
call l_box -remove
goto exit
:fi_r

REM Update: -----

if not %1 == -update goto fi_u
:then_u
REM Cold:
if not "%3" == "" goto else_u1
set GMS_COLD=%GMS_HOT%
goto fi_u1
:else_u1
set GMS_COLD=%3
:fi_u1
REM Hot:
if not "%2" == "" goto else_u2
call l_banner -no_hot g_menu -update
goto fi_u2
:else_u2
set GMS_HOT=Q
if %2 == C set GMS_HOT=C
if %2 == c set GMS_HOT=C
if %2 == 2 set GMS_HOT=C
if %2 == 0 set GMS_HOT=0
if %2 == o set GMS_HOT=0
if %2 == 3 set GMS_HOT=0
if %2 == V set GMS_HOT=V
if %2 == v set GMS_HOT=V
if %2 == 4 set GMS_HOT=V
if %2 == E set GMS_HOT=E
if %2 == e set GMS_HOT=E
if %2 == 5 set GMS_HOT=E
if %2 == B set GMS_HOT=B
if %2 == b set GMS_HOT=B
if %2 == 6 set GMS_HOT=B
if %2 == A set GMS_HOT=A
if %2 == a set GMS_HOT=A
if %2 == 7 set GMS_HOT=A
if %2 == T set GMS_HOT=T
if %2 == t set GMS_HOT=T
if %2 == 8 set GMS_HOT=T
if %2 == R set GMS_HOT=R
if %2 == r set GMS_HOT=R
if %2 == 9 set GMS_HOT=R
if %2 == L set GMS_HOT=L
if %2 == l set GMS_HOT=L
if %2 == 10 set GMS_HOT=L
if %2 == S set GMS_HOT=S
if %2 == s set GMS_HOT=S
if %2 == 11 set GMS_HOT=S
if %2 == W set GMS_HOT=W
if %2 == w set GMS_HOT=W
```



```

if %2 == 12 set GMS_HOT=W
if %2 == I set GMS_HOT=I
if %2 == i set GMS_HOT=I
if %2 == 13 set GMS_HOT=I
:fi_u2
REM Handle old, update new:
if not %GMS_HOT% == %GMS_COLD% goto else_u3
goto then_h
:else_u3
call l_menu -update %GMS_HOT% %GMS_COLD% called_by g_menu -update
:fi_u3
set GMS_RECEIVE=1
goto exit
:fi_u

REM Handle: .....

if not %1 == -handle goto fi_h
:then_h

REM Analyse:
if not %GMS_HOT% == A goto fi_ha
if "%GMS_FILE%" == "" goto then_ha1
goto fi_ha1
:then_ha1
set GMS_FILE=_folder.err
:fi_ha1
goto then_e
:fi_ha

REM View, Edit, Browse, Typeset, Read, Write, Initialize:
if %GMS_HOT% == V goto then_e
if %GMS_HOT% == E goto then_e
if %GMS_HOT% == B goto then_e
if %GMS_HOT% == T goto then_e
if %GMS_HOT% == R goto then_e
if %GMS_HOT% == W goto then_e
if %GMS_HOT% == I goto then_e
goto fi_e
:then_e
set backup=%GMS_HOT%
call l_box -remove
call l_desk -remove
REM Run program:
if "%GMS_HOT%" == "W" goto then_e11
if "%GMS_HOT%" == "I" goto then_e11
if "%GMS_FOLDER%" == "" goto else_e1
if "%GMS_FILE%" == "" goto else_e1
:then_e11
call gms %backup%
:fi_e11
goto fi_e1
:else_e1
call l_banner -no_file
:fi_e1
set REPLY_OFFSET=0
set REPLY_SIZE=0
reply -random 2
set GMS_RECEIVE=1
set GMS_TEXTMODE=1
set REPLY_MODULE=g_palet
set REPLY_ACTION=-rebuild
set GMS_RETURN=1
set backup=
goto exit
:fi_e

REM Quit:
if not %GMS_HOT% == Q goto fi_hq
call l_menu -update - Q
set REPLY_MODULE=g_good

set REPLY_ACTION=-build
set GMS_RECEIVE=1
goto exit
:fi_hq

REM Create:
if not %GMS_HOT% == C goto fi_hc
call l_menu -update - C
call l_menu -update - O
call g_dos -pwd
set GMS_LASTDIR=%PWD%
set GMS_FOLDER=%GMS_TEMPLATE%
set GMS_FILE=
set GMS_RECEIVE=1
%GMS_DRIVE%
call g_dos -chdir
call l_banner -upper -first
call l_banner -lower -folder
set GMS_FILEBOX=1. Select a template to create a copy of
call g_file -build -template
set GMS_FILEBOX=
goto exit
:fi_hc

REM Open:
if not %GMS_HOT% == O goto fi_ho
call l_menu -update - O
call l_menu -update - V
REM if not "%GMS_FOLDER%" == "%GMS_BINARIES%" goto fi_ho1
REM set GMS_FOLDER=%GMS_SETTING%
REM set GMS_FILE=
call l_banner -upper -first
call l_banner -lower -folder

REM :fi_ho1
call g_dos -chdir
set GMS_FILEBOX=Change folder and open .htm* markup file
call g_file -build -open
set GMS_FILEBOX=
goto exit
:fi_ho

REM Learn:
if not %GMS_HOT% == L goto fi_hl
set GMS_FOLDER=%GMS_TEMPLATE%\handbook
set GMS_FILE=handbook.htm
set GMS_SHORT=handbook.htm
set GMS_BASE=handbook
set GMS_REMODRV=%GMS_DRIVE%
%GMS_REMODRV%
if "%OS%" == "" cd %GMS_FOLDER%
if not "%OS%" == "" cd "%GMS_FOLDER%"
call l_banner -lower -folder
call l_banner -upper -handbook
call l_menu -update - L
call g_dos -chdir called_by g_menu
set REPLY_MODULE=g_palet
set REPLY_ACTION=-rebuild
set GMS_RETURN=1
goto exit
:fi_hl

REM Select:
if not %GMS_HOT% == S goto fi_hs
call l_menu -update - S
call l_menu -update - W
set REPLY_MODULE=g_select
set REPLY_ACTION=-build
goto exit
:fi_hs

```

```
REM Not found:
:else_h
  call l_banner -no_hotkey g_menu %GMS_HOT%
  goto exit
:fi_h

REM Not found: -----

:else
  call l_banner -no_action g_menu %1
:fi

:exit
```

g_palet.bat

```
REM g_palet.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_PALET=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_palet) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_palet %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_palet
goto exit
:fi_nd

REM Build, rebuild: -----

if %1 == -build goto then_b
if %1 == -rebuild goto then_b
goto fi_b
:then_b
set BACKUP_SIZE=12
set BACKUP_OFFSET=12
set arg=%REPLY_DESKTOP% %REPLY_PATTERN%
set arg=%arg% %REPLY_BANNER% %REPLY_TEXT%
set arg=%arg% %REPLY_SHADE% %REPLY_HOTKEY%
call g_palet -handle %arg% %REPLY_LETTER% g_palet -build
set arg=
REM call g_desk -resize:
set REPLY_OFFSET=0
set REPLY_SIZE=78
set REPLY_ITEMS=0
call l_desk -build
if not "%GMS_RETURN%" == "" goto else_b1
set REPLY_MODULE=g_wel
set REPLY_ACTION=-remove
call g_wel -build called_by g_palet %1
set GMS_RECEIVE=1
goto fi_b1
:else_b1
set GMS_RETURN=1
if not "%GMS_FILE%" == "" goto else_b2
call l_banner -upper -first called_by g_palet %1
goto fi_b2
:else_b2
call l_banner -upper -file called_by g_palet %1
if not "%OS%" == "" if not "%OS%" == "Windows_9x" title %GMS_FILE% - GMS
:fi_b2
call l_banner -lower -folder called_by g_palet %1
call g_menu -build called_by g_palet %1
:fi_b1
goto exit
:fi_b

REM Handle: -----

if not %1 == -handle goto fi_h
:then_h
REM Check if color values are acceptable:

REM 2) Background (desktop) color:
set BACKUP=%GMS_DESKTOP%
set GMS_DESKTOP=R
if %BACKUP% == 0 goto then_122
if %BACKUP% == 1 goto then_122
if %BACKUP% == 2 goto then_122
if %BACKUP% == 3 goto then_122
if %BACKUP% == 4 goto then_122
if %BACKUP% == 5 goto then_122
if %BACKUP% == 6 goto then_122
if %BACKUP% == 7 goto then_122
if %BACKUP% == R goto else_122
if %BACKUP% == r goto else_122
goto fi_122
:then_122
set REPLY_DESKTOP=%BACKUP%
set GMS_DESKTOP=%BACKUP%
goto fi_122
:else_122
if "%2" == "" goto fi_122
if %2 == 0 set REPLY_DESKTOP=%2
if %2 == 1 set REPLY_DESKTOP=%2
if %2 == 2 set REPLY_DESKTOP=%2
if %2 == 3 set REPLY_DESKTOP=%2
if %2 == 4 set REPLY_DESKTOP=%2
if %2 == 5 set REPLY_DESKTOP=%2
if %2 == 6 set REPLY_DESKTOP=%2
if %2 == 7 set REPLY_DESKTOP=%2
:fi_122

REM 3) Foreground (pattern) color:
set BACKUP=%GMS_PATTERN%
set GMS_PATTERN=R
if %BACKUP% == 0 goto then_123
if %BACKUP% == 1 goto then_123
if %BACKUP% == 2 goto then_123
if %BACKUP% == 3 goto then_123
if %BACKUP% == 4 goto then_123
if %BACKUP% == 5 goto then_123
if %BACKUP% == 6 goto then_123
if %BACKUP% == 7 goto then_123
if %BACKUP% == 8 goto then_123
if %BACKUP% == 9 goto then_123
if %BACKUP% == 10 goto then_123
if %BACKUP% == 11 goto then_123
if %BACKUP% == 12 goto then_123
if %BACKUP% == 13 goto then_123
if %BACKUP% == 14 goto then_123
if %BACKUP% == 15 goto then_123
if %BACKUP% == R goto else_123
if %BACKUP% == r goto else_123
goto fi_123
```

```

:then_123
  set REPLY_PATTERN=%BACKUP%
  set GMS_PATTERN=%BACKUP%
  goto fi_123
:else_123
  if "%3" == "" goto fi_123
  if %3 == 0 set REPLY_PATTERN=%3
  if %3 == 1 set REPLY_PATTERN=%3
  if %3 == 2 set REPLY_PATTERN=%3
  if %3 == 3 set REPLY_PATTERN=%3
  if %3 == 4 set REPLY_PATTERN=%3
  if %3 == 5 set REPLY_PATTERN=%3
  if %3 == 6 set REPLY_PATTERN=%3
  if %3 == 7 set REPLY_PATTERN=%3
  if %3 == 8 set REPLY_PATTERN=%3
  if %3 == 9 set REPLY_PATTERN=%3
  if %3 == 10 set REPLY_PATTERN=%3
  if %3 == 11 set REPLY_PATTERN=%3
  if %3 == 12 set REPLY_PATTERN=%3
  if %3 == 13 set REPLY_PATTERN=%3
  if %3 == 14 set REPLY_PATTERN=%3
  if %3 == 15 set REPLY_PATTERN=%3
:fi_123

```

```

REM 4) Banner color:
set BACKUP=%GMS_BANNER%
set GMS_BANNER=R
if %BACKUP% == 0 goto then_124
if %BACKUP% == 1 goto then_124
if %BACKUP% == 2 goto then_124
if %BACKUP% == 3 goto then_124
if %BACKUP% == 4 goto then_124
if %BACKUP% == 5 goto then_124
if %BACKUP% == 6 goto then_124
if %BACKUP% == 7 goto then_124
if %BACKUP% == R goto else_124
if %BACKUP% == r goto else_124
goto fi_124
:then_124
  set REPLY_BANNER=%BACKUP%
  set GMS_BANNER=%BACKUP%
  goto fi_124
:else_124
  if "%4" == "" goto fi_124
  if %4 == 0 set REPLY_BANNER=%4
  if %4 == 1 set REPLY_BANNER=%4
  if %4 == 2 set REPLY_BANNER=%4
  if %4 == 3 set REPLY_BANNER=%4
  if %4 == 4 set REPLY_BANNER=%4
  if %4 == 5 set REPLY_BANNER=%4
  if %4 == 6 set REPLY_BANNER=%4
  if %4 == 7 set REPLY_BANNER=%4
:fi_124

```

```

REM 5) Text color:
set BACKUP=%GMS_TEXT%
set GMS_TEXT=R
if %BACKUP% == 0 goto then_125
if %BACKUP% == 1 goto then_125
if %BACKUP% == 2 goto then_125
if %BACKUP% == 3 goto then_125
if %BACKUP% == 4 goto then_125
if %BACKUP% == 5 goto then_125
if %BACKUP% == 6 goto then_125
if %BACKUP% == 7 goto then_125
if %BACKUP% == 8 goto then_125
if %BACKUP% == 9 goto then_125
if %BACKUP% == 10 goto then_125
if %BACKUP% == 11 goto then_125
if %BACKUP% == 12 goto then_125
if %BACKUP% == 13 goto then_125

```

```

if %BACKUP% == 14 goto then_125
if %BACKUP% == 15 goto then_125
if %BACKUP% == R goto else_125
if %BACKUP% == r goto else_125
goto fi_125
:then_125
  set REPLY_TEXT=%BACKUP%
  set GMS_TEXT=%BACKUP%
  goto fi_125
:else_125
  if "%5" == "" goto fi_125
  if %5 == 0 set REPLY_TEXT=%5
  if %5 == 1 set REPLY_TEXT=%5
  if %5 == 2 set REPLY_TEXT=%5
  if %5 == 3 set REPLY_TEXT=%5
  if %5 == 4 set REPLY_TEXT=%5
  if %5 == 5 set REPLY_TEXT=%5
  if %5 == 6 set REPLY_TEXT=%5
  if %5 == 7 set REPLY_TEXT=%5
  if %5 == 8 set REPLY_TEXT=%5
  if %5 == 9 set REPLY_TEXT=%5
  if %5 == 10 set REPLY_TEXT=%5
  if %5 == 11 set REPLY_TEXT=%5
  if %5 == 12 set REPLY_TEXT=%5
  if %5 == 13 set REPLY_TEXT=%5
  if %5 == 14 set REPLY_TEXT=%5
  if %5 == 15 set REPLY_TEXT=%5
:fi_125

```

```

REM 6) Shadow color:
set BACKUP=%GMS_SHADE%
set GMS_SHADE=R
if %BACKUP% == 0 goto then_126
if %BACKUP% == 1 goto then_126
if %BACKUP% == 2 goto then_126
if %BACKUP% == 3 goto then_126
if %BACKUP% == 4 goto then_126
if %BACKUP% == 5 goto then_126
if %BACKUP% == 6 goto then_126
if %BACKUP% == 7 goto then_126
if %BACKUP% == R goto else_126
if %BACKUP% == r goto else_126
goto fi_126
:then_126
  set REPLY_SHADE=%BACKUP%
  set GMS_SHADE=%BACKUP%
  goto fi_126
:else_126
  if "%6" == "" goto fi_126
  if %6 == 0 set REPLY_SHADE=%6
  if %6 == 1 set REPLY_SHADE=%6
  if %6 == 2 set REPLY_SHADE=%6
  if %6 == 3 set REPLY_SHADE=%6
  if %6 == 4 set REPLY_SHADE=%6
  if %6 == 5 set REPLY_SHADE=%6
  if %6 == 6 set REPLY_SHADE=%6
  if %6 == 7 set REPLY_SHADE=%6
:fi_126

```

```

REM 7) Hotkey color:
set BACKUP=%GMS_HOTKEY%
set GMS_HOTKEY=R
if %BACKUP% == 0 goto then_127
if %BACKUP% == 1 goto then_127
if %BACKUP% == 2 goto then_127
if %BACKUP% == 3 goto then_127
if %BACKUP% == 4 goto then_127
if %BACKUP% == 5 goto then_127
if %BACKUP% == 6 goto then_127
if %BACKUP% == 7 goto then_127
if %BACKUP% == 8 goto then_127

```



```

set REPLY_LETTER=%BACKUP%
set GMS_LETTER=%BACKUP%
goto fi_128
:else_128
if not "%0" == "" set REPLY_LETTER=%0
:fi_128
set BACKUP=
set GMS_RECEIVE=1
goto exit
:fi_h

REM Not found: -----

:else
call l_banner -no_action g_palet %1
:fi

REM Background colors:

REM 0 black
REM 1 blue
REM 2 green

REM 3 cyan
REM 4 red
REM 5 magenta
REM 6 brown
REM 7 lightgray

REM Additional foreground colors:

REM 8 darkgray
REM 9 lightblue
REM 10 lightgreen
REM 11 lightcyan

REM 12 lightred
REM 13 lightmagenta
REM 14 yellow
REM 15 white

:exit

```

g_plug.bat

```
REM g_plug.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_PLUG=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
  echo Gerolf Markup Shredder (g_plug) . . .
  pause > nul
  gerolf
  goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
  call l_banner -debug g_plug %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
  call l_banner -no_action g_plug
  goto exit
:fi_nd

REM Rebuild list of plug-in modules: -----

if not %1 == -rebuild goto fi_rb
if "%OS%" == "" set plug=%GMS_SETTING%\plugin.cfg
if not "%OS%" == "" set plug="%GMS_SETTING%\plugin.cfg"
set str-%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
REM Display info:
set REPLY_BANNER=0
set REPLY_TEXT=7
set REPLY_OFFSET=0
set REPLY_SIZE=16
  reply -banner 17 Writing:
set REPLY_OFFSET=14
set REPLY_SIZE=68
  reply -banner 17 " %GMS_SETTING%\plugin.cfg"
REM Initial data:
echo %% plugin.cfg> %plug%
echo %% =====> %plug%
echo.>> %plug%
echo %% This TeX file was auto->> %plug%
echo %% generated by Gerolf Markup Shredder,>> %plug%
echo %% written by G. D. Brettschneider (1999-2004).>> %plug%
echo %% All rights reserved.>> %plug%
echo %% Send corrections to: MarkupShredder@Gerolf.org>> %plug%
echo %% Subject: GMS TeX macros (module loader)>> %plug%
echo.>> %plug%
echo %% This file serves for loading all GMS>> %plug%
echo %% modules that could be found in the>> %plug%
echo %% appropriate plug-in directories.>> %plug%

echo %% If you have reason to edit this file,>> %plug%
echo %% change its name to 'myplugin.cfg'.>> %plug%
echo %% The recommended way to add TeX macro files>> %plug%
echo %% to GMS, however, is to \input them in>> %plug%
echo %% 'prologue.cfg' or 'epilogue.cfg'.>> %plug%
echo.>> %plug%
echo %%str%str%>> %plug%
echo.>> %plug%
echo %% Initial data:>> %plug%
echo \def \GMSdate {%GMS_DATE%}>> %plug%
echo \def \GMSversion {%GMS_VERSION%}>> %plug%
echo \def \GMScodepage {%GMS_CODEPAGE%}>> %plug%
echo \def \GMSdebug {%GMS_DEBUG%}>> %plug%
echo \def \OS {%OS%}>> %plug%
echo.>> %plug%
echo %%str%str%>> %plug%
echo.>> %plug%
REM Modules:
echo \hyphenmessage>> %plug%
echo \message {Module loader.}>> %plug%
echo \fillmessage 42{modules}5>> %plug%
echo.>> %plug%
REM Fonts:
echo \fillmessage 6a{font}6>> %plug%
echo \MAPload>> %plug%
echo \hyphenmessage>> %plug%
echo \message {{ font.map}>> %plug%
echo \hyphenmessage>> %plug%
if "%OS%" == "Windows_9x" goto fi_9x1
echo \message {These names can be used as font face or}>> %plug%
echo \echo {font-family style names in markup files:}>> %plug%
echo \MAPfamilyshow>> %plug%
:fi_9x1
echo \message {}>> %plug%
echo \fillmessage 6-{/font}5>> %plug%
echo.>> %plug%
REM Kerning tables:
echo \fillmessage 6b{kerning}3>> %plug%
if "%OS%" == "" cd %GMS_ROOT%\data\kern > nul
if not "%OS%" == "" cd "%GMS_ROOT%\data\kern" > nul
for %i in (*.krn) do echo \KERNINGparseline (%i)\relax>> %plug%
echo \KERNINGloadtables>> %plug%
echo \fillmessage 6-{/kerning}2>> %plug%
echo.>> %plug%
REM Hyphenation patterns:
if "%OS%" == "" cd %GMS_ROOT%\tex\hyphen > nul
if not "%OS%" == "" cd "%GMS_ROOT%\tex\hyphen" > nul
echo \fillmessage 6c{language}2>> %plug%
for %i in (*.tex) do echo \LANGUAGEload %i>> %plug%
echo \LANGUAGEloadpatterns>> %plug%
echo \message {More pattern files are available at>> %plug%
echo http://www.ctan.org/tex-archive/language/}>> %plug%
echo \fillmessage 6-{/language}1>> %plug%
echo.>> %plug%
REM Unicode rows:
echo \fillmessage 6d{unicode}3>> %plug%
if "%OS%" == "" cd %GMS_ROOT%\data\row > nul
if not "%OS%" == "" cd "%GMS_ROOT%\data\row" > nul
for %i in (*.row) do echo \UNICODEload %i>> %plug%
echo \UNICODEslotsload>> %plug%
echo \fillmessage 6-{/unicode}2>> %plug%
echo.>> %plug%
REM Entity names:
echo \fillmessage 6e{entity}4>> %plug%
if "%OS%" == "" cd %GMS_ROOT%\data\ent > nul
if not "%OS%" == "" cd "%GMS_ROOT%\data\ent" > nul
```

```

for %i in (*.ent) do echo      \ENTITYYadd %i>> %plug%
echo      \ENTITYnamesload>> %plug%
if not "%OS%" == "Windows_9x" echo      \ENTITYnamesshow>> %plug%
echo      \fillmessage 6-{/entity}3>> %plug%
echo.>> %plug%
REM Glyph names:
echo      \fillmessage 6f{glyph}5>> %plug%
if "%OS%" == "" cd %GMS_ROOT%\data\gly> nul
if not "%OS%" == "" cd "%GMS_ROOT%\data\gly"> nul
for %i in (*.gly) do echo      \GLYPHSadd %i>> %plug%
echo      \GLYPHnamesload>> %plug%
echo      \fillmessage 6-{/glyph}4>> %plug%
echo.>> %plug%
REM Codepages:
echo      \fillmessage 6g{codepage}2>> %plug%
if "%OS%" == "" cd %GMS_ROOT%\data\cp > nul
if not "%OS%" == "" cd "%GMS_ROOT%\data\cp" > nul
for %i in (*.txt) do echo      \CODEPAGEadd %i>> %plug%
echo      \CODEPAGEloadthem>> %plug%
echo      \CODEPAGEencwrite>> %plug%
echo      \expandafter \CODEPAGEenable \CODEPAGE \relax>> %plug%
REM echo      \UNICODEencwrite\relax>> %plug%
echo      \fillmessage 6-{/codepage}1>> %plug%
echo.>> %plug%
echo      \fillmessage 4-{/modules}3>> %plug%
echo      \endingput>> %plug%
if "%OS%" == "" cd %GMS_BINARIES% > nul
if not "%OS%" == "" cd "%GMS_BINARIES%" > nul
set plug=
goto exit
:fi_rb

REM Build plugin.cfg, but consider myplugin.cfg: -----

REM Fixme:
if not %1 == -build goto fi_b
REM Write new plugin.cfg, if myplugin.cfg does not exist:
if "%OS%" == "" if exist %GMS_SETTING%\myplugin.cfg goto fi_i1
if not "%OS%" == "" if exist "%GMS_SETTING%\myplugin.cfg" goto fi_i1
call g_plug -rebuild called_by g_plug %1
:fi_i1

REM If myplugin.cfg exists, copy it to plugin.cfg:
if "%OS%" == "" if exist %GMS_SETTING%\myplugin.cfg goto then_i2
if not "%OS%" == "" if exist "%GMS_SETTING%\myplugin.cfg" goto else_i2
goto fi_i2
:then_i2
copy %GMS_SETTING%\myplugin.cfg %GMS_SETTING%\plugin.cfg > nul
goto fi_i2
:else_i2
copy "%GMS_SETTING%\myplugin.cfg" "%GMS_SETTING%\plugin.cfg" > nul
:fi_i2
REM Write new plugin.cfg, if plugin.cfg does not exist:
if "%OS%" == "" if exist %GMS_SETTING%\plugin.cfg goto fi_i3
if not "%OS%" == "" if exist "%GMS_SETTING%\plugin.cfg" goto fi_i3
call g_plug -rebuild g_plug %1
:fi_i3
goto exit
:fi_b

REM Remove plugin.cfg: -----

if not %1 == -remove goto fi_rp
REM If myplugin.cfg and plugin.cfg exist, delete plugin.cfg:
if "%OS%" == "" if exist %GMS_SETTING%\myplugin.cfg goto then_i4
if not "%OS%" == "" if exist "%GMS_SETTING%\myplugin.cfg" goto else_i4
goto fi_i4
:then_i4
if exist %GMS_SETTING%\plugin.cfg del %GMS_SETTING%\plugin.cfg
goto fi_i4
:else_i4
if exist "%GMS_SETTING%\plugin.cfg" del "%GMS_SETTING%\plugin.cfg"
:fi_i4
goto exit
:fi_rp

REM Not found: -----

:else
call l_banner -no_action g_plug %1
:fi

:exit

```


g_prog.bat

```
REM g_prog.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_PROG=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_prog) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_prog %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_prog
goto exit
:fi_nd

REM Resize: -----

if %1 == -resize goto then_rs
if %1 == -build goto then_rs
if not %1 == -remove goto fi_rs
:then_rs
set REPLY_OFFSET=26
set REPLY_SIZE=20
set REPLY_ITEMS=7
if %1 == -build goto then_b
if %1 == -remove goto then_r
goto exit
:fi_rs

REM Build: -----

if not %1 == -build goto fi_b
:then_b
call l_banner -lower -program
call l_box -t11_build
call l_prog -build
set GMS_HOT=Q
call l_prog -update Q Q
set REPLY_MODULE=g_prog
set REPLY_ACTION=-update
set GMS_RECEIVE=1
goto exit
:fi_b

REM Remove: -----

if not %1 == -remove goto fi_r
:then_r
call l_box -t12_remove
goto exit
:fi_r

REM Update: -----

if not %1 == -update goto fi_u
:then_u
REM Cold:
if not "%3" == "" goto else_u1
set GMS_COLD=%GMS_HOT%
goto fi_u1
:else_u1
set GMS_COLD=%3
:fi_u1
REM Hot:
if not "%2" == "" goto else_u2
call l_banner -no_hot g_prog
goto fi_u2
:else_u2
set GMS_HOT=Q
if %2 == V set GMS_HOT=V
if %2 == v set GMS_HOT=V
if %2 == 1 set GMS_HOT=V
if %2 == E set GMS_HOT=E
if %2 == e set GMS_HOT=E
if %2 == 2 set GMS_HOT=E
if %2 == B set GMS_HOT=B
if %2 == b set GMS_HOT=B
if %2 == 3 set GMS_HOT=B
if %2 == A set GMS_HOT=A
if %2 == a set GMS_HOT=A
if %2 == 4 set GMS_HOT=A
if %2 == T set GMS_HOT=T
if %2 == t set GMS_HOT=T
if %2 == 5 set GMS_HOT=T
if %2 == R set GMS_HOT=R
if %2 == r set GMS_HOT=R
if %2 == 6 set GMS_HOT=R
:fi_u2
REM Handle old, update new:
if not %GMS_HOT% == %GMS_COLD% goto else_u3
goto then_h
:else_u3
REM Set programs:
call l_banner -lower -select
if "%4" == "" goto fi_u33
if %GMS_COLD% == V set GMS_VIEWER=%4
if %GMS_COLD% == E set GMS_EDITOR=%4
if %GMS_COLD% == B set GMS_BROWSER=%4
if %GMS_COLD% == A set GMS_ANALYST=%4
if %GMS_COLD% == T set GMS_TSETTER=%4
if %GMS_COLD% == R set GMS_READER=%4
:fi_u33
REM Normalize writing:
if "%GMS_TSETTER%" == "" set GMS_TSETTER=pdfetex
if %GMS_TSETTER% == PDFETEX set GMS_TSETTER=pdfetex
if %GMS_TSETTER% == Pdfetex set GMS_TSETTER=pdfetex
if %GMS_TSETTER% == pdfTeX set GMS_TSETTER=pdfetex
if %GMS_TSETTER% == PDFTEX set GMS_TSETTER=pdfetex
if %GMS_TSETTER% == PdfTeX set GMS_TSETTER=pdfetex
if %GMS_TSETTER% == pdfTeX set GMS_TSETTER=pdfetex
if %GMS_TSETTER% == ETEX set GMS_TSETTER=etex
if %GMS_TSETTER% == ETeX set GMS_TSETTER=etex
```

```

if %GMS_TSETTER% == eTeX set GMS_TSETTER=etex
if %GMS_TSETTER% == TEX set GMS_TSETTER=tex
if %GMS_TSETTER% == Tex set GMS_TSETTER=tex
if %GMS_TSETTER% == TeX set GMS_TSETTER=tex
REM Update program box:
set arg=%GMS_HOT% %GMS_COLD%
call l_prog -update %arg%
set arg=
:fi_u3
set GMS_RECEIVE=1
goto exit
:fi_u

REM Handle: -----

if not %1 == -handle goto fi_h
:then_h

REM Quit:
if not %GMS_HOT% == Q goto fi_hq
call g_launch -build called_by g_prog %1 Q
call g_prog -remove called_by g_prog %1 Q
call g_select -update P Q called_by g_prog %1 Q
set REPLY_MODULE=g_select
set REPLY_ACTION=-update
set GMS_RECEIVE=1
goto exit
:fi_hq

REM Viewer:
if not %GMS_HOT% == V goto fi_hv
set GMS_RECEIVE=1
call l_banner -lower -viewer
call l_prog -update - V called_by g_prog %1 V
reply -question 8 %GMS_VIEWER% VV 1
goto exit
:fi_hv

REM Editor:
if not %GMS_HOT% == E goto fi_he
set GMS_RECEIVE=1
call l_banner -lower -editor
call l_prog -update - E called_by g_prog %1 E
reply -question 10 %GMS_EDITOR% EE 2
goto exit
:fi_he

REM Browser:
if not %GMS_HOT% == B goto fi_hb

set GMS_RECEIVE=1
call l_banner -lower -browser
call l_prog -update - B called_by g_prog %1 B
reply -question 12 %GMS_BROWSER% BB 3
goto exit
:fi_hb

REM Analyst:
if not %GMS_HOT% == A goto fi_ha
set GMS_RECEIVE=1
call l_banner -lower -analyst
call l_prog -update - A called_by g_prog %1 A
reply -question 14 %GMS_ANALYST% AA 4
goto exit
:fi_ha

REM Typesetter:
if not %GMS_HOT% == T goto fi_ht
set GMS_RECEIVE=1
call l_banner -lower -tsetter
call l_prog -update - T called_by g_prog %1 T
reply -question 16 %GMS_TSETTER% TT 5
goto exit
:fi_ht

REM Reader:
if not %GMS_HOT% == R goto fi_hr
set GMS_RECEIVE=1
call l_banner -lower -reader
call l_prog -update - R called_by g_prog %1 R
reply -question 18 %GMS_READER% RR 6
goto exit
:fi_hr

REM Not found:
:else_h
call l_banner -no_hotkey g_prog %2
goto exit

:fi_h

REM Not found: -----

:else
call l_banner -no_action g_prog %1
:fi

:exit

```

g_rain.bat

```
REM g_rain.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_RAIN=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_rain) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_rain %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_rain
goto exit
:fi_nd

REM Resize: -----

if %1 == -resize goto then_rs
if %1 == -build goto then_rs
if not %1 == -remove goto fi_rs
:then_rs
set REPLY_OFFSET=3
set REPLY_SIZE=17
set REPLY_ITEMS=1
if %1 == -build goto then_b
if %1 == -remove call l_rain -remove
goto exit
:fi_rs

REM Build: -----

if not %1 == -build goto fi_b
:then_b
REM Allocate memory:
set LETTER=123
set BACKUP=123
set BACKUP1=123
set BACKUP2=123
set BACKUP3=123
set BACKUP4=123
call l_rain -build
set BACKUP=
set BACKUP1=
set BACKUP2=
set BACKUP3=
set BACKUP4=
set LETTER=
goto exit
:fi_b

REM Not found: -----

:else
call l_banner -no_action g_rain %1
:fi

:exit
```

g_save.bat

```
REM g_save.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_SAVE=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_save) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_save %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_save
goto exit
:fi_nd

REM Resize: -----

if %1 == -update goto then_rs
if %1 == -resize goto then_rs
if %1 == -build goto then_rs
if not %1 == -remove goto fi_rs
:then_rs
set REPLY_OFFSET=3
set REPLY_SIZE=44
set REPLY_ITEMS=4
if %1 == -update goto then_u
if %1 == -build goto then_b
if %1 == -remove goto then_r
goto exit
:fi_rs

REM Build: -----

if not %1 == -build goto fi_b
:then_b
call l_banner -lower -folder called_by g_file %1
set REPLY_LIST=
set GMS_FILE=new_file.htm
call l_box -t8_build called_by g_file %1
call l_save -build called_by g_file %1
REM Consider existing file with same name:
set GMS_EXIST=0
if "%OS%" == "" if exist %GMS_FOLDER%\%GMS_FILE% set GMS_EXIST=1
if not "%OS%" == "" if exist "%GMS_FOLDER%\%GMS_FILE%" set GMS_EXIST=1

call l_save -update Q Q %GMS_EXIST% called_by g_save %1
set GMS_EXIST=
set REPLY_MODULE=g_save
set REPLY_ACTION=-update
set GMS_RECEIVE=1
set GMS_HOT=Q
goto exit
:fi_b

REM Remove: -----

if not %1 == -remove goto fi_r
:then_r
call l_box -t8_remove called_by g_file %1
call g_menu -resize called_by g_file %1
call l_menu -update C Q called_by g_file %1
set REPLY_MODULE=g_menu
set REPLY_ACTION=-update
set GMS_RECEIVE=1
goto exit
:fi_r

REM Update: -----

if not %1 == -update goto fi_u
:then_u
REM Cold:
if not "%3" == "" goto else_u1
set GMS_COLD=%GMS_HOT%
goto fi_u1
:else_u1
set GMS_COLD=%3
:fi_u1
REM Hot:
if not "%2" == "" goto else_u2
call l_banner -no_hot g_save
goto fi_u2
:else_u2
set GMS_HOT=Q
if %2 == C set GMS_HOT=C
if %2 == c set GMS_HOT=C
if %2 == 2 set GMS_HOT=C
if %2 == 0 set GMS_HOT=0
if %2 == o set GMS_HOT=0
if %2 == 3 set GMS_HOT=0
if %2 == N set GMS_HOT=N
if %2 == n set GMS_HOT=N
if %2 == 4 set GMS_HOT=N
:fi_u2
REM Handle old, update new:
if not %GMS_HOT% == %GMS_COLD% goto else_u3
goto then_h
:else_u3
if not "%4" == "" set GMS_FILE=%4
if not "%5" == "" set GMS_FILE=%4 %5
if not "%6" == "" set GMS_FILE=%4 %5 %6
if not "%7" == "" set GMS_FILE=%4 %5 %6 %7
if not "%8" == "" set GMS_FILE=%4 %5 %6 %7 %8
if not "%9" == "" set GMS_FILE=%4 %5 %6 %7 %8 %9
REM Update save box (and consider existing file with same name):
set GMS_EXIST=0
if "%OS%" == "" if exist %GMS_FOLDER%\%GMS_FILE% set GMS_EXIST=1
if not "%OS%" == "" if exist "%GMS_FOLDER%\%GMS_FILE%" set GMS_EXIST=1
call l_save -update %GMS_HOT% %GMS_COLD% %GMS_EXIST% called_by g_save %1
set GMS_EXIST=
:fi_u3
set GMS_RECEIVE=1
goto exit
```

```

:fi_u
REM Handle: -----
if not %1 == -handle goto fi_h
:then_h

REM Overwrite (copy template):
if not %GMS_HOT% == 0 goto fi_ho
:then_ho
set arg=%GMS_TFOLDER%\%GMS_TEMPLATEFILE%
if "%OS%" == "" goto fi_ho1
if not exist "%arg%" goto fi_ho1
copy "%arg%" "%GMS_FOLDER%\%GMS_FILE%" > nul
goto fi_ho2
:fi_ho1
if not "%OS%" == "" goto fi_ho2
if not exist "%arg%" goto fi_ho2
copy "%arg%" "%GMS_FOLDER%\%GMS_FILE%" > nul
:fi_ho2
set arg=
set REPLY_SIZE=50
set REPLY_OFFSET=0
reply -banner 1 "%GMS_FILE%"
if not "%OS%" == "" if not "%OS%" == "Windows_9x" title %GMS_FILE% - GMS
call g_save -remove called_by g_save %1
call g_dos -chdir called_by g_save %1
goto exit
:fi_ho

REM Quit:
if not %GMS_HOT% == Q goto fi_hq
if "%OS%" == "" if exist %GMS_FOLDER%\%GMS_FILE% goto then_hqg
if "%OS%" == "" if not exist %GMS_FOLDER%\%GMS_FILE% goto then_ho
if not "%OS%" == "" if exist "%GMS_FOLDER%\%GMS_FILE%" goto then_hqg
if not "%OS%" == "" if not exist "%GMS_FOLDER%\%GMS_FILE%" goto then_ho
:then_hqg
set REPLY_SIZE=50

set REPLY_OFFSET=0
reply -banner 1 "%GMS_FILE%"
if not "%OS%" == "" if not "%OS%" == "Windows_9x" title %GMS_FILE% - GMS
call g_save -remove called_by g_save %1
:fi_hqg
goto exit
:fi_hq

REM Cancel:
if not %GMS_HOT% == C goto fi_hc
set REPLY_SIZE=50
set REPLY_OFFSET=0
reply -banner 1 "%GMS_FILE%"
if not "%OS%" == "" if not "%OS%" == "Windows_9x" title %GMS_FILE% - GMS
call g_save -remove called_by g_save %1
goto exit
:fi_hc

REM Name:
if not %GMS_HOT% == N goto fi_hn
set idle=
goto exit
:fi_hn

REM Not found:
:else_h
call l_banner -no_hotkey g_save %1 %GMS_HOT%
goto exit
:fi_h

REM Not found: -----
:else
call l_banner -no_action g_save %1
:fi
:exit

```

g_select.bat

```
REM g_select.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_select=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_select) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_select %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action g_select
goto exit
:fi_nd

REM Resize: -----

if %1 == -update goto then_rs
if %1 == -resize goto then_rs
if %1 == -build goto then_rs
if not %1 == -remove goto fi_rs
:then_rs
set REPLY_OFFSET=3
set REPLY_SIZE=17
set REPLY_ITEMS=8
if %1 == -update goto then_u
if %1 == -build goto then_b
if %1 == -remove goto then_r
goto exit
:fi_rs

REM Build: -----

if not %1 == -build goto fi_b
:then_b
call l_banner -lower -select called_by g_select %1
call l_box -t11_build called_by g_select %1
if not %GMS_ANIMATE% == on goto else_b1
call l_select -build off called_by g_select %1
goto fi_b1
:else_b1
call l_select -build on called_by g_select %1
:fi_b1
set GMS_HOT=Q

call l_select -update Q Q called_by g_select %1
set REPLY_MODULE=g_select
set REPLY_ACTION=-update
set GMS_RECEIVE=1
goto exit
:fi_b

REM Remove: -----

if not %1 == -remove goto fi_r
:then_r
call l_box -t11_remove called_by g_select %1
goto exit
:fi_r

REM Update: -----

if not %1 == -update goto fi_u
:then_u
REM Cold:
if not "%3" == "" goto else_u1
set GMS_COLD=%GMS_HOT%
goto fi_u1
:else_u1
set GMS_COLD=%3
:fi_u1
REM Hot:
if not "%2" == "" goto else_u2
call l_banner -no_hot g_select called_by g_select %1
goto fi_u2
:else_u2
set GMS_HOT=Q
if %2 == A set GMS_HOT=A
if %2 == a set GMS_HOT=A
if %2 == 2 set GMS_HOT=A
if %2 == C set GMS_HOT=C
if %2 == c set GMS_HOT=C
if %2 == 3 set GMS_HOT=C
if %2 == P set GMS_HOT=P
if %2 == p set GMS_HOT=P
if %2 == 4 set GMS_HOT=P
if %2 == D set GMS_HOT=D
if %2 == d set GMS_HOT=D
if %2 == 5 set GMS_HOT=D
if %2 == X set GMS_HOT=D
if %2 == x set GMS_HOT=D
if %2 == 0 set GMS_HOT=D
if %2 == Y set GMS_HOT=D
if %2 == y set GMS_HOT=D
if %2 == o set GMS_HOT=D
if %2 == E set GMS_HOT=E
if %2 == e set GMS_HOT=E
if %2 == 6 set GMS_HOT=E
if %2 == I set GMS_HOT=E
if %2 == i set GMS_HOT=E
if %2 == U set GMS_HOT=U
if %2 == u set GMS_HOT=U
if %2 == 7 set GMS_HOT=U
if %2 == L set GMS_HOT=L
if %2 == l set GMS_HOT=L
if %2 == 8 set GMS_HOT=L
:fi_u2

REM Handle old, update new:
if not %GMS_HOT% == %GMS_COLD% goto else_u3
goto then_h
:else_u3

REM Banner:
if %GMS_HOT% == Q call l_banner -lower -first called_by g_select %1
```

```

if %GMS_HOT% == A call l_banner -lower -animate called_by g_select %1
if %GMS_HOT% == U call l_banner -lower -codepage called_by g_select %1
if %GMS_HOT% == L call l_banner -lower -codepage called_by g_select %1
if %GMS_HOT% == C call l_banner -lower -select called_by g_select %1
if %GMS_HOT% == P call l_banner -lower -select called_by g_select %1
if %GMS_HOT% == D call l_banner -lower -select called_by g_select %1
if %GMS_HOT% == E call l_banner -lower -select called_by g_select %1
REM Debugging:
if "%4" == "" goto fi_u4
if not %GMS_COLD% == D goto fi_u4
if %4 == X set GMS_DEBUG=X
if %4 == x set GMS_DEBUG=X
if %4 == Y set GMS_DEBUG=Y
if %4 == y set GMS_DEBUG=Y
if %4 == z set GMS_DEBUG=Z
if %4 == Z set GMS_DEBUG=Z
if %4 == 0 set GMS_DEBUG=0
if %4 == o set GMS_DEBUG=0
if %4 == 0 set GMS_DEBUG=0
if %4 == 0 goto then_u44
if %4 == 0 goto then_u44
if not %4 == 0 goto fi_u44
:then_u44
    REM Remove debug banners:
    set backup_offset=%REPLY_OFFSET%
    set backup_size=%REPLY_SIZE%
    set REPLY_OFFSET=0
    set REPLY_SIZE=78
    reply -stripe 2 %REPLY_PATTERN% %REPLY_LETTER%
    reply -stripe 23 %REPLY_PATTERN% %REPLY_LETTER%
    set REPLY_SIZE=%backup_size%
    set REPLY_OFFSET=%backup_offset%
    set backup_size=
    set backup_offset=
    call g_select -resize called_by g_select %1
:fi_u44
if %GMS_DEBUG% == -Z set Z=nul
call g_launch -build called_by g_select %1
:fi_u4
REM Encoding:
if "%4" == "" goto fi_u5
if not %GMS_COLD% == E goto fi_u5
set GMS_CODEPAGE=%4
REM Normalize writing:
if %4 == ARABIC set GMS_CODEPAGE=CP1256
if %4 == Arabic set GMS_CODEPAGE=CP1256
if %4 == arabic set GMS_CODEPAGE=CP1256
if %4 == ARABICBH set GMS_CODEPAGE=CP1256
if %4 == ArabicBH set GMS_CODEPAGE=CP1256
if %4 == arabicbh set GMS_CODEPAGE=CP1256
if %4 == ARABICMT set GMS_CODEPAGE=CP1256
if %4 == ArabicMT set GMS_CODEPAGE=CP1256
if %4 == arabicmt set GMS_CODEPAGE=CP1256
if %4 == ASCII set GMS_CODEPAGE=US-ASCII
if %4 == Ascii set GMS_CODEPAGE=US-ASCII
if %4 == ascii set GMS_CODEPAGE=US-ASCII
if %4 == BALTIC set GMS_CODEPAGE=CP1257
if %4 == Baltic set GMS_CODEPAGE=CP1257
if %4 == baltic set GMS_CODEPAGE=CP1257
if %4 == CANADA set GMS_CODEPAGE=CP863
if %4 == Canada set GMS_CODEPAGE=CP863
if %4 == canada set GMS_CODEPAGE=CP863
if %4 == CENTRAL set GMS_CODEPAGE=CP1250
if %4 == Central set GMS_CODEPAGE=CP1250
if %4 == central set GMS_CODEPAGE=CP1250
if %4 == cp437 set GMS_CODEPAGE=CP437
if %4 == cp850 set GMS_CODEPAGE=CP850
if %4 == cp860 set GMS_CODEPAGE=CP860
if %4 == cp863 set GMS_CODEPAGE=CP863
if %4 == cp865 set GMS_CODEPAGE=CP865
if %4 == cp874 set GMS_CODEPAGE=CP874
if %4 == cp1250 set GMS_CODEPAGE=CP1250
if %4 == cp1251 set GMS_CODEPAGE=CP1251
if %4 == cp1252 set GMS_CODEPAGE=CP1252
if %4 == cp1253 set GMS_CODEPAGE=CP1253
if %4 == cp1254 set GMS_CODEPAGE=CP1254
if %4 == cp1255 set GMS_CODEPAGE=CP1255
if %4 == cp1256 set GMS_CODEPAGE=CP1256
if %4 == cp1257 set GMS_CODEPAGE=CP1257
if %4 == cp1258 set GMS_CODEPAGE=CP1258
if %4 == CYRILLIC set GMS_CODEPAGE=CP1251
if %4 == Cyrillic set GMS_CODEPAGE=CP1251
if %4 == cyrillic set GMS_CODEPAGE=CP1251
if %4 == GREEK set GMS_CODEPAGE=CP1253
if %4 == Greek set GMS_CODEPAGE=CP1253
if %4 == greek set GMS_CODEPAGE=CP1253
if %4 == HEBREW set GMS_CODEPAGE=CP1255
if %4 == Hebrew set GMS_CODEPAGE=CP1255
if %4 == hebrew set GMS_CODEPAGE=CP1255
if %4 == INDIAN set GMS_CODEPAGE=ISCI
if %4 == Indian set GMS_CODEPAGE=ISCI
if %4 == indian set GMS_CODEPAGE=ISCI
if %4 == Iscii set GMS_CODEPAGE=ISCI
if %4 == iscii set GMS_CODEPAGE=ISCI
if %4 == ISO-646 set GMS_CODEPAGE=ISO646
if %4 == ISO_646 set GMS_CODEPAGE=ISO646
if %4 == iso-646 set GMS_CODEPAGE=ISO646
if %4 == iso_646 set GMS_CODEPAGE=ISO646
if %4 == iso646 set GMS_CODEPAGE=ISO646
if %4 == ISO-8859-1 set GMS_CODEPAGE=I8859-1
if %4 == ISO_8859-1 set GMS_CODEPAGE=I8859-1
if %4 == iso-8859-1 set GMS_CODEPAGE=I8859-1
if %4 == iso_8859-1 set GMS_CODEPAGE=I8859-1
if %4 == ISO-8859-2 set GMS_CODEPAGE=I8859-2
if %4 == ISO_8859-2 set GMS_CODEPAGE=I8859-2
if %4 == iso-8859-2 set GMS_CODEPAGE=I8859-2
if %4 == iso_8859-2 set GMS_CODEPAGE=I8859-2
if %4 == ISO-8859-3 set GMS_CODEPAGE=I8859-3
if %4 == ISO_8859-3 set GMS_CODEPAGE=I8859-3
if %4 == iso-8859-3 set GMS_CODEPAGE=I8859-3
if %4 == iso_8859-3 set GMS_CODEPAGE=I8859-3
if %4 == ISO-8859-4 set GMS_CODEPAGE=I8859-4
if %4 == ISO_8859-4 set GMS_CODEPAGE=I8859-4
if %4 == iso-8859-4 set GMS_CODEPAGE=I8859-4
if %4 == iso_8859-4 set GMS_CODEPAGE=I8859-4
if %4 == ISO-8859-5 set GMS_CODEPAGE=I8859-5
if %4 == ISO_8859-5 set GMS_CODEPAGE=I8859-5
if %4 == iso-8859-5 set GMS_CODEPAGE=I8859-5
if %4 == iso_8859-5 set GMS_CODEPAGE=I8859-5
if %4 == ISO-8859-6 set GMS_CODEPAGE=I8859-6
if %4 == ISO_8859-6 set GMS_CODEPAGE=I8859-6
if %4 == iso-8859-6 set GMS_CODEPAGE=I8859-6
if %4 == iso_8859-6 set GMS_CODEPAGE=I8859-6
if %4 == ISO-8859-7 set GMS_CODEPAGE=I8859-7
if %4 == ISO_8859-7 set GMS_CODEPAGE=I8859-7
if %4 == iso-8859-7 set GMS_CODEPAGE=I8859-7
if %4 == iso_8859-7 set GMS_CODEPAGE=I8859-7
if %4 == ISO-8859-8 set GMS_CODEPAGE=I8859-8
if %4 == ISO_8859-8 set GMS_CODEPAGE=I8859-8
if %4 == iso-8859-8 set GMS_CODEPAGE=I8859-8
if %4 == iso_8859-8 set GMS_CODEPAGE=I8859-8
if %4 == ISO-8859-9 set GMS_CODEPAGE=I8859-9
if %4 == ISO_8859-9 set GMS_CODEPAGE=I8859-9
if %4 == iso-8859-9 set GMS_CODEPAGE=I8859-9
if %4 == iso_8859-9 set GMS_CODEPAGE=I8859-9
if %4 == ISO-8859-10 set GMS_CODEPAGE=I8859-10
if %4 == ISO_8859-10 set GMS_CODEPAGE=I8859-10
if %4 == iso-8859-10 set GMS_CODEPAGE=I8859-10
if %4 == iso_8859-10 set GMS_CODEPAGE=I8859-10
if %4 == ISO-8859-11 set GMS_CODEPAGE=I8859-11
if %4 == ISO_8859-11 set GMS_CODEPAGE=I8859-11
if %4 == iso-8859-11 set GMS_CODEPAGE=I8859-11
if %4 == iso_8859-11 set GMS_CODEPAGE=I8859-11

```

```

if %4 == iso_8859-11 set GMS_CODEPAGE=I8859-11
if %4 == ISO-8859-13 set GMS_CODEPAGE=I8859-13
if %4 == ISO_8859-13 set GMS_CODEPAGE=I8859-13
if %4 == iso-8859-13 set GMS_CODEPAGE=I8859-13
if %4 == iso_8859-13 set GMS_CODEPAGE=I8859-13
if %4 == ISO-8859-14 set GMS_CODEPAGE=I8859-14
if %4 == ISO_8859-14 set GMS_CODEPAGE=I8859-14
if %4 == iso-8859-14 set GMS_CODEPAGE=I8859-14
if %4 == iso_8859-14 set GMS_CODEPAGE=I8859-14
if %4 == ISO-8859-15 set GMS_CODEPAGE=I8859-15
if %4 == ISO_8859-15 set GMS_CODEPAGE=I8859-15
if %4 == iso-8859-15 set GMS_CODEPAGE=I8859-15
if %4 == iso_8859-15 set GMS_CODEPAGE=I8859-15
if %4 == LATIN set GMS_CODEPAGE=CP1252
if %4 == Latin set GMS_CODEPAGE=CP1252
if %4 == latin set GMS_CODEPAGE=CP1252
if %4 == LATIN-1 set GMS_CODEPAGE=CP1252
if %4 == Latin-1 set GMS_CODEPAGE=CP1252
if %4 == latin-1 set GMS_CODEPAGE=CP1252
if %4 == LATIN-2 set GMS_CODEPAGE=CP1250
if %4 == Latin-2 set GMS_CODEPAGE=CP1250
if %4 == latin-2 set GMS_CODEPAGE=CP1250
if %4 == LATIN-5 set GMS_CODEPAGE=CP1254
if %4 == Latin-5 set GMS_CODEPAGE=CP1254
if %4 == latin-5 set GMS_CODEPAGE=CP1254
if %4 == MULTILINGUAL set GMS_CODEPAGE=CP850
if %4 == Multilingual set GMS_CODEPAGE=CP850
if %4 == multilingual set GMS_CODEPAGE=CP850
if %4 == NORWAY set GMS_CODEPAGE=CP865
if %4 == Norway set GMS_CODEPAGE=CP865
if %4 == norway set GMS_CODEPAGE=CP865
if %4 == PORTUGAL set GMS_CODEPAGE=CP860
if %4 == Portugal set GMS_CODEPAGE=CP860
if %4 == portugal set GMS_CODEPAGE=CP860
if %4 == THAI set GMS_CODEPAGE=CP874
if %4 == Thai set GMS_CODEPAGE=CP874
if %4 == thai set GMS_CODEPAGE=CP874
if %4 == TURKISH set GMS_CODEPAGE=CP1254
if %4 == Turkish set GMS_CODEPAGE=CP1254
if %4 == turkish set GMS_CODEPAGE=CP1254
if %4 == USA set GMS_CODEPAGE=CP437
if %4 == Usa set GMS_CODEPAGE=CP437
if %4 == usa set GMS_CODEPAGE=CP437
if %4 == US_ASCII set GMS_CODEPAGE=US-ASCII
if %4 == Us_ascii set GMS_CODEPAGE=US-ASCII
if %4 == us_ascii set GMS_CODEPAGE=US-ASCII
if %4 == Us-ascii set GMS_CODEPAGE=US-ASCII
if %4 == us-ascii set GMS_CODEPAGE=US-ASCII
if %4 == VIETNAM set GMS_CODEPAGE=VISCII
if %4 == Vietnam set GMS_CODEPAGE=VISCII
if %4 == vietnam set GMS_CODEPAGE=VISCII
if %4 == VIETNAMESE set GMS_CODEPAGE=VISCII
if %4 == Vietnamese set GMS_CODEPAGE=VISCII
if %4 == vietnamese set GMS_CODEPAGE=VISCII
if %4 == Viscii set GMS_CODEPAGE=VISCII
if %4 == viscii set GMS_CODEPAGE=VISCII
if %4 == WESTERN set GMS_CODEPAGE=CP1252
if %4 == Western set GMS_CODEPAGE=CP1252
if %4 == western set GMS_CODEPAGE=CP1252
if %4 == WINDOWS-1250 set GMS_CODEPAGE=CP1250
if %4 == windows-1250 set GMS_CODEPAGE=CP1250
if %4 == WINDOWS-1251 set GMS_CODEPAGE=CP1251
if %4 == windows-1251 set GMS_CODEPAGE=CP1251
if %4 == WINDOWS-1252 set GMS_CODEPAGE=CP1252
if %4 == windows-1252 set GMS_CODEPAGE=CP1252
if %4 == WINDOWS-1253 set GMS_CODEPAGE=CP1253
if %4 == windows-1253 set GMS_CODEPAGE=CP1253
if %4 == WINDOWS-1254 set GMS_CODEPAGE=CP1254
if %4 == windows-1254 set GMS_CODEPAGE=CP1254
if %4 == WINDOWS-1255 set GMS_CODEPAGE=CP1255
if %4 == windows-1255 set GMS_CODEPAGE=CP1255

if %4 == WINDOWS-1256 set GMS_CODEPAGE=CP1256
if %4 == windows-1256 set GMS_CODEPAGE=CP1256
if %4 == WINDOWS-1257 set GMS_CODEPAGE=CP1257
if %4 == windows-1257 set GMS_CODEPAGE=CP1257
if %4 == WINDOWS-1258 set GMS_CODEPAGE=CP1258
if %4 == windows-1258 set GMS_CODEPAGE=CP1258
call g_launch -build called_by g_select %1
:fi_u5
REM Select:
if not %GMS_ANIMATE% == on goto else_u6
set arg=%GMS_HOT% %GMS_COLD% off
call l_select -update %arg% called_by g_select %1
goto fi_u6
:else_u6
set arg=%GMS_HOT% %GMS_COLD% on
call l_select -update %arg% called_by g_select %1
:fi_u6
set arg=
:fi_u3
set GMS_RECEIVE=1
goto exit
:fi_u

REM Handle: .....
if not %1 == -handle goto fi_h
:then_h

REM Quit:
if not %GMS_HOT% == Q goto fi_hq
call l_banner -lower -folder called_by g_select %1 Q
call g_select -remove called_by g_select %1 Q
call g_menu -update S Q called_by g_select %1 Q
set REPLY_MODULE=g_menu
set REPLY_ACTION=-update
goto exit
:fi_hq

REM Animation:
if not %GMS_HOT% == A goto fi_ha
if not %GMS_ANIMATE% == on goto else_hal
set GMS_ANIMATE=off
goto fi_hal
:else_hal
set GMS_ANIMATE=on
:fi_hal
call g_launch -build called_by g_select %1 A
call g_select -update A Q called_by g_select %1 A
set REPLY_MODULE=g_select
set REPLY_ACTION=-update
goto exit
:fi_ha

REM Color:
if not %GMS_HOT% == C goto fi_hc
call l_select -update - C called_by g_select -handle C
call l_select -update - P called_by g_select -handle C
call g_select -remove called_by g_select %1 C
call g_rain -build called_by g_select %1 C
call g_color -build called_by g_select %1 C
goto exit
:fi_hc

REM Programs:
if not %GMS_HOT% == P goto fi_hp
call l_select -update - D called_by g_select %1 P
call l_select -update - P called_by g_select %1 P
call g_prog -build called_by g_select %1 P
goto exit
:fi_hp

```



```

REM Debugging:
if not %GMS_HOT% == D goto fi_hd
set GMS_RECEIVE=1
call l_banner -lower -debug called_by g_select %1 D
call l_select -update - D called_by g_select %1 D
reply -question 11 %GMS_DEBUG% DD 5
goto exit
:fi_hd

```

```

REM Encoding:
if not %GMS_HOT% == E goto fi_he
set GMS_RECEIVE=1
call l_banner -lower -encode called_by g_select %1 E
call l_select -update - E called_by g_select %1 E
reply -question 13 %GMS_CODEPAGE% EE 6
goto exit
:fi_he

```

```

REM Codepage (upper half):
if not %GMS_HOT% == U goto fi_hu :then_hu
call g_code -cp_up_build called_by g_select %1 U
goto exit

```

```
:fi_hu
```

```

REM Codepage (lower half):
if not %GMS_HOT% == L goto fi_hl
call g_code -cp_lo_build called_by g_select %1 L
goto exit
:fi_hl

```

```

REM Not found:
:else_h
call l_banner -no_hotkey g_select %2
goto exit
:fi_h

```

```
REM Not found: .....
```

```

:else
call l_banner -no_action g_select %1
:fi

```

```
:exit
```

g_vars.bat

```
REM g_vars.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateG_VARS=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_vars) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
if "%DEBUG%" == "" goto fi_db1
call l_banner -debug g_vars %1 %2 %3 %4
:fi_db1
:fi_db

REM Chapters: =====

REM Not defined:

if not (%1) == () goto fi_nd
call l_banner -no_action g_vars
goto exit
:fi_nd

REM Initialize file and folders: -----

if not (%1) == (.ini_file) goto fi_if
REM Fixme: check drive change
if "%OS%" == "" goto fi_if0
set arg##### New work file #####
if %GMS_DEBUG% == Z echo %arg#####>> %Z%
set arg=
:fi_if0
REM Dismiss -ini_file and caller(s):
if (%2) == (called_by) shift
if (%1) == (called_by) shift
if (%2) == (called_by) shift
if (%1) == (called_by) shift
if (%2) == (-passive) shift
if (%1) == (-passive) shift
if (%2) == (-a) shift
if (%1) == (-t) shift
if (%2) == (-handle) shift
shift
set GMS_RETURN=
REM Now %1 ... %9 can be a file name:
if (%1) == () goto fi_pf1
call g_dos -pwd called_by g_vars -ini_file
REM %1 ... %9 may include path, so get filename and pathname: .....
set BACKUP_FILE=%GMS_FILE%
set BACKUP_FOLDER=%GMS_FOLDER%

set GMS_FILE=%1
if "%OS%" == "" goto fi_parm
if not (%2) == () set GMS_FILE=%GMS_FILE% %2
if not (%3) == () set GMS_FILE=%GMS_FILE% %3
if not (%4) == () set GMS_FILE=%GMS_FILE% %4
if not (%5) == () set GMS_FILE=%GMS_FILE% %5
if not (%6) == () set GMS_FILE=%GMS_FILE% %6
if not (%7) == () set GMS_FILE=%GMS_FILE% %7
if not (%8) == () set GMS_FILE=%GMS_FILE% %8
if not (%9) == () set GMS_FILE=%GMS_FILE% %9
:fi_parm
set GMS_FOLDER=%GMS_FILE%
REM Get file name if %GMS_FILE% contains a path: .....
set e=%GMS_TEMP%\elif.txt
echo %GMS_FILE%> %e%
set f=%GMS_TEMP%\file.bat
if not "%GMS_DRDOS%" == "" goto elsfnl1
sed "s/\\\\/\\g;s/^\./^/;s/^\./^/;s/^\./^/" < %e% > %f%
goto fi_fnl1
:elsfnl1
sed "s/\\\\/\\g;s/^\./^/;s/^\./^/" < %e% > %f%
:fi_fnl1
if "%OS%" == "" if not exist %f% goto fi_fnl1
if not "%OS%" == "" if not exist "%f%" goto fi_fnl1
call %f%
del %f%
del %e%
:fi_fnl
set f=
set e=
set GMS_FILE=%file%
REM Get path name if %GMS_FOLDER% contains %GMS_FILE%: .....
set h=%GMS_TEMP%\htap.txt
echo %GMS_FOLDER%> %h%
set path2=%GMS_TEMP%\path2.txt
if not "%GMS_DRDOS%" == "" goto elspn1a
sed "s/\\\\/\\g;s/^\./^/" < %h% > %path2%
goto fi_pn1a
:elspn1a
sed "s/\\\\/\\g" < %h% > %path2%
:fi_pn1a
set path1=%GMS_TEMP%\path1.bat
sed "s/%GMS_FILE%$/;/s/^\./^/;s/^\./^/" <%path2% >%path1%
if "%OS%" == "" if not exist %path1% goto fi_pn1
if not "%OS%" == "" if not exist "%path1%" goto fi_pn1
call %path1%
del %path1%
del %path2%
del %h%
:fi_pn1
set path1=
set path2=
set h=
REM Get drive letter if %GMS_FOLDER% contains %GMS_FILE%: .....
set h=%GMS_TEMP%\htap.txt
echo %GMS_FOLDER%> %h%
set path2=%GMS_TEMP%\path2.txt
if not "%GMS_DRDOS%" == "" goto elspn2a
sed "s/\\\\/\\g;s/^\./^/;s/%GMS_FILE%//" < %h% > %path2%
goto fi_pn2a
:elspn2a
sed "s/\\\\/\\g;s/%GMS_FILE%//" < %h% > %path2%
:fi_pn2a
set path1=%GMS_TEMP%\path1.bat
sed "s/^\./^/;s/^\./^/" <%path2% >%path1%
if "%OS%" == "" if not exist %path1% goto fi_pn2
if not "%OS%" == "" if not exist "%path1%" goto fi_pn2
call %path1%
```

```

del %path1%
del %path2%
del %h%
set GMS_REMODRV=%DRV%
set DRV=
:fi_pn2
set path1=
set path2=
set h=
if not "%BACKUP_FOLDER%" == "" set GMS_FOLDER=%BACKUP_FOLDER%
set GMS_FILE=%BACKUP_FILE%
set BACKUP_FOLDER=
set BACKUP_FILE=
REM Check if file is in current folder: .....
if "%OS%" == "" goto fi_pf0
if not %GMS_DEBUG% == -Z goto fi_pf0
echo ***PWD=%PWD%>> %Z%
echo ***file=%file%>> %Z%
echo ***path_=%path_%>> %Z%
echo ***GMS_REMODRV=%GMS_REMODRV%>> %Z%
if not exist "%PWD%\%file%" echo File is not in current folder>> %Z%
:fi_pf0
if "%OS%" == "" if not exist %PWD%\%file% goto fi_pf2
if not "%OS%" == "" if not exist "%PWD%\%file%" goto fi_pf2
if "%OS%" == "" goto fi_pf22
if %GMS_DEBUG% == Z echo File is in current folder>> %Z%
:fi_pf22
set GMS_RETURN=1
set GMS_FILE=%file%
if not "%PWD%" == "" set GMS_FOLDER=%PWD%
set GMS_MESSAGE=%file%
call g_dos -chdir called_by_g_vars -ini_file
call g_dos -pwd called_by_g_vars -ini_file
if not "%PWD%" == "" set GMS_FOLDER=%PWD%
set file=
set path_ =
goto then_wm
:fi_pf2
REM Check if file is in a related folder: .....
if "%OS%" == "" if not exist %PWD%\%path_%\%file% goto fi_pf3
if not "%OS%" == "" if not exist "%PWD%\%path_%\%file%" goto fi_pf3
if "%OS%" == "" goto fi_pf33
if %GMS_DEBUG% == Z echo File is in a related folder>> %Z%
:fi_pf33
set GMS_RETURN=1
set GMS_FILE=%file%
if not "%PWD%" == "" set GMS_FOLDER=%PWD%\%path_%
set GMS_MESSAGE=%file%
call g_dos -chdir called_by_g_vars -ini_file
call g_dos -pwd called_by_g_vars -ini_file
if not "%PWD%" == "" set GMS_FOLDER=%PWD%
set file=
set path_ =
goto then_wm
:fi_pf3
REM Check if file is in a remote folder: .....
if "%OS%" == "" if not exist %path_%\%file% goto fi_pf4
if not "%OS%" == "" if not exist "%path_%\%file%" goto fi_pf4
if "%OS%" == "" goto fi_pf44
if %GMS_DEBUG% == Z echo File is in a remote folder>> %Z%
:fi_pf44
set GMS_RETURN=1
set GMS_FILE=%file%
if not "%PWD%" == "" set GMS_FOLDER=%path_%
set GMS_MESSAGE=%file%
set PWD=%path_%
%GMS_REMODRV%
if "%OS%" == "" goto fi_pf45
if %GMS_DEBUG% == Z echo PWD=%PWD%>> %Z%
:fi_pf45
set file=
set path_ =
goto then_wm
:fi_pf4
if %GMS_DEBUG% == Z echo Reading file and folder from gms_memo>> %Z%
:fi_rm12
if "%OS%" == "" set GMS_CMD=%GMS_SETTING%\gms_memo.bat
if not "%OS%" == "" set GMS_CMD=%GMS_SETTING%\gms_memo.bat"
if exist %GMS_CMD% call %GMS_CMD%
%GMS_REMODRV%
set GMS_CMD=
:fi_rm1
REM Fixme: Set GMS_MESSAGE, if file is in a remote folder
call g_dos -chdir called_by_g_vars -ini_file
if not "%PWD%" == "" set GMS_FOLDER=%PWD%
set GMS_RETURN=
:fi_if2
REM If no folder is present, set to current folder: .....
if not "%GMS_FOLDER%" == "" goto fi_if3
if "%OS%" == "" goto fi_if31
if %GMS_DEBUG% == Z echo File is in current folder>> %Z%
:fi_if31
call g_dos -pwd called_by_g_vars -ini_file
if not "%PWD%" == "" set GMS_FOLDER=%PWD%
set GMS_RETURN=
:fi_if3
if not %GMS_DEBUG% == Z goto fi_if5
if "%OS%" == "" goto fi_if5
echo GMS_FILE=%GMS_FILE%>> %Z%
:fi_if5
if "%OS%" == "" goto exit
set arg=#####
if %GMS_DEBUG% == Z echo %arg#####>> %Z%
set arg=
goto exit
:fi_if
REM Abort file creation check: .....
if not %1 == -abort goto fi_ab
if exist "%GMS_FILE%" goto then_wm
if "%OS%" == "" if not exist %GMS_FILE% goto else_abo
if not "%OS%" == "" if not exist "%GMS_FILE%" goto else_abo
goto then_wm
:else_abo
set GMS_MESSAGE=GMS error: Could not create file %GMS_FILE%
set GMS_FILE=
:fi_abo
goto exit
:fi_ab
REM Write file name and folder to gms_memo.bat startup file: .....
if not %1 == -write_memo goto fi_wm
:then_wm
if "%GMS_FILE%" == " folder.err" goto exit
call g_dos -chdir called_by_g_vars -write_memo
set str=%GMS_SETTING%\gms_memo.bat
if "%GMS_REMODRV%" == "" set GMS_REMODRV=%GMS_DRIVE%
if not "%OS%" == "" set str="%str%"
echo. REM gms_memo.bat> %str%
echo. REM =====>> %str%
echo. >> %str%
echo set GMS_REMODRV=%GMS_REMODRV%>> %str%
echo set GMS_FOLDER=%GMS_FOLDER%>> %str%
echo set GMS_FILE=%GMS_FILE%>> %str%

```

```

set str=
goto exit
:fi_wm

REM Clear environment variables: .....

if not %1 == -clear goto fi_c

call g_list -clear called_by g_vars -clear

REM Build new gmsdebug log file header:
if not "%OS%" == "" set Z="%GMS_SETTING%\gmshead.txt"
if "%OS%" == "" goto fi_lfh
echo gmsdebug.log> %Z%
set arg=////////////////////////////////////
if "%OS%" == "Windows 9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%%arg%>> %Z%
set arg=
echo This is the debug log file for Gerolf Markup Shredder.>> %Z%
echo To disable the run time call tracing, set GMS_DEBUG=0.>> %Z%
echo Format: called batch (parameter 1) ... (parameter 9),>> %Z%
echo followed by values of important environment variables.>> %Z%
echo shredder.bat reads user input and calls batch modules.>> %Z%
REM echo l_type modules only contribute if their compiled equi->> %Z%
REM echo valents are absent.>> %Z%
echo.>> %Z%
:fi_lfh
if not "%OS%" == "" set Z="%GMS_SETTING%\gmsdebug.log"

REM 1) General settings: .....

set GMS_ROOT=
set GMS_TEMP=
set GMS_TEMPLATE=

set GMS_BATCH=
set GMS_CHCP=
set GMS_CODEPAGE=
set GMS_CURCP=
set GMS_INICP=
set GMS_FILE=
set GMS_FOLDER=

REM 2) Browser interface variables: .....

set GMS_MAXSIZE=
set GMS_MAXFILES=
set GMS_WIDTH_ONE=
set GMS_WIDTH_TWO=
set GMS_HEIGHT_ONE=
set GMS_HEIGHT_TWO=
set GMS_SIZE_ONE=
set GMS_SIZE_TWO=

REM 3) Textmode interface variables: .....

set GMS_TEXT=
set GMS_HOTKEY=
set GMS_PATTERN=
set GMS_BANNER=
set GMS_SHADE=
set GMS_DESKTOP=
set GMS_LETTER=
set GMS_ANIMATE=

REM 4) Programs in search path (binary names): .....

set GMS_VIEWER=
set GMS_EDITOR=
set GMS_BROWSER=
set GMS_ANALYST=

```

```

set GMS_TSETTER=
set GMS_READER=

REM 5) Directories for TeX engine configuration: .....

set GMS_BINARIES=
set GMS_FONTS=

set TEXMF=
set TEXMFCNF=
set TEX_BASE=
set TEX_USER=
set TEXMFOUTPUT=
set TEXFONTS=
set TEXFORMATS=
set TEXPOOL=
set TEXPSHEADERS=

REM 6) Input file search: .....

set TEXINPUTA=
set TEXINPUTB=
set TEXINPUTS=

REM 7) Font search: .....

set AFM FONTS=
set TFM FONTS=
set T1 FONTS=
set TTF FONTS=
set VFF FONTS=

REM 8) Reply: .....

set REPLY_ACTION=
set REPLY_AFTER=
set REPLY_BANNER=
set REPLY_BEFORE=
set REPLY_COLD=
set REPLY_DESKTOP=
set REPLY_HOT=
set REPLY_HOTKEY=
set REPLY_ITEMS=
set REPLY_LETTER=
set REPLY_LIST=
set REPLY_MODULE=
set REPLY_OFFSET=
set REPLY_PATTERN=
set REPLY_SHADE=
set REPLY_SIZE=
set REPLY_TEMP=
set REPLY_TEXT=

REM 9) Other: .....

set any=
set arg=
set BACKUP_OFFSET=
set BACKUP_SIZE=
set backup_base=
set backup_file=
set backup_folder=
set backup_REPLY_TEMP=
set backup_short=
set dash=
set debug=
set dlist=
set gms=
set i=
set l=
set LEFT=

```

```

set m=
set myname=
set path_ =
set PWD=
set RIGHT=
set return=
set str=

set GMS_BASE=
set GMS_COLD=
set GMS_DATE=
set GMS_DRIVE=
set GMS_FreeDOS=
set GMS_HOT=
set GMS_ERRBASE=
set GMS_ERRSHORT=
set GMS_FILETYPE=
set GMS_LASTDIR=
set GMS_LINES=
set GMS_LINKS=
set GMS_LOGBASE=
set GMS_LOGSHORT=
set GMS_MESSAGE=
set GMS_NOREMO=
set GMS_PROGRAM=
set GMS_RECEIVE=
set GMS_REMODRV=
set GMS_RETURN=
set GMS_SHORT=
set GMS_SPACY=
set GMS_STARTUP=
set GMS_STATE=

```

```

set GMS_TEMPLATEFILE=
set GMS_TFOLDER=
set GMS_VERSION=

REM 10 ) Restore os, prompt and path: .....

if not "%GMS_DRDOS%" == "" goto fi_drd
    if not "%GMS_PROMPT%" == "" prompt %GMS_PROMPT%
:fi_drd
if "%GMS_DRDOS%" == "" if not "%GMS_PROMPT%" == "" prompt %GMS_PROMPT%
if not "%GMS_DRDOS%" == "" prompt $P$G
set GMS_PROMPT=
if not "%GMS_PATH%" == "" if "%GMS_DRDOS%" == "" set PATH=%GMS_PATH%
if not "%GMS_PATH%" == "" if not "%GMS_DRDOS%" == "" path %GMS_PATH%
set GMS_PATH=
if "%OS%" == "" set windir=
if not "%OS%" == "" if not "%OS%" == "Windows_9x" title Console > nul
if not "%GMS_DRDOS%" == "" set OS=DRDOS
set GMS_DRDOS=
if "%OS%" == "Windows_9x" set OS=
if "%OS%" == "Windows_XP" set OS=Windows_NT

goto exit

REM Not found: .....

:else_c
    call l_banner -no_action g_vars %1
:fi_c

:exit

```

g_wel.bat

```
REM g_wel.bat                                     :fi_rs
REM =====
REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).
REM set GMSdateG_WEL=20060927
REM Prologue: =====
REM Not running:
if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (g_wel) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr
REM Debug: -----
if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug g_wel %1 %2 %3 %4
:fi_db
REM Chapters: =====
REM Not defined:
if not "%1" == "" goto fi_nd
call l_banner -no_action g_wel
goto exit
:fi_nd
REM Resize: -----
if %1 == -resize goto then_rs
if %1 == -build goto then_rs
if not %1 == -remove goto fi_rs
:then_rs
set REPLY_OFFSET= 2
set REPLY_SIZE=25
set REPLY_ITEMS=1
if %1 == -build goto then_b
if %1 == -remove goto then_r
goto exit
:then_b
set backup_desktop=%REPLY_DESKTOP%
call l_wel -build g_wel -build
set backup_desktop=
set GMS_HOT=Q
set GMS_COLD=Q
set GMS_RECEIVE=1
goto exit
:fi_b
REM Remove: -----
if not %1 == -remove goto fi_r
:then_r
set backup_desktop=%REPLY_DESKTOP%
call l_wel -remove g_wel -remove
set REPLY_DESKTOP=%backup_desktop%
REM Remove error messages from l_gms bug:
set REPLY_SIZE=50
reply -stripe 23 %REPLY_PATTERN% %REPLY_LETTER%
reply -stripe 2 %REPLY_PATTERN% %REPLY_LETTER%
set backup_desktop=
REM Debug: -----
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == X goto then_dbr
if not %GMS_DEBUG% == Y goto fi_dbr
:then_dbr
call l_banner -debug g_wel %1 %2 %3 %4
:fi_dbr
call l_banner -upper -first g_wel -remove
call l_banner -lower -first g_wel -remove
call g_menu -build g_wel -remove
call l_banner -upper -file g_wel -remove
call l_banner -lower -folder g_wel -remove
goto exit
:fi_r
REM Not found: -----
:else
call l_banner -no_action g_wel %1
:fi
:exit
```

gms.bat

```
echo off
if not "%2" == "-quiet" if not "%3" == "-quiet" cls

REM gms.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateGMS=20060927

REM Prologue: =====

REM Start shell:

if "%1" == "-shell" goto th_sh
if "%comspec%" == "" echo Please set COMSPEC variable. Press any key ...
if "%comspec%" == "" pause > nul
if "%comspec%" == "" goto fi_sh
    %comspec% /E:4096 /C %0 -shell %1 %2 %3 %4 %5 %6 %7 %8 %9
goto exit
:th_sh
    shift
:fi_sh

REM Trace launcher parameters in debug log file:

if not (%1) == (-trace_launcher) goto fi_trace
    if "%OS%" == "" goto exit
    if %Z% == "" goto exit
    shift
    echo gerolf (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8) (%9)>> %Z%
    goto exit
:fi_trace

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
    set REPLY_BANNER=2
    set REPLY_TEXT=15
    call l_banner -debug gms %1 %2 %3 %4
:fi_db

REM 1. Check common variables: =====

REM If necessary, read launcher script to load environment:

if not %GMS_BATCH% == "" goto fi_ll
call gerolf -passive %1 %2 %3 %4 %5 %6 %7 %8 %9
if "%GMS_STARTUP%" == "" set GMS_STARTUP=%PWD%
cls
if "%OS%" == "" type %GMS_SETTING%\launch_3.scn
if not "%OS%" == "" type "%GMS_SETTING%\launch_3.scn"
if not %GMS_BATCH% == "" goto fi_lll
cls
echo GMS error: Could not set environment variables.
echo Run "gmssetup" to create GMS launcher script "gerolf"!
echo.
goto exit
:fi_lll
:fi_ll

REM 2. Check chapter variable: =====

REM GMS is not yet running:

if not (%1) == () goto fi_nr
cls
if "%OS%" == "" type %GMS_SETTING%\welcome.scn
if not "%OS%" == "" type "%GMS_SETTING%\welcome.scn"
call l_gms -welcome called_by gms
pause > nul
goto help
:fi_nr

REM Remove optional "-mime" argument at first position (browser interface):

if not %1 == -mime goto fi_mf
if (%2) == () goto else_mr
if not %2 == -read goto else_mr
    echo Content-type: application\pdf
    goto fi_mr
:else_mr
    echo Content-type: text\plain
:fi_mr
    shift
:fi_mf

REM Chapters: =====

if (%1) == () goto exit

REM Quit: -----

if (%1) == (-g) goto then_q
if (%1) == (-Q) goto then_q
if (%1) == (-.g) goto then_q
if (%1) == (-.Q) goto then_q
if (%1) == (/g) goto then_q
if (%1) == (/Q) goto then_q
if (%1) == (g) goto then_q
if (%1) == (Q) goto then_q
if (%1) == (gg) goto then_q
if not (%1) == (QQ) goto fi_q
:then_q
    cls
    REM call l_gms -goodbye called_by gms -Q
    if not %GMS_TEXTMODE% == "" goto fi_q1
    if "%OS%" == "" type %GMS_SETTING%\goodbye.scn
    if not "%OS%" == "" type "%GMS_SETTING%\goodbye.scn"
:fi_q1
    REM Remove temporary files:
    if (%1) == (-Q) goto then_q2
    if (%1) == (/Q) goto then_q2
    if (%1) == (Q) goto then_q2
    if not (%1) == (QQ) goto fi_q2
:then_q2
    if exist *.log del *.log
    if exist *.ok del *.ok
    if exist *.err del *.err
:fi_q2
    REM Return to startup folder:
    if "%GMS_STARTUP%" == "" goto fi_q3
    if "%OS%" == "" if exist %GMS_STARTUP%\%any% cd %GMS_STARTUP% > nul
    if not "%OS%" == "" if exist "%GMS_STARTUP%\%any%" cd "%GMS_STARTUP%"> nul
:fi_q3
    call g_vars -write_memo
    REM Restore environment:
    call g_vars -clear called_by gms -g
    if %3 == "g_good" goto exit
    REM Final unsets:
```

```

set GMS_BREAK=
set GMS_DEBUG=
set GMS_SETTING=
set Z=
REM pause
goto exit
:fi_q

REM Create: -----

if (%1) == (-c) goto then_c
if (%1) == (-C) goto then_c
if (%1) == (--c) goto then_c
if (%1) == (--C) goto then_c
if (%1) == (/c) goto then_c
if (%1) == (/C) goto then_c
if (%1) == (c) goto then_c
if (%1) == (C) goto then_c
if (%1) == (cc) goto then_c
if not (%1) == (CC) goto fi_c
:then_c
set GMS_FILE=%2
set GMS_MESSAGE=%GMS_FILE%
REM Check if there is an old file with that name:
if "%OS%" == "" if not exist %GMS_FILE% goto fi_c1
if not "%OS%" == "" if not exist "%GMS_FILE%" goto fi_c1
set GMS_MESSAGE=GMS error: Did not overwrite existing file %GMS_FILE%
goto fi_c5
:fi_c1
REM Look for template in current directory and copy it:
if (%3) == () goto else_c5
if "%OS%" == "" if not exist %3 goto fi_c2
if not "%OS%" == "" if not exist "%3" goto fi_c2
if "%OS%" == "" copy %3 %GMS_FILE% > nul
if not "%OS%" == "" copy "%3" "%GMS_FILE%" > nul 2> nul
call g_vars -ini_file %2
rem call g_vars -abort called_by gms -c
goto fi_c5
:fi_c2
REM Look for template in template directory and copy it:
if "%OS%" == "" if not exist %GMS_TEMPLATE%\%3\%3.htm goto fi_c3
if not "%OS%" == "" if not exist "%GMS_TEMPLATE%\%3\%3.htm" goto fi_c3
if "%OS%" == "" copy %GMS_TEMPLATE%\%3\%3.htm %GMS_FILE% > nul
if not "%OS%" == "" copy "%GMS_TEMPLATE%\%3\%3.htm" "%GMS_FILE%" > nul
call g_vars -ini_file %2
rem call g_vars -abort called_by gms -c
goto fi_c5
:fi_c3
if "%OS%" == "" if not exist %GMS_TEMPLATE%\%3 goto else_c5
if not "%OS%" == "" if not exist "%GMS_TEMPLATE%\%3" goto else_c5
if "%OS%" == "" copy %GMS_TEMPLATE%\%3 %GMS_FILE% > nul
if not "%OS%" == "" copy "%GMS_TEMPLATE%\%3" "%GMS_FILE%" > nul
call g_vars -ini_file %2
rem call g_vars -abort called_by gms -c
goto fi_c5
:else_c5
set GMS_MESSAGE=GMS: Creating default file. There is no template %3
if "%OS%" == "" copy %GMS_TEMPLATE%\default\default.htm %GMS_FILE% > nul
if not "%OS%" == "" copy "%GMS_TEMPLATE%\default\default.htm" "%GMS_FILE%" > nul
:fi_c5
goto help
:fi_c

REM Open: -----

if (%1) == (-o) goto then_o
if (%1) == (-O) goto then_o
if (%1) == (--o) goto then_o
if (%1) == (--O) goto then_o
if (%1) == (/o) goto then_o
if (%1) == (/O) goto then_o

```

```

if (%1) == (o) goto then_o
if (%1) == (O) goto then_o
if (%1) == (oo) goto then_o
if not (%1) == (OO) goto fi_o
:then_o
shift
if (%1) == () set GMS_FILE=
if (%1) == () goto help
call g_vars -ini_file %1 %2 %3 %4 %5 %6 %7 %8 %9
goto help
:fi_o

REM View: -----

if (%1) == (-v) goto then_v
if (%1) == (-V) goto then_v
if (%1) == (--v) goto then_v
if (%1) == (--V) goto then_v
if (%1) == (/v) goto then_v
if (%1) == (/V) goto then_v
if (%1) == (v) goto then_v
if (%1) == (V) goto then_v
if (%1) == (vv) goto then_v
if not (%1) == (VV) goto fi_v
:then_v
set GMS_COMMAND=%GMS_VIEWER%
%GMS_REMODRV%
goto then_x
:fi_v

REM Edit: -----

if (%1) == (-e) goto then_e
if (%1) == (-E) goto then_e
if (%1) == (--e) goto then_e
if (%1) == (--E) goto then_e
if (%1) == (/e) goto then_e
if (%1) == (/E) goto then_e
if (%1) == (e) goto then_e
if (%1) == (E) goto then_e
if (%1) == (ee) goto then_e
if not (%1) == (EE) goto fi_e
:then_e
set GMS_PROGRAM=text editor
set GMS_COMMAND=%GMS_EDITOR%
goto then_x
:fi_e

REM Browse: -----

if (%1) == (-b) goto then_b
if (%1) == (-B) goto then_b
if (%1) == (--b) goto then_b
if (%1) == (--B) goto then_b
if (%1) == (/b) goto then_b
if (%1) == (/B) goto then_b
if (%1) == (b) goto then_b
if (%1) == (B) goto then_b
if (%1) == (bb) goto then_b
if not (%1) == (BB) goto fi_b
:then_b
if "%GMS_FILE%" == "_folder.err" goto help
set GMS_PROGRAM=markup file browser
set GMS_COMMAND=%GMS_BROWSER%
goto then_x
:fi_b

REM Analyse folder kernel: -----

if not (%1) == (-analyse_kernel) goto fi_ak
REM Check file:

```



```

    echo %slashline%%slashline%%slashes%> errfoot.err
    echo %GMS_FOLDER% >> errfoot.err
set slashline=
set slashes=
copy errhead.err + %GMS_BASE%.tmp + errfoot.err %GMS_BASE%.err > nul
:fi_a2
if "%GMS_MODE%" == "quiet" ren %GMS_BASE%.tmp %GMS_BASE%.err > nul
if not "%GMS_MODE%" == "quiet" if exist %GMS_BASE%.pmt del %GMS_BASE%.pmt
if not "%GMS_MODE%" == "quiet" if exist %GMS_BASE%.tmp del %GMS_BASE%.tmp
if exist errhead.err del errhead.err
if exist errfoot.err del errfoot.err
if "%GMS_MODE%" == "quiet" goto fi_a2
cls
if not "%GMS_MODE%" == "quiet" if "%GMS_FreeDOS%" == "" mode con lines=25>nul
if not "%GMS_MODE%" == "quiet" if not "%GMS_FreeDOS%" == "" mode co80,25
if not "%2" == "-quiet" echo. %GMS_FILE%
if "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" if not "%2" == "-quiet" type "%GMS_SETTING%\desktop.scn"
if not "%2" == "-quiet" echo %GMS_FOLDER%
set GMS_PROGRAM=%GMS_TSETTER% engine
if "%OS%" == "Windows_9x" set GMS_PROGRAM=%GMS_TSETTER% engine - please wait
if not "%GMS_MODE%" == "quiet" call l_gms -desktop -clear called_by gms -t
set GMS_PROGRAM=%GMS_TSETTER% engine
set GMS_FILE=%backup_file%
set TEXINPUTS=.\;%TEXMFCNF%;%TEX_USER%\
set WEB2C=%GMS_SETTING%
if not %GMS_DEBUG% == Z goto fi_t0
if "%OS%" == "" goto fi_t0
    echo GMS_TSETTER=%GMS_TSETTER%>> %Z%
    echo GMS_REMODRV=%GMS_REMODRV%>> %Z%
    echo GMS_FOLDER=%GMS_FOLDER%>> %Z%
    echo GMS_FILE=%GMS_FILE%>> %Z%
    echo GMS_SHORT=%GMS_SHORT%>> %Z%
    echo GMS_BASE=%GMS_BASE%>> %Z%
    echo TEXINPUTS=%TEXINPUTS%>> %Z%
:fi_t0
%GMS_REMODRV%
if "%OS%" == "" if exist %GMS_FOLDER%\%any% cd %GMS_FOLDER%
if not "%OS%" == "" if exist "%GMS_FOLDER%\%any%" cd "%GMS_FOLDER%"
if not "%GMS_DRDOS%" == "" cd %GMS_FOLDER%
if not "%GMS_MODE%" == "quiet" call g_dos -begin_runtime called_by gms -t
if "%OS%" == "" echo > gmserr.log
if "%OS%" == "" call %GMS_TSETTER% -programe=gerolf %GMS_FILE%
if not "%OS%" == "" call %GMS_TSETTER% -programe=gerolf %GMS_FILE% 2>gmserr.log
if not "%GMS_MODE%" == "quiet" call g_dos -end_runtime called_by gms -t
set TEXINPUTS=
set WEB2C=
set GMS_FILE=%GMS_BASE%.log
if "%GMS_MODE%" == "quiet" goto exit
set slashline=////////////////////////////////////
if "%OS%" == "Windows_9x" set slashline=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set slashline=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
set slashes=////////////////////////////////////
if "%OS%" == "Windows_9x" set slashes=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set slashes=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %GMS_FILE% - GMS: Running %GMS_PROGRAM% ...> loghd.log
echo %slashline%%slashline%%slashes%>> loghd.log
if exist logft.log del logft.log
if exist readme.txt type readme.txt >> loghd.log
if not "%GMS_MODE%" == "quiet" echo GMS run time: %REPLY_AFTER%>logft.log
echo %slashline%%slashline%%slashes%>> logft.log
echo %GMS_FOLDER%>> logft.log
echo.>> logft.log
set slashline=
set slashes=
if exist %GMS_FILE% ren %GMS_FILE% %GMS_BASE%.pmt
set a0=(format
set a1=entering extended mode
set a2=\\write18 enabled.
sed "s/%a0%/\\n&/;s/%a1%/;/s/%a2%/;/\^$/d" <%GMS_BASE%.pmt >%GMS_BASE%.pmu
sed "s/</>/g;/s/>/>/g" <%GMS_BASE%.pmt >%GMS_BASE%.tmp
set a2=
set a1=
set a0=
if "%OS%" == "Windows_9x" echo. > gmserr.log
if not exist gmserr.log echo.>gmserr.log
if "%GMS_MODE%" == "quiet" goto fi_t23
copy loghd.log + %GMS_BASE%.tmp + logft.log +gmserr.log %GMS_BASE%.pmt>nul
:fi_t23
if "%GMS_MODE%" == "quiet" copy %GMS_BASE%.tmp+gmserr.log %GMS_BASE%.pmt>nul
sed "$d" < %GMS_BASE%.pmt > %GMS_FILE%
if exist %GMS_BASE%.pmt del %GMS_BASE%.pmt
if exist %GMS_BASE%.pmu del %GMS_BASE%.pmu

```

```

cls
if not "%GMS_MODE%" == "quiet" if "%GMS_FreeDOS%" == "" mode con lines=25>nul
if not "%GMS_MODE%" == "quiet" if not "%GMS_FreeDOS%" == "" mode co80,25
if not "%2" == "-quiet" echo. %GMS_FILE%
if "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" if not "%2" == "-quiet" type "%GMS_SETTING%\desktop.scn"
if not "%2" == "-quiet" echo %GMS_FOLDER%
set GMS_PROGRAM=%GMS_TSETTER% engine
if "%OS%" == "Windows_9x" set GMS_PROGRAM=%GMS_TSETTER% engine - please wait
if not "%GMS_MODE%" == "quiet" call l_gms -desktop -clear called_by gms -t
set GMS_PROGRAM=%GMS_TSETTER% engine
set GMS_FILE=%backup_file%
set TEXINPUTS=.\;%TEXMFCNF%;%TEX_USER%\
set WEB2C=%GMS_SETTING%
if not %GMS_DEBUG% == Z goto fi_t0
if "%OS%" == "" goto fi_t0
    echo GMS_TSETTER=%GMS_TSETTER%>> %Z%
    echo GMS_REMODRV=%GMS_REMODRV%>> %Z%
    echo GMS_FOLDER=%GMS_FOLDER%>> %Z%
    echo GMS_FILE=%GMS_FILE%>> %Z%
    echo GMS_SHORT=%GMS_SHORT%>> %Z%
    echo GMS_BASE=%GMS_BASE%>> %Z%
    echo TEXINPUTS=%TEXINPUTS%>> %Z%
:fi_t0
%GMS_REMODRV%
if "%OS%" == "" if exist %GMS_FOLDER%\%any% cd %GMS_FOLDER%
if not "%OS%" == "" if exist "%GMS_FOLDER%\%any%" cd "%GMS_FOLDER%"
if not "%GMS_DRDOS%" == "" cd %GMS_FOLDER%
if not "%GMS_MODE%" == "quiet" call g_dos -begin_runtime called_by gms -t
if "%OS%" == "" echo > gmserr.log
if "%OS%" == "" call %GMS_TSETTER% -programe=gerolf %GMS_FILE%
if not "%OS%" == "" call %GMS_TSETTER% -programe=gerolf %GMS_FILE% 2>gmserr.log
if not "%GMS_MODE%" == "quiet" call g_dos -end_runtime called_by gms -t
set TEXINPUTS=
set WEB2C=
set GMS_FILE=%GMS_BASE%.log
if "%GMS_MODE%" == "quiet" goto exit
set slashline=////////////////////////////////////
if "%OS%" == "Windows_9x" set slashline=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set slashline=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
set slashes=////////////////////////////////////
if "%OS%" == "Windows_9x" set slashes=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set slashes=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %GMS_FILE% - GMS: Running %GMS_PROGRAM% ...> loghd.log
echo %slashline%%slashline%%slashes%>> loghd.log
if exist logft.log del logft.log
if exist readme.txt type readme.txt >> loghd.log
if not "%GMS_MODE%" == "quiet" echo GMS run time: %REPLY_AFTER%>logft.log
echo %slashline%%slashline%%slashes%>> logft.log
echo %GMS_FOLDER%>> logft.log
echo.>> logft.log
set slashline=
set slashes=
if exist %GMS_FILE% ren %GMS_FILE% %GMS_BASE%.pmt
set a0=(format
set a1=entering extended mode
set a2=\\write18 enabled.
sed "s/%a0%/\\n&/;s/%a1%/;/s/%a2%/;/\^$/d" <%GMS_BASE%.pmt >%GMS_BASE%.pmu
sed "s/</>/g;/s/>/>/g" <%GMS_BASE%.pmt >%GMS_BASE%.tmp
set a2=
set a1=
set a0=
if "%OS%" == "Windows_9x" echo. > gmserr.log
if not exist gmserr.log echo.>gmserr.log
if "%GMS_MODE%" == "quiet" goto fi_t23
copy loghd.log + %GMS_BASE%.tmp + logft.log +gmserr.log %GMS_BASE%.pmt>nul
:fi_t23
if "%GMS_MODE%" == "quiet" copy %GMS_BASE%.tmp+gmserr.log %GMS_BASE%.pmt>nul
sed "$d" < %GMS_BASE%.pmt > %GMS_FILE%
if exist %GMS_BASE%.pmt del %GMS_BASE%.pmt
if exist %GMS_BASE%.pmu del %GMS_BASE%.pmu

```

```

if exist %GMS_BASE%.tmp del %GMS_BASE%.tmp
if exist loghd.log del loghd.log
if exist logft.log del logft.log
if exist gmserr.log del gmserr.log
cls
if not "%GMS_MODE%" == "quiet" if "%GMS_FreeDOS%" == "" mode con lines=25>nul
if not "%GMS_MODE%" == "quiet" if not "%GMS_FreeDOS%" == "" mode co80,25
if not "%2" == "-quiet" echo. %GMS_FILE%
if "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" if not "%2" == "-quiet" type "%GMS_SETTING%\desktop.scn"
if not "%2" == "-quiet" echo. %GMS_FOLDER%
if not "%GMS_LOGSHORT%" == "" goto fi_t1
call g_dos -chdir called_by gms -t
set GMS_LOGBASE=%GMS_BASE%
set GMS_LOGSHORT=%GMS_SHORT%
:fi_t1
set GMS_BASE=%GMS_LOGBASE%
set GMS_SHORT=%GMS_LOGSHORT%
goto then_x
:fi_t

REM Initialize TeX format: -----

if (%1) == (-i) goto then_i
if (%1) == (-I) goto then_i
if (%1) == (-.i) goto then_i
if (%1) == (-.I) goto then_i
if (%1) == (/i) goto then_i
if (%1) == (/I) goto then_i
if (%1) == (i) goto then_i
if (%1) == (I) goto then_i
if (%1) == (ii) goto then_i
if not (%1) == (II) goto fi_i
:then_i
%GMS_DRIVE%
call g Plug -build called_by gms -i
set backup_folder=%GMS_FOLDER%
set backup_file=%GMS_FILE%
set backup_base=%GMS_BASE%
set backup_short=%GMS_SHORT%
set GMS_FOLDER=%TEXFORMATS%
set GMS_FILE=gerolf.log
set GMS_BASE=gerolf
set GMS_SHORT=gerolf.log
set GMS_PROGRAM=%GMS_TSETTER% engine
set GMS_COMMAND=%GMS_VIEWER%
set GEROLF=*gerolf
if %GMS_TSETTER% == tex set GEROLF=gerolf
if %GMS_TSETTER% == pdftex set GEROLF=gerolf
set TEXINPUTS=%TEX_BASE%\;%TEXMFCNF%;%GMS_ROOT%\data\;%GMS_BINARIES%
set WEB2C=%GMS_SETTING%
set GMS_NOREMO=1
REM if "%OS%" == "" if not "%winbootdir%" == "." set TEXINPUTS=%GMS_ROOT%\
call g_dos -chdir called_by gms -i
if not "%2" == "-quiet" if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%2" == "-quiet" if not "%GMS_FreeDOS%" == "" mode co80,25
if not "%2" == "-quiet" call l_gms -desktop -clear called_by gms -i
set slashline=////////////////////////////////////
if "%OS%" == "Windows_9x" set slashline=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set slashline=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
set slashes=////////////////////////////////////
if "%OS%" == "Windows_9x" set slashes=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set slashes=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %GMS_FILE% - GMS: Running %GMS_PROGRAM% ...>gmshd.log
echo %slashline%slashline%slashes%>> gmshd.log
if not "%2" == "-quiet" call g_dos -begin_runtime called_by gms -i
if "%OS%" == "" goto else_iv
if "%OS%" == "Windows_9x" goto else_iv
echo \dump | "%GMS_BINARIES%\%GMS_TSETTER%" -ini %GEROLF% 2>gmserr.log
goto fi_ii
:else_iv

if not "%2" == "-quiet" goto else_v
echo \dump | "%GMS_BINARIES%\%GMS_TSETTER%" -ini %GEROLF% > nul
goto fi_ii
:else_v
echo \dump | "%GMS_BINARIES%\%GMS_TSETTER%" -ini %GEROLF%
goto fi_ii
:else_ii
if not "%2" == "-quiet" goto elseiii
echo \dump | "%GMS_BINARIES%\%GMS_TSETTER%" -ini %GEROLF% > nul
goto fi_ii
:elseiii
echo \dump | "%GMS_BINARIES%\%GMS_TSETTER%" -ini %GEROLF%
:fi_ii
if not "%2" == "-quiet" call g_dos -end_runtime called_by gms -i
if not "%2" == "-quiet" echo GMS run time: %REPLY_AFTER% >> gerolf.log
if exist gerolf.log ren gerolf.log gms.pmt
echo %slashline%slashline%slashes%>> gmsft.log
echo %GMS_FOLDER%>> gmsft.log
set slashline=
set slashes=
sed "s/(INITEX)/\n&/" < gms.pmt > gms.tmp
if not exist gmserr.log echo.>gmserr.log
copy gmshd.log + gms.tmp + gmsft.log + gmserr.log gerolf.log > nul
if exist gmshd.log del gmshd.log
if exist gms.pmt del gms.pmt
if exist gms.tmp del gms.tmp
if exist gmserr.log del gmserr.log
if exist gmsft.log del gmsft.log
REM Move .log and .enc files to appropriate locations:
set logf=gerolf.log
if "%OS%" == "" if exist %logf% copy %logf% %GMS_SETTING% > nul
if not "%OS%" == "" if exist %logf% copy %logf% "%GMS_SETTING%" > nul
if exist %logf% del %logf%
set logf=
REM if exist *.enc copy *.enc %GMS_ROOT%\data\enc > nul
if "%OS%" == "" if exist *.enc copy *.enc %GMS_ROOT%\data\enc > nul
if not "%OS%" == "" if exist *.enc copy *.enc "%GMS_ROOT%\data\enc" > nul
if exist *.enc del *.enc
if exist *.lst del *.lst
if not "%2" == "-quiet" if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%2" == "-quiet" if not "%GMS_FreeDOS%" == "" mode co80,25
if not "%2" == "-quiet" echo. %GMS_FILE%
if "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" if not "%2" == "-quiet" type "%GMS_SETTING%\desktop.scn"
if not "%2" == "-quiet" echo. %GMS_SETTING%
set TEXINPUTS=
set WEB2C=
set GMS_FOLDER=%GMS_SETTING%
set GEROLF=
call g_dos -chdir called_by gms -i
call g Plug -remove called_by gms -i
goto then_x
:fi_i

REM Read output file: -----

if (%1) == (-r) goto then_r
if (%1) == (-R) goto then_r
if (%1) == (-.r) goto then_r
if (%1) == (-.R) goto then_r
if (%1) == (/r) goto then_r
if (%1) == (/R) goto then_r
if (%1) == (r) goto then_r
if (%1) == (R) goto then_r
if (%1) == (rr) goto then_r
if not (%1) == (RR) goto fi_r
:then_r
if "%GMS_FILE%" == "_folder.err" goto help
set backup_file=%GMS_FILE%
set GMS_FILE=%GMS_BASE%.pdf
set GMS_PROGRAM=portable document file reader

```

```

set GMS_COMMAND=%GMS_READER%
cls
if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%GMS_FreeDOS%" == "" mode co80,25
if not "%2" == "-quiet" echo %GMS_FILE% · Running %GMS_PROGRAM% ...
if "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" if not "%2" == "-quiet" type "%GMS_SETTING%\desktop.scn"
if not "%2" == "-quiet" echo. %GMS_FOLDER%
goto then_x
:fi_r

```

REM Learn:

```

if (%1) == (-1) goto then_l
if (%1) == (-L) goto then_l
if (%1) == (-·1) goto then_l
if (%1) == (-·L) goto then_l
if (%1) == (/1) goto then_l
if (%1) == (/L) goto then_l
if (%1) == (1) goto then_l
if (%1) == (L) goto then_l
if (%1) == (11) goto then_l
if not (%1) == (1L) goto fi_l
:then_l
set GMS_MESSAGE=handbook.htm           The
set GMS_MESSAGE=%GMS_MESSAGE% Gerolf Markup Shredder Handbook
set GMS_REMODRV=%GMS_DRIVE%
%GMS_REMODRV%
set GMS_FOLDER=%GMS_TEMPLATE%\handbook
set GMS_FILE=handbook.htm
set GMS_SHORT=handbook.htm
set GMS_BASE=handbook
call g_dos ·chdir called_by gms ·l
goto help
:fi_l

```

REM Select:

```

if (%1) == (-s) goto then_s
if (%1) == (-S) goto then_s
if (%1) == (-·s) goto then_s
if (%1) == (-·S) goto then_s
if (%1) == (/s) goto then_s
if (%1) == (/S) goto then_s
if (%1) == (s) goto then_s
if (%1) == (S) goto then_s
if (%1) == (ss) goto then_s
if not (%1) == (SS) goto fi_s
:then_s
set GMS_MESSAGE= gerolf: The GMS launcher script. Here
set GMS_MESSAGE=%GMS_MESSAGE% you can edit the programs
set GMS_MESSAGE=%GMS_MESSAGE% to use via GMS
set GMS_REMODRV=%GMS_DRIVE%
%GMS_REMODRV%
set GMS_FOLDER=%GMS_SETTING%
call g_dos ·chdir called_by gms ·s
set GMS_FILE=gerolf.bat
set GMS_SHORT=gerolf.bat
set GMS_BASE=gerolf
goto help
:fi_s

```

REM Write font map:

```

if (%1) == (-w) goto then_w
if (%1) == (-W) goto then_w
if (%1) == (-·w) goto then_w
if (%1) == (-·W) goto then_w
if (%1) == (/w) goto then_w
if (%1) == (/W) goto then_w
if (%1) == (w) goto then_w

```

```

if (%1) == (w) goto then_w
if (%1) == (ww) goto then_w
if not (%1) == (WW) goto fi_w
REM %2: empty or "-quiet"
:then_w
set backup_folder=%GMS_FOLDER%
set backup_file=%GMS_FILE%
set backup_base=%GMS_BASE%
set backup_short=%GMS_SHORT%
set GMS_FOLDER=%GMS_SETTING%
set GMS_FILE=font.map
set GMS_BASE=font
set GMS_SHORT=font.map
set GMS_COMMAND=%GMS_VIEWER%
set GMS_PROGRAM=font installer
set GMS_NOEMO=1
%GMS_DRIVE%
call g_dos ·chdir called_by gms ·w
if not "%2" == "-quiet" if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%2" == "-quiet" if not "%GMS_FreeDOS%" == "" mode co80,25
if not "%2" == "-quiet" echo. %GMS_FILE%
if "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" if not "%2" == "-quiet" type "%GMS_SETTING%\desktop.scn"
if not "%2" == "-quiet" echo. %GMS_FOLDER%
if not "%2" == "-quiet" call g_dos ·begin_runtime called_by gms ·w
if "%OS%" == "" goto else_w1
if "%OS%" == "Windows_9x" goto else_w1
call g_font ·build_all %2 2> gmserr.log
goto fi_w1
:else_w1
call g_font ·build_all %2
:fi_w1
set slashline=////////////////////////////////////
if "%OS%" == "Windows_9x" set slashline=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set slashes=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
set slashes=////////////////////////////////////
if "%OS%" == "Windows_9x" set slashes=XXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set slashes=XXXXXXXXXXXXXXXXXXXX
if not "%2" == "-quiet" call g_dos ·end_runtime called_by gms ·w
if not "%2" == "-quiet" echo.>> font.map
if not "%2" == "-quiet" echo %% GMS run time: %REPLY_AFTER%>> font.map
echo.>> font.map
if "%OS%" == "" goto fi_w2
echo %% Warnings, errors, missing glyph list: see font.log>>font.map
echo.>> font.map
echo %% %slashline%slashes%>> font.map
:fi_w2
echo %% %GMS_FOLDER%>>font.map
if not exist gmserr.log echo.>gmserr.log
sed "/^$/d; s/^\% /; s/^\% $/ /" < gmserr.log >> font.log
if exist gmserr.log del gmserr.log
if "%GMS_DRDOS%" == "" echo %% %slashline%slashes%>> font.log
echo %% %GMS_FOLDER%>> font.log
set slashline=
set slashes=
goto then_x
:fi_w

```

REM Execute command:

```

if not (%1) == (-execute) goto fi_x
:then_x
if "%GMS_FILE%" == "" goto else_x1
if "%OS%" == "" if not exist %GMS_FOLDER%\%GMS_FILE% goto else_x1
if not "%OS%" == "" if not exist "%GMS_FOLDER%\%GMS_FILE%" goto else_x1
if "%GMS_MODE%" == "quiet" goto fi_x2
if "%2" == "-quiet" goto fi_x2
if not "%GMS_COMMAND%" == "%GMS_VIEWER%" goto else_x2
REM Contribute to gmsdebug.log:
if not %GMS_DEBUG% == Z goto fi_x21
if "%OS%" == "" goto fi_x21

```

```

    echo  GMS_COMMAND=%GMS_COMMAND%>> %Z%
    echo  GMS_SHORT=%GMS_SHORT%>> %Z%
:fi_x21
REM Change codepage, color, line mode; run viewer:
if not "%OS%" == "" if not "%OS%" == "Windows_9x" color F7
cls
echo.
if not "%OS%" == "" chcp %GMS_CHCP% > nul
cls
if not "%OS%" == "" if not "%OS%" == "Windows_9x" color F0
if "%GMS_FreeDOS%" == "" mode con lines=50 > nul
if not "%GMS_FreeDOS%" == "" mode co80,50
cls
if "%GMS_NOREMO%" == "" set GMS_ACTDRV=%GMS_REMODRV%
if not "%GMS_NOREMO%" == "" set GMS_ACTDRV=%GMS_DRIVE%
    %GMS_ACTDRV%
    browse %GMS_SHORT%
    %GMS_REMODRV%
set GMS_NOREMO=
set GMS_ACTDRV=
cls
if not "%OS%" == "" if not "%OS%" == "Windows_9x" color F7
cls
echo.
if not "%OS%" == "" chcp %GMS_INICP% > nul
cls
if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%GMS_FreeDOS%" == "" mode co80,25
if not "%2" == "-quiet" echo. %GMS_FILE%
if "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn%
if not "%2" == "-quiet" echo. %GMS_FOLDER%
if not "%OS%" == "" if not "%OS%" == "Windows_9x" color 07
goto fi_x2
:else_x2
if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%GMS_FreeDOS%" == "" mode co80,25
if not "%2" == "-quiet" echo. %GMS_FILE% - Running %GMS_PROGRAM% ...
if "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn%
if not "%2" == "-quiet" echo. %GMS_FOLDER%
REM Contribute to gmsdebug.log:
if not %GMS_DEBUG% == Z goto fi_x22
if "%OS%" == "" goto fi_x22
    echo  GMS_COMMAND=%GMS_COMMAND%>> %Z%
    echo  GMS_REMODRV=%GMS_REMODRV%>> %Z%
    echo  GMS_FOLDER=%GMS_FOLDER%>> %Z%
    echo  GMS_FILE=%GMS_FILE%>> %Z%
:fi_x22
REM Execute command:
if "%GMS_COMMAND%" == "default" set GMS_COMMAND=
%GMS_REMODRV%
if "%OS%" == "" if not "%GMS_FOLDER%" == "" cd %GMS_FOLDER%>nul
if not "%OS%" == "" if not "%GMS_FOLDER%" == "" cd "%GMS_FOLDER%">nul
if "%OS%" == "" call %GMS_COMMAND% %GMS_FOLDER%\%GMS_FILE%
if not "%OS%" == "" call %GMS_COMMAND% "%GMS_FOLDER%\%GMS_FILE%"
if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%GMS_FreeDOS%" == "" mode co80,25
    if not "%2" == "-quiet" echo. %GMS_FILE% - Running %GMS_PROGRAM% ...
    if "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" if not "%2" == "-quiet" type %GMS_SETTING%\desktop.scn%
    if not "%2" == "-quiet" echo. %GMS_FOLDER%
:fi_x2
goto fi_x1
:else_x1
cls
echo %GMS_FILE% - GMS error: File not found
if "%OS%" == "" type %GMS_SETTING%\desktop.scn
if not "%OS%" == "" type %GMS_SETTING%\desktop.scn"
echo %GMS_FOLDER%
if "%OS%" == "" goto fi_x11
set mymsg= GMS error: File not found
if %GMS_DEBUG% == Z echo %mymsg% (%GMS_FOLDER%\%GMS_FILE%)." >> %Z%
set mymsg=
:fi_x11
reply -sleep
:fi_x1
set GMS_COMMAND=
set GMS_PROGRAM=
if not "%backup_short%" == "" set GMS_SHORT=%backup_short%
if not "%backup_base%" == "" set GMS_BASE=%backup_base%
if not "%backup_file%" == "" set GMS_FILE=%backup_file%
if not "%backup_folder%" == "" set GMS_FOLDER=%backup_folder%
if not "%backup_folder%" == "" call g_dos -chdir called_by gms -execute
set backup_short=
set backup_base=
set backup_file=
set backup_folder=
goto help
:fi_x

REM Show error: .....
:else
if "%OS%" == "" if exist %GMS_FOLDER%\%GMS_FILE% goto fi
if not "%OS%" == "" if exist "%GMS_FOLDER%\%GMS_FILE%" goto fi
set GMS_MESSAGE=GMS error: No file or command line option %1
reply -sleep
if "%OS%" == "" goto exit
if %GMS_DEBUG% == Z echo GMS error: No file or command line option %1>>%Z%
goto exit
:fi

REM Epilogue: =====
:help
if "%2" == "-quiet" goto exit
if not "%GMS_TEXTMODE%" == "" goto exit
cls
if (%GMS_FILE%) == () echo GMS: No markup file opened.
if not (%GMS_FILE%) == () echo %GMS_FILE%
if "%OS%" == "" type %GMS_SETTING%\menu.scn
if not "%OS%" == "" type %GMS_SETTING%\menu.scn"
:exit

```

I_banner.bat

```
REM l_banner.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_BANNER=20060927

REM Prologue: =====

REM Debug: Banners are out of the X/Y-debugging system (which uses banners)...

if "%OS%" == "" goto fi_db1
if %GMS_DEBUG% == - goto exit
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
:fi_db1

REM Size: -----

set BACKUP_SIZE=%REPLY_SIZE%
set REPLY_SIZE=78

set BACKUP_OFFSET=%REPLY_OFFSET%
set REPLY_OFFSET=0

REM Chapters: =====

REM Not running:

if not %1 == "" goto fi_se
echo Gerolf Markup Shredder (l_banner) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_se

REM Debug: -----

if not %1 == -debug goto fi_db
REM Upper:
set REPLY_OFFSET=0
set REPLY_SIZE=22
reply -banner 2 "Module: %2"
set REPLY_OFFSET=19
set REPLY_SIZE=22
reply -banner 2 "Action: %3"
set REPLY_OFFSET=38
set REPLY_SIZE=22
reply -banner 2 "Hotkey: %4"
set REPLY_OFFSET=57
set REPLY_SIZE=25
reply -banner 2 "Coldkey: %5"
REM Lower:
set REPLY_OFFSET=0
set REPLY_SIZE=22
reply -banner 23 "Offset: %BACKUP_OFFSET%"
set REPLY_OFFSET=19
set REPLY_SIZE=22
reply -banner 23 "Size: %BACKUP_SIZE%"
set REPLY_OFFSET=38
set REPLY_SIZE=22
reply -banner 23 "Items: %REPLY_ITEMS%"
set REPLY_OFFSET=57
set REPLY_SIZE=25
reply -banner 23 "Data: %6"
if %GMS_DEBUG% == Y reply -sleep
goto exit
:fi_db

REM Diverse: -----

REM No module:
if not %1 == -no_module goto fi_nm
reply -banner 1 "GMS error: No module '%2'"
set REPLY_OFFSET=57
set REPLY_SIZE=25
reply -banner 1 "Gerolf Markup Shredder"
reply -sleep
goto exit
:fi_nm

REM No action:
if not %1 == -no_action goto fi_na
reply -banner 1 "GMS error: No action '%3'"
set REPLY_OFFSET=57
set REPLY_SIZE=25
reply -banner 1 "Module: '%2'"
reply -sleep
goto exit
:fi_na

REM No hot:
if not %1 == -no_hot goto fi_nh
reply -banner 1 "GMS error: No hotkey '%3'"
set REPLY_OFFSET=57
set REPLY_SIZE=25
reply -banner 1 "Module: '%2'"
reply -sleep
goto exit
:fi_nh

REM No cold:
if not %1 == -no_cold goto fi_nc
reply -banner 1 "GMS error: No coldkey '%3'"
set REPLY_OFFSET=57
set REPLY_SIZE=25
reply -banner 1 "Module: '%2'"
reply -sleep
goto exit
:fi_nc

REM No file:
if not %1 == -no_file goto fi_nf
reply -banner 1 "GMS error: No file."
set REPLY_OFFSET=57
set REPLY_SIZE=25
reply -banner 1 "Module: '%2'"
reply -sleep
goto exit
:fi_nf

REM Upper: -----

if not %1 == -upper goto fi_u

REM First upper banner:
if not %2 == -first goto fi_ul
reply -banner 1 "Gerolf Markup Shredder %GMS_VERSION%"
set REPLY_OFFSET=54
set REPLY_SIZE=27
reply -banner 1 "MarkupShredder@Gerolf.org "
goto exit
:fi_ul

REM Last upper banner:
if not %2 == -last goto fi_ult
```

```

reply -banner 1 "MarkupShredder@Gerolf.org"
set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 1 "www.Gerolf.org"
goto exit
:fi_ult

REM Write handbook title on upper banner:
if not %2 == -handbook goto fi_uht
reply -banner 1 "%GMS_FILE%"
set REPLY_OFFSET=44
set REPLY_SIZE=38
reply -banner 1 "The Gerolf Markup Shredder Handbook"
goto exit
:fi_uht

REM Write current file on upper banner:
REM This does not work on DOS:
if not %2 == -file goto fi_uf
reply -banner 1 "%GMS_FILE%"
set REPLY_OFFSET=57
set REPLY_SIZE=25
reply -banner 1 "Gerolf Markup Shredder "
goto exit
:fi_uf

REM Write current template on upper banner:
if not %2 == -template goto fi_ut
reply -banner 1 "Template: %GMS_TEMPLATEFILE%"
set REPLY_OFFSET=57
set REPLY_SIZE=25
reply -banner 1 "Gerolf Markup Shredder"
goto exit
:fi_ut

REM Write help text for codepage (upper half):
if not %2 == -cp_up goto fi_ucu
reply -banner 1 "Terminal codepage (upper half): cp%GMS_CURCP%"
set REPLY_OFFSET=57
set REPLY_SIZE=24
reply -banner 1 "Gerolf Markup Shredder "
goto exit
:fi_ucu

if not %2 == -cp_up_dos goto fi_ucv
reply -banner 1 "Upper half"
set REPLY_OFFSET=57
set REPLY_SIZE=24
reply -banner 1 "Gerolf Markup Shredder "
goto exit
:fi_ucv

REM Write help text for codepage (lower half):
if not %2 == -cp_lo goto fi_ucl
reply -banner 1 "American Standard Code for Information Interchange"
set REPLY_OFFSET=51
set REPLY_SIZE=31
reply -banner 1 " (only if in range 32 - 126) " X
goto exit
:fi_ucl

:fi_u

REM Lower: -----
if not %1 == -lower goto else_1

REM Write domain on lower banner:
if not %2 == -domain goto fi_ldo
set REPLY_OFFSET=65
set REPLY_SIZE=16

reply -banner 24 "www.Gerolf.org"
goto exit
:fi_ldo

REM First lower banner:
if not %2 == -first goto fi_lft
reply -banner 24 "The typesetting program that uses TeX for"
set REPLY_OFFSET=43
set REPLY_SIZE=38
reply -banner 24 "document conversion from HTML to PDF "
goto exit
:fi_lft

REM Write current folder on lower banner:
if not %2 == -folder goto fi_lfo
REM There must be one space before the second quoting mark for 'C:\' etc.:
reply -banner 24 "%GMS_FOLDER% "
set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 24 "www.Gerolf.org "
goto exit
:fi_lfo

REM Write warning for select dialog:
if not %2 == -select goto fi_lse
reply -banner 24 "Warning: Changes done here will be written"
set REPLY_OFFSET=44
set REPLY_SIZE=38
reply -banner 24 "to the GMS launcher script 'gerolf'"
goto exit
:fi_lse

REM Write help text for menu animation:
if not %2 == -animate goto fi_lan
reply -banner 24 "Menu animation is currently turned %GMS_ANIMATE%"
set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 24 "www.Gerolf.org"
goto exit
:fi_lan

REM Write help text for debugging:
if not %2 == -debug goto fi_ldb
reply -banner 24 "X: trace modules, Y: slow motion, Z: write log,"
set REPLY_OFFSET=49
set REPLY_SIZE=32
reply -banner 24 "0: no debug. www.Gerolf.org"
goto exit
:fi_ldb

REM Write help text for encoding:
if not %2 == -encode goto fi_len
reply -banner 24 "CP1250 East, CP1251 Cyril, CP1252 West,"
set REPLY_OFFSET=41
set REPLY_SIZE=41
reply -banner 24 "CP1253 Greek, CP1254 Turk, CP1256 Arab"
goto exit
:fi_len

REM Write help text for color dialog:
if not %2 == -color goto fi_lcr
reply -banner 24 "Enter a color or ASCII letter number"
set REPLY_OFFSET=38
set REPLY_SIZE=44
reply -banner 24 "(or 'R' for random changes of appearance) "
goto exit
:fi_lcr

REM Write help text for codepage:
if not %2 == -codepage goto fi_lcp
reply -banner 24 "Show terminal codepage"

```

```

set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 24 "www.Gerolf.org"
goto exit
:fi_lcp

REM Write help text for terminal font:
if not %2 == -terminal goto fi_ltl
reply -banner 24 "You may have to change the terminal"
set REPLY_OFFSET=37
set REPLY_SIZE=45
reply -banner 24 "font to display the appropriate characters "
goto exit
:fi_ltl

REM Write help text for programs:
if not %2 == -program goto fi_lpm
reply -banner 24 "Select programs to associate with GMS menu items"
set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 24 "www.Gerolf.org"
goto exit
:fi_lpm

REM Write help text for viewer:
REM Fixme: should depend on OS:
if not %2 == -viewer goto fi_lvw
reply -banner 24 "Text viewers: more, browse, list, ..."
set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 24 "www.Gerolf.org"
goto exit
:fi_lvw

REM Write help text for editor:
REM Fixme: should depend on OS:
if not %2 == -editor goto fi_led
reply -banner 24 "Text editors: notepad, wordpad, html-kit, frontpg, ..."
set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 24 "www.Gerolf.org"
goto exit
:fi_led

REM Write help text for browser:
if not %2 == -browser goto fi_lbr
reply -banner 24 "HTML browsers: lynx, links, mozilla, iexplore"
set REPLY_OFFSET=37
set REPLY_SIZE=45
reply -banner 24 "netscape, opera, amaya, ..."

goto exit
:fi_lbr

REM Write help text for analyst:
if not %2 == -analyst goto fi_lay
reply -banner 24 "HTML syntax checker: tidy"
set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 24 "www.Gerolf.org"
goto exit
:fi_lay

REM Write help text for typesetter:

if not %2 == -tsetter goto fi_lts
reply -banner 24 "HTML typesetters: pdftex, pdftex, etex, tex"
set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 24 "www.Gerolf.org"
goto exit
:fi_lts

REM Write help text for reader:

if not %2 == -reader goto fi_lrr
reply -banner 24 "PDF readers: acroread, gv, ghostview, xpdf, ..."
set REPLY_OFFSET=65
set REPLY_SIZE=16
reply -banner 24 "www.Gerolf.org"
goto exit
:fi_lrr

REM Error: .....

:else_l
echo GMS error: No l_banner %1 %2
reply -sleep
:fi

:exit

REM Epilogue: =====

set REPLY_OFFSET=%BACKUP_OFFSET%
set BACKUP_OFFSET=

set REPLY_SIZE=%BACKUP_SIZE%
set BACKUP_SIZE=

```


I_box.bat

```

REM I_box.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_BOX=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_box) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call I_banner -debug I_box %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call I_banner -no_action I_box
goto exit
:fi_nd

REM Idle: -----

if not %1 == -idle goto fi_id
reply -item 4 " " " "
goto exit
:fi_id

REM Build: =====

REM Box cycle:

if not %1 == -cyc_build goto fi_bc
if not %2 == 0 reply -top %2
if not %3 == 0 reply -item %3
if not %4 == 0 reply -shadow %4
if not %5 == 0 reply -bottom %5
if not %6 == 0 reply -item %6
goto exit
:fi_bc

REM Box, y-offset=13: -----

if not %1 == -build goto fi_b
if not %GMS_ANIMATE% == on goto fi_bb
REM
REM , - Offset
call I_box -cyc_build 12 0 13 0 0
call I_box -cyc_build 11 12 14 13 12
call I_box -cyc_build 10 11 15 14 13
call I_box -cyc_build 9 10 16 15 14

call I_box -cyc_build 8 9 17 16 15
call I_box -cyc_build 7 8 18 17 16
call I_box -cyc_build 6 7 19 18 17
call I_box -cyc_build 5 6 20 19 18
call I_box -cyc_build 4 5 21 20 19
call I_box -cyc_build 3 4 22 21 20
:fi_bb
goto exit
:fi_b

REM Top box, y-offset=12: -----

if not %1 == -t12_build goto fi_b12
if not %GMS_ANIMATE% == on goto fi_b1b
REM
REM , - Offset
call I_box -cyc_build 11 0 12 0 0
call I_box -cyc_build 10 11 13 12 11
call I_box -cyc_build 9 10 14 13 12
call I_box -cyc_build 8 9 15 14 13
call I_box -cyc_build 7 8 16 15 14
call I_box -cyc_build 6 7 17 16 15
call I_box -cyc_build 5 6 18 17 16
call I_box -cyc_build 4 5 19 18 17
call I_box -cyc_build 3 4 20 19 18
:fi_b1b
goto exit
:fi_b12

REM Top box, y-offset=11: -----

if not %1 == -t11_build goto fi_b11
if not %GMS_ANIMATE% == on goto fi_b1ab
REM
REM , - Offset
call I_box -cyc_build 10 0 11 0 0
call I_box -cyc_build 9 10 12 11 10
call I_box -cyc_build 8 9 13 12 11
call I_box -cyc_build 7 8 14 13 12
call I_box -cyc_build 6 7 15 14 13
call I_box -cyc_build 5 6 16 15 14
call I_box -cyc_build 4 5 17 16 15
call I_box -cyc_build 3 4 18 17 16
:fi_b1ab
goto exit
:fi_b11

REM Top box, y-offset=8: -----

if not %1 == -t8_build goto fi_b8
if not %GMS_ANIMATE% == on goto fi_b8b
REM
REM , - Offset
call I_box -cyc_build 7 0 8 0 0
call I_box -cyc_build 6 7 9 8 7
call I_box -cyc_build 5 6 10 9 8
call I_box -cyc_build 4 5 11 10 9
call I_box -cyc_build 3 4 12 11 10
call I_box -cyc_build 0 0 13 12 11
:fi_b8b
goto exit
:fi_b8

REM Remove: =====

REM Box cycle:

if not %1 == -cyc_remove goto fi_cr
if not %2 == 0 reply -bottom %2
if not %3 == 0 reply -shadow %3
if not %4 == 0 reply -stripe %4 %REPLY_PATTERN% %REPLY_LETTER%

```

```

if not %5 == 0 reply -top %5
if not %6 == 0 reply -stripe %6 %REPLY_PATTERN% %REPLY_LETTER%
goto exit
:fi_cr

REM Box, y-offset=13: -----
if not %1 == -remove goto fi_r
if not %GMS_ANIMATE% == on goto else_rr
call l_box -cyc_remove 20 21 22 4 3
call l_box -cyc_remove 19 20 21 5 4
call l_box -cyc_remove 18 19 20 6 5
call l_box -cyc_remove 17 18 19 7 6
call l_box -cyc_remove 16 17 18 8 7
call l_box -cyc_remove 15 16 17 9 8
call l_box -cyc_remove 14 15 16 10 9
call l_box -cyc_remove 13 14 15 11 10
call l_box -cyc_remove 12 13 14 12 11
call l_box -cyc_remove 0 12 13 0 12
REM      ` - Offset
goto exit
:else_rr
for %i in (3 22 4) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (21 5 20) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (6 19 7) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (18 8 17) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (9 16 10) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (15 11 14) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (12 13) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
:fi_rr
goto exit
:fi_r

REM Top box, y-offset=12: -----
if not %1 == -t12_remove goto fi_r2
if not %GMS_ANIMATE% == on goto else_r2r
call l_box -cyc_remove 18 19 20 4 3
call l_box -cyc_remove 17 18 19 5 4
call l_box -cyc_remove 16 17 18 6 5
call l_box -cyc_remove 15 16 17 7 6
call l_box -cyc_remove 14 15 16 8 7
call l_box -cyc_remove 13 14 15 9 8
call l_box -cyc_remove 12 13 14 10 9
call l_box -cyc_remove 11 12 13 11 10
call l_box -cyc_remove 0 11 12 0 11
REM      ` - Offset
goto exit
:else_r2r
for %i in (3 20 4) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (19 5 18) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (6 17 7) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (16 8 15) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (9 14 10) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (13 11 12) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
:fi_r2r
goto exit
:fi_r2

```

```

REM Top box, y-offset=11: -----
if not %1 == -t11_remove goto fi_r1
if not %GMS_ANIMATE% == on goto else_r1r
call l_box -cyc_remove 16 17 18 4 3
call l_box -cyc_remove 15 16 17 5 4
call l_box -cyc_remove 14 15 16 6 5
call l_box -cyc_remove 13 14 15 7 6
call l_box -cyc_remove 12 13 14 8 7
call l_box -cyc_remove 11 12 13 9 8
call l_box -cyc_remove 10 11 12 10 9
call l_box -cyc_remove 0 10 11 0 10
REM      ` - Offset
goto exit
:else_r1r
for %i in (3 18 4) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (17 5 16) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (6 15 7) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (14 8 13) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (9 12) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (10 11) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
:fi_r1r
goto exit
:fi_r1

REM Top box, y-offset=8: -----
if not %1 == -t8_remove goto fi_r8
if not %GMS_ANIMATE% == on goto else_r8r
call l_box -cyc_remove 13 0 0 0 0
call l_box -cyc_remove 12 13 0 0 0
call l_box -cyc_remove 11 12 13 3 0
call l_box -cyc_remove 10 11 12 4 3
call l_box -cyc_remove 9 10 11 5 4
call l_box -cyc_remove 8 9 10 6 5
call l_box -cyc_remove 7 8 9 7 6
call l_box -cyc_remove 0 7 8 0 7
REM      ` - Offset
goto exit
:else_r8r
for %i in (3 13 4) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (12 5 11) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (6 10 7) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (9 8) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
:fi_r8r
goto exit
:fi_r8

REM Not found: -----
:else
call l_banner -no_action l_box %1
:fi

:exit

```

I_code.bat

```
REM I_code.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_CODE=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_code) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call I_banner -debug I_code %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call I_banner -no_action I_code
goto exit
:fi_nd

REM Code column (upper half): -----

if not %1 == -code_up goto fi_cu
REM Character:
set REPLY_SIZE=1
set REPLY_OFFSET=%OFFSET_A%
reply -stripe 5 %REPLY_HOTKEY% %2
reply -stripe 6 %REPLY_HOTKEY% %3
reply -stripe 7 %REPLY_HOTKEY% %4
reply -stripe 8 %REPLY_HOTKEY% %5
reply -stripe 9 %REPLY_HOTKEY% %6
reply -stripe 10 %REPLY_HOTKEY% %7
reply -stripe 11 %REPLY_HOTKEY% %8
if "%OS%" == "" if "%9" == "7" goto fi_7
reply -stripe 12 %REPLY_HOTKEY% %9
:fi_7
REM Gap:
set REPLY_SIZE=1
set REPLY_OFFSET=%OFFSET_B%
reply -stripe 12 0 32
reply -stripe 11 0 32
reply -stripe 10 0 32
reply -stripe 9 0 32
reply -stripe 8 0 32
reply -stripe 7 0 32
reply -stripe 6 0 32
reply -stripe 5 0 32
REM Number:
set REPLY_SIZE=7
set REPLY_OFFSET=%OFFSET_C%
reply -banner 20 %9
reply -banner 19 %8
reply -banner 18 %7
reply -banner 17 %6
reply -banner 16 %5
reply -banner 15 %4
reply -banner 14 %3
reply -banner 13 %2
goto exit
:fi_cl

REM Codepage: -----

if %1 == -cp_lo goto then_cp
if not %1 == -cp_up goto fi_cp
:then_cp
REM call g_code -resize:
set REPLY_OFFSET=3
set REPLY_SIZE=72
REM Clear desktop area for codepage pane:
for %i in (12 13 11 14 10 15) do reply -stripe %i 0 32
for %i in (9 16 8 17 7 18) do reply -stripe %i 0 32
for %i in (6 19 5 20 4 21) do reply -stripe %i 0 32
REM Code rows:
set PAGE=%1
REM Column 5:
set OFFSET_A=45
set OFFSET_B=44
set REPLY_OFFSET=%OFFSET_C%
reply -banner 5 %2
reply -banner 6 %3
reply -banner 7 %4
reply -banner 8 %5
reply -banner 9 %6
reply -banner 10 %7
reply -banner 11 %8
reply -banner 12 %9
goto exit
:fi_cu

REM Code column (lower half): -----

if not %1 == -code_lo goto fi_cl
REM Character:
set REPLY_SIZE=1
set REPLY_OFFSET=%OFFSET_A%
reply -stripe 20 %REPLY_HOTKEY% %9
reply -stripe 19 %REPLY_HOTKEY% %8
reply -stripe 18 %REPLY_HOTKEY% %7
reply -stripe 17 %REPLY_HOTKEY% %6
reply -stripe 16 %REPLY_HOTKEY% %5
reply -stripe 15 %REPLY_HOTKEY% %4
reply -stripe 14 %REPLY_HOTKEY% %3
reply -stripe 13 %REPLY_HOTKEY% %2
REM Gap:
set REPLY_SIZE=1
set REPLY_OFFSET=%OFFSET_B%
reply -stripe 13 0 32
reply -stripe 14 0 32
reply -stripe 15 0 32
reply -stripe 16 0 32
reply -stripe 17 0 32
reply -stripe 18 0 32
reply -stripe 19 0 32
reply -stripe 20 0 32
REM Number:
set REPLY_SIZE=7
set REPLY_OFFSET=%OFFSET_C%
reply -banner 20 %9
reply -banner 19 %8
reply -banner 18 %7
reply -banner 17 %6
reply -banner 16 %5
reply -banner 15 %4
reply -banner 14 %3
reply -banner 13 %2
goto exit
:fi_cl
```

```

set OFFSET_C=41
if not %PAGE% == -cp_lo goto else_cp1
  call l_code -code_up 64 65 66 67 68 69 70 71
  call l_code -code_lo 72 73 74 75 76 77 78 79
  goto fi_cp1
:else_cp1
  call l_code -code_up 192 193 194 195 196 197 198 199
  call l_code -code_lo 200 201 202 203 204 205 206 207
:fi_cp1
REM Column 4:
set OFFSET_A=36
set OFFSET_B=35
set OFFSET_C=32
if not %PAGE% == -cp_lo goto else_cp2
  call l_code -code_lo 56 57 58 59 60 61 62 63
  call l_code -code_up 48 49 50 51 52 53 54 55
  goto fi_cp2
:else_cp2
  call l_code -code_lo 184 185 186 187 188 189 190 191
  call l_code -code_up 176 177 178 179 180 181 182 183
:fi_cp2
REM Column 6:
set OFFSET_A=54
set OFFSET_B=53
set OFFSET_C=50
if not %PAGE% == -cp_lo goto else_cp3
  call l_code -code_up 80 81 82 83 84 85 86 87
  call l_code -code_lo 88 89 90 91 92 93 94 95
  goto fi_cp3
:else_cp3
  call l_code -code_up 208 209 210 211 212 213 214 215
  call l_code -code_lo 216 217 218 219 220 221 222 223
:fi_cp3
REM Column 3:
set OFFSET_A=27
set OFFSET_B=26
set OFFSET_C=23
if not %PAGE% == -cp_lo goto else_cp4
  call l_code -code_lo 40 41 42 43 44 45 46 47
  call l_code -code_up 32 33 34 35 36 37 38 39
  goto fi_cp4
:else_cp4
  call l_code -code_lo 168 169 170 171 172 173 174 175
  call l_code -code_up 160 161 162 163 164 165 166 167
:fi_cp4
REM Column 7:
set OFFSET_A=63
set OFFSET_B=62
set OFFSET_C=59
if not %PAGE% == -cp_lo goto else_cp5
  call l_code -code_up 96 97 98 99 100 101 102 103
  call l_code -code_lo 104 105 106 107 108 109 110 111
  goto fi_cp5
:else_cp5
  call l_code -code_up 224 225 226 227 228 229 230 231
  call l_code -code_lo 232 233 234 235 236 237 238 239
:fi_cp5
REM Column 2:
set OFFSET_A=18
set OFFSET_B=17
set OFFSET_C=14
if not %PAGE% == -cp_lo goto else_cp6
  call l_code -code_lo 24 25 26 27 28 29 30 31
  call l_code -code_up 16 17 18 19 20 21 22 23
  goto fi_cp6
:else_cp6
  call l_code -code_lo 152 153 154 155 156 157 158 159
  call l_code -code_up 144 145 146 147 148 149 150 151
:fi_cp6
REM Column 8:
set OFFSET_A=72
set OFFSET_B=71
set OFFSET_C=68
if not %PAGE% == -cp_lo goto else_cp7
  call l_code -code_up 112 113 114 115 116 117 118 119
  call l_code -code_lo 120 121 122 123 124 125 126 127
  goto fi_cp7
:else_cp7
  call l_code -code_up 240 241 242 243 244 245 246 247
  call l_code -code_lo 248 249 250 251 252 253 254 255
:fi_cp7
REM Column 1:
set OFFSET_A=9
set OFFSET_B=8
set OFFSET_C=5
if not %PAGE% == -cp_lo goto else_cp8
  call l_code -code_up 0 1 2 3 4 5 6 7
  call l_code -code_lo 8 9 10 11 12 13 14 15
  goto fi_cp8
:else_cp8
  call l_code -code_up 128 129 130 131 132 133 134 135
  call l_code -code_lo 136 137 138 139 140 141 142 143
:fi_cp8
REM Unset:
set PAGE=
goto exit
:fi_cp

REM Not found: .....

:else
  call l_banner -no_action l_code %1
:fi

:exit

```

I_color.bat

```
REM I_color.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_COLOR=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_color) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug:

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call I_banner -debug I_color %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call I_banner -no_action I_color
goto exit
:fi_nd

REM Update: -----

if not %1 == -update goto fi_u
REM Cold:
if not "%3" == "" goto fi_uc0
call I_banner -no_cold I_color
:fi_uc0
if not %3 == - goto fi_uc1
reply -item 6 " " " "
goto fi_uc
:fi_uc1
if not %3 == T goto fi_uc2
reply -item 7 "Text: 0...15" T 1
reply -item 8 %GMS_TEXT%
goto fi_uc
:fi_uc2
if not %3 == H goto fi_uc3
reply -item 9 "Hotkey: 0...15" H 2
reply -item 10 %GMS_HOTKEY%
goto fi_uc
:fi_uc3
if not %3 == P goto fi_uc4
reply -item 11 "Pattern: 0...15" P 3
reply -item 12 %GMS_PATTERN%
goto fi_uc
:fi_uc4
if not %3 == B goto fi_uc5 :then_uc5
reply -item 13 "Banner: 0...7" B 4
reply -item 14 %GMS_BANNER%
goto fi_uc
:fi_uc5
if not %3 == S goto fi_uc6
reply -item 15 "Shadow: 0...7" S 5
reply -item 16 %GMS_SHADE%
goto fi_uc
:fi_uc6
if not %3 == D goto fi_uc7
reply -item 17 "Desktop: 0...7" D 6
reply -item 18 %GMS_DESKTOP%
goto fi_uc
:fi_uc7
if not %3 == L goto fi_uc8
reply -item 19 "Letter: 32..126" L 7
reply -item 20 %GMS_LETTER%
goto fi_uc
:fi_uc8
if not %3 == Q goto else_uc9
reply -item 5 "Quit dialog []->" Q 8
goto fi_uc
:else_uc9
call I_banner -no_cold I_color %3
:fi_uc

REM Hot: .....
if not "%2" == "" goto fi_uh0
call I_banner -no_hot I_color
:fi_uh0
if not %2 == - goto fi_uh1
reply -item 6 " " " "
goto exit
:fi_uh1
if not %2 == T goto fi_uh2
reply -item 7 "Text: 0...15" TT 1
goto exit
:fi_uh2
if not %2 == H goto fi_uh3
reply -item 9 "Hotkey: 0...15" HH 2
goto exit
:fi_uh3
if not %2 == P goto fi_uh4
reply -item 11 "Pattern: 0...15" PP 3
goto exit
:fi_uh4
if not %2 == B goto fi_uh5
reply -item 13 "Banner: 0...7" BB 4
goto exit
:fi_uh5
if not %2 == S goto fi_uh6
reply -item 15 "Shadow: 0...7" SS 5
goto exit
:fi_uh6
if not %2 == D goto fi_uh7
reply -item 17 "Desktop: 0...7" DD 6
goto exit
:fi_uh7
if not %2 == L goto fi_uh8
reply -item 19 "Letter: 32..126" LL 7
goto exit
:fi_uh8
if not %2 == Q goto else_uh9
reply -item 5 "Quit dialog []->" QQ 8
goto exit
:else_uh9
call I_banner -no_hot I_color %2
goto exit
:fi_uh9
:fi_u
```

```

REM Build: -----
if not %1 == -build goto fi_b
  reply -top 3 "-- Select color --"
  reply -item 4 " " " "
  reply -item 5 "Quit dialog []->" Q 8
  reply -item 6 " " " "
  reply -item 7 "Text: 0...15" T 1
  reply -item 8 "%GMS_TEXT%"
  reply -item 9 "Hotkey: 0...15" H 2
  reply -item 10 "%GMS_HOTKEY%"
  reply -item 11 "Pattern: 0...15" P 3
  reply -item 12 "%GMS_PATTERN%"
  reply -item 13 "Banner: 0...7" B 4
  reply -item 14 "%GMS_BANNER%"
  reply -item 15 "Shadow: 0...7" S 5
  reply -item 16 "%GMS_SHADE%"

  reply -item 17 "Desktop: 0...7" D 6
  reply -item 18 "%GMS_DESKTOP%"
  reply -item 19 "Letter: 32..126" L 7
  reply -item 20 "%GMS_LETTER%"
  reply -bottom 21
  reply -shadow 22
  goto exit
:fi_b

REM Not found: -----

:else
  call l_banner -no_action l_color %1
:fi

:exit

```

I_desk.bat

```
REM l_desk.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_DESK=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (l_desk) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug l_desk %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action l_desk
goto exit
:fi_nd

REM Build: -----

if not %1 == -build goto fi_b
if not %GMS_ANIMATE% == on goto fi_b2
for %i in (2 23 3 22 4 21 5 20) do reply -stripe %i %REPLY_PATTERN% 32
for %i in (6 19 7 18 8 17 9 16) do reply -stripe %i %REPLY_PATTERN% 32
for %i in (10 15 11 14 12 13) do reply -stripe %i %REPLY_PATTERN% 32
:fi_b2
for %i in (2 23 3 22) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (4 21 5 20) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (6 19 7 18) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (8 17 9 16) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (10 15 11) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (14 12 13) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
goto exit
:fi_b

REM Remove: -----

if not %1 == -remove goto fi_r
set REPLY_OFFSET=0
set REPLY_SIZE=78
set REPLY_ITEMS=0
if not %GMS_ANIMATE% == on goto fi_ra
for %i in (12 13 11 14 10 15) do reply -stripe %i %REPLY_PATTERN% 32
for %i in (9 16 8 17 7 18 6 19) do reply -stripe %i %REPLY_PATTERN% 32
for %i in (5 20 4 21 3 22 2 23) do reply -stripe %i %REPLY_PATTERN% 32
:fi_ra
set REPLY_DESKTOP=0
for %i in (12 13 11 14 10 15 9 16 8 17 7) do reply -stripe %i 7 47
for %i in (18 6 19 5 20 4 21 3 22 2 23) do reply -stripe %i 7 47
set REPLY_SIZE=82
set REPLY_BANNER=0
set REPLY_TEXT=7
if "%2" == ".nobanners" goto exit
reply -banner 1 "%GMS_FILE%"
REM There must be one space before the second quoting mark for 'C:\' etc.:
reply -banner 24 "%GMS_FOLDER%"
goto exit
:fi_r

REM Not found: -----

:else
call l_banner -no_action l_desk %1
:fi

:exit
```

I_file.bat

```
REM I_file.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateI_file=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_file) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call I_banner -debug I_file %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call I_banner -no_action I_file
goto exit
:fi_nd

REM Update: -----

if not %1 == -update goto fi_u

REM Cold state:
if not "%3" == "" goto fi_uc0
call I_banner -no_cold I_file
goto fi_uc
:fi_uc0
if not %3 == 1 goto fi_uc1
reply -item 11 "1 %GMS1% %REPLY1%" 1 1
goto fi_uc
:fi_uc1
if not %3 == 2 goto fi_uc2
reply -item 12 "2 %GMS2% %REPLY2%" 2 2
goto fi_uc
:fi_uc2
if not %3 == 3 goto fi_uc3
reply -item 13 "3 %GMS3% %REPLY3%" 3 3
goto fi_uc
:fi_uc3
if not %3 == 4 goto fi_uc4
reply -item 14 "4 %GMS4% %REPLY4%" 4 4
goto fi_uc
:fi_uc4
if not %3 == 5 goto fi_uc5
reply -item 15 "5 %GMS5% %REPLY5%" 5 5
goto fi_uc
:fi_uc5

if not %3 == 6 goto fi_uc6
reply -item 16 "6 %GMS6% %REPLY6%" 6 6
goto fi_uc
:fi_uc6
if not %3 == 7 goto fi_uc7
reply -item 17 "7 %GMS7% %REPLY7%" 7 7
goto fi_uc
:fi_uc7
if not %3 == 8 goto fi_uc8
reply -item 18 "8 %GMS8% %REPLY8%" 8 8
goto fi_uc
:fi_uc8
if not %3 == 9 goto fi_uc9
reply -item 19 "9 %GMS9% %REPLY9%" 9 9
goto fi_uc
:fi_uc9
if not %3 == - goto fi_ucd
reply -item 4 " " " "
goto fi_uc
:fi_ucd
if not %3 == Q goto fi_ucq
reply -item 5 "Q Quit dialog" Q 10
goto fi_uc
:fi_ucq
if not %3 == P goto fi_ucp
reply -item 7 "P <- Previous files" P 11
goto fi_uc
:fi_ucp
if not %3 == N goto fi_ucn
reply -item 8 "N -> Next files" N 12
goto fi_uc
:fi_ucn
if not %3 == H goto else_uc
reply -item 9 "H .. Higher level" H 13
goto fi_uc
:else_uc
call I_banner -no_cold I_file %3
:fi_uc

REM Hot state:
if not "%2" == "" goto fi_uh0
call I_banner -no_hot I_file
goto exit
:fi_uh0
if not %2 == 1 goto fi_uh1
reply -item 11 "1 %GMS1% %REPLY1%" 11 1
goto exit
:fi_uh1
if not %2 == 2 goto fi_uh2
reply -item 12 "2 %GMS2% %REPLY2%" 22 2
goto exit
:fi_uh2
if not %2 == 3 goto fi_uh3
reply -item 13 "3 %GMS3% %REPLY3%" 33 3
goto exit
:fi_uh3
if not %2 == 4 goto fi_uh4
reply -item 14 "4 %GMS4% %REPLY4%" 44 4
goto exit
:fi_uh4
if not %2 == 5 goto fi_uh5
reply -item 15 "5 %GMS5% %REPLY5%" 55 5
goto exit
:fi_uh5
if not %2 == 6 goto fi_uh6
reply -item 16 "6 %GMS6% %REPLY6%" 66 6
goto exit
:fi_uh6
```



```

if not %2 == 7 goto fi_uh7
  reply -item 17 "7 %GMS7% %REPLY7%" 77 7
  goto exit
:fi_uh7
if not %2 == 8 goto fi_uh8
  reply -item 18 "8 %GMS8% %REPLY8%" 88 8
  goto exit
:fi_uh8
if not %2 == 9 goto fi_uh9
  reply -item 19 "9 %GMS9% %REPLY9%" 99 9
  goto exit
:fi_uh9
if not %2 == Q goto fi_uhq
  reply -item 5 "Q Quit dialog" QQ 10
  goto exit
:fi_uhq
if not %2 == P goto fi_uhp
  reply -item 7 "P Previous files" PP 11
  goto exit
:fi_uhp
if not %2 == N goto fi_uhn
  reply -item 8 "N -> Next files" NN 12
  goto exit
:fi_uhn
if not %2 == H goto else_uh
  reply -item 9 "H .. Higher level" HH 13
  goto exit
:else_uh
  call l_banner -no_hot l_file %2
  goto exit

```

```

:fi_uh
  goto exit
:fi_u

REM Build: -----

if not %1 == -build goto fi_b
  reply -top 3 "%GMS_FILEBOX%"
  reply -item 4 " " " "
  reply -item 5 "Q Quit dialog" Q 10
  reply -item 6 " " " "
  reply -item 7 "P <- Previous files" P 11
  reply -item 8 "N -> Next files" N 12
  reply -item 9 "H .. Higher level" H 13
  for %i in (10 11 12 13 14 15) do reply -item %i " " " "
  for %i in (16 17 18 19 20) do reply -item %i " " " "
  reply -bottom 21
  reply -shadow 22
  goto exit
:fi_b

REM Not found: -----

:else
  call l_banner -no_action l_file %1
:fi

:exit

```

I_gms.bat

```
REM I_gms.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_GMS=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_gms) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Chapters: =====

REM Desktop:

if not %1 == -desktop goto fi_dt
cls
echo. %GMS_FILE%
if "%2" == "-draw" goto fi_x00
for %i in (1 2 3 4 5 6 7 8 9 10 11) do echo.
for %i in (1 2 3 4 5 6 7 8 9 10 11) do echo.
:fi_x00
if not "%3" == "-nofolder" if not "%GMS_FOLDER%" == "" echo %GMS_FOLDER%
set REPLY_BANNER=0
set REPLY_TEXT=7
set REPLY_OFFSET=0
set REPLY_SIZE=82
if "%GMS_PROGRAM%" == "" goto else_x2
if "%OS%" == "" goto fi_x2
reply -banner 1 "%GMS_FILE% - GMS: Running %GMS_PROGRAM% ..."
goto fi_x2
:else_x2
reply -banner 1 "%GMS_FILE%"
:fi_x2
set REPLY_DESKTOP=0
set REPLY_PATTERN=7
if not "%2" == "-clear" set REPLY_OFFSET=1
set REPLY_SIZE=78
if not "%2" == "-clear" set REPLY_SIZE=76
reply -stripe 23 7 47
set REPLY_OFFSET=0
set REPLY_SIZE=82
REM There must be one space before the second quote mark for 'C:\' etc.:
if not "%3" == "-nofolder" reply -banner 24 "%GMS_FOLDER% "
if not "%2" == "-clear" set REPLY_OFFSET=1
set REPLY_SIZE=78
if not "%2" == "-clear" set REPLY_SIZE=76
reply -stripe 2 7 47
goto exit
:fi_dt

REM Welcome: -----

if not %1 == -welcome goto fi_w

if "%GMS_TEXTMODE%" == "" goto then_ww
if %GMS_TEXTMODE% == 1 goto fi_ww
:then_ww
REM Show version number and date:
set REPLY_BANNER=0
set REPLY_DESKTOP=0
set REPLY_SHADE=0
set REPLY_TEXT=7
set REPLY_HOTKEY=7
set REPLY_OFFSET=11
set REPLY_SIZE=17
reply -item 12 "Version %GMS_VERSION%" X X
set REPLY_OFFSET=8
set REPLY_SIZE=20
reply -item 16 %GMS_DATE% X X
REM Set prompt on line 24:
set REPLY_OFFSET=2
set REPLY_SIZE=26
reply -bottom 23
:fi_ww
goto exit
:fi_w

REM Write %5 %6 %7 ... %11 %12 %13 on line %2 with offset %3 and size %4: ----

if not %1 == -write_line goto fi_gl
set REPLY_BANNER=0
set REPLY_TEXT=7
set REPLY_OFFSET=%3
set REPLY_SIZE=%4
set str=%2
for %i1 in (1, 2, 3, 4) do shift
reply -banner %str% "%1 %2 %3 %4 %5 %6 %7 %8 %9"
set str=
goto exit
:fi_gl

REM Not found: -----

:else
set GMS_MESSAGE= GMS error (I_gms): No action %1
reply -sleep
call I_gms -help
:fi

REM Epilogue: =====

:exit

REM Debug:

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
set REPLY_BANNER=3
set REPLY_TEXT=15
call I_banner -debug I_gms %1 %2 %3 %4
:fi_db
```

I_good.bat

```
REM I_good.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2008).

REM set GMSdateL_GOOD=20080107

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_good) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug I_good %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action I_good
goto exit
:fi_nd

REM Update: -----

if not %1 == -update goto fi_u
REM Cold:
reply -item 6
REM Hot:
if not "%2" == "" goto fi_u1
call l_banner -no_hot I_good
goto exit

:fi_u1
if not %2 == - goto fi_u2
reply -item 5 " Gerolf Markup Shredder" Q
goto exit
:fi_u2
if not %2 == Q goto else_u3
reply -item 5 " Gerolf Markup Shredder" QQ
goto exit
:else_u3
call l_banner -no_hot I_good %2
:fi_u3
goto exit
:fi_u

REM Build: -----

if not %1 == -build goto fi_b
reply -top 3 "----- About: GMS %GMS_VERSION% -----"
reply -item 4 " "
reply -item 5 " Gerolf Markup Shredder" Q
reply -item 6 " "
reply -item 7 " Copyright (c) 1999-2008 by G. D. Brett-"
reply -item 8 " schneider, Luchtbergstr. 27, D-28237"
reply -item 9 " Bremen. All rights reserved. This GMS"
reply -item 10 " software comes without ANY warranty."
reply -item 11 " You may freely distribute and use it."
reply -item 12 " "
reply -item 13 " "
reply -item 14 " "
reply -item 15 " "
reply -item 16 " "
reply -item 17 " "
reply -item 18 " "
reply -item 19 " "
reply -item 20 " "
reply -bottom 21
reply -shadow 22
goto exit
:fi_b

REM Not found: -----

:else
call l_banner -no_action I_good %1
:fi

:exit
```

I_list.bat

```
REM I_list.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_LIST=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_list) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call I_banner -debug I_list %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call I_banner -no_action I_list
goto exit
:fi_nd

REM Build: -----

if not %1 == -build goto fi_b
reply -item 11 "1 %GMS1% %REPLY1%" 1 1
reply -item 12 "2 %GMS2% %REPLY2%" 2 2
reply -item 13 "3 %GMS3% %REPLY3%" 3 3
reply -item 14 "4 %GMS4% %REPLY4%" 4 4
reply -item 15 "5 %GMS5% %REPLY5%" 5 5
reply -item 16 "6 %GMS6% %REPLY6%" 6 6
reply -item 17 "7 %GMS7% %REPLY7%" 7 7
reply -item 18 "8 %GMS8% %REPLY8%" 8 8
reply -item 19 "9 %GMS9% %REPLY9%" 9 9
goto exit
:fi_b

REM Not found: -----

:else
call I_banner -no_action I_list %1
:fi

:exit
```

I_menu.bat

```
REM l_menu.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2007).

REM set GMSdateL_MENU=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (l_menu) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug l_menu %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action l_menu
goto exit
:fi_nd

REM Update: -----

if not %1 == -update goto fi_u
REM Cold:
if not "%3" == "" goto fi_uc0
call l_banner -no_cold l_menu
goto fi_uc
:fi_uc0
if not %3 == Q goto fi_ucq
reply -item 5 "Quit GMS program" Q 1
goto fi_uc
:fi_ucq
if not %3 == C goto fi_ucc
reply -item 7 "Create markup file" C 2
goto fi_uc
:fi_ucc
if not %3 == O goto fi_uco
reply -item 8 "Open folder/file" O 3
goto fi_uc
:fi_uco
if not %3 == V goto fi_ucv
reply -item 9 "View markup file" V 4
goto fi_uc
:fi_ucv
if not %3 == E goto fi_uce
reply -item 10 "Edit markup file" E 5
goto fi_uc
:fi_uce
if not %3 == B goto fi_uchb
reply -item 11 "Browse markup file" B 6
goto fi_uc
:fi_uchb
if not %3 == A goto fi_uha
reply -item 12 "Analyse folder/file" AA 7
goto fi_uc
:fi_uha
if not %3 == T goto fi_uct
reply -item 13 "Typeset markup file" T 8
goto fi_uc
:fi_uct
if not %3 == R goto fi_ucr
reply -item 14 "Read output file" R 9
goto fi_uc
:fi_ucr
if not %3 == L goto fi_ucl
reply -item 16 "Learn GMS tricks" L 10
goto fi_uc
:fi_ucl
if not %3 == S goto fi_ucs
reply -item 17 "Select GMS setting" S 11
goto fi_uc
:fi_ucs
if not %3 == W goto fi_ucw
reply -item 18 "Write GMS fontmap" W 12
goto fi_uc
:fi_ucw
if not %3 == I goto fi_ucI
reply -item 19 "Init GMS format" I 13
goto fi_uc
:fi_ucI
else uc
call l_banner -no_cold l_menu %3
:fi_uc
REM Hot: -----
if not "%2" == "" goto fi_uh0
call l_banner -no_hot l_menu
goto exit
:fi_uh0
if not %2 == Q goto fi_uhq
reply -item 5 "Quit GMS program" QQ 1
goto exit
:fi_uhq
if not %2 == - goto fi_uhm
reply -item 6 " " " " " "
goto exit
:fi_uhm
if not %2 == C goto fi_uhc
reply -item 7 "Create markup file" CC 2
goto exit
:fi_uhc
if not %2 == O goto fi_uho
reply -item 8 "Open folder/file" OO 3
goto exit
:fi_uho
if not %2 == V goto fi_uhv
reply -item 9 "View markup file" VV 4
goto exit
:fi_uhv
if not %2 == E goto fi_uhe
reply -item 10 "Edit markup file" EE 5
goto exit
:fi_uhe
if not %2 == B goto fi_uhb
reply -item 11 "Browse markup file" BB 6
goto exit
:fi_uhb
if not %2 == A goto fi_uha
reply -item 12 "Analyse folder/file" AA 7
goto fi_uc
:fi_uha
```

```

goto exit
:fi_uha
if not %2 == T goto fi_uht
    reply -item 13 "Typeset markup file" TT 8
    goto exit
:fi_uht
if not %2 == R goto fi_uhr
    reply -item 14 "Read output file" RR 9
    goto exit
:fi_uhr
if not %2 == L goto fi_uhl
    reply -item 16 "Learn G.M.S. tricks" LL 10
    goto exit
:fi_uhl
if not %2 == S goto fi_uhs
    reply -item 17 "Select GMS setting" SS 11
    goto exit
:fi_uhs
if not %2 == W goto fi_uhw
    reply -item 18 "Write GMS fontmap" WW 12
    goto exit
:fi_uhw
if not %2 == I goto else_uh
    reply -item 19 "Init GMS format" II 13
    goto exit
:else_uh
    call l_banner -no_hot l_menu %2
:fi_uh
goto exit
:fi_u
REM Build: -----

```

```

if not %1 == -build goto fi_b
reply -top 3 "----- GMS -----"
reply -item 4 " " " "
reply -item 5 "Quit GMS program" Q 1
reply -item 6 " " " "
reply -item 7 "Create markup file" C 2
reply -item 8 "Open folder/file" O 3
reply -item 9 "View markup file" V 4
reply -item 10 "Edit markup file" E 5
reply -item 11 "Browse markup file" B 6
reply -item 12 "Analyse folder/file" A 7
reply -item 13 "Typeset markup file" T 8
reply -item 14 "Read output file" R 9
reply -item 15 " " " "
reply -item 16 "Learn GMS tricks" L 10
reply -item 17 "Select GMS setting" S 11
reply -item 18 "Write GMS fontmap" W 12
reply -item 19 "Init GMS format" I 13
reply -item 20 " " " "
reply -bottom 21
reply -shadow 22
goto exit
:fi_b
REM Not found: -----
:else
    call l_banner -no_action l_menu %1
:fi
:exit

```

I_prog.bat

```
REM I_prog.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_PROG=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_prog) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call I_banner -debug I_prog %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call I_banner -no_action I_prog
goto exit
:fi_nd

REM Update: -----

if not %1 == -update goto fi_u
REM Cold state:
if not "%3" == "" goto fi_uc0
call I_banner -no_cold I_prog
goto fi_uc
:fi_uc0
if not %3 == - goto fi_ucm
reply -item 6 " " " "
goto fi_uc
:fi_ucm
if not %3 == V goto fi_ucv
reply -item 7 "Viewer: any text" V 1
reply -item 8 "%GMS_VIEWER%"
goto fi_uc
:fi_ucv
if not %3 == E goto fi_uce
reply -item 9 "Editor: any text" E 2
reply -item 10 "%GMS_EDITOR%"
goto fi_uc
:fi_uce
if not %3 == B goto fi_uch
reply -item 11 "Browser: HTML" B 3
reply -item 12 "%GMS_BROWSER%"
goto fi_uc
:fi_uch
if not %3 == A goto fi_uca
reply -item 13 "Analyst: HTML" A 4
reply -item 14 "%GMS_ANALYST%"
goto fi_uc
:fi_uca
if not %3 == T goto fi_uct
reply -item 15 "Typesetter: HTML" T 5
reply -item 16 "%GMS_TSETTER%"
goto fi_uc
:fi_uct
if not %3 == R goto fi_ucr
reply -item 17 "Reader: PDF" R 6
reply -item 18 "%GMS_READER%"
goto fi_uc
:fi_ucr
if not %3 == Q goto else_uc
reply -item 5 "Quit dialog []->" Q 7
goto fi_uc
:else_uc
call I_banner -no_cold I_prog %3
:fi_uc

REM Hot state:
if not "%2" == "" goto fi_uh0
call I_banner -no_hot I_prog
goto fi_uh
:fi_uh0
if not %2 == - goto fi_uhm
reply -item 6 " " " "
goto fi_uh
:fi_uhm
if not %2 == V goto fi_uhv
reply -item 7 "Viewer: any text" VV 1
reply -item 8 "%GMS_VIEWER%"
goto fi_uh
:fi_uhv
if not %2 == E goto fi_uhe
reply -item 9 "Editor: any text" EE 2
reply -item 10 "%GMS_EDITOR%"
goto fi_uh
:fi_uhe
if not %2 == B goto fi_uhb
reply -item 11 "Browser: HTML" BB 3
reply -item 12 "%GMS_BROWSER%"
goto fi_uh
:fi_uhb
if not %2 == A goto fi_uha
reply -item 13 "Analyst: HTML" AA 4
reply -item 14 "%GMS_ANALYST%"
goto fi_uh
:fi_uha
if not %2 == T goto fi_uht
reply -item 15 "Typesetter: HTML" TT 5
reply -item 16 "%GMS_TSETTER%"
goto fi_uh
:fi_uht
if not %2 == R goto fi_uhr
reply -item 17 "Reader: PDF" RR 6
reply -item 18 "%GMS_READER%"
goto fi_uh
:fi_uhr
if not %2 == Q goto else_uh
reply -item 5 "Quit dialog []->" QQ 7
goto fi_uh
:else_uh
call I_banner -no_hot I_prog %2
:fi_uh
goto exit
:fi_u

REM Build: -----
```

```

if not %1 == -build goto fi_b
reply -top 3 "--- Programs ---"
reply -item 4 " "
reply -item 5 "Quit dialog []->" Q 7
reply -item 6 " "
reply -item 7 "Viewer: any text" V 1
reply -item 8 "%GMS_VIEWER%"
reply -item 9 "Editor: any text" E 2
reply -item 10 "%GMS_EDITOR%"
reply -item 11 "Browser: HTML" B 3
reply -item 12 "%GMS_BROWSER%"
reply -item 13 "Analyst: HTML" A 4
reply -item 14 "%GMS_ANALYST%"
reply -item 15 "Typesetter: HTML" T 5
reply -item 16 "%GMS_TSETTER%"

reply -item 17 "Reader: PDF" R 6
reply -item 18 "%GMS_READER%"
reply -bottom 19
reply -shadow 20
goto exit
:fi_b

REM Not found: .....

:else
call l_banner -no_action l_prog %1
:fi

:exit

```


I_rain.bat

```
REM l_rain.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_RAIN=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (l_rain) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug l_rain %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call l_banner -no_action l_rain
goto exit
:fi_nd

REM Rainbow column: -----

if not %1 == -raincol goto fi_rc
set REPLY_SIZE=1
set REPLY_DESKTOP=%2
REM Get offset and letter number:
set REPLY_OFFSET=%2
if %2 == 0 set REPLY_OFFSET=6
if %2 == 0 set LETTER=48
if %2 == 1 set REPLY_OFFSET=7
if %2 == 1 set LETTER=49
if %2 == 2 set REPLY_OFFSET=8
if %2 == 2 set LETTER=50
if %2 == 3 set REPLY_OFFSET=9
if %2 == 3 set LETTER=51
if %2 == 4 set REPLY_OFFSET=10
if %2 == 4 set LETTER=52
if %2 == 5 set REPLY_OFFSET=11
if %2 == 5 set LETTER=53
if %2 == 6 set REPLY_OFFSET=12
if %2 == 6 set LETTER=54
if %2 == 7 set REPLY_OFFSET=13
if %2 == 7 set LETTER=55
REM Draw column:
reply -stripe 5 0 %LETTER%
reply -stripe 6 1 %LETTER%
reply -stripe 7 2 %LETTER%
reply -stripe 8 3 %LETTER%
reply -stripe 9 4 %LETTER%
reply -stripe 10 5 %LETTER%
reply -stripe 11 6 %LETTER%
reply -stripe 12 7 %LETTER%
reply -stripe 13 8 %LETTER%
reply -stripe 14 9 %LETTER%
reply -stripe 15 10 %LETTER%
reply -stripe 16 11 %LETTER%
reply -stripe 17 12 %LETTER%
reply -stripe 18 13 %LETTER%
reply -stripe 19 14 %LETTER%
reply -stripe 20 15 %LETTER%
set LETTER=
goto exit
:fi_rc

REM Rainbow: -----

if not %1 == -build goto fi_b
REM Save colors:
set BACKUP1=%REPLY_DESKTOP%
set BACKUP2=%REPLY_PATTERN%
set BACKUP3=%REPLY_BANNER%
set BACKUP4=%REPLY_TEXT%
REM Clear desktop area for pane:
for %i in (12 13 11 14 10 15) do reply -stripe %i 0 32
for %i in ( 9 16 8 17 7 18) do reply -stripe %i 0 32
for %i in ( 6 19 5 20 4 21) do reply -stripe %i 0 32
for %i in ( 3 22 ) do reply -stripe %i 0 32
REM Draw heading:
set REPLY_BANNER=%BACKUP1%
set REPLY_TEXT=%REPLY_HOTKEY%
reply -banner 3 " Back- & Fore- "
reply -banner 22 " -ground colors "
REM Draw columns with background color numbers:
for %i in ( 0 1 2 3) do call l_rain -raincol %i
for %i in ( 4 5 6 7) do call l_rain -raincol %i
REM Clear overwritten area:
set REPLY_OFFSET=14
set REPLY_SIZE=5
set REPLY_DESKTOP=%BACKUP1%
set REPLY_PATTERN=%BACKUP2%
for %i in ( 5 6 7 8 9 10) do reply -stripe %i 0 32
for %i in (11 12 13 14 15 16) do reply -stripe %i 0 32
for %i in (17 18 19 20 ) do reply -stripe %i 0 32
REM Draw rows with foreground color numbers:
set REPLY_BANNER=%REPLY_DESKTOP%
set REPLY_SIZE=3
REM One digit:
set REPLY_OFFSET=18
set REPLY_TEXT=0
reply -banner 5 0
set REPLY_TEXT=1
reply -banner 6 1
set REPLY_TEXT=2
reply -banner 7 2
set REPLY_TEXT=3
reply -banner 8 3
set REPLY_TEXT=4
reply -banner 9 4
set REPLY_TEXT=5
reply -banner 10 5
set REPLY_TEXT=6
reply -banner 11 6
set REPLY_TEXT=7
reply -banner 12 7
set REPLY_TEXT=8
reply -banner 13 8
set REPLY_TEXT=9
```

```

reply ·banner 14 9
REM Two digits:
set REPLY_OFFSET=17
set REPLY_TEXT=10
reply ·banner 15 10
set REPLY_TEXT=11
reply ·banner 16 11
set REPLY_TEXT=12
reply ·banner 17 12
set REPLY_TEXT=13
reply ·banner 18 13
set REPLY_TEXT=14
reply ·banner 19 14
set REPLY_TEXT=15
reply ·banner 20 15
REM Reset colors:
set REPLY_BANNER=%BACKUP3%
set REPLY_TEXT=%BACKUP4%
set BACKUP1=
set BACKUP2=
set BACKUP3=
set BACKUP4=
goto exit
:fi_b

```

```

REM Rainbow (and codepage) remove: -----

```

```

if not %1 == -remove goto fi_r
REM Clear pane:
if not %GMS_ANIMATE% == on goto fi_rr
for %i in (4 21 5 20) do reply -stripe %i 0 32
for %i in (6 19 7 18 8 17) do reply -stripe %i 0 32
for %i in (9 16 10 15 11 14) do reply -stripe %i 0 32
for %i in (12 13) do reply -stripe %i 0 32
:fi_rr
REM Remove pane:
for %i in (3 22 4) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (21 5 20) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (6 19 7) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (18 8 17) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (9 16 10) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (15 11 14) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
for %i in (12 13) do reply -stripe %i %REPLY_PATTERN% %REPLY_LETTER%
goto exit
:fi_r
REM Not found: -----
:else
call l_banner -no_action l_rain %1
:fi
:exit

```

I_save.bat

```
REM I_save.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_SAVE=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_save) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call I_banner -debug I_save %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call I_banner -no_action I_save
goto exit
:fi_nd

REM Update: -----

if not %1 == -update goto fi_u
if not "%4" == "1" goto else_ue

:then_ue
REM There exists an old file that can be overwritten:
reply -item 8 "Overwrite existing file" 0 3
REM Cold state:
if not "%3" == "" goto fi_uec0
call I_banner -no_cold I_save
goto fi_uec
:fi_uec0
if not %3 == Q goto fi_uecg
reply -item 5 "Quit dialog" Q 1
goto fi_uec
:fi_uecg
if not %3 == C goto fi_uecc
reply -item 7 "Cancel file creation" C 2
goto fi_uec
:fi_uecc
if not %3 == 0 goto fi_ueco
reply -item 8 "Overwrite existing file" 0 3
goto fi_uec
:fi_ueco
if not %3 == N goto else_uec
reply -item 10 "%GMS_FILE%" N 4
goto fi_uec
:else_uec

call I_banner -no_cold I_save %3
:fi_uec

REM Hot state:
if not "%2" == "" goto fi_ueh0
call I_banner -no_hot I_save
goto fi_ueh
:fi_ueh0
if not %2 == Q goto fi_uehg
reply -item 5 "Quit dialog" QQ 1
goto fi_ueh
:fi_uehg
if not %2 == C goto fi_uehc
reply -item 7 "Cancel file creation" CC 2
goto fi_ueh
:fi_uehc
if not %2 == 0 goto fi_ueho
reply -item 8 "Overwrite existing file" OO 3
goto fi_ueh
:fi_ueho
if not %2 == N goto else_ueh
reply -question 10 "%GMS_FILE%" NN 4
goto fi_ueh
:else_ueh
call I_banner -no_hot I_save %2
:fi_ueh
goto exit

:else_ue
REM There does not exist an old file that can be overwritten:
REM Cold state:
if not "%3" == "" goto fi_unc0
call I_banner -no_cold I_save
goto fi_unc
:fi_unc0
if not %3 == Q goto fi_uncq
reply -item 5 "Quit dialog" Q 1
goto fi_unc
:fi_uncq
if not %3 == C goto fi_uncc
reply -item 7 "Cancel file creation" C 2
goto fi_unc
:fi_uncc
if not %3 == 0 goto fi_unc0
reply -item 8 " " " " 3
goto fi_unc
:fi_unc0
if not %3 == N goto else_unc
reply -item 10 "%GMS_FILE%" N 4
goto fi_unc
:else_unc
call I_banner -no_cold I_save %3
:fi_unc

REM Hot state:
if not "%2" == "" goto fi_unh0
call I_banner -no_hot I_save
goto fi_unh
:fi_unh0
if not %2 == Q goto fi_unhg
reply -item 5 "Quit dialog" QQ 1
goto fi_unh
:fi_unhg
if not %2 == C goto fi_unhc
reply -item 7 "Cancel file creation" CC 2
goto fi_unh
:fi_unhc
if not %2 == 0 goto fi_unho
reply -item 8 " " " 00 3
goto fi_unh
```

```

:fi_unho
if not %2 == N goto else_unh
  reply -question 10 "%GMS_FILE%" NN 4
  goto fi_unh
:else_unh
  call l_banner -no_hot l_save %2
:fi_unh
:fi_ue
goto exit
:fi_u

```

REM Build: -----

```

if not %1 == -build goto fi_b
reply -top 3 " 3. Enter a better name for this new file "
reply -item 4 " " " "
reply -item 5 "Quit dialog " Q 1
reply -item 6 " " " "
reply -item 7 "Cancel file creation" C 2

```

```

reply -item 8 "0 " 0 3
reply -item 8 " " " "
reply -item 9 " " " "
reply -item 10 "%GMS_FILE%" N 4
reply -item 11 " " " "
reply -bottom 12
reply -shadow 13
goto exit
:fi_b

```

REM Not found: -----

```

:else
  call l_banner -no_action l_save %1
:fi

:exit

```

I_select.bat

```
REM I_select.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_SELECT=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_select) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call I_banner -debug I_select %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call I_banner -no_action I_select
goto exit
:fi_nd

REM Update: -----

if not %1 == -update goto fi_u
REM Cold:
if not "%3" == "" goto fi_uc0
call I_banner -no_cold I_select
goto fi_uc
:fi_uc0
if not %3 == Q goto fi_ucq
reply -item 5 "Quit dialog >" Q 1
goto fi_uc
:fi_ucq
if not %3 == A goto fi_uca
reply -item 7 "Animation %4" A 2
goto fi_uc
:fi_uca
if not %3 == C goto fi_ucc
reply -item 8 "Colors >" C 3
goto fi_uc
:fi_ucc
if not %3 == P goto fi_ucp
reply -item 9 "Programs >" P 4
goto fi_uc
:fi_ucp
if not %3 == D goto fi_ucd
reply -item 10 "Debugging:" D 5
if not %GMS_DEBUG% == X goto fi_ucdx
reply -item 11 "%GMS_DEBUG%" X
goto fi_ucdz
:fi_ucdx
if not %GMS_DEBUG% == Y goto fi_ucdy
reply -item 11 "%GMS_DEBUG%" Y
goto fi_ucdz
:fi_ucdy
if not %GMS_DEBUG% == Z goto elseucdz
reply -item 11 "%GMS_DEBUG%" Z
goto fi_ucdz
:elseucdz
reply -item 11 "%GMS_DEBUG%" 0
:fi_ucdz
goto fi_uc
:fi_ucd
if not %3 == E goto fi_uce
reply -item 12 "Encoding:" E 6
reply -item 13 "%GMS_CODEPAGE%" I
goto fi_uc
:fi_uce
if not %3 == I goto fi_uci
reply -item 13 "%GMS_CODEPAGE%" I
goto fi_uc
:fi_uci
if not %3 == U goto fi_uu
reply -item 14 "Upper half >" U 7
goto fi_uc
:fi_uu
if not %3 == L goto else_uc
reply -item 15 "Lower half >" L 8
goto fi_uc
:else_uc
call I_banner -no_cold I_select %3
:fi_uc

REM Hot: -----
if not "%2" == "" goto fi_uh0
call I_banner -no_hot I_select
goto fi_uh
:fi_uh0
if not %2 == - goto fi_uhm
reply -item 4 " " " "
goto fi_uh
:fi_uhm
if not %2 == Q goto fi_uhq
reply -item 5 "Quit dialog >" QQ 1
goto fi_uh
:fi_uhq
if not %2 == A goto fi_uha
reply -item 7 "Animation %4" AA 2
goto fi_uh
:fi_uha
if not %2 == C goto fi_uhc
reply -item 8 "Colors >" CC 3
goto fi_uh
:fi_uhc
if not %2 == P goto fi_uhp
reply -item 9 "Programs >" PP 4
goto fi_uh
:fi_uhp
if not %2 == D goto fi_uhd
reply -item 10 "Debugging:" DD 5
goto fi_uh
:fi_uhd
if not %2 == E goto fi_uhe
reply -item 12 "Encoding:" EE 6
goto fi_uh
:fi_uhe
if not %2 == U goto fi_uhu
reply -item 14 "Upper half >" UU 7
goto fi_uh
```

```

:fi_uhu
if not %2 == L goto else_uh
  reply -item 15 "Lower half >" LL 8
  goto fi_uh
:else_uh
  call l_banner -no_hot l_select %2
:fi_uh
goto exit
:fi_u

```

REM Build:

```

if not %1 == -build goto fi_b
reply -top 3 " Select/Show "
reply -item 4 " " " "
reply -item 5 "Quit dialog >" Q 1
reply -item 6 " " " "
reply -item 7 "Animation %2" A 2
reply -item 8 "Colors >" C 3
reply -item 9 "Programs >" P 4
reply -item 10 "Debugging: " D 5
if not %GMS_DEBUG% == X goto fi_bbx
  reply -item 11 "%GMS_DEBUG%" X
  goto fi_bb

```

```

:fi_bbx
if not %GMS_DEBUG% == Y goto else_bb
  reply -item 11 "%GMS_DEBUG%" Y
  goto fi_bb
:else_bb
  reply -item 11 "%GMS_DEBUG%" 0
:fi_bb
reply -item 12 "Encoding: " E 6
reply -item 13 "%GMS_CODEPAGE%" I
reply -item 14 "Upper half >" U 7
reply -item 15 "Lower half >" L 8
reply -item 16
reply -bottom 17
reply -shadow 18
goto exit
:fi_b

```

REM Not found:

```

:else
  call l_banner -no_action l_select %1
:fi

:exit

```

I_wel.bat

```
REM I_wel.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateL_WEL=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (I_wel) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
REM if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call I_banner -debug I_wel %1 %2 %3 %4
:fi_db

REM Chapters: =====

REM Not defined:

if not "%1" == "" goto fi_nd
call I_banner -no_action I_wel
goto exit
:fi_nd

REM Build: -----

if not %1 == -build goto fi_b
REM Cycle 1:
set REPLY_DESKTOP=0
reply -shadow 1
set REPLY_DESKTOP=%backup_desktop%
set REPLY_SIZE=26
reply -shadow 2
set REPLY_DESKTOP=0
reply -shadow 24
set REPLY_DESKTOP=%backup_desktop%
REM Cycle 2:
set REPLY_DESKTOP=0
reply -top 1
set REPLY_DESKTOP=%backup_desktop%
reply -shadow 2
reply -bottom 23
REM Cycle 3:
reply -shadow 3
reply -item 2 "<Welcome>"
reply -item 22 "</Welcome>"
REM Cycle 4:
reply -shadow 4
set REPLY_OFFSET=2
reply -bottom 3
set REPLY_OFFSET=5
set REPLY_SIZE=23
reply -item 3
set REPLY_OFFSET=2
set REPLY_SIZE=26
reply -bottom 21
set REPLY_OFFSET=5
set REPLY_SIZE=23
reply -item 21
REM Cycle 5:
reply -shadow 5
reply -item 4 to
reply -item 20 Enjoy!
REM Cycle 6:
reply -shadow 6
reply -item 5
reply -item 19
REM Cycle 7:
reply -shadow 7
reply -item 6 "<Gerolf>"
set REPLY_SIZE=23
reply -item 18 "</Gerolf>"
REM Cycle 8:
reply -shadow 8
reply -bottom 7
set REPLY_OFFSET=8
set REPLY_SIZE=20
reply -item 7
set REPLY_OFFSET=5
set REPLY_SIZE=23
reply -bottom 17
set REPLY_SIZE=20
set REPLY_OFFSET=8
reply -item 17
REM Cycle 9:
reply -shadow 9
reply -item 8 Markup
reply -item 16 %GMS_DATE% X X
REM Cycle 10:
reply -shadow 10
reply -item 9
reply -item 15
REM Cycle 11:
reply -shadow 11
reply -item 10 "<Shredder>"
reply -item 14 "</Shredder>"
REM Cycle 12:
reply -shadow 12
reply -bottom 11
set REPLY_OFFSET=11
set REPLY_SIZE=17
reply -item 11
REM Cycle 13:
set REPLY_OFFSET=8
set REPLY_SIZE=20
reply -bottom 13
set REPLY_OFFSET=11
set REPLY_SIZE=17
reply -item 13
reply -item 12 "Version %GMS_VERSION%" X X
REM Debug:
if -%GMS_DEBUG% == - goto fi_dbb
if %GMS_DEBUG% == X goto then_dbb
if not %GMS_DEBUG% == Y goto fi_dbb
:then_dbb
call I_banner -debug I_wel %1 %2 %3 %4
:fi_dbb
REM Wait:
set REPLY_OFFSET=2
set REPLY_SIZE=26
reply -item 2 "<Welcome>"
```

```

set REPLY_DESKTOP=0
  reply -top 1 - XX
set REPLY_DESKTOP=%backup_desktop%
goto exit
:fi_b

```

REM Remove cycle: -----

```

if not %1 == -cyc_remove goto fi_cr
  reply -stripe %2 %REPLY_PATTERN% %REPLY_LETTER%
  if %3 == 1 set REPLY_DESKTOP=0
    reply -shadow %3
    if %3 == 1 set REPLY_DESKTOP=%backup_desktop%
    reply -stripe %4 %REPLY_PATTERN% %REPLY_LETTER%

```

REM Debug:

```

if %GMS_DEBUG% == . goto fi_dbc
if %GMS_DEBUG% == X goto then_dbc
if not %GMS_DEBUG% == Y goto fi_dbc
:then_dbc
  call l_banner -debug l_wel %1 %2 %3 %4
:fi_dbc

```

```

goto exit
:fi_cr

```

REM Remove: -----

```

if not %1 == -remove goto fi_r

```

REM Cycle 1:

```

  set REPLY_OFFSET=8
  set REPLY_SIZE=20
  reply -shadow 12

```

REM Cycles 2 - 12:

```

call l_wel -cyc_remove 13 11 12
call l_wel -cyc_remove 14 10 11
call l_wel -cyc_remove 15 9 10

```

```

set REPLY_OFFSET=5
set REPLY_SIZE=23
call l_wel -cyc_remove 16 8 9
call l_wel -cyc_remove 17 7 8
call l_wel -cyc_remove 18 6 7
call l_wel -cyc_remove 19 5 6
set REPLY_OFFSET=2
set REPLY_SIZE=26
call l_wel -cyc_remove 20 4 5
call l_wel -cyc_remove 21 3 4
call l_wel -cyc_remove 22 2 3
call l_wel -cyc_remove 23 1 2

```

REM Cycle 13:

```

  set REPLY_DESKTOP=0
  reply -stripe 24 0 32
  reply -stripe 1 0 32

```

REM Debug:

```

if %GMS_DEBUG% == . goto fi_dbr
if %GMS_DEBUG% == X goto then_dbr
if not %GMS_DEBUG% == Y goto fi_dbr
:then_dbr
  call l_banner -debug l_wel %1 %2 %3 %4
:fi_dbr

```

```

goto exit
:fi_r

```

REM Not found: -----

```

:else
  call l_banner -no_action l_wel %1
:fi

```

```

:exit

```


reader.bat

```
REM reader.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateREADER=20060927

REM =====

REM Debug:
if "%OS%" == "" goto fi_db
if %GMS_DEBUG% == . goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> "%Z%"
:fi_db

REM =====

REM Error message on Dos:

if not "%OS%" == "" goto fi_dos
set GMS_RDR=
set GMS_TEST=
REM Search Adobe Acrobat Reader:
REM Windows 3x:
set GMS_TEST=%windir%\..\acrobat3\acrord16.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
set GMS_TEST=\acrobat3\acrord16.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
set GMS_TEST=C:\acrobat3\acrord16.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%

set GMS_TEST=%windir%\..\acrobat3\read16\acrord16.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
set GMS_TEST=\acrobat3\read16\acrord16.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
set GMS_TEST=C:\acrobat3\read16\acrord16.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%

if "%GMS_RDR%" == "" goto then_dos
if "%windir%" == "" goto fi_sw
cls
echo. %GMS_FILE%
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%arg%
echo.
echo. You may have to quit Windows 3x
echo. before starting Markup Shredder.
echo.
echo. Press any key ...
pause > nul
set arg=
goto exit
:fi_sw
win %GMS_RDR% %GMS_FOLDER%\%GMS_FILE%
goto fi_dos
:then_dos
set GMS_TEST=
set GMS_RDR=
cls
echo. %GMS_FILE%
set arg=////////////////////////////////////
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%arg%
echo.
echo. Could not find PDF reader.

echo.
echo. Acrobat Reader for Windows 3x can be downloaded from:
echo. http://vetusware.com
echo. It should be installed at 'C:\acrobat3'.

echo.
echo. Otherwise add the path to the reader binary to the
echo. PATH environment variable in 'C:\autoexec.bat'
echo. or add a startup batch file to:
echo. %GMS_BINARIES%

echo.
echo. Then change the 'Reader' entry in the 'Select Programs'
echo. dialog box or the GMS_READER environment variable in:
echo. %GMS_SETTING%\gerolf.bat

echo.
echo. You may have to quit Windows before starting GMS.

echo.
echo. Press any key ...

echo. %arg%arg%
echo. %GMS_FOLDER%
set arg=
pause > nul
goto exit
:fi_dos

REM Error message on Windows:

if "%OS%" == "" goto fi_win
set GMS_RDR=
set GMS_TEST=
REM Search Adobe Acrobat Reader:
REM Windows 9x:
set GMS_TEST=%windir%\..\Programs\Adobe\Acroba-1.0\Reader\AcroRd32.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
set GMS_TEST=%windir%\..\Progra-1\Adobe\Acroba-1.0\Reader\AcroRd32.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
set GMS_TEST=%windir%\..\Programs\Adobe\Acroba-2.0\Reader\AcroRd32.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
set GMS_TEST=%windir%\..\Progra-1\Adobe\Acroba-2.0\Reader\AcroRd32.exe
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
REM Windows XP:
set GMS_TEST=%ALLUSERSPROFILE%\Desktop\Acroba-1.lnk
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
set GMS_TEST=%ALLUSERSPROFILE%\Desktop\Acroba-2.lnk
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
set GMS_TEST=%ALLUSERSPROFILE%\Desktop\Adobe Reader 6.0.lnk
if exist %GMS_TEST% set GMS_RDR=%GMS_TEST%
if not "%GMS_RDR%" == "" "%GMS_RDR%" "%GMS_FOLDER%\%GMS_FILE%"
if not "%GMS_RDR%" == "" goto fi_win
set GMS_TEST=
set GMS_RDR=
cls
echo. %GMS_FILE%
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%arg%
echo.
echo. Problem: Could not find PDF Reader.
echo.
echo. Advice: Try Acrobat Reader or Xpdf.
echo. Add the path to the reader binary to the
echo. PATH environment variable in:
if "%OS%" == "Windows_9x" set arg1=C:\autoexec.bat
if not "%OS%" == "Windows_9x" set arg1=%SystemRoot%\System32\autoexec.nt
echo %arg1%
set arg1=
echo or add a startup .BAT or .LNK file to:
```


shredder.bat

```
REM shredder.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateSHREDDER=20060927

REM Prologue: =====

REM Not running:

if not "%GMS_BATCH%" == "" goto fi_nr
echo Gerolf Markup Shredder (shredder.bat) . . .
pause > nul
gerolf 2> nul
goto exit
:fi_nr

REM Check if module is given: -----

if not "%1" == "" goto fi_0
if not exist ..\gerolf goto else_0e
REM Load launcher (if not loaded by launcher):
call ..\gerolf -passive shredder
set REPLY_MODULE=g_palet
set REPLY_ACTION=-build
:else_0e
REM Not set up:
echo.
echo Please run 'gmssetup' script to initialize Gerolf Markup Shredder.
goto exit
:fi_0e
:fi_0

REM Help message: -----

if %1 == /help goto then_h
if not %1 == -help goto else_h
:then_h
echo.
echo This is Gerolf Markup Shredder,
echo written by G. D. Brettschneider (www.Gerolf.org)
echo Please send bug reports or corrections to
echo MarkupShredder@Gerolf.org - Thank you.
goto exit
:else_h
REM Initialize loop:
set REPLY_MODULE=%1
set REPLY_ACTION=%2
:fi_h

REM Debug: -----

if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == - goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
if %GMS_DEBUG% == X goto then_db
if not %GMS_DEBUG% == Y goto fi_db
:then_db
call l_banner -debug %1 %2 %3 %4
:fi_db

REM Build: -----

set REPLY_TEMP=%GMS_TEMP%\replytmp.bat
if "%OS%" == "" if not exist %GMS_TEMP%\replytmp.bat goto fi_bb
if not "%OS%" == "" if not exist "%GMS_TEMP%\replytmp.bat" goto fi_bb

del %GMS_TEMP%\replytmp.bat
:fi_bb
set REPLY_MODULE=g_palet
set REPLY_ACTION=-build
set REPLY_OFFSET=0
set REPLY_SIZE=0
set REPLY_ITEMS=0
reply -random 2
set GMS_RECEIVE=1
set GMS_TEXTMODE=1

REM Textmode interface master loop: =====

if "%OS%" == "" goto repeat
if not %GMS_DEBUG% == Z goto repeat
echo ----- Entering textmode interface master loop>> %Z%
:repeat

set REPLY_DATA=

REM Load and remove old temporary file 'replytmp': -----

if -%GMS_RECEIVE% == - goto fi_r
if not -%GMS_RECEIVE% == -1 goto fi_r
if "%OS%" == "" if not exist %REPLY_TEMP% goto fi_rr
if not "%OS%" == "" if not exist "%REPLY_TEMP%" goto fi_rr
call %REPLY_TEMP%
del %REPLY_TEMP%
set GMS_RECEIVE=0
:fi_rr
:fi_r

REM Debug: -----

if -%GMS_DEBUG% == - goto fi_db2
set arg=(%REPLY_MODULE%) (%REPLY_ACTION%)
set arg=%arg% (%REPLY_HOT%) (%REPLY_COLD%) (%REPLY_DATA%)
if "%OS%" == "" goto fi_db20
if %GMS_DEBUG% == Z echo ----- %0 %arg%>> %Z%
:fi_db20
if %GMS_DEBUG% == X goto then_db3
if not %GMS_DEBUG% == Y goto fi_db3
:then_db3
set arg=%REPLY_MODULE% %REPLY_ACTION%
set arg=%arg% %REPLY_HOT% %REPLY_COLD% %REPLY_DATA%
call l_banner -debug %arg%
:fi_db3
set arg=
:fi_db2

REM Load module: -----

if "%OS%" == "" if not exist %GMS_BATCH%\%REPLY_MODULE%.bat goto else_l
if not "%OS%" == "" if not exist "%GMS_BATCH%\%REPLY_MODULE%.bat" goto else_l
call %REPLY_MODULE% %REPLY_ACTION% %REPLY_HOT% %REPLY_COLD% %REPLY_DATA%
goto fi_l

REM Not found: -----

:else_l
call l_banner -no_module %REPLY_MODULE% shredder
goto exit
:fi_l

if -%GMS_BREAK% == - goto repeat
if not %GMS_BREAK% == 1 goto repeat
```

```

:taeper
REM gmsdebug.log footer:
if "%OS%" == "" goto fi_left
if not %GMS_DEBUG% == Z goto fi_left
echo ..... Leaving textmode interface master loop>> %Z%
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%%arg%%arg%>> %Z%
set arg=
echo. %GMS_SETTING%>> %Z%
:fi_left
REM Final screen:
cls
if "%OS%" == "" goto else_gb
echo Gerolf Markup Shredder . . .
goto fi_gb
:else_gb

echo Please support the author of Markup Shredder - www.Gerolf.org
:fi_gb
for %i in (1 2 3 4 5 6 7 8 9 10 11) do echo.
for %i in (1 2 3 4 5 6 7 8 9 10) do echo.
if "%OS%" == "" echo.

:exit
REM Final unsets:
if "%OS%" == "Windows_9x" set OS=
if "%OS%" == "Windows_XP" set OS=Windows_NT
if not "%GMS_DOSMODE%" == "" set windir=
set GMS_BREAK=
set GMS_DEBUG=
set GMS_DOSMODE=
set GMS_SETTING=
set Z=

```



```
if exist UPCIBI.TT_ expand %source%\UPCIBI.TT_ UPCIBI.TTF
if exist UPCII.TT_ expand %source%\UPCII.TT_ UPCII.TTF
if exist UPCIL.TT_ expand %source%\UPCIL.TT_ UPCIL.TTF
if exist UPCJB.TT_ expand %source%\UPCJB.TT_ UPCJB.TTF
if exist UPCJBI.TT_ expand %source%\UPCJBI.TT_ UPCJBI.TTF
if exist UPCJI.TT_ expand %source%\UPCJI.TT_ UPCJI.TTF
if exist UPCJL.TT_ expand %source%\UPCJL.TT_ UPCJL.TTF
if exist UPCKB.TT_ expand %source%\UPCKB.TT_ UPCKB.TTF
if exist UPCKBI.TT_ expand %source%\UPCKBI.TT_ UPCKBI.TTF
if exist UPCKI.TT_ expand %source%\UPCKI.TT_ UPCKI.TTF
if exist UPCKL.TT_ expand %source%\UPCKL.TT_ UPCKL.TTF
if exist UPCLB.TT_ expand %source%\UPCLB.TT_ UPCLB.TTF
if exist UPCLBI.TT_ expand %source%\UPCLBI.TT_ UPCLBI.TTF
```

```
if exist UPCLI.TT_ expand %source%\UPCLI.TT_ UPCLI.TTF
if exist UPCLL.TT_ expand %source%\UPCLL.TT_ UPCLL.TTF
if exist VERDANA.TT_ expand %source%\VERDANA.TT_ VERDANA.TTF
if exist VERDANAB.TT_ expand %source%\VERDANAB.TT_ VERDANAB.TTF
if exist VERDANAI.TT_ expand %source%\VERDANAI.TT_ VERDANAI.TTF
if exist VERDANAZ.TT_ expand %source%\VERDANAZ.TT_ VERDANAZ.TTF
if exist WEBDINGS.TT_ expand %source%\WEBDINGS.TT_ WEBDINGS.TTF
if exist WINGDING.TT_ expand %source%\WINGDING.TT_ WINGDING.TTF
```

:exit

```
set source=
set target=
```

viewer.bat

```
REM viewer.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2006).

REM set GMSdateVIEWER=20060927

REM =====

REM Debug:
if "%OS%" == "" goto fi_db
if -%GMS_DEBUG% == . goto fi_db
if %GMS_DEBUG% == Z echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8) (%9)>>%Z%
:fi_db

REM Select codepage on Windows 32 (Dos: 437, 850; Windows: 1250 ... 1258):
if not "%OS%" == "" chcp 1252 > nul

REM Select number of console text lines (25, 43 or 50):

if "%GMS_FreeDOS%" == "" mode con lines=50 > nul
if not "%GMS_FreeDOS%" == "" mode co80,50

REM Set viewer colors on Windows 32 (black on white):
if not "%OS%" == "" if not "%OS%" == "Windows_9x" color F0

REM Execute external program (filename as %1 or %GMS_SHORT%, in 8.3 form):
browse %GMS_FOLDER%\%GMS_SHORT%

REM Reset viewer colors on Windows 32:
if not "%OS%" == "" if not "%OS%" == "Windows_9x" color 07

REM Reset console text lines:
if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%GMS_FreeDOS%" == "" mode co80,25

REM Reset codepage on Windows 32:
if not "%OS%" == "" chcp 850 > nul
```

Codepage Files

cp437.txt	i8859-2.txt
cp850.txt	i8859-3.txt
cp860.txt	i8859-4.txt
cp863.txt	i8859-5.txt
cp865.txt	i8859-6.txt
cp874.txt	i8859-7.txt
cp1250.txt	i8859-8.txt
cp1251.txt	i8859-9.txt
cp1252.txt	i8859-10.txt
cp1253.txt	i8859-11.txt
cp1254.txt	i8859-13.txt
cp1255.txt	i8859-14.txt
cp1256.txt	i8859-15.txt
cp1257.txt	iscii.txt
cp1258.txt	iso646.txt
i8859-1.txt	us-ascii.txt
	viscii.txt

[GMS_ROOT]/data/enc

Encoding Files, Hand-Written from Code Page

arabicbh.enc

psy.enc

arabicmt.enc

pzd.enc

Encoding Files, Generated from Code Page

cp437.enc	i8859-2.enc
cp850.enc	i8859-3.enc
cp860.enc	i8859-4.enc
cp863.enc	i8859-5.enc
cp865.enc	i8859-6.enc
cp874.enc	i8859-7.enc
cp1250.enc	i8859-8.enc
cp1251.enc	i8859-9.enc
cp1252.enc	i8859-10.enc
cp1253.enc	i8859-11.enc
cp1254.enc	i8859-13.enc
cp1255.enc	i8859-14.enc
cp1256.enc	i8859-15.enc
cp1257.enc	iscii.enc
cp1258.enc	iso646.enc
i8859-1.enc	us-ascii.enc
	viscii.enc

Encoding Files, Generated from Unicode Row, with Glyph Names

g0000.enc

g0100.enc

g0200.enc

g0300.enc

g0400.enc

g0500.enc

g0600.enc

g0e00.enc

g1e00.enc

g2000.enc

g2100.enc

g2200.enc

g2300.enc

g2400.enc

g2500.enc

g2600.enc

g2700.enc

gf800.enc

Encoding Files, Generated from Unicode Row, with Unicode Names

u0000.enc

u0100.enc

u0200.enc

u0300.enc

u0400.enc

u0500.enc

u0600.enc

u0e00.enc

u1e00.enc

u2000.enc

u2100.enc

u2200.enc

u2300.enc

u2400.enc

u2500.enc

u2600.enc

u2700.enc

uf800.enc

[GMS_ROOT]/data/ent

Entity Lists

HTMLlat1.ent

HTMLspec.ent

HTMLsymb.ent

[GMS_ROOT]/data/gly

Glyph Lists

gms.gly

dingbats.gly

[GMS_ROOT]/data/krn

Kerning Tables

phv.krn

ptm.krn

[GMS_ROOT]/data/lang

Language Strings (**Web Browser** Interface)

english.txt

```
language_name|English|
button_accept|Accept|
button_accept_|Accept modifications and go on|
button_select|Select|
button_select_|Select a few system properties|
button_analyse|Analyse|
button_analyse_|Analyse hypertext markup syntax|
button_browse|Browse|
button_browse_|Browse through the markup file (*.htm)|
button_quit|Quit|
button_quit_|Quit the current markup file|
button_create|Create|
button_create_|Create a new markup file (*.htm)|
button_edit|Edit|
button_edit_|Edit the markup file (*.htm)|
button_join|Join|
button_join_|Join workspaces|
button_learn|Learn|
button_learn_|Learn to create documents with Markup Shredder|
button_download|Download|
button_download_|Download Gerolf Markup Shredder|
button_open|Open|
button_open_|Open an existing markup file (*.htm)|
button_open_hint|For security reasons most browsers delete your file selection
when calling this side.|
button_open_hint_|You can use the back button to come here again.|
button_open_left|Open a markup file <br />example (*.htm)|
button_open_picture|subfiles to be included|
button_open_right|Open an existing markup file (*.htm)|
button_read|Read|
button_read_|Read the portable document (*.pdf)|
button_reset|Reset|
button_reset_|Discard all modifications|
button_save|Save|
button_save_|Save the markup file (*.htm)|
button_split|Split|
button_split_|Split workspace|
button_get|Get|
button_get_|Get Markup Shredder by download|
button_typeset|Typeset|
button_typeset_|Typeset a portable document according to the hypertext markup|
button_view|View|
button_view_|View the markup file (*.htm)|
file_characters|letters of text|
file_cleared|There is no markup file (*.htm) given at present|
file_contains|contains|
file_is_no_text|does not contain text|
file_not_there|does not exist yet|
file_protected|but write-protected|
file_readable|is readable|
file_unreadable|can not be read|
file_writeable|and can be modified|
language_|In English, please!|
language_abbreviation|en|
own_file_|Open own markup file|
sample|Example|
sample_|Show example|
select_area_|Workspace|
select_files_|Maximum <i>number of files</i><br />allowed for transfer|
select_font|Size|
select_height|Height|
select_joined_|joined|
select_size_|Maximum <i>total size of files</i><br />allowed for transfer<br
/>(number of characters)|
select_split_|split|
select_width|Width|
template|Template|
template_|Choose template|
template_myfile_|File name|
template_empty|Empty page|
template_empty_html|Empty HTML page|
template_empty_html_meta|Empty HTML page with meta info|
template_empty_xhtml|Empty XHTML page|
template_empty_xhtml_meta|Empty XHTML page with meta info|
template_codepage|Code page|
template_letter|Letter|
template_story|Story|
template_one_column_left|One column (left)|
template_one_column_center|One column (center)|
template_one_column_right|One column (right)|
welcome_welcome|Welcome|
welcome_to|to|
welcome_version|Version|
welcome_enjoy|Enjoy!|
welcome_hint|This is an example of a 'well-formed markup list'|
```

french.txt

```
language_name|Franccedil;ais|
button_accept|Accepter|
button_accept_title|Accepter modifications et continuer|
button_select|Choisir|
button_select_|Choisir quelques propri&eacute;t&eacute;s du syst&egrave;me|
button_analyse|Examiner|
button_analyse_|Examiner l'hypertexte|
button_browse|Naviger|
button_browse_|Naviger dans l'hypertexte|
button_download|Download|
button_download_|Download Gerolf Markup Shredder|
button_create|Cr&eacute;er|
button_create_|Cr&eacute;er un hypertexte nouveau|
button_edit|R&eacute;diger|
button_edit_|R&eacute;diger l'hypertexte|
button_join|Joindre|
button_join_|Joindre les fen&ecirc;tres de travailler|
button_learn|Apprendre|
button_learn_|Apprendre &agrave; cr&eacute;er des documents avec Markup Shredder|
button_open|Ouvrir|
button_open_|Ouvrir un hypertexte existant (*.htm)|
button_open_hint|En raison de la s&eacute;curit&eacute;, la plupart des navigateurs &eacute;teint votre selection des fichiers en ouvrant cette page.|
button_open_hint_|Vous pouvez utiliser le bouton &agrave; retourner pour revenir ici.|
button_open_left|Ouvrir un hyperexte<br />exemplaire|
button_open_picture|sous-fichiers &agrave; ins&eacute;rer|
button_open_right|Ouvrir un hypertexte existant (*.htm)|
button_quit|Quitter|
button_quit_|Quitter l'hypertexte courant|
button_read|Lire|
button_read_|Lire le document portable (*.pdf)|
button_reset|Reculer|
button_reset_|Rejeter toutes modifications|
button_save|Sauver|
button_save_|Sauver l'hypertexte|
button_split|Fendre|
button_split_|Fendre le fen&ecirc;tre de travailler|
button_typeset|Composer|
button_typeset_|Composer un document portable suivant les marques de l'hypertexte|
button_view|Regarder|
button_view_|Regarder l'hypertexte|
file_characters|lettres de texte|
file_cleared|Il n'ya pas d'hypertexte (*.htm) &agrave; pr&eacute;sent|
file_contains|contient|
file_is_no_text|ne contient pas de texte|
file_not_there|n'existe pas encore|
file_protected|mais est prot&eacute;g&eacute; contre des modifications|
file_readable|peut &ecirc;tre lit|
file_unreadable|ne peut pas &ecirc;trelit|
file_writeable|et modifi&eacute;|
language_|En franccedil;ais, s'il vous pla&icirc;t!|
language_abbreviation|fr|
own_file_|Ouvrir un propre hypertexte|
sample|Exemple|
sample_|Montrer exemple|
select_area_|Panneau de travail|
select_split_|fendu|
select_files_|Maximum <i>nombre des fichiers</i><br />permis pour le transfert|
select_font|Grandeur|
select_height|Hauteur|
select_joined_|joint|
select_size_|Maximum <i>largeur total des fichiers</i><br />permis pour le transfert (nombre de signes)|
select_width|Largeur|
template|Mod&egrave;le|
template_|Choisir le mod&egrave;le|
template_myfile_|Nom du fichier|
template_empty|Empty page|
template_empty_html|Empty HTML page|
template_empty_html_meta|Empty HTML page with meta info|
template_empty_xhtml|Empty XHTML page|
template_empty_xhtml_meta|Empty XHTML page with meta info|
template_codepage|Code page|
template_letter|Letter|
template_story|Story|
template_one_column_left|One column (left)|
template_one_column_center|One column (center)|
template_one_column_right|One column (right)|
welcome_welcome|Bienvenue|
welcome_to|chez|
welcome_version|Version|
welcome_enjoy|Amusez-vous!|
welcome_hint|C'est un exemple pour un 'hypertexte bien-form&eacute;|'
```

german.txt

```
language_name|Deutsch|
button_accept|Annehmen|
button_accept_|&Auml;nderungen &uuml;bernehmen und fortfahren|
button_select|Ausw&auml;hlen|
button_select_|Einige Eigenschaften der Arbeitsumgebung ausw&auml;hlen|
button_analyse|Pr&uuml;fen|
button_analyse_|Bau der Auszeichnungsmarken pr&uuml;fen|
button_browse|&Uuml;berfliegen|
button_browse_|Markendatei (*.htm) im Brauser &uuml;berfliegen|
button_create|Erzeugen|
button_create_|Neue Markendatei (*.htm) erzeugen|
button_download|Herunterladen|
button_download_|GMS aus dem Netz herunterladen|
button_edit|Bearbeiten|
button_edit_|Markendatei (*.htm) bearbeiten|
button_join|Vereinigen|
button_join_|Arbeitsbereiche vereinigen|
button_learn|Lernen|
button_learn_|Lernen, wie man Schriftst&uuml;cke mit GMS erstellt|
button_open|&Ouml;ffnen|
button_open_|Bestehende Markendatei (*.htm) &uuml;ffnen|
button_open_hint|Aus Sicherheitsgr&uuml;nden l&uuml;schen die meisten Brauser Ihre Dateiauswahl bei erneutem Aufruf dieser Seite.|
button_open_hint_|Verwenden Sie gegebenenfalls die R&uuml;cktaaste, um wieder hierher zu gelangen.|
button_open_left|&Ouml;ffnen einer Beispiel-<br />Markendatei (*.htm)|
button_open_picture|einzuf&uuml;gende Unterdateien|
button_open_right|&Ouml;ffnen einer bestehenden Markendatei (*.htm)|
button_quit|Verlassen|
button_quit_|Jetzige Markendatei (*.htm) verlassen|
button_read|Lesen|
button_read_|&Uuml;bertragbares Schriftst&uuml;ck (*.pdf) lesen|
button_reset|R&uuml;cksetzen|
button_reset_|Alle &Auml;nderungen verwerfen|
button_save|Sichern|
button_save_|Markendatei (*.htm) sichern|
button_split|Teilen|
button_split_|Arbeitsbereich teilen|
button_typeset|Setzen|
button_typeset_|&Uuml;bertragbares Schriftst&uuml;ck gem&auml; &szlig; der Auszeichnungsmarken setzen|
button_view|Betrachten|
button_view_|Markendatei (*.htm) betrachten|
file_characters|Buchstaben|
file_cleared|Keine Markendatei (*.htm) gegeben|
file_contains|enth&auml;lt|
file_is_no_text|enth&auml;lt nichts Lesbares|
file_not_there|gibt es noch nicht|
file_protected|ist aber schreibgesch&uuml;tzt|
file_readable|kann gelesen werden|
file_unreadable|ist nicht lesbar und|
file_writable|und ist &uuml;nderbar|
language_|Auf Deutsch, bitte!|
language_abbreviation|de|
own_file_|Eigene Markendatei &uuml;ffnen|
sample|Beispiel|
sample_|Beispiel anzeigen|
select_area_|Arbeitsbereich|
select_files_|H&uuml;chstzul&auml;ssige <i>Anzahl</i><br />hinaufladbarer Dateien|
select_font|Gr&uuml; &szlig; e|
select_height|H&uuml;he|
select_joined_|vereinigt|
select_size_|H&uuml;chstzul&auml;ssige <i>Gesamtgr&uuml; &szlig; e</i><br />hinaufladbarer Dateien (Zeichenzahl)|
select_split_|geteilt|
select_width|Breite</i>|
template|Vorlage|
template_|Vorlage ausw&auml;hlen|
template_myfile_|Dateiname|
template_empty|Leere Seite|
template_empty_html|Leere HTML-Seite|
template_empty_html_meta|Leere HTML-Seite mit Zusatz|
template_empty_xhtml|Leere XHTML-Seite|
template_empty_xhtml_meta|Leere XHTML-Seite mit Zusatz|
template_codepage|Zeichensatz|
template_letter|Brief|
template_story|Geschichte|
template_one_column_left|Eine Spalte (links)|
template_one_column_center|Eine Spalte (mittig)|
template_one_column_right|Eine Spalte (rechts)|
welcome_welcome|Willkommen|
welcome_to|bei|
welcome_version|Fassung|
welcome_enjoy|Viel Vergn&uuml;gen!|
welcome_hint|Dies ist ein Beispiel f&uuml;r eine 'wohlgeformte Auszeichnungsliste'|
```

[GMS_ROOT]/data/row

Unicode Rows

u0000.row

u0100.row

u0200.row

u0300.row

u0400.row

u0500.row

u0600.row

u0a00.row

u0e00.row

u1e00.row

u2000.row

u2100.row

u2200.row

u2300.row

u2400.row

u2500.row

u2600.row

u2700.row

u3000.row

u3100.row

u3200.row

u3300.row

uf600.row

uf700.row

uf800.row

ufb00.row

ufc00.row

ufd00.row

ufe00.row

uff00.row

[GMS_ROOT]/doc/reply

Ansi-C Source (Text Mode Interface)

reply.c

```

/* reply.c
=====

This file is part of Gerolf Markup Shredder,
written by G. D. Brettschneider (1999-2006). All rights reserved.
Send corrections to: MarkupShredder(at)Gerolf.org (www.Gerolf.org)
Subject: Reply shell script interface */

#define GMSdateREPLY "20061102"
#define GMSversionREPLY "0.05a"

/* ===== */

/* Required libraries: */

#ifndef CONIO
#include <conio.h>
#include <dos.h>
#endif

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

/* Console routines for compatibility with conio.h: ----- */

/* Get single character from keyboard (without waiting for "enter"): */
#ifndef CONIO
static char getch (void) {
    static unsigned char old_c = 0;
    unsigned char new_c = 0;
    /* Return special key (which has been backed up from previous call): */
    if (old_c) {new_c = old_c; old_c = 0; return new_c;}
    /* Get a new character (which may be a multi-byte escape sequence): */
    scanf ("%c", &new_c); old_c = 0;
    if (new_c == 27) {scanf ("%c", &new_c); scanf ("%c", &new_c);
    /* Mask arrow keys: */
        if (new_c == 'A') {old_c = 'H';}
        else if (new_c == 'B') {old_c = 'P';}
        else if (new_c == 'C') {old_c = 'M';}
        else if (new_c == 'D') {old_c = 'K';}
        else {old_c = 0;}
        new_c = 0;}
    return new_c;}
#endif

/* Clear screen: */
#ifndef CONIO
static void clrscr (void) {printf ("\033[2J");}
#endif

/* Normvideo: */
#ifndef CONIO
static void normvideo (void) {
    printf ("\033[0m");}
#endif

/* Set cursor absolutely: */
#ifndef CONIO
static void gotoxy (int x, int y) {printf ("\033[%d;%dH", y, x);}
#endif
#endif

/* Set cursor relatively (one column or row): */
static void cursor_up (void) {
    #ifndef CONIO
        gotoxy (wherex (), (wherey () - 1));
    #else
        printf ("\033[1A");
    #endif
}
static void cursor_down (void) {
    #ifndef CONIO
        gotoxy (wherex (), (wherey () + 1));
    #else
        printf ("\033[1B");
    #endif
}
static void cursor_right (void) {
    #ifndef CONIO
        gotoxy (wherex () + 1, (wherey ()));
    #else
        printf ("\033[1C");
    #endif
}
static void cursor_left (void) {
    #ifndef CONIO
        gotoxy (wherex () - 1, (wherey ()));
    #else
        printf ("\033[1D");
    #endif
}

/* Set textbackground and textcolor: */
/* Background (black ... lightgray): */
static char my_background [8] [5] = {
    "0;40", // 0 black
    "0;44", // 1 blue
    "0;42", // 2 green
    "0;46", // 3 cyan
    "0;41", // 4 red
    "0;45", // 5 magenta
    "0;43", // 6 brown
    "0;47"; // 7 lightgray
}
/*Foreground (black ... white): */
static char my_foreground [16] [5] = {
    "30", // 0 black
    "34", // 1 blue
    "32", // 2 green
    "36", // 3 cyan
    "31", // 4 red
    "35", // 5 magenta
    "33", // 6 brown
    "37", // 7 lightgray
    "1;30", // 8 darkgray
    "1;34", // 9 lightblue
    "1;32", // 10 lightgreen
    "1;36", // 11 lightcyan
    "1;31", // 12 lightred
    "1;35", // 13 lightmagenta
    "1;33", // 14 yellow
    "1;37"; // 15 white
}

```

```

/* Text background: */
#ifdef CONIO
    static void textbackground (int color) {
        printf ("\033[%sm", my_background [color]);}
#endif
/* Text color: */
#ifdef CONIO
    static void textcolor (int color) {
        printf ("\033[%sm", my_foreground [color]);}
#endif

/* Colorize: */
static void colorize (int back, int fore) {
    textbackground (back); textcolor (fore);}

/* Sleep one second: */
#ifdef CONIO
    static void sleep_ (void) {delay (1000);}
#endif

/* Global variables: ----- */

#define line_length 80
#define lines 24

typedef unsigned char uchar;
typedef unsigned char boolean;

static int error = 0;
static int line_number = 0;
static int filler = 0;

static int box_width = 24;
static int box_offset = 30;

static int item_number = 0;
static int item_amount = 5;
static int item_index = 0;

static char item_hotkey = '\0';
static char item_next = '\0';

static boolean item_background = 0;
static boolean item_question = 0;
static boolean item_selected = 0;
static boolean item_measurement = 0;

static char *mode = ".help";

static char item_hotstr [line_length + 1];
static char item_entry [line_length + 1];
static char line [line_length + 1];
static char item_ [line_length + 1];

/* Background colors: */
static int color_back; // desktop
static int color_banner;
static int color_shadow;

/* Foreground colors: */
static int color_fore; // pattern
static int color_text;
static int color_hotkey;

/* Time measurement: */
static long after;
static long before;

/* Environment variables: ----- */

static char *REPLY_ITEMS;
static char *REPLY_OFFSET;
static char *REPLY_SIZE;
static char *REPLY_MODULE;
static char *REPLY_ACTION;
static char *REPLY_HOT;
static char *REPLY_COLD;
static char *REPLY_TEMP;
static char *REPLY_TEXT;

/* Background colors: */
static char *REPLY_DESKTOP;
static char *REPLY_BANNER;
static char *REPLY_SHADE;

/* Foreground colors: */
static char *REPLY_PATTERN;
static char *REPLY_TEXT;
static char *REPLY_HOTKEY;

/* Time measurement: */
static char *REPLY_BEFORE;

/*Get environment variables: -----*/

#define reply_error "REPLY error: Environment"
#define need_num "must be set to a number"

static void get_environment (void) {
    /* Name of menu-building script: */
    REPLY_MODULE = getenv ("REPLY_MODULE");
    REPLY_ACTION = getenv ("REPLY_ACTION");
    /* Name of temporary answering script: */
    REPLY_TEMP = getenv ("REPLY_TEMP");
    /* Width of dialog box, or height of background: */
    REPLY_SIZE = getenv ("REPLY_SIZE");
    if ((REPLY_SIZE != NULL) && (sscanf
        ((char *) REPLY_SIZE, "%d", &box_width) == 0))
        printf (" %s REPLY_SIZE %s. ", reply_error, need_num);
    /* Horizontal offset of dialog box: */
    REPLY_OFFSET = getenv ("REPLY_OFFSET");
    if ((REPLY_OFFSET != NULL) && (sscanf
        ((char *) REPLY_OFFSET, "%d", &box_offset) == 0))
        printf (" %s REPLY_OFFSET %s. ", reply_error, need_num);
    /* Number of menu items: */
    REPLY_ITEMS = getenv ("REPLY_ITEMS");
    if ((REPLY_ITEMS != NULL) && (sscanf
        ((char *) REPLY_ITEMS, "%d", &item_amount) == 0))
        printf (" %s REPLY_ITEMS %s. ", reply_error, need_num);
    /* Number of desktop color (0..7): */
    REPLY_DESKTOP = getenv ("REPLY_DESKTOP");
    if ((REPLY_DESKTOP !=NULL) && (sscanf
        ((char *) REPLY_DESKTOP, "%d", &color_back) == 0))
        printf (" %s REPLY_DESKTOP %s. ", reply_error, need_num);
    /* Number of pattern color (0..15): */
    REPLY_PATTERN = getenv ("REPLY_PATTERN");
    if ((REPLY_PATTERN !=NULL) && (sscanf
        ((char *) REPLY_PATTERN, "%d", &color_fore) == 0))
        printf (" %s REPLY_PATTERN %s. ", reply_error, need_num);
    /* Number of banner color (0..7): */
    REPLY_BANNER = getenv ("REPLY_BANNER");
    if ((REPLY_BANNER != NULL) && (sscanf
        ((char *) REPLY_BANNER, "%d", &color_banner) == 0))
        printf (" %s REPLY_BANNER %s. ", reply_error, need_num);
    /* Number of text color (0..15): */
    REPLY_TEXT = getenv ("REPLY_TEXT");
    if ((REPLY_TEXT != NULL) && (sscanf
        ((char *) REPLY_TEXT, "%d", &color_text) == 0))
        printf (" %s REPLY_TEXT %s. ", reply_error, need_num);
    /* Number of shadow color (0..7): */
    REPLY_SHADE = getenv ("REPLY_SHADE");
}

```

```

if ((REPLY_SHADE != NULL) && (sscanf
    ((char *) REPLY_SHADE, "%d", &color_shadow) == 0))
    printf (" %s REPLY_SHADE %s. ", reply_error, need_num);
/* Number of hotkey color (0..15): */
REPLY_HOTKEY = getenv ("REPLY_HOTKEY");
if ((REPLY_HOTKEY != NULL) && (sscanf
    ((char *) REPLY_HOTKEY, "%d", &color_hotkey) == 0))
    printf (" %s REPLY_HOTKEY %s. ", reply_error, need_num);
/* Time measurement: */
REPLY_BEFORE = getenv ("REPLY_BEFORE");
if ((REPLY_BEFORE != NULL) && (sscanf
    ((char *) REPLY_BEFORE, "%ld", &before) == 0))
    printf (" %s REPLY_BEFORE %s. ", reply_error, need_num);

#undef need_num

/* Command line parameters: ----- */

static char **P_argv;
static int Argc;

/* (1) Get mode: */
static void get_mode (void) {
    if (Argc < 2) return;
    mode = P_argv [1];}

/* (n) Get number parameter: */
#define msg1 "REPLY error: Number ex"
#define msg2 "pected. Parameter number "
#define msg3 "must be the "
#define msg4 " , not "
static int get_number (int par_number,
    char *par_name) {
    int i, error;
    if (Argc < par_number + 1) return (0);
    error = (sscanf ((char *) P_argv [par_number], "%d", &i) == 0);
    if (strcmp (P_argv [par_number], "") || (error == 0)) return (i);
    printf (" %s%s%d %s%s%s\n", msg1, msg2, par_number,
        msg3, par_name, msg4, P_argv [par_number]);
    return (0);}
#undef msg1
#undef msg2
#undef msg3
#undef msg4

/* (5) Get text for banner, menu item, default answer, fill sign: */
static void get_text (void) {
    int n, limit;
    if (Argc < 4) return;
    strcpy (item_, P_argv [3]);
    limit = strlen (P_argv [3]);
    for (n = 0; n < limit; n++) {if (item_[n] == '=' item_[n] = ' ');}

/* (6) Get item hotkey: */
static void get_item_hotkey (void) {
    if (Argc < 5) return;
    strcpy (item_hotstr, P_argv [4]);
    if (*item_hotstr != '\0') item_hotkey = item_hotstr [0];
    else item_hotkey = '\0';
    item_selected = (strlen (item_hotstr) > 1);}

/* Get command line parameters: */
static void get_parameters (int argc, char *argv []) {
    Argc = argc;
    P_argv = argv;
    get_mode (); // 1
    line_number = get_number (2, "line number");
    get_text (); // 3
    get_item_hotkey (); // 4
    item_number = get_number (5, "item number");}

/* Answering file handling: ----- */

/*Check whether file exists: */
static boolean file_exists (char *name) {
    FILE *p_file = NULL;
    if ((p_file = fopen (name, "rt")) == NULL) return 0; else return 1;}

/* Write file error message: */
static void file_error (char *action, char *name) {
    error = 1;
    gotoxy (1, 1);
    printf (" REPLY error: Cannot %s %s", action, name);
    getch ();}

/* Write replytmp or replytmp.bat: */
static void write_replytmp (char *first, char *second) {
    int n;
    char *my_str = item_;
    FILE *replytmp = NULL;
    /* Search old file: */
    if (file_exists (REPLY_TEMP)) {
        file_error ("overwrite", REPLY_TEMP); return;}
    /* Rewrite file: */
    if ((replytmp = fopen (REPLY_TEMP, "wt")) == NULL) {
        file_error ("rewrite", REPLY_TEMP); return;}
    /* Write content (1: header line): */
    if (fprintf (replytmp, "%s\n", second) <= 0) {
        file_error ("write header line to", REPLY_TEMP); return;}
    /* Okay... omit error check for the next lines: */
    /* Write content (2: hot and cold state, if not background): */
    if (!item_background) {
        if ((int) item_next < 32) {
            fprintf (replytmp, "%s REPLY_HOT=%d\n", first, item_next);}
        else {fprintf (replytmp, "%s REPLY_HOT=%c\n", first, item_next);}
        if ((int) item_hotkey < 32) {
            fprintf (replytmp, "%s REPLY_COLD=%d\n", first, item_hotkey);}
        else {fprintf (replytmp, "%s REPLY_COLD=%c\n", first, item_hotkey);}
    }
    /* Write content (3: data (if question): */
    if (item_question) {
        fprintf (replytmp, "%s REPLY_DATA=%s\n", first, my_str);}
    /* Write content (4: color palette (if background): */
    if (item_background) {
        fprintf (replytmp, "%s REPLY_DESKTOP=%d\n", first, color_back);
        fprintf (replytmp, "%s REPLY_PATTERN=%d\n", first, color_fore);
        fprintf (replytmp, "%s REPLY_BANNER=%d\n", first, color_banner);
        fprintf (replytmp, "%s REPLY_TEXT=%d\n", first, color_text);
        fprintf (replytmp, "%s REPLY_SHADE=%d\n", first, color_shadow);
        fprintf (replytmp, "%s REPLY_HOTKEY=%d\n", first, color_hotkey);
        fprintf (replytmp, "%s REPLY_LETTER=%d\n", first, filler);}
    if (item_measurement) {
        printf (replytmp, "%s REPLY_AFTER=%ld\n", first, after);}
    /* Close file: */
    if (fclose (replytmp) != 0) {
        file_error ("close", REPLY_TEMP); return;}

/* Write user input to temporary files in Unix and Dos shell script styles: */
static void write_user_input (void) {
    char *bat = ".bat"; char *Bat = ".Bat"; char *BAT = ".BAT";
    if (strstr (REPLY_TEMP, bat) != NULL
        || strstr (REPLY_TEMP, Bat) != NULL
        || strstr (REPLY_TEMP, BAT) != NULL)
        write_replytmp ("set", "@echo off");
    else write_replytmp ("export", "#!/bin/sh");}

/* Line drawing functions: ----- */

/* Fill line: */
static void fill_line (int yoffset) {
    int i;
    gotoxy (1 + box_offset, line_number + yoffset);}

```

```

colorize (color_back, color_fore);
for (i = 0; i < box_width + 2; i++)
#ifdef CONIO
    printf ("%c", filler);
#else
    printf ("%c", filler);
#endif
return;}

/* Draw definite background: */
static void definite_background (void) {
    int n; char *tab = "\t";
    char col_back [3]; char col_fore [3];
    char col_banner [3]; char col_text [3];
    char col_shadow [3]; char col_hotkey [3];
    char char_filler [3];
    sprintf (col_back, "%d", color_back);
    sprintf (col_fore, "%d", color_fore);
    sprintf (col_banner, "%d", color_banner);
    sprintf (col_text, "%d", color_text);
    sprintf (col_shadow, "%d", color_shadow);
    sprintf (col_hotkey, "%d", color_hotkey);
    sprintf (char_filler, "%d", filler);
    colorize (color_back, color_fore);
    /* box_width here means background height: */
    for (n = 0; n < box_width; n++) fill_line (n);
    /* Save background information to temp file: */
    item_background = 1; item_next = ' '; item_hotkey = ' ';
    //strcpy (item_, "-g palet");
    strcat (item_, tab); strcat (item_, col_back);
    strcat (item_, tab); strcat (item_, col_fore);
    strcat (item_, tab); strcat (item_, col_banner);
    strcat (item_, tab); strcat (item_, col_text);
    strcat (item_, tab); strcat (item_, col_shadow);
    strcat (item_, tab); strcat (item_, col_hotkey);
    strcat (item_, tab); strcat (item_, char_filler);
    write_user_input ();}

/* Color pairs that have enough contrast to be readable ("rainbow"): */
// 0 1 2 3 4 5 6 7
// BK BL GN CY RD MA BN LGR
static int pairs [16] [8] = {
#ifdef CONIO
    {0, 0, 1, 1, 0, 0, 1, 1}, // 0 BK
    {0, 0, 0, 0, 0, 0, 0, 1}, // 1 BL
    {0, 0, 0, 0, 0, 0, 0, 0}, // 2 GN
    {0, 0, 0, 0, 0, 0, 0, 0}, // 3 CY

    {0, 0, 1, 1, 0, 0, 0, 1}, // 4 RD
    {0, 0, 1, 1, 0, 0, 0, 1}, // 5 MA
    {0, 0, 0, 0, 0, 0, 0, 0}, // 6 BN
    {1, 1, 0, 0, 1, 1, 0, 0}, // 7 LGR

    {0, 0, 0, 0, 0, 0, 0, 0}, // 8 DGR
    {0, 0, 0, 0, 0, 0, 0, 1}, // 9 LBL
    {1, 1, 0, 0, 1, 1, 1, 0}, //10 LGN
    {1, 1, 0, 0, 1, 1, 0, 0}, //11 LCY

    {0, 0, 1, 1, 0, 0, 0, 1}, //12 LRD
    {0, 0, 0, 0, 0, 0, 0, 0}, //13 LMA
    {1, 1, 0, 0, 1, 1, 1, 0}, //14 YE
    {1, 1, 0, 0, 1, 1, 1, 0}, //15 WH
#else
    {0, 0, 1, 1, 0, 0, 1, 1}, // 0 BK
    {0, 0, 0, 0, 0, 0, 0, 1}, // 1 BL
    {0, 0, 0, 0, 0, 0, 0, 0}, // 2 GN
    {0, 0, 0, 0, 0, 0, 0, 0}, // 3 CY

    {0, 0, 1, 1, 0, 0, 1, 1}, // 4 RD
    {0, 0, 0, 0, 0, 0, 0, 1}, // 5 MA
#endif
};

/* Draw random background: */
static void random_back (void) {
    int i; struct tm *time_now; time_t secs_now;
    char weekday [2]; // BK, BL, CY, GN, BN, RD, LGR
    int color_of_day [8] = {0, 1, 3, 2, 6, 4, 7};
    /* Randomize: */
    srand ((unsigned) time (&secs_now));
    /* Get time: */
    tzset (); time (&secs_now); time_now = localtime (&secs_now);
    strftime (weekday, 2, "%w", time_now);
    /* Get three different background colors: */
    /* Background color: */
    color_back = color_of_day [weekday [0] - 48];
    i = 1000;
    do {i = 0;
        /* Banner color: */
        do color_banner = (8.0 * rand () / (RAND_MAX + 1.0));
        while (color_banner == color_back);
        /* Shadow color: */
        do color_shadow = (8.0 * rand () / (RAND_MAX + 1.0));
        while ((color_shadow == color_back) || (color_shadow == color_banner));
        /* Get readable paired foreground colors: */
        /* Foreground color: */
        do color_fore = (16.0 * rand () / (RAND_MAX + 1.0));
        while (!pairs [color_fore] [color_back]);
        /* Text color: */
        do color_text = (16.0 * rand () / (RAND_MAX + 1.0));
        while (!pairs [color_text] [color_banner]);
        /* Hotkey color: */
        do {i++; // avoid hang-up
            color_hotkey = (16.0 * rand () / (RAND_MAX + 1.0));
            while (((!pairs [color_hotkey] [color_back])
                || (!pairs [color_hotkey] [color_banner])
                || (!pairs [color_hotkey] [color_shadow])) && (i < 1000));
        } while (i > 999);
        /* Fill pattern character: */
        filler = (95.0 * rand () / (RAND_MAX + 1.0)) + 32.0;
        definite_background ();}

/* Time measurement: */
static void measure_time (void) {
    struct tm *time_now; time_t t;
    t = time (NULL);
    item_measurement = 1;
    after = t - before;}

/* Write text of item and whitespace: ----- */
/* Write single character: */
static void write_char (char character, uchar back, uchar fore) {
    colorize (back, fore);
#ifdef CONIO
    printf ("%c", character);
#else
    printf ("%c", character);
#endif
};

```

```

#endif
}
/* Fill up with whitespace or fill character, if text fits into box: */
static void write_aftertext (char character, uchar back, uchar fore) {
    long i, limit;
    colorize (back, fore);
    limit = box_width - 4;
    for (i = strlen(item_) + 1; i <= limit; i++)
#ifdef CONIO
        cprintf ("%c", character);
#else
        printf ("%c", character);
#endif
}
/* Write single whitespace or character fill character: */
static void write_gap (char character, uchar back, uchar fore) {
    colorize (back, fore);
    if (*item_ != '\0') {
#ifdef CONIO
        cprintf ("%c", character);
#else
        printf ("%c", character);
#endif
    } else {
#ifdef CONIO
        cprintf ("%c", character);
#else
        printf ("%c", character);
#endif
    }
}
/* Write text of item: */
static void write_item_text (uchar back, uchar fore,
    uchar hotback, uchar hotfore) {
    long i, limit; int found = 0;
    colorize (back, fore);
    limit = strlen (item_);
    for (i = 0; i < limit; i++) {
        if (item_[i] == item_hotkey && found == 0) {
            found = 1;
            /* Colorize hotkey: */
            colorize (hotback, hotfore);
#ifdef CONIO
            cprintf ("%c", item_[i]);
#else
            printf ("%c", item_[i]);
#endif
            colorize (back, fore);}
        else {
            /* Write other letters of item text: */
#ifdef CONIO
            cprintf ("%c", item_[i]);
#else
            printf ("%c", item_[i]);
#endif
        }
    }
}
/* Write static line: */
static void write_static_line (char character, uchar back, uchar fore,
    uchar hotback, uchar hotfore) {
    if (strlen (item_) < box_width - 3) {
        write_gap (character, back, fore);
        write_item_text (back, fore, hotback, hotfore);
        write_gap (character, back, fore);}
    else
        write_item_text (back, fore, hotback, hotfore);
    write_aftertext (character, back, hotfore);}

/* Edit text of item: ..... */

/* Cursor movement: */

/* Move cursor left: */

static void move_left (void) {
    if (item_index > 0) {cursor_left ();item_index --;}}
/* Move cursor right: */
static void move_right (void) {
    if (item_index < strlen (item_)) {cursor_right (); item_index ++;}}
/* Move cursor up: */
static void move_up (void) {
    if (item_number == 1) item_number = item_amount; else item_number --;
    item_next = (char) item_number;}
/* Move cursor down: */
static void move_down (void) {
    if (item_number == item_amount) item_number = 1; else item_number ++;
    item_next = (char) item_number;}

/* Text deletion and insertion: */

/* Move cursor back (and delete what is there): */
static void move_back (uchar back, uchar fore) {
    int i;
    if (item_index > 0) {
        /* Set cursor and index: */
        cursor_left (); item_index --;
        /* Move right and shift letters: */
        for (i = item_index; i < strlen (item_); i++) {
            item_ [i] = item_ [i + 1]; write_char (item_ [i], back, fore);}
        /* Overwrite last letter: */
        write_char (' ', back, fore);
        /* Move left to index position: */
        for (i = strlen (item_); i >= item_index; i--) cursor_left ();}}

/* Move cursor forth (and insert new text): */
static void move_forth (char character, uchar back, uchar fore) {
    int i;
    if ((strlen (item_) < box_width - 4)
        && (character > 31) && (character != 127)) {
        /* Move right to end of text: */
        for (i = item_index; i <= strlen (item_); i++) cursor_right ();
        /* Move left and shift letters: */
        for (i = strlen (item_); i > item_index; i--) {
            item_ [i] = item_ [i - 1]; cursor_left ();
            write_char (item_ [i], back, fore); cursor_left ();}
        /* Set cursor and index: */
        cursor_left (); item_index ++;
        /* Insert new letter: */
        item_ [item_index - 1] = character;
        write_char (character, back, fore);}

/* Edit item text: */
static void edit_item_text (uchar back, uchar fore) {
    unsigned char c = '\0'; unsigned char d = '\0';
    colorize (back, fore);
    do {c = getch ();
        if (!c) {/* It's a special character: */
            d = getch ();
            /* Unmask arrow keys: */
            if (d == 'K') move_left ();
            else if (d == 'M') move_right ();
            else if (d == 'H') {move_up (); break;}
            else if (d == 'P') {move_down (); break;}}
        else if (c == 8 && item_index > 0) {
            move_back (back, fore);
#ifdef CONIO
            cursor_left ();
#endif
        }
        else if (c == 127) move_back (back, fore);
        else move_forth (c, back, fore);}
    while (1);}

/* Edge below line needs special care for 3D effects: */

```

```

/* Goto edge below (to set cursor at a nice place): */
static void goto_edge_below (void) {cursor_left (); cursor_down ();}
/* Draw edge below: */
static void draw_edge_below (void) {
    cursor_left (); cursor_down (); cursor_left ();
    write_char ('\|', color_shadow, color_hotkey);}
/* Remove edge below: */
static void remove_edge_below (void) {
    if (!item_selected) {return;}
    cursor_left ();
    write_char ('|', color_shadow, color_hotkey);
    cursor_left ();}
/* Goto start point: */
static void goto_start (void) {remove_edge_below ();
    gotoxy ((int) (box_offset + 1), (int) line_number);}

/* Distinguish different line types: ..... */

/* Line definition to fixed colors and fill character: */
#define msg1 "REPLY error (-background,"
#define msg2 " -stripe): Number expected."
#define par0 "Parameter number"
#define par3 "the foreground color"
#define par4 "an ASCII code"
static void fix_color_and_filler (void) {
    int i, err3, err4;
    /* Take parameter 3 as color index: */
    err3 = (sscanf (item_, "%d", &color_fore) == 0);
    if (!strcmp (item_, "") || err3) {
        printf (" %s%s ", msg1, msg2);
        printf ("%s 3 must be %s", par0, par3);
        printf (" , not \"%s\". ", item_); return;}
    /* Take parameter 4 as fill character code: */
    err4 = (sscanf (item_hotstr, "%d", &filler) == 0);
    if (!strcmp (item_hotstr, "") || err4) {
        printf (" %s%s ", msg1, msg2);
        printf ("%s 4 must be %s", par0, par4);
        printf (" , not \"%s\". ", item_hotstr); return;}}
#undef msg1
#undef msg2
#undef par0
#undef par3
#undef par4

/* Stripe: */
static void stripe (void) {
    item_selected = 0; fix_color_and_filler (); fill_line (0);}

/* Clear: */
static void clear (void) {
    item_selected = 0; filler = ' ';
#ifdef CONIO
    color_back = 0; // black DOS
    color_fore = 15; // white
#else
    color_back = 15; // white Unix
    color_fore = 0; // black
#endif
    normvideo (); fill_line (0);}

/* Background: */
static void background (void) {
    item_selected = 0; fix_color_and_filler (); definite_background ();}

/* Banner: */
static void banner (void) {goto_start ();
    write_static_line ('|', color_banner, color_text,
        color_banner, color_text);}

/* Top of box: */
static void top (void) {goto_start ();
    write_char ('+', color_banner, color_hotkey);
    write_static_line ('.', color_banner, color_hotkey,
        color_banner, color_hotkey);
    write_char ('+', color_banner, color_hotkey);
    write_char ('\|', color_back, color_hotkey);
    cursor_right ();
    goto_edge_below ();}

/* Item box: */
static void item (void) {goto_start ();
    if (item_selected) {
        write_char ('\|', color_back, color_hotkey);
        write_char ('|', color_shadow, color_hotkey);
        write_static_line ('|', color_shadow, color_hotkey,
            color_banner, color_text);
        write_char ('\|', color_shadow, color_hotkey);
        write_char ('|', color_back, color_hotkey);
        draw_edge_below ();
        return;}
    write_char ('|', color_banner, color_hotkey);
    write_static_line ('|', color_banner, color_text,
        color_shadow, color_hotkey);
    write_char ('|', color_banner, color_hotkey);
    write_char ('|', color_shadow, color_banner);
    write_char ('|', color_back, color_hotkey);
    goto_edge_below ();}

/* Question box: */
static void question (void) {
    int i; item_question = 1; goto_start ();
    if (item_selected) {
        write_char ('\|', color_back, color_hotkey);
        write_char ('|', color_shadow, color_hotkey);
        write_static_line ('|', color_shadow, color_hotkey,
            color_banner, color_text);
        write_char ('\|', color_shadow, color_hotkey);
        write_char ('|', color_back, color_hotkey);
        draw_edge_below (); cursor_up ();
        item_index = strlen (item_);
        for (i = item_index + 1; i <= box_width - 2; i++) {cursor_left ();}
        edit_item_text (color_shadow, color_hotkey); return;}
    write_char ('|', color_banner, color_hotkey);
    write_static_line ('|', color_banner, color_text,
        color_shadow, color_hotkey);
    write_char ('|', color_banner, color_hotkey);
    write_char ('|', color_shadow, color_banner);
    write_char ('|', color_back, color_hotkey);
    goto_edge_below ();}

/* Bottom of box: */
static void bottom (void) {goto_start ();
    write_char ('+', color_banner, color_hotkey);
    write_static_line ('.', color_banner, color_hotkey,
        color_banner, color_hotkey);
    write_char ('+', color_banner, color_hotkey);
    write_char ('|', color_shadow, color_banner);
    write_char ('|', color_back, color_hotkey);
    goto_edge_below ();}

/* Shadow of box: */
static void shadow (void) {goto_start ();
    write_char ('\|', color_back, color_hotkey);
    write_char ('_', color_shadow, color_hotkey);
    write_char ('_', color_shadow, color_hotkey);
    write_char ('_', color_shadow, color_hotkey);
    write_aftertext ('_', color_shadow, color_hotkey);
    write_char ('\|', color_shadow, color_hotkey);
    write_char ('|', color_back, color_hotkey);
    goto_edge_below ();}

```

```

/* Main program: ----- */
}

/* Analyze mode and draw background or line: */
static void draw_screen (void) {
    /* strcmp returns 0 if strings are equal: */
    if (!strcmp (mode, "-banner")) banner ();
    else if (!strcmp (mode, "-clear")) clear ();
    #ifdef CONIO
    else if (!strcmp (mode, "-sleep")) sleep_ ();
    #endif
    else if (!strcmp (mode, "-background")) background ();
    else if (!strcmp (mode, "-time")) measure_time ();
    else if (!strcmp (mode, "-random")) random_back ();
    else if (!strcmp (mode, "-top")) top ();
    else if (!strcmp (mode, "-item")) item ();
    else if (!strcmp (mode, "-question")) question ();
    else if (!strcmp (mode, "-bottom")) bottom ();
    else if (!strcmp (mode, "-shadow")) shadow ();
    else if (!strcmp (mode, "-stripe")) stripe ();
    else
    #define msg1 "This is REPLY Version"
    #define msg2 "the tiny shell script user"
    #define msg3 "interface for DOS and UNIX"
    #define msg4 "written by G. D. Brettschneider. Refer"
    #define msg5 "to Gerolf Markup Shredder"
    #define msg6 "for examples of usage. -- www.Gerolf.org"
    #ifdef CONIO
    printf ("\r\n %s %s (%s), %s\r\n %s, %s\r\n %s %s\r\n\r\n",
        msg1, GMSversionREPLY, GMSdateREPLY, msg2, msg3, msg4, msg5, msg6);
    #else
    printf ("\n %s %s (%s), %s\n %s, %s\n %s %s\n\n",
        msg1, GMSversionREPLY, GMSdateREPLY, msg2, msg3, msg4, msg5, msg6);
    #endif
    #undef msg1
    #undef msg2
    #undef msg3
    #undef msg4

/* Get user input: */
static void read_keyboard (void) {
    char d = '\0';
    item_next = getch ();
    if (item_next == 0) { /* It's a special sign, but which one? */
        d = getch ();
        if (d == 'H') move_up ();
        else if (d == 'P') move_down ();}
    else if ((item_next == '\t') // Tabulator
    || (item_next == '\r')) // Enter
        item_next = (char) item_number;}

/* Execute: */
int main (int argc, char *argv []) {
    get_parameters (argc, argv);
    get_environment ();
    draw_screen ();
    /* Get user input: */
    if (item_selected) {
        if (!item_question) {
            #ifdef CONIO
            _setcursortype (_NOCURSOR);
            #endif
            read_keyboard ();
            #ifdef CONIO
            _setcursortype (_NORMALCURSOR);
            #endif
        }
        write_user_input ();}
    if (item_measurement) { write_user_input ();}
    normvideo (); return (error);}

/* end */
}

```

mk_reply

```
#!/bin/sh

# mk_reply
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider(1999-2006). All rights reserved.
# Send corrections to: MarkupShredder(at)Gerolf.de (www.Gerolf.de)

GMSdateMK_REPLY=20061224

# =====

# This script just helps to compile 'reply.c' (expected to be placed in
# '$GMS_ROOT/doc/reply), to view error messages, and to move 'reply' to
# the binary folder ('$GMS_ROOT/bin/linux'):

# Start shell:
if [ "$1" = "" ]; then
    xterm -fg black -bg white -e "$0" -shell
    exit 1
fi

# Compile reply.c, if present:
replydir=$(dirname "$0")
cd "$replydir"
if [ -f "$replydir/reply.c" ]; then
    echo; echo "Compiling 'reply'..."
    gcc -o reply reply.c -lstring 2> reply.log
else
    echo; echo "Could not find '$replydir/reply.c!'"
    sleep 1
    exit 2
fi

# Check for success:
if [ -f "$replydir/reply" ]; then
    echo; echo "Compilation successful."
else
    echo; echo "Compilation failed."; echo
    less < "$replydir/reply.log"
    sleep 1
    exit 3
fi

# Move 'reply' to ../bin/linux, if that folder is present:
OLDDIR=$PWD
if [ -d ../bin/linux ]; then
    cd ../bin/linux
    NEWDIR=$PWD
    cd "$OLDDIR"
    echo; echo "Moving 'reply' to $NEWDIR ..."
    mv "$replydir/reply" "$NEWDIR"
    echo; echo "Done."
    NEWDIR=
    sleep 1
fi
OLDDIR=
```


[GMS_ROOT]/etc

Shell Scripts (Linux)

folder

```
#!/bin/sh

# folder
# =====

# Folder structure file for Gerolf Markup Shredder

# Entries with a leading hash mark are disabled for the command line and text
# mode interfaces of GMS only. The browser interface parses this file, it does
# not execute it as a shell script. So the "commented out" entries must be set
# correctly for the browser interface, and it is not allowed to use variables
# other than GMS_ROOT and SRV_ROOT in it, if the directory names are not given
# absolutely (see setting.pm). Other variables are not recognized by the
# browser interface parser.

# =====

# 1) GMS directories:

# Server subdirectories:
# export GMS_ROOT_SHORT=
# export GMS_ROOT=

# Shell scripts (Unix):
export GMS_SHELL="$GMS_ROOT/shell"

# Temporary files:
export GMS_TEMP="/tmp/gms"

# Template markup files:
export GMS_TEMPLATE="$GMS_ROOT/doc"

# Encoding:
# export SRV_ENCODING=$SRV_ROOT/data/enc

# Handbook:
# export SRV_HANDBOOK=$SRV_ROOT/doc/handbook

# Language strings:
# export GMS_LANGUAGE=$GMS_ROOT/data/lang

# Process (must be writable for any user):
# export GMS_PROCESS=$GMS_ROOT/tmp
# export SRV_PROCESS=$SRV_ROOT/tmp

# Rotation of banners:
# export GMS_ROTATION=$GMS_ROOT/doc/rotation

# Setting:
# export SRV_SETTING=$SRV_ROOT/etc

# Style:
# export SRV_STYLE=$SRV_ROOT/data/css

# 2) Directories for TeX engine configuration: -----

# : path separator
# / directory separator

# // subdirectory inclusion

# TeX Main Folder:
export TEXMF="$GMS_ROOT"

# Configuration files texmf.cnf and pdftex.cfg:
export TEXMFCNF="$GMS_ROOT/etc"

# Binaries:
export GMS_BINARIES="$GMS_ROOT/bin/linux"

# Directory to place .log files, if current directory is read-only:
export TEXMFOUTPUT="$GMS_ROOT/tmp"

# Dump files (.fmt/.efmt) for virtex:
export TEXFORMATS="$GMS_BINARIES"

# String pools (.pool) for initex:
export TEXPOOL="$GMS_BINARIES"

# 3) TeX input file search: -----

# This must be one single directory:
export TEX_USER="$GMS_ROOT/doc"

# This is needed only during initex run (single directory):
export TEX_BASE="$GMS_ROOT/tex"

# Encodings (.enc), PostScript fonts, font maps):
export TEXPSHEADERS="$GMS_ROOT/etc":"$GMS_ROOT/data/enc":"$GMS_ROOT/fonts/"

# 4) TeX font search: -----

# This is needed only during initex run (single directory):
export GMS_FONTS="$GMS_ROOT/fonts"

# TeX font metric files:
export TEXFONTS="$GMS_FONTS/tfm/"

# TeX virtual fonts:
export VPFONTS="$GMS_FONTS/vf/"

# PostScript Adobe metric files:
export AFMFONTS="$GMS_FONTS/afm/"

# PostScript Type 1 outline fonts:
export T1FONTS="$GMS_FONTS/type1/"

# TrueType outline fonts:
export TTFONTS="$GMS_FONTS/ttf/"

# 5) Contribute to gmsdebug.log: -----

if [ "$GMS_DEBUG" = "Z" -a "$GMS_MODE" != "quiet" ]; then
  echo " folder ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi
```

gmssetup

```
#!/bin/sh

# gmssetup
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2008). All rights reserved.
# Send corrections to: MarkupShredder(at)Gerolf.org (www.Gerolf.org)

export GMS_DATE=20080107
export GMS_VERSION="0.08a"

# =====

# Initialize debugging mode, main codepage, linking mode:

export GMS_DEBUG="0"
export GMS_CODEPAGE="CP1252"
export GMS_LINKS="internal"

# Welcome message: -----

if [ "$1" != "-shell" -a "$2" != "-shell" ]; then
    xterm -fg black -bg white -e "$0" -shell "$1"
    exit
fi
shift

echo
setterm -clear -reset -cursor off
left=" Running:      GMS setup ..."
if [ "$1" != "" ]; then
    left=" Running:      GMS quick setup ..."; fi
right="
    Gerolf Markup Shredder $GMS_VERSION"
if [ "$1" != "" ]; then
    right="
    Gerolf Markup Shredder $GMS_VERSION"; fi
arg="/////////////////////////////////////"
echo "$left$right"
echo " $arg$arg"
echo

# Root user check: -----

if [ "$USER" != "root" ]; then
    echo "      This application wants to create startup links in "
    echo "      '\usr\bin' directory."
    echo
    echo "      You may have to log in as user 'root' to run "
    echo "      GMS setup."
    echo
    echo -n "      Press [Enter] to continue or $quit"
    read
    setterm -reset
    echo "$left$right"
    echo " $arg$arg"
    echo
fi

unset left
unset right
unset arg

# Get folder location: =====

# Get GMS_SETTING (current directory):

# Check if current directory differs from script path:
cd .. ###

test=$(pwd) ###
cd - > "$test/nil" ###
if [ -f "$test/nil" ]; then rm "$test/nil"; fi ###
if [ ! -f "$PWD/gmssetup" ]; then
    if [ -f "$0" ]; then export PWD=$(dirname "$0"); fi; fi
test=
export GMS_SETTING=$PWD

# Debug: -----

# Called scripts may contribute to gmsdebug.log.
# Renaming to gmssetup.log takes place at the end of this gmssetup script.
export Z="gmsdebug.log"
# Remove old log files in curent folder:
if [ -f gmssetup.log ]; then rm gmssetup.log; fi
if [ -f gmsdebug.log ]; then rm gmsdebug.log; fi
# gmssetup log file header:
echo " gmssetup.log" > $Z
arg="/////////////////////////////////////"
echo " $arg$arg" >> $Z
unset arg
echo " This is the setup log file for Gerolf Markup Shredder." >> $Z
echo " To write the run time debug log file, set GMS_DEBUG=Z." >> $Z
echo " Format: called script (parameter 1) ... (parameter 9)," >> $Z
echo " followed by values of important environment variables." >> $Z
echo " 'shredder': reads user input and calls script modules." >> $Z
echo >> $Z
# gmssetup parameters and environment variables:
echo " $0 ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> $Z
echo "   GMS_VERSION=$GMS_VERSION" >> $Z
echo "   GMS_CODEPAGE=$GMS_CODEPAGE" >> $Z
echo "   GMS_SETTING=$GMS_SETTING" >> $Z
export Z="$GMS_SETTING/gmsdebug.log"

# Go up one level and call it GMS_ROOT: -----

cd "$PWD"
cd ..
export PWD=$(pwd)
echo " Defining:      Installation directories ..."
echo
export GMS_ROOT=$PWD
echo " GMS_ROOT:      $PWD"
echo "   GMS_ROOT=$PWD" >> "$Z"
echo

# Load folders and check if there's a folder for temorary files: -----

if [ -d "$GMS_SETTING" ]; then cd "$GMS_SETTING"; fi
source "$GMS_SETTING/folder" called by gmssetup
if [ ! -d "$GMS_TEMP" ]; then mkdir "$GMS_TEMP"; fi

# Setting variables: =====

# Browser interface variables:

# Maximal size and number of files:
export GMS_MAXSIZE=1234567
export GMS_MAXFILES=123
# Width of text area (join/split):
export GMS_WIDTH_ONE=60
export GMS_WIDTH_TWO=40
# Height of text area (join/split):
export GMS_HEIGHT_ONE=30
export GMS_HEIGHT_TWO=40
# Font size, in point (join/split):
export GMS_SIZE_ONE=10
export GMS_SIZE_TWO=8
```

```

if [ ! -d "$GMS_ROOT/tmp" ]; then mkdir "$GMS_ROOT/tmp"; fi
if [ -d "$GMS_ROOT/tmp" ]; then chmod 777 "$GMS_ROOT/tmp"; fi ###

# Textmode interface variables: -----

# Initialize colors & pattern (R: set by random) and animation mode:

# Foreground colors:
export GMS_TEXT=R
export GMS_HOTKEY=R
export GMS_PATTERN=R
# Background colors:
export GMS_BANNER=R
export GMS_SHADE=R
export GMS_DESKTOP=R
# Fill character:
export GMS_LETTER=R
# Menu animation:
export GMS_ANIMATE="on"

# Programs: -----

export GMS_VIEWER=less
export GMS_EDITOR=vi
if [ -f /usr/bin/gedit ]; then export GMS_EDITOR=gedit; fi
if [ -f /usr/bin/kedit ]; then export GMS_EDITOR=kedit; fi
if [ -f /usr/bin/emacs ]; then export GMS_EDITOR=emacs; fi
if [ -f /usr/bin/xemacs ]; then export GMS_EDITOR=xemacs; fi
if [ -f /usr/bin/quanta ]; then export GMS_EDITOR=quanta; fi
if [ -f /opt/kde/bin/kwrite ]; then export GMS_EDITOR=kwrite; fi # SuSE
if [ -f /opt/kde2/bin/kwrite ]; then export GMS_EDITOR=kwrite; fi # "
if [ -f /opt/kde3/bin/kwrite ]; then export GMS_EDITOR=kwrite; fi # "
if [ -f /usr/bin/kwrite ]; then export GMS_EDITOR=kwrite; fi
if [ -f /usr/bin/nedit ]; then export GMS_EDITOR=nedit; fi
export GMS_BROWSER=lynx
if [ -f /usr/bin/links ]; then export GMS_BROWSER=links; fi
if [ -f /usr/bin/opera ]; then export GMS_BROWSER=opera; fi
if [ -f /opt/kde/bin/konqueror ]; then export GMS_BROWSER=konqueror; fi # SuSE
if [ -f /opt/kde2/bin/konqueror ]; then export GMS_BROWSER=konqueror; fi # "
if [ -f /opt/kde3/bin/konqueror ]; then export GMS_BROWSER=konqueror; fi # "
if [ -f /usr/bin/konqueror ]; then export GMS_BROWSER=konqueror; fi
if [ -f /usr/bin/netscape ]; then export GMS_BROWSER=netscape; fi
if [ -f /usr/bin/mozilla ]; then export GMS_BROWSER=mozilla; fi
export GMS_ANALYST=tidy
export GMS_TSETPER=pdfetex
export GMS_READER=acroread
if [ -f /usr/bin/gv ]; then export GMS_READER=gv; fi
if [ -f /usr/bin/ghostview ]; then export GMS_READER=ghostview; fi
if [ -f /usr/bin/xpdf ]; then export GMS_READER=xpdf; fi
if [ -f /usr/bin/kghostview ]; then export GMS_READER=kghostview; fi
if [ -f /usr/bin/kpdf ]; then export GMS_READER=kpdf; fi
if [ -f /usr/bin/acroread ]; then export GMS_READER=acroread; fi

# Set script and binary permissions: =====

for i in g_color g_file g_font g_good g_launch g_list g_menu g_palet g_plug \
g_prog g_rain g_save g_select g_vars g_wel gms l_banner l_box l_code \
l_color l_desk l_file l_gms l_good l_list l_menu l_prog l_rain l_save \
l_select l_wel shredder;
do chmod 755 "$GMS_SHELL/$i"; done

for i in afm2tfm pdfetex reply tidy ttf2afm vptovf;
do chmod 755 "$GMS_BINARIES/$i"; done

chmod 755 "$GMS_ROOT/doc/reply/mk_reply"

# Launcher scrip: =====

# Create a new 'gerolf' script:

```

```

echo " Building:      GMS launcher script ..."
echo "                $GMS_SETTING/gerolf"
echo
source "$GMS_SHELL/g_launch" -build called by gmssetup
if [ ! -f "gerolf" ]; then
    echo "                Error: Could not build launcher script."
    exit
fi

# Show lower desktop border: -----

export REPLY_BANNER=15; export REPLY_TEXT=0
export REPLY_OFFSET=0; export REPLY_SIZE=80
export arg=////////////////////////////////////
"$GMS_BINARIES/reply" -banner 23 "$arg$arg"
unset arg
"$GMS_BINARIES/reply" -banner 24 "Running under $OSTYPE ..."
export REPLY_OFFSET=79; export REPLY_SIZE=3
"$GMS_BINARIES/reply" -banner 11 " "
echo

# Create a link from 'gerolf' and 'gms' to /usr/bin: -----

echo " Creating:      Links from 'gerolf' and 'gms' to ..."
echo "                /usr/bin"
if [ -d "/usr/bin" ]; then
    if [ -L "/usr/bin/gms" ]; then rm "/usr/bin/gms"; fi
    ln -s "$GMS_SHELL/gms" "/usr/bin/gms"
    if [ -L "/usr/bin/gerolf" ]; then rm "/usr/bin/gerolf"; fi
    ln -s "$GMS_SETTING/gerolf" "/usr/bin/gerolf"
fi

# Check if links have been created: -----

if [ ! -L "/usr/bin/gms" ]; then
    echo
    echo "                Could not link 'gms' script to"
    echo "                '/usr/bin' directory."
fi
if [ ! -L "/usr/bin/gerolf" ]; then
    echo
    echo "                Could not link 'gerolf' script to"
    echo "                '/usr/bin' directory."
fi

# Create KDE menu directories: -----

function KDE_menu {
    GMS_KDE=$1
    if [ -d "$GMS_KDE" ]; then
        if [ -d "$GMS_KDE/share/applnk" ]; then
            GMS_KDE="$GMS_KDE/share/applnk/Office"
            if [ ! -d "$GMS_KDE" ]; then mkdir "$GMS_KDE"; fi
            GMS_KDE="$GMS_KDE/Markup Shredder"
            if [ ! -d "$GMS_KDE" ]; then mkdir "$GMS_KDE"; fi
        fi
        GMS_LINK="$GMS_KDE/gerolf.kdelnk"
        echo "# KDE Config File" > "$GMS_LINK"
        echo "[KDE Desktop Entry]" >> "$GMS_LINK"
        echo "Exec=\"$GMS_ROOT/gerolf\" >> \"$GMS_LINK"
        echo "Icon=run" >> "$GMS_LINK"
        echo "Name=gerolf" >> "$GMS_LINK"
        echo "Terminal=true" >> "$GMS_LINK"
        echo "Type=Application" >> "$GMS_LINK"
        chmod 755 "$GMS_LINK"
        GMS_LINK="$GMS_KDE/gmssetup.kdelnk"
        echo "# KDE Config File" > "$GMS_LINK"
        echo "[KDE Desktop Entry]" >> "$GMS_LINK"
        echo "Exec=\"$GMS_ROOT/gmssetup\" >> \"$GMS_LINK"
    fi
}

```

```

    echo "Icon=run" >> "$GMS_LINK"
    echo "Name=gmssetup" >> "$GMS_LINK"
    echo "Terminal=true" >> "$GMS_LINK"
    echo "Type=Application" >> "$GMS_LINK"
    chmod 755 "$GMS_LINK"
fi
GMS_KDE=
}

# Get KDE directory:

if [ -d "/var/lib/menu/kde" ]; then # Aurox Linux 9x
    KDE_menu "/var/lib/menu/kde"; fi
if [ -d "/opt/kde3" ]; then # SuSE Linux 9x
    KDE_menu "/opt/kde3"; fi
if [ -d "/opt/kde2" ]; then # SuSE Linux 8x
    KDE_menu "/opt/kde2"; fi
if [ -d "/opt/kde" ]; then # SuSE Linux 7x
    KDE_menu "/opt/kde"; fi
if [ -d "$KDEDIR" ]; then
    KDE_menu "$KDEDIR"; fi

# Create [GMS_ROOT] level link to gerolf: -----
GMS_LINK="$GMS_ROOT/gerolf"
GMS_GEROLF="$GMS_ROOT/etc/gerolf"
echo "#!/bin/sh" > "$GMS_LINK"
echo "\$GMS_GEROLF\" \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9" >> "$GMS_LINK"
chmod 755 "$GMS_LINK"

GMS_LINK="$HOME/Desktop/gerolf"
GMS_GEROLF="$GMS_ROOT/etc/gerolf"
echo "#!/bin/sh" > "$GMS_LINK"
echo "\$GMS_GEROLF\" \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9" >> "$GMS_LINK"
chmod 755 "$GMS_LINK"

# Create [GMS_ROOT] level link to gmssetup: -----
GMS_LINK="$GMS_ROOT/gmssetup"
echo "#!/bin/sh" > "$GMS_LINK"
echo >> "$GMS_LINK"
echo " cd \"\$GMS_ROOT/etc\" >> \"$GMS_LINK"
echo "\$GMS_ROOT/etc/gmssetup\" \$1" >> "$GMS_LINK"

GMS_LINK="$HOME/Desktop/gmssetup"
echo "#!/bin/sh" > "$GMS_LINK"
echo >> "$GMS_LINK"
echo " cd \"\$GMS_ROOT/etc\" >> \"$GMS_LINK"
echo "\$GMS_ROOT/etc/gmssetup\" \$1" >> "$GMS_LINK"
chmod 755 "$GMS_LINK"

GMS_GEROLF=
GMS_LINK=

# Font map and format creation: =====

# Get Windows fonts: -----

if [ ! -d "$GMS_ROOT/fonts" ]; then
    mkdir "$GMS_ROOT/fonts"; fi
    chmod 755 "$GMS_ROOT/fonts"
if [ ! -d "$GMS_ROOT/fonts/ttf" ]; then
    mkdir "$GMS_ROOT/fonts/ttf"; fi
    chmod 755 "$GMS_ROOT/fonts/ttf"

echo
echo
echo \
"
    Please enter the path to your Windows True Type fonts here,"
echo \

"
    if this is a dual boot system, e.g. \"/hda1/WINDOWS/Fonts\"."
echo
echo -n " Your answer: "
setterm -cursor on
    read winfonts
setterm -cursor off

if [ -d "$winfonts" ]; then cd "$winfonts"
echo " Copying:      $winfonts ..." ###
for j in *.ttf; do
    if [ -f $j ]; then
        cp $j "$GMS_ROOT/fonts/ttf"
        chmod 755 "$GMS_ROOT/fonts/ttf/$j"
    fi
done
if [ -d "$GMS_ROOT/fonts/ttf" ]; then
    cd "$GMS_ROOT/fonts/ttf"
    for j in u*.ttf; do
        if [ -f $j ]; then rm $j; fi; done ###
    fi
else
    if [ "$winfonts" = "" ]; then winfonts="True type font directory." ###
        echo "          Missing path to standard Windows fonts." ###
    else echo "
          $winfonts"; fi ###
    echo -n ###
    echo "          Press [Ctrl+C] to break and restart gmssetup ..." ###
    sleep 2s ###
fi
winfonts=

# Write font map:

if [ "$1" != "-quick" ]; then
    source "$GMS_SHELL/gms" -w -quiet called_by gmssetup
    if [ ! -f "$GMS_SETTING/font.map" ]; then
        echo " Error:      Could not build font map."
    fi
fi

# Initialize format: -----

export REPLY_OFFSET=0; export REPLY_SIZE=17
"$GMS_BINARIES/reply" -banner 18 Initializing:
export REPLY_OFFSET=14; export REPLY_SIZE=68
"$GMS_BINARIES/reply" \
    -banner 18 "TeX format file (this may take some time) ..."
source "$GMS_SHELL/gms" -i -quiet called_by gmssetup
export REPLY_BANNER=15; export REPLY_TEXT=0
export REPLY_OFFSET=14; export REPLY_SIZE=68
"$GMS_BINARIES/reply" -banner 18 "$GMS_BINARIES/gerolf.(e)fmt"

if [ ! -f "$GMS_BINARIES/gerolf.efmt" -a \
! -f "$GMS_BINARIES/gerolf.fmt" ]; then
    echo " Error:      Could not build TeX format file."
fi

# Write memo (load default.htm): -----

export GMS_FOLDER="$GMS_TEMPLATE/default"
export GMS_FILE=default.htm
source "$GMS_SHELL/g_vars" -write_memo called_by gmssetup

# Epilogue: =====

echo; echo
echo " Startup call: gerolf ... textmode interface (random colors)"
echo "          gms ... command line interface (black/white)"
echo
echo -n " ///////////////////////////////////////"
echo " ///////////////////////////////////////"

```

```

# Draw startup/break info on screen:
echo -n " Done:          GMS setup."
echo -n "          Press [Enter] to start or [Ctrl+C] to quit"
# Set break cursor at a nice place:
export REPLY_OFFSET=13; export REPLY_SIZE=0
"$GMS_BINARIES/reply" -banner 22
# Clean up environment:
source "$GMS_SHELL/g_vars" -clear called_by gmssetup
# Write gmsdebug.log footer:
arg="/////////////////////////////////////"
echo " $arg$arg" >> "$Z"
echo -n " $GMS_SETTING" >> "$Z"
unset arg
# Rename gmsdebug.log to gmssetup.log:

if [ -f "$Z" ]; then mv "$Z" "$GMS_SETTING/gmssetup.log"; fi
# Final unsets:
export Z=
export GMS_DEBUG=
export GMS_ROOT=
export GMS_SETTING=
export backup_base=
export backup_file=
export backup_folder=
# Allow user interruption or start textmode interface:
setterm -cursor on
read
gerolf called_by gmssetup

```

Batch Files (Dos, Windows)

folder.bat

```

REM folder.bat
REM =====

REM Folder structure file for Gerolf Markup Shredder (Dos/Windows)

REM Entries with a leading "REM" mark are disabled for the command line and
REM text mode interfaces of GMS only. The browser interface parses this file,
REM it does not execute it as a batch script. So the "commented out" entries
REM must be set correctly for the browser interface, and it is not allowed to
REM use variables other than GMS_ROOT and SRV_ROOT in it, if the directory
REM names are not given absolutely (see "setting.pm"). Other variables are not
REM recognized by the browser interface parser.

REM =====

REM CAUTION: NO WHITESPACE is allowed to follow values assigned to variables!

REM 1) GMS directories:

REM Server subdirectories:
REM set GMS_ROOT_SHORT=
REM set SRV_ROOT=

REM Shell scripts (Dos):
set GMS_BATCH=%GMS_ROOT%\batch

REM Temporary files:
set GMS_TEMP=%GMS_DRIVE%\tmp
REM If the temporary files folder is not on top level of the tree, you may
REM have to create it manually before running gmssetup.
REM Temporary files can be written to a RAM drive on Dos:
REM set GMS_TEMP=R:
REM In order to do that, an entry like this is required in c:\config.sys:
REM device=c:\dos\ramdrive.sys 16

REM Template markup files:
set GMS_TEMPLATE=%GMS_ROOT%\doc
REM Fixme: Should be possible to move the template folder to %USERPROFILE%

REM Encoding:
REM set SRV_ENCODING=%SRV_ROOT%\data\enc

REM Handbook:
REM set SRV_HANDBOOK=%SRV_ROOT%\doc\handbook

REM Language strings:
REM set GMS_LANGUAGE=%GMS_ROOT%\data\lang

REM Process (must be writable for any user):
REM set GMS_PROCESS=%GMS_ROOT%\tmp
REM set SRV_PROCESS=%SRV_ROOT%\tmp

REM Rotation of banners:
REM set GMS_ROTATION=%GMS_ROOT%\doc\rotation

REM Setting:
REM set SRV_SETTING=%SRV_ROOT%\etc

REM Style:
REM set SRV_STYLE=%SRV_ROOT%\data\css

REM 2) Directories for TeX engine configuration: -----
REM ; path separator

REM \ directory separator
REM \\ subdirectory inclusion

REM TeX Main Folder:
set TEXMF=%GMS_ROOT%

REM Configuration files texmf.cnf and pdftex.cfg:
set TEXMFCNF=%GMS_ROOT%\etc

REM Binaries:
if "%OS%" == "" set GMS_BINARIES=dos
if "%windir%" == "" set GMS_BINARIES=dos
if not "%OS%" == "" if not "%windir%" == "" set GMS_BINARIES=win
if "%GMS_DOSMODE%" == "1" set GMS_BINARIES=dos
set GMS_BINARIES=%GMS_ROOT%\bin\%GMS_BINARIES%

REM Directory to place .log files, if current directory is read-only:
set TEXMFOUTPUT=%GMS_ROOT%\tmp

REM Dump files (.fmt/.efmt on Windows, .fmt/.efm on Dos) for virtex:
set TEXFORMATS=%GMS_BINARIES%

REM String pools (.pool on Windows, .poo on Dos) for initex:
set TEXPOOL=%GMS_BINARIES%

REM 3) TeX input file search:-----
REM This must be one single directory:
set TEX_USER=%GMS_ROOT%\doc

REM This is needed only during initex run (single directory):
set TEX_BASE=%GMS_ROOT%\tex

REM Encodings (.enc), PostScript fonts, font maps):
set TEXPSHEADERS=%GMS_ROOT%\etc
set TEXPSHEADERS=%TEXPSHEADERS%;%GMS_ROOT%\data\enc
set TEXPSHEADERS=%TEXPSHEADERS%;%GMS_ROOT%\fonts\\

REM 4) TeX font search: -----
REM This is needed only during initex run (single directory):
set GMS_FONTS=%GMS_ROOT%\fonts

REM TeX font metric files:
set TEXFONTS=%GMS_FONTS%\tfm\\

REM TeX virtual fonts:
set VFFONTS=%GMS_FONTS%\vf\\

REM PostScript Adobe metric files:
set AFM_FONTS=%GMS_FONTS%\afm\\

REM PostScript Type 1 outline fonts:
set T1FONTS=%GMS_FONTS%\type1\\

REM TrueType outline fonts:
set TTFONTS=%GMS_FONTS%\ttf\\

REM 5) Contribute to gmsdebug.log: -----
if "%OS%" == "" goto exit
if not %Z% == "" echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8)>> %Z%
:exit

```

gerolf.bat

```
@echo off

REM gerolf.bat
REM =====

REM Launcher script for Gerolf Markup Shredder (Dos).

REM Web: www.Gerolf.org
REM eMail: MarkupShredder@Gerolf.org

REM #####

REM 1) General settings: -----

REM Setting directory:
set GMS_SETTING=F:\wamp\www\etc
if (%2) == (-noclearscreen) goto fi_fdg
if (%GMS_MODE%) == (quiet) goto fi_fdg
mode con lines=25
:fi_fdg
if not (%1) == (-passive) type "%GMS_SETTING%\launch_1.scn"
REM Main directories (can be web server subfolders):
set GMS_ROOT=F:\wamp\www
set GMS_DRIVE=F:
REM Debugging mode (0, 1, 2, 3, ... or X, Y, Z. Not empty!):
set GMS_DEBUG=0
REM Rewrite gmsdebug log file:
set Z="%GMS_SETTING%\gmsdebug.log"
if "%4" == "gmssetup" goto fi_debug
if exist %Z% del %Z%
set gmshead="%GMS_SETTING%\_gmshead.txt"
if exist %gmshead% ren %gmshead% gmsdebug.log
set gmshead=
goto fi_debug
:fi_debug
REM Main codepage name:
set GMS_CODEPAGE=CP1252
REM Main codepage number (for chcp-command, if %OS% defined):
set GMS_CHCP=1252
REM Initial codepage number (before running gmssetup):
set GMS_INICP=850

REM 2) Programs in search path: -----

REM Plain text viewer and editor:
set GMS_VIEWER=viewer
set GMS_EDITOR=notepad
REM HTML browser, syntax checker, typesetting engine:
set GMS_BROWSER=default
set GMS_ANALYST=tidy
set GMS_TSETTER=pdfetex
REM PDF reader:
set GMS_READER=default

REM 3) Textmode interface variables: -----

REM Menu animation ('on' or 'off'):
set GMS_ANIMATE=on
REM Colors and pattern ('R'== 'Random'):
REM Foreground colors (0 to 15):
set GMS_TEXT=R
set GMS_HOTKEY=R
set GMS_PATTERN=R
REM Initial values:
set REPLY_TEXT=15
set REPLY_HOTKEY=14

set REPLY_PATTERN=13
REM Background colors (0 to 7):
set GMS_BANNER=R
set GMS_SHADE=R
set GMS_DESKTOP=R
REM Initial values:
set REPLY_BANNER=1
set REPLY_SHADE=2
set REPLY_DESKTOP=3
REM Fill character (ASCII code 32 to 126):
set GMS_LETTER=R
set REPLY_LETTER=123

REM 4) Browser interface variables:-----

REM Maximal size and number of files:
set GMS_MAXSIZE=1234567
set GMS_MAXFILES=123
REM Width of text area (join/split):
set GMS_WIDTH_ONE=60
set GMS_WIDTH_TWO=40
REM Height of text area (join/split):
set GMS_HEIGHT_ONE=30
set GMS_HEIGHT_TWO=40
REM Font size, in point (join/split):
set GMS_SIZE_ONE=10
set GMS_SIZE_TWO=8
REM Linking to internal or external target:
set GMS_LINKS=internal

REM 5) Start program: -----

REM Do not change the rest of this file:

REM Set version number and date:
set GMS_VERSION=0.05a
set GMS_DATE=20060224
REM System variables:
set OS=Windows_XP
set any=*. *
if not "%GMS_MODE%" == "quiet" title GMS 0.05a
REM Get folder structure:
call "%GMS_SETTING%\folder" called_by gerolf
REM If necessary, backup prompt and path:
if not "%GMS_PROMPT%" == "" goto fi_pmt
set GMS_PROMPT=%PROMPT%
set PROMPT= $P$G
:fi_pmt
if not "%GMS_PATH%" == "" goto fi_pth
set GMS_PATH=%PATH%
set PATH=%GMS_BATCH%;%GMS_BINARIES%;%PATH%
:fi_pth
cls
if not (%1) == (-passive) type "%GMS_SETTING%\launch_2.scn"
REM Get work file:
if "%GMS_MODE%" == "quiet" goto fi_ini
call g_vars -ini_file called_by gerolf %1 %2 %3 %4 %5 %6 %7 %8 %9
:fi_ini
cls
if not (%1) == (-passive) type "%GMS_SETTING%\launch_3.scn"
REM Launch textmode interface:
if not (%1) == (-passive) call shredder called_by gerolf
:exit

REM End of GMS launcher script -----
```


gms_memo.bat

```
REM gms_memo.bat  
REM =====
```

```
set GMS_REMODRV=F:
```

```
set GMS_FOLDER=F:\wamp\www  
set GMS_FILE=gerolf.php
```

gmssetup.bat

```
echo off

REM gmssetup.bat
REM =====

REM This file is part of Gerolf Markup Shredder,
REM written by G. D. Brettschneider (1999-2008). All rights reserved.
REM Send corrections to: MarkupShredder(at)Gerolf.org (www.Gerolf.org)

set GMS_DATE=20080107
set GMS_VERSION=0.08a

REM =====

REM Initialize debugging mode, main codepage, linking mode:

set GMS_DEBUG=0
set GMS_CODEPAGE=CP1252
set GMS_LINKS=internal

REM Start shell: -----

cls
echo Running ...
if "%1" == "-shell" goto fi_sh
if "%2" == "-shell" goto fi_sh
if "%comspec%" == "" echo Please set COMSPEC variable. Press any key ...
if "%comspec%" == "" pause > nul
if "%comspec%" == "" goto fi_sh
%comspec% /E:4096 /C %0 -shell %1
set GMS_VERSION=
set GMS_DEBUG=
set GMS_CODEPAGE=
set GMS_LINKS=
goto tixe
:fi_sh
shift

REM Check operating system: -----

REM Assume support for long filenames if environment variable OS is defined:
set any=*.
if "%OS%" == "" set any=nul

REM FreeDOS:
if not "%OS%" == "" goto fi_fd
if exist C:\kernel.sys if not exist C:\msdos.sys set GMS_FreeDOS=FreeDOS
:fi_fd

REM DRDOS:
if "%OS%" == "DRDOS" set GMS_DRDOS=6
if not "%DRDOSCFG%" == "" set GMS_DRDOS=7
if "%OS%" == "DRDOS" set OS=
if not "%GMS_DRDOS%" == "" goto fi_os

REM Windows 9x:
if "%OS%" == "" if not "%winbootdir%" == "" set OS=Windows_9x
if "%OS%" == "Windows_9x" if "%winbootdir%" == "." set OS=
if "%OS%" == "Windows_9x" set any=nul
if "%windir%" == "" if not "%winbootdir%" == "" set GMS_DOSMODE=1
if "%windir%" == "" if "%winbootdir%" == "." set GMS_DOSMODE=
if not "%GMS_DOSMODE%" == "" set windir=%winbootdir%
if exist "%windir%\Desktop\%any%" set GMS_LINKDIR=%windir%\Desktop
if exist "%windir%\SendTo\%any%" set GMS_SENDDIR=%windir%\SendTo
if not exist "%windir%\STARTM-1\%any%" goto fi_no9xA
set GMS_MENUDIR=%windir%\STARTM-1
:fi_no9xA
if not exist "%windir%\STARTM-1\PROGRAMS\%any%" goto fi_no9xB

set GMS_MENUDIR=%windir%\STARTM-1\PROGRAMS
:fi_no9xB
if not exist "%windir%\STARTM-1\PROGRA-1\%any%" goto fi_no9xC
set GMS_MENUDIR=%windir%\STARTM-1\PROGRA-1
:fi_no9xC

REM Windows NT:
if "%USERPROFILE%" == "" goto fi_nontC
if not exist "%USERPROFILE%\Desktop\%any%" goto fi_nont0
set GMS_LINKDIR=%USERPROFILE%\Desktop
:fi_nont0
if not exist "%USERPROFILE%\SendTo\%any%" goto fi_nont1
set GMS_SENDDIR=%USERPROFILE%\SendTo
:fi_nont1
if not exist "%USERPROFILE%\STARTM-1\%any%" goto fi_nontA
set GMS_MENUDIR=%USERPROFILE%\STARTM-1
:fi_nontA
if not exist "%USERPROFILE%\STARTM-1\PROGRAMS\%any%" goto fi_nontB
set GMS_MENUDIR=%USERPROFILE%\STARTM-1\PROGRAMS
:fi_nontB
if not exist "%USERPROFILE%\STARTM-1\PROGRA-1\%any%" goto fi_nontC
set GMS_MENUDIR=%USERPROFILE%\STARTM-1\PROGRA-1
:fi_nontC

REM Windows XP:
if "%ALLUSERSPROFILE%" == "" goto fi_noxpC
if not exist "%ALLUSERSPROFILE%\Desktop\%any%" goto fi_noxp0
set GMS_LINKDIR=%ALLUSERSPROFILE%\Desktop
if "%OS%" == "Windows_NT" set OS=Windows_XP
:fi_noxp0
if not exist "%ALLUSERPROFILE%\STARTM-1\%any%" goto fi_noxpA
set GMS_MENUDIR=%ALLUSERPROFILE%\STARTM-1
:fi_noxpA
if not exist "%ALLUSERPROFILE%\STARTM-1\PROGRAMS\%any%" goto fi_noxpB
set GMS_MENUDIR=%ALLUSERPROFILE%\STARTM-1\PROGRAMS
:fi_noxpB
if not exist "%ALLUSERPROFILE%\STARTM-1\PROGRA-1\%any%" goto fi_noxpC
set GMS_MENUDIR=%ALLUSERPROFILE%\STARTM-1\PROGRA-1
:fi_noxpC
:fi_os

REM Windows 3x (set "WINDIR" to the content of "windir"):
set | find "windir" > }{.bat
echo set windir=%1> windir.bat
call }{
del }{.bat
del windir.bat

REM Welcome message: -----

if "%GMS_FreeDOS%" == "" mode con lines=25 > nul
if not "%GMS_FreeDOS%" == "" mode co80,25
cls
set left=Running: GMS setup ...
if not "%1" == "" set left=Running: GMS quick setup ...
if "%1" == "-no_shell" set left=Running: GMS setup ...
set right=Gerolf Markup Shredder %GMS_VERSION%
if "%1" == "" echo %left% %right%
if not "%1" == "" echo %left% %right%
set left=
set right=
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%arg%
set arg=
echo.
```

```

REM Assume support for long filenames if environment variable OS is defined:
REM if "%OS%" == "" if not "%winbootdir%" == "" set OS=Windows_9x
REM if "%OS%" == "Windows_9x" if "%winbootdir%" == "." set OS=

set hallo=www.ctan.org, www.tug.org, www.dante.de
if "%OS%" == "Windows_9x" goto fi_w91
if not "%OS%" == "" title Welcome to the TeX World - %hallo%> nul
:fi_w91
set hallo=

REM Windows 9x DOSmode message: -----
goto fi_dm
if "%GMS_DOSMODE%" == "" goto fi_dm
echo This application does not support DOS mode
echo. on %OS%/ME, sorry.
echo.
echo Please start the graphical user interface
echo by entering the 'win' command and run
echo [GMS_ROOT]\etc\gmssetup.bat
echo in a console window by clicking on it in
echo Windows Explorer.
echo.
set any=
set arg=
set GMSdateGMSUNZIP=
set GMS_CODEPAGE=
set GMS_DATE=
set GMS_DEBUG=
set GMS_DOSMODE=
set GMS_LINKDIR=
set GMS_MENUDIR=
set GMS_SENDDIR=
set GMS_VERSION=
set left=
set OS=
set right=
set windir=
goto tixe
:fi_dm

REM Get folder location: =====
REM Get GMS_SETTING (current directory):
REM pwd.sys just contains the command "set PWD=":
copy pwd.sys pwd_tmp.bat > nul
cd>> pwd_tmp.bat
if exist pwd_tmp.bat call pwd_tmp
if exist pwd_tmp.bat del pwd_tmp.bat
set GMS_SETTING=%PWD%

REM Debug: -----
REM Called batches may contribute to gmsdebug.log.
REM Renaming to gmssetup.log takes place at the end of this gmssetup script.
set Z=gmsdebug.log
if "%OS%" == "" goto fi_debug
REM Remove old log files in curent folder:
if exist gmssetup.log del gmssetup.log
if exist gmsdebug.log del gmsdebug.log
REM gmssetup log file header:
echo gmssetup.log> %Z%
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%%arg%>> %Z%
set arg=
echo This is the setup log file for Gerolf Markup Shredder.>> %Z%
echo To write the run time debug log file, set GMS_DEBUG=Z.>> %Z%
echo Format: called batch (parameter 1) ... (parameter 9),>> %Z%

echo followed by values of important environment variables.>> %Z%
echo shredder.bat reads user input and calls batch modules.>> %Z%
echo l_type modules only contribute if their compiled equi->> %Z%
echo valents are absent.>> %Z%
echo.>> %Z%
REM gmssetup parameters and environment variables:
echo %0 (%1) (%2) (%3) (%4) (%5) (%6) (%7) (%8) (%9)>> %Z%
echo GMS_VERSION=%GMS_VERSION%>> %Z%
echo GMS_CODEPAGE=%GMS_CODEPAGE%>> %Z%
echo GMS_SETTING=%GMS_SETTING%>> %Z%
:fi_debug
if not "%OS%" == "" set Z=%GMS_SETTING%\gmsdebug.log"

REM Go up one level and call it GMS_ROOT: -----
cd ..
echo Defining: Installation directories ...
if "%OS%" == "" copy %GMS_SETTING%\pwd.sys pwd_tmp.bat > nul
if not "%OS%" == "" copy "%GMS_SETTING%\pwd.sys" pwd_tmp.bat > nul
cd>> pwd_tmp.bat
if exist pwd_tmp.bat call pwd_tmp.bat
if exist pwd_tmp.bat del pwd_tmp.bat
if exist pwd.bat del pwd.bat
REM Directory name shall not end with a slash:
if "%PWD%" == "A:\\" set PWD=A:
if "%PWD%" == "B:\\" set PWD=B:
if "%PWD%" == "C:\\" set PWD=C:
if "%PWD%" == "D:\\" set PWD=D:
if "%PWD%" == "E:\\" set PWD=E:
if "%PWD%" == "F:\\" set PWD=F:
if "%PWD%" == "G:\\" set PWD=G:
if "%PWD%" == "H:\\" set PWD=H:
if "%PWD%" == "I:\\" set PWD=I:
if "%PWD%" == "J:\\" set PWD=J:
if "%PWD%" == "K:\\" set PWD=K:
if "%PWD%" == "L:\\" set PWD=L:
if "%PWD%" == "M:\\" set PWD=M:
if "%PWD%" == "N:\\" set PWD=N:
if "%PWD%" == "O:\\" set PWD=O:
if "%PWD%" == "P:\\" set PWD=P:
if "%PWD%" == "Q:\\" set PWD=Q:
if "%PWD%" == "R:\\" set PWD=R:
if "%PWD%" == "S:\\" set PWD=S:
if "%PWD%" == "T:\\" set PWD=T:
if "%PWD%" == "U:\\" set PWD=U:
if "%PWD%" == "V:\\" set PWD=V:
if "%PWD%" == "W:\\" set PWD=W:
if "%PWD%" == "X:\\" set PWD=X:
if "%PWD%" == "Y:\\" set PWD=Y:
if "%PWD%" == "Z:\\" set PWD=Z:
set GMS_ROOT=%PWD%
echo GMS_ROOT: %PWD%
if not "%OS%" == "" echo GMS_ROOT=%GMS_ROOT%>> %Z%

REM Load folders, check if there's a folder for temorary files, backup path: -
if "%OS%" == "" if exist %GMS_SETTING%\%any% cd %GMS_SETTING%
if not "%OS%" == "" if exist "%GMS_SETTING%\%any%" cd "%GMS_SETTING%"
if not "%GMS_DRDOS%" == "" cd %GMS_SETTING%
if not "%OS%" == "" if not "%OS%" == "Windows_9x" cd "%GMS_SETTING%" > nul
if not exist folder.bat echo Missing: Folder configuration file.
if exist folder.bat call folder called_by gmssetup
if not "%GMS_DRDOS%" == "" echo Delete me! > %GMS_TEMP%\deleteme.txt > nul
if "%OS%" == "" if not exist %GMS_TEMP%\%any% md %GMS_TEMP% > nul
if not "%OS%" == "" if not exist "%GMS_TEMP%\%any%" md "%GMS_TEMP%" > nul

if not "%GMS_PATH%" == "" goto fi_path
set GMS_PATH=%PATH%
if "%DRDOSCFG%" == "" goto elsedr7
path %GMS_BATCH%;%GMS_BINARIES%;%PATH%

```

```

goto fi_dr7
:else_dr7
if not "%OS%" == "Windows_9x" set PATH=%GMS_BATCH%;%GMS_BINARIES%;%PATH%
if "%OS%" == "Windows_9x" set PATH=%GMS_BINARIES%;"%PATH%"
if "%OS%" == "Windows_9x" set PATH=%GMS_BATCH%;"%PATH%"
:fi_dr7
if "%OS%" == "" goto fi_path
echo GMS_PATH=%GMS_PATH%>> %Z%
echo PATH=%PATH%>> %Z%
:fi_path

REM Get GMS_DRIVE: -----

REM drv.sys just contains the command "set drv=":
set dlist=%GMS_TEMP%\drv.lst
set batch=%GMS_TEMP%\drv.bat
if "%OS%" == "" copy %GMS_SETTING%\drv.sys %batch% > nul
if not "%OS%" == "" copy "%GMS_SETTING%\drv.sys" "%batch%" > nul
cd > %dlist%
if "%OS%" == "" %GMS_BINARIES%\sed "s/./:/" < %dlist% >> %batch%
if not "%OS%" == "" "%GMS_BINARIES%\sed" "s/./:/" < %dlist% >> %batch%
if exist %batch% call %batch%
set GMS_DRIVE=%DRV%
if not "%OS%" == "" echo GMS_DRIVE=%GMS_DRIVE%>> %Z%
set DRV=
if "%OS%" == "" goto else_gd
if "%OS%" == "Windows_9x" goto else_gd
if exist %dlist% del %dlist% > nul 2> nul
if exist %batch% del %batch% > nul 2> nul
goto fi_gd
:else_gd
if exist %dlist% del %dlist% > nul
if exist %batch% del %batch% > nul
:fi_gd
set dlist=
set batch=

REM Setting variables: =====

REM Browser interface variables:
REM Maximal size and number of files:
set GMS_MAXSIZE=1234567
set GMS_MAXFILES=123
REM Width of text area (join/split):
set GMS_WIDTH_ONE=60
set GMS_WIDTH_TWO=40
REM Height of text area (join/split):
set GMS_HEIGHT_ONE=30
set GMS_HEIGHT_TWO=40
REM Font size, in point (join/split):
set GMS_SIZE_ONE=10
set GMS_SIZE_TWO=8

REM Textmode interface variables: -----

REM Initialize colors & pattern (R: set by random) and animation mode:
REM Foreground colors:
set GMS_TEXT=R
set GMS_HOTKEY=R
set GMS_PATTERN=R
REM Background colors:
set GMS_BANNER=R
set GMS_SHADE=R
set GMS_DESKTOP=R
REM Fill character:
set GMS_LETTER=R
REM Menu animation:
set GMS_ANIMATE=on
if "%OS%" == "" set GMS_ANIMATE=off
if "%OS%" == "Windows_9x" set GMS_ANIMATE=off

```

```

REM Programs: -----

set GMS_VIEWER=viewer
set GMS_ANALYST=tidy
set GMS_TSETTER=pdfetex

if "%OS%" == "" set GMS_EDITOR=editor
if "%OS%" == "" if not "%DRDOSCFG%" == "" set GMS_EDITOR=edit
if not "%OS%" == "" set GMS_EDITOR=notepad

if not "%OS%" == "" set GMS_BROWSER=default
if "%OS%" == "" set GMS_BROWSER=browser
if "%OS%" == "Windows_9x" set GMS_BROWSER=browser

if not "%OS%" == "" set GMS_READER=default
if "%OS%" == "Windows_9x" set GMS_READER=reader
if "%OS%" == "" set GMS_READER=reader

REM Launcher script: =====

REM Create a new "gerolf" script:
echo.
echo Building: GMS launcher script ...
echo %GMS_SETTING%\gerolf.bat
call g_dos -current_cp
call g_launch -build called by gmssetup
if "%OS%" == "" if exist %GMS_SETTING%\gerolf.bat goto fi_cg
if not "%OS%" == "" if exist "%GMS_SETTING%\gerolf.bat" goto fi_cg
echo Error: Could not build launcher script.
goto tixe
:fi_cg

REM Show lower desktop border: -----

set REPLY_BANNER=0
set REPLY_TEXT=7
set REPLY_OFFSET=0
set REPLY_SIZE=82
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
reply -banner 23 "%arg%arg%"
if "%OS%" == "" set arg=DOS
if "%OS%" == "" if not "%windir%" == "" set arg=Windows_3x
if not "%OS%" == "" set arg=%OS%
if not "%GMS_FreeDOS%" == "" set arg=FreeDOS (www.freedos.org)
if not "%GMS_DRDOS%" == "" set arg=DRDOS Version %GMS_DRDOS%
echo Running under %arg% ...
set arg=
set REPLY_OFFSET=79
set REPLY_SIZE=3
reply -banner 10 " "

REM Get a system directory that can be supposed to be included in search path:
REM -----

if not "%windir%" == "" goto fi_ss
set windir=%GMS_DRIVE%
if exist %TEMP%\%any% set windir=%TEMP%
if "%GMS_FreeDOS%" == "" goto fi_sfd
if exist %GMS_DRIVE%\fdos\%any% set windir=%GMS_DRIVE%\fdos
if exist %GMS_DRIVE%\fdos\bin\%any% set windir=%GMS_DRIVE%\fdos\bin
if exist %GMS_DRIVE%\freedos\%any% set windir=%GMS_DRIVE%\freedos
if exist C:\fdos\%any% set windir=C:\fdos
if exist C:\fdos\bin\%any% set windir=C:\fdos\bin
if exist C:\freedos\%any% set windir=C:\freedos
goto fi_s
:fi_sfd
if "%GMS_DRDOS%" == "" goto fi_sdr
if exist %GMS_DRIVE%\drdos\*. * set windir=%GMS_DRIVE%\drdos
if exist %GMS_DRIVE%\drdos5\*. * set windir=%GMS_DRIVE%\drdos5

```

```

if exist %GMS_DRIVE%\drdos5\*. * set windir=%GMS_DRIVE%\drdos5
if exist %GMS_DRIVE%\drdos6\*. * set windir=%GMS_DRIVE%\drdos6
if exist %GMS_DRIVE%\drdos60\*. * set windir=%GMS_DRIVE%\drdos60
if exist C:\drdos\*. * set windir=C:\drdos
if exist C:\drdos5\*. * set windir=C:\drdos5
if exist C:\drdos50\*. * set windir=C:\drdos50
if exist C:\drdos6\*. * set windir=C:\drdos6
if exist C:\drdos60\*. * set windir=C:\drdos60
if "%DRDOSCFG%" == "" goto fi_s
    if exist %GMS_DRIVE%\drdos7\%any% set windir=%GMS_DRIVE%\drdos7
    if exist %GMS_DRIVE%\drdos70\%any% set windir=%GMS_DRIVE%\drdos70
    if exist %GMS_DRIVE%\drdos701\%any% set windir=%GMS_DRIVE%\drdos701
    if exist %GMS_DRIVE%\drdos702\%any% set windir=%GMS_DRIVE%\drdos702
    if exist %GMS_DRIVE%\drdos703\%any% set windir=%GMS_DRIVE%\drdos703
    if exist C:\drdos7\%any% set windir=C:\drdos7
    if exist C:\drdos70\%any% set windir=C:\drdos70
    if exist C:\drdos701\%any% set windir=C:\drdos701
    if exist C:\drdos702\%any% set windir=C:\drdos702
    if exist C:\drdos703\%any% set windir=C:\drdos703
goto fi_s
:fi_sdr
if exist %GMS_DRIVE%\dos\%any% set windir=%GMS_DRIVE%\dos
if exist %GMS_DRIVE%\dos5\%any% set windir=%GMS_DRIVE%\dos5
if exist %GMS_DRIVE%\dos50\%any% set windir=%GMS_DRIVE%\dos50
if exist %GMS_DRIVE%\dos6\%any% set windir=%GMS_DRIVE%\dos6
if exist %GMS_DRIVE%\dos60\%any% set windir=%GMS_DRIVE%\dos60
if exist %GMS_DRIVE%\dos61\%any% set windir=%GMS_DRIVE%\dos61
if exist %GMS_DRIVE%\dos62\%any% set windir=%GMS_DRIVE%\dos62
if exist %GMS_DRIVE%\dos7\%any% set windir=%GMS_DRIVE%\dos7
if exist %GMS_DRIVE%\dos70\%any% set windir=%GMS_DRIVE%\dos70
if exist %GMS_DRIVE%\dos71\%any% set windir=%GMS_DRIVE%\dos71
if exist %GMS_DRIVE%\dos72\%any% set windir=%GMS_DRIVE%\dos72
if exist %GMS_DRIVE%\msdos\%any% set windir=%GMS_DRIVE%\msdos
if exist %GMS_DRIVE%\ms-dos\%any% set windir=%GMS_DRIVE%\ms-dos
if exist %GMS_DRIVE%\msdos5\%any% set windir=%GMS_DRIVE%\msdos5
if exist %GMS_DRIVE%\msdos50\%any% set windir=%GMS_DRIVE%\msdos50
if exist %GMS_DRIVE%\msdos6\%any% set windir=%GMS_DRIVE%\msdos6
if exist %GMS_DRIVE%\msdos60\%any% set windir=%GMS_DRIVE%\msdos60
if exist %GMS_DRIVE%\msdos61\%any% set windir=%GMS_DRIVE%\msdos61
if exist %GMS_DRIVE%\msdos62\%any% set windir=%GMS_DRIVE%\msdos62
if exist %GMS_DRIVE%\msdos7\%any% set windir=%GMS_DRIVE%\msdos7
if exist %GMS_DRIVE%\msdos70\%any% set windir=%GMS_DRIVE%\msdos70
if exist %GMS_DRIVE%\msdos71\%any% set windir=%GMS_DRIVE%\msdos71
if exist %GMS_DRIVE%\msdos72\%any% set windir=%GMS_DRIVE%\msdos72
if exist %GMS_DRIVE%\pcdos\%any% set windir=%GMS_DRIVE%\pcdos
if exist %GMS_DRIVE%\mwin\%any% set windir=%GMS_DRIVE%\mwin
if exist %GMS_DRIVE%\mwin3\%any% set windir=%GMS_DRIVE%\mwin3
if exist %GMS_DRIVE%\mwin31\%any% set windir=%GMS_DRIVE%\mwin31
if exist %GMS_DRIVE%\mwin311\%any% set windir=%GMS_DRIVE%\mwin311
if exist %GMS_DRIVE%\win\%any% set windir=%GMS_DRIVE%\win
if exist %GMS_DRIVE%\win3\%any% set windir=%GMS_DRIVE%\win3
if exist %GMS_DRIVE%\win31\%any% set windir=%GMS_DRIVE%\win31
if exist %GMS_DRIVE%\win311\%any% set windir=%GMS_DRIVE%\win311
if exist %GMS_DRIVE%\windows\%any% set windir=%GMS_DRIVE%\windows
if exist C:\dos\%any% set windir=C:\dos
if exist C:\dos5\%any% set windir=C:\dos5
if exist C:\dos50\%any% set windir=C:\dos50
if exist C:\dos6\%any% set windir=C:\dos6
if exist C:\dos60\%any% set windir=C:\dos60
if exist C:\dos61\%any% set windir=C:\dos61
if exist C:\dos62\%any% set windir=C:\dos62
if exist C:\dos7\%any% set windir=C:\dos7
if exist C:\dos70\%any% set windir=C:\dos70
if exist C:\dos71\%any% set windir=C:\dos71
if exist C:\dos72\%any% set windir=C:\dos72
if exist C:\msdos\%any% set windir=C:\msdos
if exist C:\ms-dos\%any% set windir=C:\ms-dos
if exist C:\msdos5\%any% set windir=C:\msdos5
if exist C:\msdos50\%any% set windir=C:\msdos50

```

```

if exist C:\msdos6\%any% set windir=C:\msdos6
if exist C:\msdos60\%any% set windir=C:\msdos60
if exist C:\msdos61\%any% set windir=C:\msdos61
if exist C:\msdos62\%any% set windir=C:\msdos62
if exist C:\msdos7\%any% set windir=C:\msdos7
if exist C:\msdos70\%any% set windir=C:\msdos70
if exist C:\msdos71\%any% set windir=C:\msdos71
if exist C:\msdos72\%any% set windir=C:\msdos72
if exist C:\pcdos\%any% set windir=C:\pcdos
if exist C:\mwin\%any% set windir=C:\mwin
if exist C:\mwin3\%any% set windir=C:\mwin3
if exist C:\mwin31\%any% set windir=C:\mwin31
if exist C:\mwin311\%any% set windir=C:\mwin311
if exist C:\win\%any% set windir=C:\win
if exist C:\win3\%any% set windir=C:\win3
if exist C:\win31\%any% set windir=C:\win31
if exist C:\win311\%any% set windir=C:\win311
if exist C:\windows\%any% set windir=C:\windows
if not "%OS%" == "" echo windir=%windir%>> %Z%
:fi_s

REM if not %windir% == C: goto fi_ss
if not %windir% == %GMS_DRIVE% goto fi_ss
set i=%GMS_DRIVE%\gerolf.bat
set j=%GMS_DRIVE%\gms.bat
set k=%GMS_DRIVE%
echo          Could not find binary folder %k%\dos or %k%\windows.
echo          Please link or copy %i% and %j% to a
echo          folder which is included in the executables search path!
set i=
set j=
set k=
:fi_ss

REM Create desktop link (Windows): -----
echo Linking:      gerolf.bat and gms.bat to ...
if "%OS%" == "" goto fi_nodl
if "%GMS_LINKDIR%" == "" goto fi_nodl
if not exist "%GMS_LINKDIR%\%any%" goto fi_nodl
set gerolf="%GMS_LINKDIR%\gerolf.bat"
if not "%GMS_DRDOS%" == "" echo echo off > %gerolf%
if not "%GMS_DRDOS%" == "" echo cls >> %gerolf%
if "%GMS_DRDOS%" == "" echo @echo off > %gerolf%
echo.>> %gerolf%
echo REM gerolf.bat Link script to Gerolf Markup Shredder>>%gerolf%
echo REM =====>> %gerolf%
echo.>> %gerolf%
rem echo %GMS_DRIVE% >> %gerolf%
if "%OS%" == "" echo call %GMS_SETTING%\gerolf %1 %2 %3 %4>> %gerolf%
if not "%OS%" == "" echo call "%GMS_SETTING%\gerolf" %1 %2 %3 %4>>%gerolf%
:fi_nodl

REM Context Menu and Start Menu: -----
if "%OS%" == "" goto fi_noos
if "%GMS_SENDDIR%" == "" goto fi_nosd
if not exist "%GMS_SENDDIR%\%any%" goto fi_nosd
copy %gerolf% "%GMS_SENDDIR%" > nul
:fi_nosd
if "%GMS_MENUDIR%" == "" goto fi_nomd2
if not exist "%GMS_MENUDIR%\%any%" goto fi_nomd2
if exist "%GMS_MENUDIR%\Markup Shredder\%any%" goto fi_nomd1
md "%GMS_MENUDIR%\Markup Shredder" > nul
:fi_nomd1
copy %gerolf% "%GMS_MENUDIR%\Markup Shredder" > nul
:fi_nomd2
:fi_noos

REM Copy 'sort.exe' to binary folder on Windows 9x: -----

```

```

if not "%OS%" == "Windows_9x" goto fi_sort
  if exist "%GMS_BINARIES%\sort.exe" goto fi_sort
  copy "%windir%\command\sort.exe" "%GMS_BINARIES%" > nul
:fi_sort

REM Writing link scripts to system directory: -----

echo          %windir%
REM set gerolf=%windir%\gerolf.bat
set gerolf=..\gerolf.bat

if not "%GMS_DRDOS%" == "" echo echo off > %gerolf%
if not "%GMS_DRDOS%" == "" echo cls >> %gerolf%
if "%GMS_DRDOS%" == "" echo @echo off > %gerolf%
echo.>> %gerolf%
echo REM gerolf.bat Link script to Gerolf Markup Shredder>>%gerolf%
echo REM ===== Must be placed in search path>> %gerolf%
echo.>> %gerolf%
if "%OS%" == "" echo call %GMS_SETTING%\gerolf %1 %2 %3 %4>> %gerolf%
if not "%OS%" == "" echo call "%GMS_SETTING%\gerolf" %1 %2 %3 %4>>%gerolf%

if "%OS%" == "" copy %gerolf% %windir% > nul
if "%OS%" == "" goto fi_xp1
if "%OS%" == "Windows_9x" copy %gerolf% %windir% > nul
if "%OS%" == "Windows_9x" goto fi_xp1
  copy "%gerolf%" "%windir%" > nul 2> nul
:fi_xp1
set gerolf=

REM set gms=%windir%\gms.bat
set gms=..\gms.bat

if not "%GMS_DRDOS%" == "" echo echo off > %gms%
if not "%GMS_DRDOS%" == "" echo cls >> %gms%
if "%GMS_DRDOS%" == "" echo @echo off > %gms%
echo.>> %gms%
echo REM gms.bat Link script to Gerolf Markup Shredder>> %gms%
echo REM ===== Must be placed in search path>> %gms%
echo.>> %gms%
if "%OS%" == "" echo call %GMS_BATCH%\gms %1 %2 %3 %4 %5 %6>> %gms%
if not "%OS%" == "" echo call "%GMS_BATCH%\gms" %1 %2 %3 %4 %5 %6>>%gms%

if "%OS%" == "" copy %gms% %windir% > nul
if "%OS%" == "" goto fi_xp2
if "%OS%" == "Windows_9x" copy %gms% %windir% > nul
if "%OS%" == "Windows_9x" goto fi_xp2
  if not "%OS%" == "" copy "%gms%" "%windir%" > nul 2> nul
:fi_xp2
set gms=

REM Check if links have been created: -----

if "%OS%" == "" if exist %windir%\gerolf.bat goto fi_n11
if not "%OS%" == "" if exist "%windir%\gerolf.bat" goto fi_n11
  echo.
  echo          Could not build link from
  echo          %GMS_SETTING%\gerolf.bat to
  echo          %windir%
  echo          Link or copy gerolf.bat into the search path.
:fi_n11

if "%OS%" == "" if exist %windir%\gms.bat goto fi_n12
if not "%OS%" == "" if exist "%windir%\gms.bat" goto fi_n12
  echo.
  echo          Could not build link from
  echo          %GMS_BATCH%\gms.bat to
  echo          %windir%
  echo          Link or copy gms.bat into the search path.
:fi_n12

```

```

REM Create desktop link for quick setup (Windows): -----

if "%OS%" == "" goto fi_nolnk
if "%GMS_LINKDIR%" == "" goto fi_nolnk

REM Writing test:
if not "%OS%" == "Windows_9x" goto else_x1
  echo GMS writing test (2).> "%GMS_LINKDIR%\gms.002"
  echo This file can be removed.>> "%GMS_LINKDIR%\gms.002"
  goto fi_x1
:else_x1
  echo GMS writing test (2).> "%GMS_LINKDIR%\gms.002" 2> nul
  echo This file can be removed.>> "%GMS_LINKDIR%\gms.002" 2> nul
:fi_x1
if exist "%GMS_LINKDIR%\gms.002" goto fi_notxt
  echo          Cannot write start links to
  echo          %GMS_LINKDIR%.
  echo.
  if not "%OS%" == "Windows_NT" if not "%OS%" == "Windows_XP" goto fi_x2
  echo          Please log in as administrator to install GMS.
  goto tixe
:fi_x2
  echo          Press any key to continue or [Ctrl+C] to break ...
  pause > nul
  goto fi_nowin
:fi_notxt
if exist "%GMS_LINKDIR%\gms.002" del "%GMS_LINKDIR%\gms.002"
:fi_nowin

REM Recreate desktop link for setup (Windows): -----

set GMS_LINK=..\gmssetup.bat
if not "%GMS_DRDOS%" == "" echo echo off > %GMS_LINK%
if not "%GMS_DRDOS%" == "" echo cls >> %GMS_LINK%
if "%GMS_DRDOS%" == "" echo @echo off > %GMS_LINK%
echo.>> %GMS_LINK%
echo REM gmssetup.bat Link script to:>> %GMS_LINK%
echo REM ===== Gerolf Markup Shredder setup>> %GMS_LINK%
echo.>> %GMS_LINK%
echo %GMS_DRIVE%>> %GMS_LINK%
echo cd "%GMS_SETTING%" >> %GMS_LINK%
echo "%GMS_SETTING%\gmssetup.bat" %1>> %GMS_LINK%
copy "%GMS_LINK%" "%GMS_LINKDIR%" > nul
if not exist "%GMS_MENUDIR%\%any%" goto fi_nosm3
  copy %GMS_LINK% "%GMS_MENUDIR%\Markup Shredder" > nul
:fi_nosm3
set GMS_LINK=
set GMS_LINKDIR=
set GMS_MENUDIR=
:fi_nolnk

REM Batch compiling, font map and format creation: =====

REM Batch compiling:
echo.
if "%OS%" == "" if exist %GMS_BATCH%\%any% cd %GMS_BATCH%
if not "%OS%" == "" if exist "%GMS_BATCH%\%any%" cd "%GMS_BATCH%"
if "%GMS_DRDOS%" == "6" echo.
if "%GMS_DRDOS%" == "6" echo.
if "%GMS_DRDOS%" == "6" goto fi_bccc
  echo Compiling: batch command scripts ...
  call compiler -quiet called by gmssetup
if "%OS%" == "" goto fi_bcc
  echo see '%GMS_BATCH%\compiler.log' for syntax check results>> %Z%
:fi_bcc
  echo          %GMS_BATCH%\1*.bat to 1*.com
:fi_bccc
set GMS_FOLDER=%GMS_SETTING%
if "%OS%" == "" if exist %GMS_FOLDER%\%any% cd %GMS_FOLDER%
if not "%OS%" == "" if exist "%GMS_FOLDER%\%any%" cd "%GMS_FOLDER%"

```

```

REM Copy Windows true type fonts: -----
copy "%windir%\system\msvcr40.dll" "%windir%\system\msvcr.dll" > nul
:fi_dll

if "%OS%" == "" goto fi_nofnt
echo.
echo Copying: *.ttf - True Type Fonts from/to %windir%\Fonts ...
call g_font -exchange_fonts called_by gmssetup
:fi_nofnt

if not "%OS%" == "" goto fi_nofnw
if not exist c:\windows\system\*.ttf goto fi_nofnw
copy c:\windows\system\*.ttf %GMS_FONTS%\ttf > nul
:fi_nofnw

REM Write font map (if not in quick mode): -----
if "%1" == "-quick" goto fi_nomap
if "%1" == "-quick" goto fi_nomap
if "%1" == "/quick" goto fi_nomap
if "%1" == "quick" goto fi_nomap
if "%1" == "-QUICK" goto fi_nomap
if "%1" == "-QUICK" goto fi_nomap
if "%1" == "/QUICK" goto fi_nomap
if "%1" == "QUICK" goto fi_nomap
call gms -shell -w -quiet called_by gmssetup
:fi_nomap

set REPLY_OFFSET=0
set REPLY_SIZE=82
reply -banner 22
reply -banner 21
reply -banner 20
reply -banner 19

REM Check if pool file has appropriate name: -----
if not "%OS%" == "" goto fipool0
if "%OS%" == "" if not exist %GMS_BINARIES%\pdfetex.poo goto fipool0
ren %GMS_BINARIES%\pdfetex.poo pdfetex.tmp
ren %GMS_BINARIES%\pdfetex.tmp pdfetex.pool
:fipool0

if "%OS%" == "" if exist %GMS_BINARIES%\pdfetex.po* goto fipool2
if not "%OS%" == "" if not exist "%GMS_BINARIES%\pdfete-1.po*" goto fipool1
if "%OS%" == "" ren %GMS_BINARIES%\pdfete-1.po* pdfetex.poo
if not "%OS%" == "" ren "%GMS_BINARIES%\pdfete-1.po*" pdfetex.pool
:fipool1

if "%OS%" == "" if exist %GMS_BINARIES%\pdfetex.po* goto fipool2
if not "%OS%" == "" if exist "%GMS_BINARIES%\pdfetex.po*" goto fipool2
echo Error: No message string file '%GMS_BINARIES%\pdfetex.po*'.
pause > nul
:fipool2

if "%OS%" == "" goto fipool3
if exist "%GMS_BINARIES%\pdfetex.pool" goto fipool3
if exist "%GMS_BINARIES%\pdfetex.poo" goto elspool3
if exist "%GMS_BINARIES%\pdfete-1.poo" goto elspool3
echo Error: No message string file '%GMS_BINARIES%\pdfetex.pool'.
goto fipool3
:elspool3
ren "%GMS_BINARIES%\pdfetex.poo" pdfetex.pool
goto fipool3
:elspool3
ren "%GMS_BINARIES%\pdfete-1.poo" pdfetex.pool
:fipool3

REM Check if DLL required by Tidy is missing on Windows-95: -----
if not "%OS%" == "Windows_9x" goto fi_dll
if exist "%windir%\system\msvcr.dll" goto fi_dll
if not exist "%windir%\system\msvcr40.dll" goto fi_dll
copy "%windir%\system\msvcr40.dll" "%windir%\system\msvcr.dll" > nul
:fi_dll

REM Define maximal number of input files on Dos: -----
if not "%OS%" == "" goto fifiles
set arg=%GMS_SETTING%\files.cfg
echo. > %arg%
echo %% files.cfg>> %arg%
echo %% =====>> %arg%
echo.>> %arg%
echo %% This file defines the maximal number of input>> %arg%
echo %% files for Gerolf Markup Shredder on Dos.>> %arg%
echo.>> %arg%
echo \message {-}>> %arg%
echo.>> %arg%
echo \def \NOfilemax {222}>> %arg%
echo.>> %arg%
echo \message {Maximal number of input or image files: \NOfilemax}>> %arg%
echo.>> %arg%
echo \endinginput>> %arg%
:fifiles

REM Initialize format: -----
set REPLY_OFFSET=0
set REPLY_SIZE=17
reply -banner 18 Initializing:
set REPLY_OFFSET=14
set REPLY_SIZE=68
reply -banner 18 "TeX format file (this may take some time) ..."
if "%GMS_DRDOS%" == "" call gms -shell -i -quiet called_by gmssetup > nul
if not "%GMS_DRDOS%" == "" call gms -shell -i -quiet called_by gmssetup
set REPLY_BANNER=0
set REPLY_TEXT=7
set REPLY_OFFSET=14
set REPLY_SIZE=68
if "%OS%" == "" reply -banner 18 " %GMS_BINARIES%\gerolf.(efm/fmt)"
if not "%OS%" == "" reply -banner 18 " %GMS_BINARIES%\gerolf.(e)fmt"

REM Write memo (load default.htm): -----
set GMS_FOLDER=%GMS_TEMPLATE%\default
set GMS_FILE=default.htm
call g_vars -write_memo called_by gmssetup

REM Goodbye message: =====
echo.
echo Startup call: gerolf ... textmode interface (random colors)
echo.
echo.
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%%arg%
set arg=

:exit
set GMS_FOLDER=%GMS_SETTING%
REM Fixme for spacy names:
if "%OS%" == "" if exist %GMS_FOLDER%\%any% cd %GMS_FOLDER%
if not "%OS%" == "" if exist "%GMS_FOLDER%\%any%" cd "%GMS_FOLDER%"
REM Draw startup/break info on screen:
set arg= Press any key to start or [Ctrl+C] to quit
echo Done: GMS setup. %arg%
set arg=
REM Draw gms startup info on screen:
set REPLY_OFFSET=14
set REPLY_SIZE=68
reply -banner 21 "gms ... command line interface (black/white)"

```

```

REM Set break cursor at a nice place:
set REPLY_OFFSET=13
set REPLY_SIZE=0
reply -banner 22
REM Clean up environment:
call g_vars -clear called_by gmssetup
REM Rename gmsdebug.log to gmssetup.log:
if not "%OS%" == "" if exist %Z% if exist gmssetup.log del gmssetup.log
if not "%OS%" == "" if exist %Z% ren %Z% gmssetup.log
REM Write gmsdebug.log footer:
if "%OS%" == "" goto fi_noft
set arg=////////////////////////////////////
if "%OS%" == "Windows_9x" set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
if "%winbootdir%" == "." set arg=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
echo %arg%%arg%>> %Z%
echo. %GMS_SETTING%>> %Z%
set arg=
:fi_noft
REM First Steps readme: .....
REM pause > nul
REM cls
REM if "%OS%" == "DRDOS" set GMS_DRDOS=6

```

```

REM if not "%DRDOCFG%" == "" set GMS_DRDOS=7
REM if "%OS%" == "DRDOS" set OS=
REM type readme.txt
REM Allow user interruption or start textmode interface:
if "%OS%" == "" goto fi_w
if "%OS%" == "Windows_9x" goto fi_w
set arg=Welcome to Gerolf Markup Shredder
if not "%OS%" == "" title %arg% · www.Gerolf.org > nul
set arg=
:fi_w
REM Final unsets: .....
set Z=
set GMS_DEBUG=
set GMS_INST=
set GMS_MISS=
set GMS_SETTING=
set GMS_SENDDIR=
pause > nul
if not "%OS%" == "" if not "%OS%" == "Windows_9x" color 07
gerolf called_by gmssetup
:tixe
echo.

```


Configuration Files

alias.cfg

```
% alias.cfg
% =====

% This file is part of Gerolf Markup Shredder.

% =====

\hyphenmessage \echo {- User-defined font names (typeface aliases):}

% These definitions are necessary for slanted or extended fonts as well as
% for fonts with irregularities in their PostScript font names, e.g. "ArialMT"
% and "Arial-BoldMT" instead of "ArialMT-Bold", or the same name ("Baramond")
% for all four faces.

% Syntax:
% \MAPfontadd <desired HTML name> (<Roman font file name in font.map col. 1>)
% \MAPfontaddpair <HTML name> (<Roman font file>, <Italic font file>)
% \MAPfontaddmates <HTML name> (<Roman font file>, <Bold font file>)
% \MAPfontaddfamily <HTML name> (<Roman>, <Italic>, <Bold>, <BoldItal>)
% There must always be at least one space behind the comma
% in the above definitions.
% There are two short forms for \MAPfontaddfamily:
% \MAPfontaddfamilystd <HTML name> (<Roman font file name>)
% Font file names must be "foo, fooi, foobd, foobi" for an
% "\MAPfontaddfamilystd Foo (foo)" command.
% \MAPfontaddfamilyuni <HTML name> (<Roman font file name> <Unicode row hex.>)
% Font file names must be "fooXX, fooiXX, foobdXX, foobiXX" for an
% "\MAPfontaddfamilyuni Foo (foo XX)" command. A two letter abbreviation
% of another codepage name is acceptable too for "XX".

\message {Arabic,}
\MAPfontaddfamily LucidaSansTypewriterArabic (hlsrta, hlsota,
hlsbta, hlsbota)
\MAPfontaddfamilyuni ArialArabic (arial W6)
\MAPfontaddfamilyuni CourierArabic (cour W6)
\MAPfontaddmates TahomaArabic (tahomaW6, tahomabdW6)
\MAPfontaddfamilyuni TimesArabic (times W6)

\message {Cyrillic,}
\MAPfontaddfamily ArialCyrillic (arialW1, arialiW1, arialbdW1, arialbiW1)
\MAPfontaddfamily BaskervilleCyrillic (baskernW1, baskeriW1,
baskerbW1, baskerbiW1)
\MAPfontaddmates ComicSansCyrillic (comicW1, comicbdW1)
\MAPfontaddfamily CMSuperCapsCyrillic (fcmccW1, fcmccoW1, fcmxcW1, fcmxcoW1)
\MAPfontaddpair CMSuperDunhillCyrillic (fcmhxW1, fcmhxcoW1)
\MAPfontaddpair CMSuperFibonacciCyrillic (fcmfbcW1, fcmfbcocW1)
\MAPfontaddpair CMSuperFunnyCyrillic (fcmffuW1, fcmffiW1)
\MAPfontaddfamily CMSuperRomanCyrillic (fcmrW1, fcmtiW1, fcmxW1, fcmbiW1)
\MAPfontaddfamily CMSuperSansCyrillic (fcmssW1, fcmssocW1, fcmxsW1, fcmxocW1)
\MAPfontaddpair CMSuperTypewriterCyrillic (fcmttW1, fcmttocW1)
\MAPfontaddpair CMSuperVariableCyrillic (fcmvtW1, fcmvtocW1)
\MAPfontaddfamilyuni CourierCyrillic (cour W1)
\MAPfontaddpair FranklinGothicCyrillic (framdW1, framditW1)
\MAPfontaddfamily GaramondCyrillic (garaW1, garaitW1, garabdW1,
garabiW1)% miss last
\MAPfontaddfamily GeorgiaCyrillic (georgiaW1, georgiaiW1,
georgiabW1, georgiazW1)
\MAPfontaddfamily GothicCyrillic (gothicW1, gothiciW1, gothicbW1, gothicbiW1)
\MAPfontadd ImpactCyrillic (impactW1)
\MAPfontaddmates TahomaCyrillic (tahomaW1, tahomabdW1)
\MAPfontaddfamilyuni TimesCyrillic (times W1)
\MAPfontaddfamily TrebuchetCyrillic (trebucW1, trebucitW1,
trebucbdW1, trebucbiW1)

\MAPfontaddfamily VerdanaCyrillic (verdanaW1, verdanaiW1,
verdanabW1, verdanazW1)

\message {Greek,}
\MAPfontaddfamilyuni ArialGreek (arial W3)
\MAPfontadd CardoGreek (cardoW3)
\MAPfontaddmates ComicSansGreek (comicW3, comicbdW3)
\MAPfontaddfamilyuni CourierGreek (cour W3)
\MAPfontaddpair FranklinGothicGreek (framdW3, framditW3)
\MAPfontaddfamily GaramondGreek (garaw3, garaitW3,
garabdW3, garabiW3)% miss last
\MAPfontaddpair GentiumGreek (genrW3, geniW3)
\MAPfontaddfamily GeorgiaGreek (georgiaW3, georgiaiW3,
georgiabW3, georgiazW3)
\MAPfontaddfamily GothicGreek (gothicW3, gothiciW3, gothicbW3, gothicbiW3)
\MAPfontadd ImpactGreek (impactW3)
\MAPfontadd LucidaConsoleGreek (luconW3)
\MAPfontadd LucidaSansGreek (l_10646W3)
\MAPfontaddfamily LucidaSansTypewriterGreek (hlsrtW3, hlsotW3,
hlsbtW3, hlsbotW3)
\MAPfontadd SylfaenGreek (sylfaenW3)
\MAPfontaddmates TahomaGreek (tahomaW3, tahomabdW3)
\MAPfontaddfamilyuni TimesGreek (times W3)
\MAPfontaddfamily TrebuchetGreek (trebucW3, trebucitW3,
trebucbdW3, trebucbiW3)
\MAPfontaddfamily VerdanaGreek (verdanaW3, verdanaiW3,
verdanabW3, verdanazW3)

\message {Hebrew,}
\MAPfontadd Aharoni (ahronbd)
\MAPfontaddmates Levenim (lvnm, lvnmbd)
\MAPfontaddfamilyuni ArialHebrew (arial W5)
\MAPfontaddfamilyuni CourierHebrew (cour W5)
\MAPfontadd RodTransparent (rodtr) % named Rod in font.map col. [2]
\MAPfontaddmates TahomaHebrew (tahomaW5, tahomabdW5)
\MAPfontaddfamilyuni TimesHebrew (times W5)
\MAPfontaddfamily VerdanaHebrew (verdanaW5, verdanaiW5,
verdanabW5, verdanazW5)

\message {Indian,}
\MAPfontadd ShreeAss001 (ass1) \MAPfontadd ShreeAss002 (ass2)
\MAPfontadd ShreeBan001 (ban1) \MAPfontadd ShreeBan002 (ban2)
\MAPfontadd ShreeDev001 (dev1) \MAPfontadd ShreeDev002 (dev2)
\MAPfontadd ShreeGuj001 (guj1) \MAPfontadd ShreeGuj002 (guj2)
\MAPfontadd ShreeKan001 (kan1) \MAPfontadd ShreeKan002 (kan2)
\MAPfontadd ShreeMal001 (mal1) \MAPfontadd ShreeMal002 (mal2)
\MAPfontadd ShreeOri001 (ori1) \MAPfontadd ShreeOri002 (ori2)
\MAPfontadd ShreePun001 (pun1) \MAPfontadd ShreePun002 (pun2)
\MAPfontadd ShreeTam001 (tam1) \MAPfontadd ShreeTam002 (tam2)
\MAPfontadd ShreeTel001 (tel1) \MAPfontadd ShreeTel002 (tel2)

\message {Latin,}
\MAPfontaddpair Antiqua (uaqr, uaqro)
\MAPfontaddfamilystd Arial (arial)
\MAPfontaddpair ArialBlack (ariblk, ariblki)
\MAPfontaddpair Augie (faumm, faumno)
\MAPfontaddfamily Avantgarde (uagk, uagko, uagd, uagdo)% URWGothicL
\MAPfontaddpair BrushScript (fbsm, fbsmi)
\MAPfontaddfamily CMSuperCaps (fcmcc, fcmcco, fcmxc, fcmxco)
\MAPfontaddpair CMSuperDunhill (fcmhx, fcmhxco)
\MAPfontaddpair CMSuperFibonacci (fcmfbc, fcmfbcoc)
\MAPfontaddpair CMSuperFunny (fcmffu, fcmffi)
\MAPfontaddfamily CMSuperRoman (fcmr, fcmti, fcmx, fcmbi)
```

```

\MAPfontaddfamily CMSuperSans (fcmss, fcmss0, fcmxs, fcmx0)
\MAPfontaddpair CMSuperTypewriter (fcmst, fcmst0)
\MAPfontaddpair CMSuperVariable (fcmvt, fcmvt0)
\MAPfontaddfamily CourierNarrow (ncrrc, ncrrco, ncrbc, ncrbco)
\MAPfontaddfamily CourierSlim (scrrc, scrrco, scrbc, scrbco)
\MAPfontaddmates ComicSans (comic, comicbd)
\MAPfontaddfamilystd CourierNew (cour)
\MAPfontaddpair DayRoman (fdrr, fdrr0)
\MAPfontaddpair DayRomanExpert (fdxr, fdxr0)
\MAPfontaddfamily Garamond (fgmr, fgmr1, fgmb, fgmbi)
\MAPfontaddpair Gutenberg (fgcmx, fgcmx0)% GoodCityModern
\MAPfontaddpair Grotesk (ugkb, ugkbo)
\MAPfontadd LucidaHandwriting (lhandw)
\MAPfontaddpair LucidaSansUnicode (hlsr, hlsri) % *** Karl Berry's names
\MAPfontaddfamily LucidaSansUnicode (l_10464, lsansi, lsansd, lsansdi) % ***
% *** The Lucida Sans italic etc. typefaces are *not* unicode!
% You may have to delete lsansi/hlsri here.
\MAPfontaddfamily LucidaSansTypewriter (hlsrt, hlsot, hlsbt, hlsbot)
\MAPfontaddfamily LuxiMono (hlmr, hlmr0, hlmb, hlmb0)
\MAPfontaddfamily LuxiMonoNarrow (hlmr, hlmr0, hlmb, hlmb0)
\MAPfontaddfamily Mono (ucrrc, ucrrco, ucrbc, ucrbco)
\MAPfontaddpair NationalFirst (fnfrx, fnfrx0)
\MAPfontaddfamily Sanity (fsarx, fsarx0, fsabx, fsabx0)
\MAPfontaddfamilystd TimesNewRoman (times)
\MAPfontaddfamily Trebuchet (trebuc, trebucit, trebucbd, trebuchi)
\MAPfontaddmates ZapfChancery (bzcm, bzcb)

\message {Thai,}
\MAPfontadd CourMonoThai (crmtH)
\MAPfontadd CourPropThai (crpTH)

\message {Unicode,}

\MAPfontaddfamilyuni Arial (arial 00)
% The above line is a short form of:
% \MAPfontaddfamily Arial (arial00, ariali00, arialbd00, arialbi00)
\MAPfontaddfamilyuni Arial (arial 01)
\MAPfontaddfamilyuni Arial (arial 02)
\MAPfontaddfamilyuni Arial (arial 03)
\MAPfontaddfamilyuni Arial (arial 04)
\MAPfontaddfamilyuni Arial (arial 05)
\MAPfontaddfamilyuni Arial (arial 06)
\MAPfontaddfamilyuni Arial (arial 0E)
\MAPfontaddfamilyuni Arial (arial 1E)
\MAPfontaddfamilyuni Arial (arial 20)
\MAPfontaddfamilyuni Arial (arial 21)
\MAPfontaddfamilyuni Arial (arial 22)
\MAPfontaddfamilyuni Arial (arial 23)
\MAPfontaddfamilyuni Arial (arial 24)
\MAPfontaddfamilyuni Arial (arial 25)
\MAPfontaddfamilyuni Arial (arial 26)
\MAPfontaddfamilyuni Arial (arial 27)

\MAPfontaddfamilyuni CourierNew (cour 00)
\MAPfontaddfamilyuni CourierNew (cour 01)
\MAPfontaddfamilyuni CourierNew (cour 02)
\MAPfontaddfamilyuni CourierNew (cour 03)
\MAPfontaddfamilyuni CourierNew (cour 04)
\MAPfontaddfamilyuni CourierNew (cour 05)
\MAPfontaddfamilyuni CourierNew (cour 06)
\MAPfontaddfamilyuni CourierNew (cour 0E)
\MAPfontaddfamilyuni CourierNew (cour 1E)
\MAPfontaddfamilyuni CourierNew (cour 20)
\MAPfontaddfamilyuni CourierNew (cour 21)
\MAPfontaddfamilyuni CourierNew (cour 22)
\MAPfontaddfamilyuni CourierNew (cour 23)
\MAPfontaddfamilyuni CourierNew (cour 24)
\MAPfontaddfamilyuni CourierNew (cour 25)
\MAPfontaddfamilyuni CourierNew (cour 26)

\MAPfontaddfamilyuni CourierNew (cour 27)
\MAPfontaddfamilyuni TimesNewRoman (times 00)
\MAPfontaddfamilyuni TimesNewRoman (times 01)
\MAPfontaddfamilyuni TimesNewRoman (times 02)
\MAPfontaddfamilyuni TimesNewRoman (times 03)
\MAPfontaddfamilyuni TimesNewRoman (times 04)
\MAPfontaddfamilyuni TimesNewRoman (times 05)
\MAPfontaddfamilyuni TimesNewRoman (times 06)
\MAPfontaddfamilyuni TimesNewRoman (times 0E)
\MAPfontaddfamilyuni TimesNewRoman (times 1E)
\MAPfontaddfamilyuni TimesNewRoman (times 20)
\MAPfontaddfamilyuni TimesNewRoman (times 21)
\MAPfontaddfamilyuni TimesNewRoman (times 22)
\MAPfontaddfamilyuni TimesNewRoman (times 23)
\MAPfontaddfamilyuni TimesNewRoman (times 24)
\MAPfontaddfamilyuni TimesNewRoman (times 25)
\MAPfontaddfamilyuni TimesNewRoman (times 26)
\MAPfontaddfamilyuni TimesNewRoman (times 27)

\message {Vietnamese,}
\MAPfontaddfamilyuni ArialVietnam (arial VI)
\MAPfontaddfamilyuni CourierVietnam (cour VI)
\MAPfontaddmates TahomaVietnam (tahomaVI, tahomabdVI)
\MAPfontaddfamilyuni TimesVietnam (times VI)
\MAPfontaddfamily VerdanaVietnam (verdanaVI, verdanaVI,
verdanabVI, verdanzVI)

\message {Family aliases,} %-----
% These definitions are useful if a font family shall be called by a shorter
% name, e.g. "Arial" instead of "ArialMT".

% Format:
% 1: Desired HTML name
% 2: PostScript name in font.map column [2]

\MAPfontaddalias Aharoni (AharoniBold)
\MAPfontaddalias Antiqua (URWAntiquaT)
\MAPfontaddalias Arial (ArialMT)
\MAPfontaddalias Baskerville (BaskervilleBT)
\MAPfontaddalias Baskerville (BaskervilleWin95)
\MAPfontaddalias Beautiful (BeautifulES)
\MAPfontaddalias Bodoni (BodoniBT)
\MAPfontaddalias Bookman (URWBookmanL)
\MAPfontaddalias CenturySchoolbook (CenturySchL)
\MAPfontaddalias Champagne (CACChampagne)
\MAPfontaddalias Charter (CharterBT)
\MAPfontaddalias CourierNarrow (IBMCourier)
\MAPfontaddalias Garamond (Baramond)
\MAPfontaddalias Grotesk (URWGroteskT)
\MAPfontaddalias Helvetica (NimbusSanL)
\MAPfontaddalias LucidaSans (LucidaSansUnicode)
\MAPfontaddalias Mono (NimbusMonL)
\MAPfontaddalias NationalFirst (National)
\MAPfontaddalias Palatino (URWPalladioL)
\MAPfontaddalias ParkAvenue (ParkAvenueBT)
\MAPfontaddalias Sanity (SanityWide)
\MAPfontaddalias SanityWide (SanityWideBold)
\MAPfontaddalias TimesNewRoman (TimesNewRomanPS)
\MAPfontaddalias Trebuchet (TrebuchetMS)
\MAPfontaddalias ZapfChancery (ZapfChanceryITCbyBT)
\MAPfontaddalias ZapfDingbats (ITCZapfDingbats)

\message {family-specific encodings,} %-----

\MAPfontaddspecific Symbol
\MAPfontaddspecific ZapfDingbats

```

```

\MAPfontaddencoding ArabicBH (CP1256)
\MAPfontaddencoding ArabicMT (CP1256)

\message {encoding aliases.}%-----

% These definitions are useful if a document's codepage differs from the
% encodings of the installed fonts.

% Format:
% 1: Document encoding (HTML name) to be replaced. It is given like
%    '<meta http-equiv = "Content-Type"
%      content = "text/html; charset=iso-8859-1" />'

% 2: Desired codepage (HTML name) like 'windows-1252'
% 3: Desired codepage (DOS name) like 'CP1252'

% West Europe:
\CODEPAGEaddalias iso-8859-1 (windows-1252, CP1252)
\CODEPAGEaddalias windows-1252 (iso-8859-1, I8859-1)

% Thai (access West European fonts):
\CODEPAGEaddalias windows-874 (windows-1252, CP1252)

\endinput

```

encoding.cfg

```
# encoding.cfg
# =====

# User-defined default font encodings for non-latin fonts.
# This file is part of Gerolf Markup Shredder.

# Format:
# [1]: base name of font file
# [2] desired encoding

# =====

# Arabic:

# ArabicMT and ArabicBH are compatible to CP1256,
# but use non-Adobe glyph names in encoding files.

# Andalus:
  andlso: ArabicMT
  ANDLSO: ArabicMT
# ArabicTransparent:
  artro: ArabicMT
  ARTRO: ArabicMT
  artrbdo: ArabicMT
  ARTRBDO: ArabicMT
# SimplifiedArabic:
  simpo: ArabicMT
  SIMPO: ArabicMT
  simpbdo: ArabicMT
  SIMPBDO: ArabicMT
  simpfxo: ArabicMT
  SIMPFXO: ArabicMT
# TraditionalArabic:
  trado: ArabicMT
  TRADO: ArabicMT
  tradbdo: ArabicMT
  TRADBDO: ArabicMT

# Hebrew: -----
# Aharoni:
  ahronbd: CP1255
  AHRONBD: CP1255
# David:
  david: CP1255
  DAVID: CP1255
  davidbd: CP1255
  DAVIDBD: CP1255
  davidtr: CP1255
  DAVIDTR: CP1255
# FrankRuehl:
  frank: CP1255
  FRANK: CP1255
# Levenim:
  lvnm: CP1255
  LVNM: CP1255
  lvnmbd: CP1255
  LVNMBD: CP1255
# Miriam:
  mriam: CP1255
  MRIAM: CP1255
  mriamc: CP1255
  MRIAMC: CP1255
  mriamfx: CP1255
  MRIAMFX: CP1255
  mriamtr: CP1255
  MRIAMTR: CP1255
# Narkisim:
  nrkis: CP1255
  NRKIS: CP1255
# Rod:
  rod: CP1255
  ROD: CP1255
  rodtr: CP1255
  RODTR: CP1255
```

epilogue.cfg

```
% epilogue.cfg
% =====

% This is the last file to be loaded
% when Gerolf Markup Shredder is initialized.

# =====

% Undo change in prologue.cfg:

  \let \write \backupwrite

% Write encoding files according to loaded unicode rows (enabled by default):

\CODEPAGEencwrite % .enc files like cp125x.enc, i8859-x.enc
\UNICODEencwrite % .enc files with Unicode char numbers or glyph names

% Load third-party macros (disabled by default):

% \input anyother.tex

% Start GMS (enabled by default; expecting TeX input files if disabled):

\shredder
```

files.cfg

font.cfg

```
# font.cfg
# =====

# This file defines additional font transformations for Gerolf Markup Shredder

# Syntax:

# [1] base name of font file (e.g. 'cour' from 'cour.ttf')
# [2] encoding
# [3] core/embed/corefamily/embedfamily mark
# [4] extension factor (must be set to 1.0 for True Type fonts)
# [5] slanting factor (must be set to 0.0 for True Type fonts)
# [6] new name (or 'none')
# [7] new suffix (optional, used to mark encoding in new font file name)

# Lines for alphabets that are not needed should be commented out or deleted
# in order to save mamory.
# The "W" in "W1", "W3" etc. stands for "windows-125" etc.

# =====

# Arabic:

# Monotype and Bigelow & Holmes use other glyph names than those in the
# Adobe Glyph List, so different encoding files are required.
# (Pixme: Could not identify tatweel, Arabic comma, Arabic semicolon,
# Arabic question mark. This must be corrected in ArabicMT.enc)

# Arial:
    arial CP1256 embedfamily 1.0 0.0 none W6
# CourierNew:
    cour CP1256 embedfamily 1.0 0.0 none W6
# LucidaSansTypewriter:
    hlsbot ArabicBH embed 1.0 0.0 hlsbota
    hlsbt ArabicBH embed 1.0 0.0 hlsbta
    hlsot ArabicBH embed 1.0 0.0 hlsota
    hlsrt ArabicBH embed 1.0 0.0 hlsrta
# Tahoma:
    tahoma CP1256 embedfamily 1.0 0.0 none W6
# TimesNewRoman:
    times CP1256 embedfamily 1.0 0.0 none W6
# Verdana:
    verdana CP1256 embedfamily 1.0 0.0 none W6

# Cyrillic: -----

# Arial:
    arial CP1251 embedfamily 1.0 0.0 none W1
# Baskerville:
    basker CP1251 embedfamily 1.0 0.0 none W1
# CenturyGothic:
    gothic CP1251 embedfamily 1.0 0.0 none W1
# CMSuperCaps:
    fcmcc CP1251 embed 1.0 0.0 fcmccW1
    fcmcc CP1251 embed 1.0 0.2 fcmccoW1
    fcmxc CP1251 embed 1.0 0.0 fcmxcW1
    fcmxc CP1251 embed 1.0 0.2 fcmxcoW1
# CMSuperDunhill:
    fcmdh CP1251 embed 1.0 0.0 fcmdhxW1
    fcmdh CP1251 embed 1.0 0.2 fcmdhxoW1
# CMSuperFibonacci:
    fcmfb CP1251 embed 0.8 0.0 fcmfbcW1
    fcmfb CP1251 embed 0.8 0.2 fcmfbcO1
# CMSuperFunny:
    fcmff CP1251 embed 1.3 0.1 fcmffuW1
    fcmfi CP1251 embed 1.05 0.1 fcmffiW1
# CMSuperRoman:
    fcmr CP1251 embed 1.0 0.0 fcmrW1

    fcmti CP1251 embed 1.0 0.0 fcmtiW1
    fcmbr CP1251 embed 1.0 0.0 fcmbrW1
    fcmbi CP1251 embed 1.0 0.0 fcmbiW1
# CMSuperSans:
    fcmsg CP1251 embed 1.0 0.0 fcmsgW1
    fcmsg CP1251 embed 1.0 0.2 fcmsgoW1
    fcmsx CP1251 embed 1.0 0.0 fcmsxW1
    fcmsx CP1251 embed 1.0 0.2 fcmsxoW1
# CMSuperTypewriter:
    fcmtt CP1251 embed 1.0 0.0 fcmttW1
    fcmtt CP1251 embed 1.0 0.2 fcmttoW1
# CMSuperVariable:
    fcmvt CP1251 embed 1.0 0.0 fcmvtW1
    fcmvt CP1251 embed 1.0 0.2 fcmvtoW1
# ComicSans:
    comic CP1251 embedfamily 1.0 0.0 none W1
# CourierNew:
    cour CP1251 embedfamily 1.0 0.0 none W1
# FranklinGothic:
    framd CP1251 embedfamily 1.0 0.0 none W1
# Garamond:
    gara CP1251 embedfamily 1.0 0.0 none W1
# Georgia:
    georgia CP1251 embedfamily 1.0 0.0 none W1
# Impact:
    impact CP1251 embed 1.0 0.0 none W1
# Sylfaen:
    sylfaen CP1251 embed 1.0 0.0 none W1
# Tahoma:
    tahoma CP1251 embedfamily 1.0 0.0 none W1
# TimesNewRoman:
    times CP1251 embedfamily 1.0 0.0 none W1
# Trebuchet:
    trebuc CP1251 embedfamily 1.0 0.0 none W1
# Verdana:
    verdana CP1251 embedfamily 1.0 0.0 none W1

# Greek: -----

# Arial:
    arial CP1253 embedfamily 1.0 0.0 none W3
# Cardo:
    cardo CP1253 embed 1.0 0.0 none W3
# Century Gothic:
    gothic CP1253 embedfamily 1.0 0.0 none W3
# ComicSans:
    comic CP1253 embedfamily 1.0 0.0 none W3
# CourierNew:
    cour CP1253 embedfamily 1.0 0.0 none W3
# FranklinGothic:
    framd CP1253 embedfamily 1.0 0.0 none W3
# Garamond:
    gara CP1253 embedfamily 1.0 0.0 none W3
# Gentium:
    gen CP1253 embedfamily 1.0 0.0 none W3
# Georgia:
    georgia CP1253 embedfamily 1.0 0.0 none W3
# Impact:
    impact CP1253 embed 1.0 0.0 none W3
# LucidaSansTypewriter:
    hls CP1253 embedfamily 1.0 0.0 none W3
# LucidaConsole:
    lucon CP1253 embed 1.0 0.0 none W3
# LucidaSans:
    l_10646 CP1253 embed 1.0 0.0 none W3
# Sylfaen:
    sylfaen CP1253 embed 1.0 0.0 none W3
# Tahoma:
    tahoma CP1253 embedfamily 1.0 0.0 none W3
```

```

    tahoma CP1253 embedfamily 1.0 0.0 none W3
# TimesNewRoman:
    times CP1253 embedfamily 1.0 0.0 none W3
# Trebuchet:
    trebuc CP1253 embedfamily 1.0 0.0 none W3
# Verdana:
    verdana CP1253 embedfamily 1.0 0.0 none W3
# Hebrew: -----
# Arial:
    arial CP1255 embedfamily 1.0 0.0 none W5
# CourierNew:
    cour CP1255 embedfamily 1.0 0.0 none W5
# LucidaSans:
    l_10646 CP1255 embed 1.0 0.0 none W5
# Tahoma:
    tahoma CP1255 embedfamily 1.0 0.0 none W5
# TimesNewRoman:
    times CP1255 embedfamily 1.0 0.0 none W5
# Verdana:
    verdana CP1255 embedfamily 1.0 0.0 none W5
# Latin: -----
# Antiqua:
    uaqr %GMS_CODEPAGE% embed 1.1 0.0 uaqr
    uaqr %GMS_CODEPAGE% embed 1.1 0.2 uaqr
# Augie:
    faum %GMS_CODEPAGE% embed 0.9 0.1 faumn
    faum %GMS_CODEPAGE% embed 0.9 0.5 faumno
# Avantgarde:
    uagd %GMS_CODEPAGE% embed 1.0 0.25 uagdo
    uagk %GMS_CODEPAGE% embed 1.0 0.25 uagko
# BrushScript:
    fbsmi %GMS_CODEPAGE% embed 1.0 -0.4 fbsm
# CMSuperCaps:
    fcmcc %GMS_CODEPAGE% embed 1.0 0.2 fcmcco
    fcmxc %GMS_CODEPAGE% embed 1.0 0.2 fcmxco
# CMSuperDunhill:
    fcmdh %GMS_CODEPAGE% embed 1.0 0.0 fcmdhx
    fcmdh %GMS_CODEPAGE% embed 1.0 0.2 fcmdhxo
# CMSuperFibonacci:
    fcmfb %GMS_CODEPAGE% embed 0.8 0.0 fcmfbc
    fcmfb %GMS_CODEPAGE% embed 0.8 0.2 fcmfbco
# CMSuperFunny:
    fcmff %GMS_CODEPAGE% embed 1.3 0.1 fcmffu
    fcmfi %GMS_CODEPAGE% embed 1.05 0.1 fcmffi
# CMSuperSans:
    fcms %GMS_CODEPAGE% embed 1.0 0.2 fcmsso
    fcmsx %GMS_CODEPAGE% embed 1.0 0.2 fcmsxo
# CMSuperTypewriter:
    fcmtt %GMS_CODEPAGE% embed 1.0 0.2 fcmtto
# CMSuperVariable:
    fcmvt %GMS_CODEPAGE% embed 1.0 0.2 fcmvto
# DayRoman:
    fdrr %GMS_CODEPAGE% embed 1.0 0.2 fdrrro
# DayRomanExpert:
    fdxr %GMS_CODEPAGE% embed 1.0 0.2 fdxrro
# Grotesk:
    ugkb %GMS_CODEPAGE% embed 1.0 0.2 ugkbro
# Gutenberg:
    # fgcm %GMS_CODEPAGE% embed 1.4 0.0 fgcmx
    # fgcm %GMS_CODEPAGE% embed 1.4 0.2 fgcmxo
    fgcm %GMS_CODEPAGE% embed 1.0 0.0 fgcmx
    fgcm %GMS_CODEPAGE% embed 1.0 0.2 fgcmxo
# IMBCourier (CourierNarrow):
    ncrb %GMS_CODEPAGE% embed 0.75 0 ncrbc
    ncrb %GMS_CODEPAGE% embed 0.75 0.25 ncrbco
    ncrb %GMS_CODEPAGE% embed 0.75 0 ncrbc
    ncrb %GMS_CODEPAGE% embed 0.75 0.25 ncrbcro
# IMBCourier (CourierSlim):
    ncrb %GMS_CODEPAGE% embed 0.5 0 ncrbc
    ncrb %GMS_CODEPAGE% embed 0.5 0.25 ncrbco
    ncrb %GMS_CODEPAGE% embed 0.5 0 ncrbc
    ncrb %GMS_CODEPAGE% embed 0.5 0.25 ncrbcro
# LuxiMono:
    hlmb %GMS_CODEPAGE% embed 1.0 0.25 hlmbco
    hlmb %GMS_CODEPAGE% embed 1.0 0.25 hlmbro
# LuxiMonoNarrow:
    hlmb %GMS_CODEPAGE% embed 0.75 0.0 hlmbc
    hlmb %GMS_CODEPAGE% embed 0.75 0.25 hlmbco
    hlmb %GMS_CODEPAGE% embed 0.75 0.0 hlmbc
    hlmb %GMS_CODEPAGE% embed 0.75 0.25 hlmbco
# Mono:
    ucrb %GMS_CODEPAGE% embed 0.75 0 ucrbc
    ucrb %GMS_CODEPAGE% embed 0.75 0.25 ucrbco
    ucrb %GMS_CODEPAGE% embed 0.75 0 ucrbc
    ucrb %GMS_CODEPAGE% embed 0.75 0.25 ucrbco
# NationalFirst:
    fnfr %GMS_CODEPAGE% embed 1.2 0.0 fnfrx
    fnfr %GMS_CODEPAGE% embed 1.2 0.2 fnfrxo
# Sanity:
    fsab %GMS_CODEPAGE% embed 1.1 0.0 fsabx
    fsab %GMS_CODEPAGE% embed 1.1 0.2 fsabxo
    fsar %GMS_CODEPAGE% embed 1.2 0.0 fsarx
    fsar %GMS_CODEPAGE% embed 1.2 0.2 fsarxo
# Thai: -----
# Courier Mono/Proportional Thai replace Latin supplement A (CP1252)
# by CP874 Thai characters without changing glyph names:
# CourierMonoThai:
    crmthai CP1252 embed 1.0 0.0 crmth
# CourierProportionalThai:
    crpthai CP1252 embed 1.0 0.0 crpth
# Unicode Fonts: -----
# Arial:
    arial G0000 embedfamily 1.0 0.0 none 00
    arial G0100 embedfamily 1.0 0.0 none 01
    arial G0200 embedfamily 1.0 0.0 none 02
    arial G0300 embedfamily 1.0 0.0 none 03
    arial G0400 embedfamily 1.0 0.0 none 04
    arial G0500 embedfamily 1.0 0.0 none 05
    arial G0600 embedfamily 1.0 0.0 none 06
    arial G1E00 embedfamily 1.0 0.0 none 1E
    arial G2000 embedfamily 1.0 0.0 none 20
    arial G2100 embedfamily 1.0 0.0 none 21
    arial G2200 embedfamily 1.0 0.0 none 22
    arial G2300 embedfamily 1.0 0.0 none 23
    arial G2500 embedfamily 1.0 0.0 none 25
    arial G2600 embedfamily 1.0 0.0 none 26
# CourierNew:
    cour G0000 embedfamily 1.0 0.0 none 00
    cour G0100 embedfamily 1.0 0.0 none 01
    cour G0200 embedfamily 1.0 0.0 none 02
    cour G0300 embedfamily 1.0 0.0 none 03
    cour G0400 embedfamily 1.0 0.0 none 04
    cour G0500 embedfamily 1.0 0.0 none 05
    cour G0600 embedfamily 1.0 0.0 none 06
    cour G1E00 embedfamily 1.0 0.0 none 1E
    cour G2000 embedfamily 1.0 0.0 none 20
    cour G2100 embedfamily 1.0 0.0 none 21
    cour G2200 embedfamily 1.0 0.0 none 22
    cour G2300 embedfamily 1.0 0.0 none 23
    cour G2500 embedfamily 1.0 0.0 none 25
    cour G2600 embedfamily 1.0 0.0 none 26

```

```

# TimesNewRoman:
times G0000 embedfamily 1.0 0.0 none 00
times G0100 embedfamily 1.0 0.0 none 01
times G0200 embedfamily 1.0 0.0 none 02
times G0300 embedfamily 1.0 0.0 none 03
times G0400 embedfamily 1.0 0.0 none 04
times G0500 embedfamily 1.0 0.0 none 05
times G0600 embedfamily 1.0 0.0 none 06
times G1E00 embedfamily 1.0 0.0 none 1E
times G2000 embedfamily 1.0 0.0 none 20
times G2100 embedfamily 1.0 0.0 none 21
times G2200 embedfamily 1.0 0.0 none 22
times G2300 embedfamily 1.0 0.0 none 23
times G2500 embedfamily 1.0 0.0 none 25
times G2600 embedfamily 1.0 0.0 none 26

# Vietnamese: -----
# Arial:
arial VISCII embedfamily 1.0 0.0 none VI
# CourierNew:
cour VISCII embedfamily 1.0 0.0 none VI
# Tahoma:
tahoma VISCII embedfamily 1.0 0.0 none VI
# TimesNewRoman:
times VISCII embedfamily 1.0 0.0 none VI
# Verdana:
verdana VISCII embedfamily 1.0 0.0 none VI

# -----
# A future version of GMS might support this:
# faum %GMS_CODEPAGE% embed 0.99 0.3 faumo-10
# faum %GMS_CODEPAGE% embed 1.01 0.3 faumo+10
# faum %GMS_CODEPAGE% embed 1.01 0.3 faumo+20
# faum %GMS_CODEPAGE% embed 0.08 0 faum -20
# faum %GMS_CODEPAGE% embed 0.09 0 none -10
# faum %GMS_CODEPAGE% embed 1.01 0 faum +10
# faum %GMS_CODEPAGE% embed 1.02 0 none +20
# ttf2afm cannot process ArialUnicodeMS:
# "No names available in 'post' table, printing by index forced"
# MAU G0000 embed 1.0 0.0 MAU-00
# pdfTeX 1.0b fails embedding Code2000 with error:
# "Unknown version of OS/2 table (3)"
# CODE2000 U0000 embed 1.0 0.0 C2-00
# CODE2000 U0100 embed 1.0 0.0 C2-01
# CODE2000 U0200 embed 1.0 0.0 C2-02
# CODE2000 U0300 embed 1.0 0.0 C2-03
# CODE2000 U2000 embed 1.0 0.0 C2-20
# This won't work as desired (run ttf2pt1 to convert True Type to PostScript):
# COMIC %GMS_CODEPAGE% embed 1.01 -0.1667 S -10

```

pdftex.cfg

```
% pdftex.cfg
% =====

% This file is part of Gerolf Markup Shredder.

% Documentation to the options below is available at:
% http://www.tug.org/applications/pdftex/pdftex-a.pdf

% =====

% Output file format:
% output_format 0 % DVI output
% output_format 1 % PDF output

% Output compression level:
% compress_level 0 % *no* compression (human-readable PDF)
% compress_level 1 % fast compression
% compress_level 9 % best compression

% Output pixel resolution:
% image_resolution 300
% pk_resolution 300

% Numerical precision:
% decimal_digits 3

% Printer offset:
% horigin 1in % TeX standard
% vorigin 1in

% Page size:
% page_height 297mm
% page_width 210mm

% Available fonts:
% map font.map % Generated by GMS
% map +cm.map % Computer Modern fonts (for plain.tex)
% map +extra.map % Additional user fonts
```


plugin.cfg

```
% plugin.cfg
% =====

% This TeX file was auto-
% generated by Gerolf Markup Shredder,
% written by G. D. Brettschneider (1999-2006),
% on Sat, 2008/01/05, 18:33:20 CET.
% All rights reserved.

% This file serves for loading all GMS
% modules that could be found in the
% appropriate plug-in directories.

% Initial data:
  \def \GMSdate {20080107}
  \def \GMSversion {0.08a}
  \def \GMScodepage {cp1252}
  \def \GMSdebug {0}
  \def \OS {WINNT}

\hyphenmessage
\message {Module loader.}
\fillmessage 42{modules}5

\fillmessage 6a{font}6
  \MAPload
  \hyphenmessage
  \message {{ font.map}
  \hyphenmessage
  \message {These names can be used as font face or}
  \echo {font-family style names in markup files:}
  \MAPfamilyshow
  \message {}
\fillmessage 6-{/font}5

\fillmessage 6b{kerning}3
  \KERNINGparseline (phv.krn)\relax
  \KERNINGparseline (ptm.krn)\relax
  \KERNINGloadtables
\fillmessage 6-{/kerning}2

\fillmessage 6c{language}2
  \LANGUAGEadd ca.tex
  \LANGUAGEadd cs.tex
  \LANGUAGEadd da.tex
  \LANGUAGEadd de-rf.tex
  \LANGUAGEadd de.tex
  \LANGUAGEadd el.tex
  \LANGUAGEadd en-UK.tex
  \LANGUAGEadd en-US.tex
  \LANGUAGEadd es.tex
  \LANGUAGEadd et.tex
  \LANGUAGEadd eu.tex
  \LANGUAGEadd fi.tex
  \LANGUAGEadd fr.tex
  \LANGUAGEadd ga.tex
  \LANGUAGEadd hr.tex
  \LANGUAGEadd hu.tex
  \LANGUAGEadd ia.tex
  \LANGUAGEadd id.tex
  \LANGUAGEadd is.tex
  \LANGUAGEadd it.tex
  \LANGUAGEadd la.tex
  \LANGUAGEadd nl.tex
  \LANGUAGEadd no.tex
  \LANGUAGEadd pl.tex

  \LANGUAGEadd pt.tex
  \LANGUAGEadd ru.tex
  \LANGUAGEadd sr.tex
  \LANGUAGEadd sv.tex
  \LANGUAGEadd tr.tex
  \LANGUAGEadd uk.tex
  \LANGUAGEadd wen.tex
  \LANGUAGEloadpatterns
  \message {More pattern files are available at
  http://www.ctan.org/tex-archive/language/}
\fillmessage 6-{/language}1

\fillmessage 6d{unicode}3
  \UNICODEadd u0000.row
  \UNICODEadd u0100.row
  \UNICODEadd u0200.row
  \UNICODEadd u0300.row
  \UNICODEadd u0400.row
  \UNICODEadd u0500.row
  \UNICODEadd u0600.row
  \UNICODEadd u0900.row
  \UNICODEadd u0e00.row
  \UNICODEadd u1e00.row
  \UNICODEadd u2000.row
  \UNICODEadd u2100.row
  \UNICODEadd u2200.row
  \UNICODEadd u2300.row
  \UNICODEadd u2400.row
  \UNICODEadd u2500.row
  \UNICODEadd u2600.row
  \UNICODEslotsload
\fillmessage 6-{/unicode}2

\fillmessage 6e{entity}4
  \ENTITYadd HTMLlat1.ent
  \ENTITYadd HTMLspec.ent
  \ENTITYadd HTMLsymb.ent
  \ENTITYnamesload
  \ENTITYnamesshow
\fillmessage 6-{/entity}3

\fillmessage 6f{glyph}5
  \GLYPHsadd dingbats.gly
  \GLYPHsadd gms.gly
  \GLYPHsnamesload
\fillmessage 6-{/glyph}4

\fillmessage 6g{codepage}2
  \CODEPAGEadd cp1250.txt
  \CODEPAGEadd cp1251.txt
  \CODEPAGEadd cp1252.txt
  \CODEPAGEadd cp1253.txt
  \CODEPAGEadd cp1254.txt
  \CODEPAGEadd cp1255.txt
  \CODEPAGEadd cp1256.txt
  \CODEPAGEadd cp1257.txt
  \CODEPAGEadd cp1258.txt
  \CODEPAGEadd cp437.txt
  \CODEPAGEadd cp850.txt
  \CODEPAGEadd cp860.txt
  \CODEPAGEadd cp863.txt
  \CODEPAGEadd cp865.txt
  \CODEPAGEadd cp874.txt
  \CODEPAGEadd i8859-1.txt
  \CODEPAGEadd i8859-10.txt
  \CODEPAGEadd i8859-11.txt
```

```
\CODEPAGEadd i8859-13.txt
\CODEPAGEadd i8859-14.txt
\CODEPAGEadd i8859-15.txt
\CODEPAGEadd i8859-2.txt
\CODEPAGEadd i8859-3.txt
\CODEPAGEadd i8859-4.txt
\CODEPAGEadd i8859-5.txt
\CODEPAGEadd i8859-6.txt
\CODEPAGEadd i8859-7.txt
\CODEPAGEadd i8859-8.txt
\CODEPAGEadd i8859-9.txt
```

```
\CODEPAGEadd iscii.txt
\CODEPAGEadd iso646.txt
\CODEPAGEadd us-ascii.txt
\CODEPAGEadd viscii.txt
\CODEPAGEloadthem
\CODEPAGEencwrite
\expandafter \CODEPAGEenable \CODEPAGE \relax
\fillmessage 6-{/codepage}1

\fillmessage 4-{/modules}3
\endinginput
```

prologue.cfg

```
% prologue.cfg
% =====

% This TeX file permits loading of third-party macros before
% Gerolf Markup Shredder gets initialized.
% GMS uses macros defined in Donald Knuth's 'plain.tex'.
% The file 'hyphen.tex' which is loaded by 'plain.tex'
% should be set to '\endinginput'.

% =====

% Basic character categories of plain.tex:

\catcode\{=1 % begin-group
\catcode\}=2 % end-group
\catcode\#=6 % macro parameter
\catcode \@ = 11 % letter

% Disable plain.tex's register logging (which makes 'gerolf.log' unreadable):

\let \backupwrite \write
\countdef \m@ne = 22 \m@ne = -1

\def \write #1#2{%
  \edef \first {#1}%
  \def \test {\m@ne}%
  \ifx \first \test \else \immediate \backupwrite #1{#2}\fi}

% Disable preloading of 'more fonts' in 'plain.tex':

\let \backupfont \font
\long \def \gulp #1cmsbx10 scaled \magstep 2{}

\def \gulpfont {%
  \ifx \second \test \let \next \gulp \else
  \let \next \relax \fi \next}

\def \testfont {%
  \ifx \second \test \let \next \dofont \else
  \let \next \relax \fi \next}

\def \font #1=#2 {%
  \def \second {#2}%
  \def \dofont {\backupfont #1=#2}%
  \def \test {cmr10}\testfont
  \def \test {cmr7}\testfont
  \def \test {cmr5}\testfont
  \def \test {cmmi10}\testfont

\def \test {cmmi7}\testfont
\def \test {cmmi5}\testfont
\def \test {cmbx10}\testfont
\def \test {cmbx7}\testfont
\def \test {cmbx5}\testfont
\def \test {cmsy10}\testfont
\def \test {cmsy7}\testfont
\def \test {cmsy5}\testfont
\def \test {cmex10}\testfont
\def \test {cmsl10}\testfont
\def \test {cmti10}\testfont
\def \test {cmtt10}\testfont
\def \test {cmdunh10}\gulpfont}

% Load plain.tex:

% Another wrapper for plain.tex (not necessary, disabled by default:
% \input etex.src

% Required (but may be loaded via etex.src), enabled by default:
\input plain.tex

% Hide above definitions:

\let \font \backupfont
\let \backupfont \undefined
\let \gulp \undefined
\let \gulpfont \undefined
\let \testfont \undefined

% Load other TeX macro definitions (disabled by default):

% \input anyother.tex

% Now comes Markup Shredder code.

% This will display register numbers in 'gerolf.log' if defined
% (only for debugging purposes, disabled by default):

% \def \GMSdisplayregs {1}

% This will switch off a few GMS features if defined
% (only for debugging purposes, disabled by default):

% \def \GMSfastini {1}

% \endinginput
```

tidy.cfg

```
# tidy.cfg
# =====

# HTML Tidy configuration file for Gerolf Markup Shredder.
# HTML Tidy was originally written by Dave Raggett.
# Documentation: http://www.w3.org/People/Raggett/tidy
# CFG Reference: http://tidy.sourceforge.net/docs/quickref.html (quoted below)
# Any option name must start at the very beginning of the line to become valid

# =====

# HTML, XHTML, XML Options:

# add-xml-decl: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should add the XML declaration when
# outputting XML or XHTML. Note that if the input already includes an
# <?xml ... ?> declaration then this option will be ignored.

# add-xml-decl: no

# add-xml-pi: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option is the same as the add-xml-decl option.

# add-xml-pi: no

# add-xml-space: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should add xml:space="preserve" to elements
# such as <PRE>, <STYLE> and <SCRIPT> when generating XML. This is needed if
# the whitespace in such elements is to be parsed appropriately without
# having access to the DTD.

# add-xml-space: no

# alt-text: .....

# Type: String
# Default: -none-
# This option specifies the default "alt=" text Tidy uses for <IMG>
# attributes. This feature is dangerous as it suppresses further
# accessibility warnings. You are responsible for making your documents
# accessible to people who can not see the images!

# alt-text: ""

# assume-xml-procins: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should change the parsing of processing
# instructions to require ?> as the terminator rather than >. This option is
# automatically set if the input is in XML.

# assume-xml-procins: no

# bare: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should strip Microsoft specific HTML from
# Word 2000 documents, and output spaces rather than non-breaking spaces
# where they exist in the input.

# bare: yes

# break-before-br: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should output a line break before each <BR>
# element.

# break-before-br: no

# clean: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should strip out surplus presentational tags
# and attributes replacing them by style rules and structural markup as
# appropriate. It works well on the HTML saved by Microsoft Office products.

# clean: no

# doctype: .....

# Type: DocType
# Default: auto
# Example: auto, omit, strict, loose, transitional, user specified fpi
# (string)
# This option specifies the DOCTYPE declaration generated by Tidy. If set to
# "omit" the output won't contain a DOCTYPE declaration. If set to "auto"
# (the default) Tidy will use an educated guess based upon the contents of
# the document. If set to "strict", Tidy will set the DOCTYPE to the strict
# DTD. If set to "loose", the DOCTYPE is set to the loose (transitional)
# DTD. Alternatively, you can supply a string for the formal public
# identifier (FPI), for example: "-//ACME/DTD HTML 3.14159/EN".
# If you specify the FPI for an XHTML document, Tidy will set the system
# identifier to the empty string. Tidy leaves the DOCTYPE for generic XML
# documents unchanged. "--doctype omit" implies "--numeric-entities yes".

# doctype: auto

# drop-empty-paras: .....

# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should discard empty paragraphs. If set to
# no, empty paragraphs are replaced by a pair of <BR> elements as HTML4
# precludes empty paragraphs.

# drop-empty-paras: yes

# drop-font-tags: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should discard <FONT> and <CENTER> tags
# rather than creating the corresponding style rules, but only if the clean
```

```

# option is also set to yes.
# drop-font-tags: no
# drop-proprietary-attributes: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should strip out proprietary attributes,
# such as MS data binding attributes.
# drop-proprietary-attributes: no
# enclose-block-text: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should insert a <P> element to enclose any
# text it finds in any element that allows mixed content for HTML
# transitional but not HTML strict.
# enclose-block-text: no
# enclose-text: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should enclose any text it finds in the body
# element within a <P> element. This is useful when you want to take
# existing HTML and use it with a style sheet.
# enclose-text: no
# escape-cdata: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should convert <![CDATA[]]> sections to
# normal text.
# escape-cdata: no
# fix-bad-comments: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should replace unexpected hyphens with "-"
# characters when it comes across adjacent hyphens. The default is yes. This
# option is provided for users of Cold Fusion which uses the comment syntax:
# <!-- --->
# fix-bad-comments: yes
# fix-uri: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should check attribute values that carry
# URIs for illegal characters and if such are found, escape them as HTML 4
# recommends.
# fix-uri: yes
# hide-comments: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should print out comments.
# hide-comments: no
# hide-endtags: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should omit optional end-tags when
# generating the pretty printed markup. This option is ignored if you are
# outputting to XML.
# hide-endtags: no
# indent-cdata: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should indent <![CDATA[]]> sections.
# indent-cdata: no
# input-xml: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should use the XML parser rather than the
# error correcting HTML parser.
# input-xml: no
# join-classes: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0 join-styles
# repeated-attributes
# This option specifies if Tidy should combine class names to generate a
# single new class name, if multiple class assignments are detected on an
# element.
# join-classes: no
# join-styles: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0 join-classes
# repeated-attributes
# This option specifies if Tidy should combine styles to generate a single
# new style, if multiple style values are detected on an element.
# join-styles: yes
# logical-emphasis: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should replace any occurrence of <I> by <EM>
# and any occurrence of <B> by <STRONG>. In both cases, the attributes are
# preserved unchanged. This option can be set independently of the clean and
# drop-font-tags options.
# logical-emphasis: no

```

```

# lower-literals: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should convert the value of an attribute
# that takes a list of predefined values to lower case. This is required for
# XHTML documents.

# lower-literals: yes

# ncr: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should allow numeric character references.

# ncr: yes

# new-blocklevel-tags: .....
# Type: Tag names
# Default: -none-
# Example: tagX, tagY, ...
# This option specifies new block-level tags. This option takes a space or
# comma separated list of tag names. Unless you declare new tags, Tidy will
# refuse to generate a tidied file if the input includes previously unknown
# tags. Note you can't change the content model for elements such as
# <TABLE>, <UL>, <OL> and <DL>.

# new-empty-tags: .....
# Type: Tag names
# Default: -none-
# Example: tagX, tagY, ...new-blocklevel-tags
# new-inline-tags
# This option specifies new empty inline tags. This option takes a space or
# comma separated list of tag names. Unless you declare new tags, Tidy will
# refuse to generate a tidied file if the input includes previously unknown
# tags. Remember to also declare empty tags as either inline or blocklevel.

# new-empty-tags: pageno

# new-inline-tags: .....
# Type: Tag names
# Default: -none-
# Example: tagX, tagY, ...
# This option specifies new non-empty inline tags. This option takes a space
# or comma separated list of tag names. Unless you declare new tags, Tidy
# will refuse to generate a tidied file if the input includes previously
# unknown tags.

# new-pre-tags: .....
# Type: Tag names
# Default: -none-
# Example: tagX, tagY, ...
# This option specifies new tags that are to be processed in exactly the
# same way as HTML's <PRE> element. This option takes a space or comma
# separated list of tag names. Unless you declare new tags, Tidy will refuse
# to generate a tidied file if the input includes previously unknown tags.
# Note you can not as yet add new CDATA elements (similar to <SCRIPT>).

# numeric-entities: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0

# This option specifies if Tidy should output entities other than the
# built-in HTML entities (&amp;, &lt;, &gt; and &quot;) in the numeric
# rather than the named entity form.

# numeric-entities: no

# output-xhtml: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should generate pretty printed output,
# writing it as extensible HTML. This option causes Tidy to set the DOCTYPE
# and default namespace as appropriate to XHTML. If a DOCTYPE or namespace
# is given they will be checked for consistency with the content of the
# document. In the case of an inconsistency, the corrected values will
# appear in the output. For XHTML, entities can be written as named or
# numeric entities according to the setting of the "numeric-entities"
# option. The original case of tags and attributes will be preserved,
# regardless of other options.

# output-xhtml: no

# output-xml: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should pretty print output, writing it as
# well-formed XML. Any entities not defined in XML 1.0 will be written as
# numeric entities to allow them to be parsed by a XML parser. The original
# case of tags and attributes will be preserved, regardless of other
# options.

# output-xml: no

# quote-ampersand: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should output unadorned & characters as
# &amp;.

# quote-ampersand: yes

# quote-marks: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should output " characters as &quot; as is
# preferred by some editing environments. The apostrophe character ' is
# written out as &#39; since many web browsers don't yet support &apos;.

# quote-marks: no

# quote-nbsp: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should output non-breaking space characters
# as entities, rather than as the Unicode character value 160 (decimal).

# quote-nbsp: yes

# repeated-attributes: .....
# Type: -
# Default: keep-last

```

```

# Example: keep-first, keep-lastjoin-classes
# join-styles
# This option specifies if Tidy should keep the first or last attribute, if
# an attribute is repeated, e.g. has two align attributes.

# repeated-attributes: keep-last

# replace-color: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should replace numeric values in color
# attributes by HTML/XHTML color names where defined, e.g. replace "#ffffff"
# with "white".

# replace-color: no

# show-body-only: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should print only the contents of the body
# tag as an HTML fragment. Useful for incorporating existing whole pages as
# a portion of another page.

# show-body-only: no

# slide-style: .....
# Type: Name
# Default: none
# split
# Currently not used.

# split: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should create a sequence of slides from the
# input, splitting the markup prior to each successive <H2>. The slides are
# written to "slide001.html", "slide002.html" etc.

# split: no

# uppercase-attributes: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should output attribute names in upper case.
# The default is no, which results in lower case attribute names, except for
# XML input, where the original case is preserved.

# uppercase-attributes: no

# uppercase-tags: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should output tag names in upper case. The
# default is no, which results in lower case tag names, except for XML
# input, where the original case is preserved.

# uppercase-tags: no

# word-2000: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should go to great pains to strip out all
# the surplus stuff Microsoft Word 2000 inserts when you save Word documents
# as "Web pages". Doesn't handle embedded images or VML.

# word-2000: yes

# Diagnostics Options: =====
# gnu-emacs: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should change the format for reporting
# errors and warnings to a format that is more easily parsed by GNU Emacs.

# gnu-emacs: no

# quiet: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should output the summary of the numbers of
# errors and warnings, or the welcome or informational messages.

# quiet: no

# show-errors: .....
# Type: Integer
# Default: 6
# Example: 0, 1, 2, ...
# This option specifies the number Tidy uses to determine if further errors
# should be shown. If set to 0, then no errors are shown.

# show-errors: 6

# show-warnings: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should suppress warnings. This can be useful
# when a few errors are hidden in a flurry of warnings.

# show-warnings: yes

# Pretty Print Options: =====
# indent:
# Type: AutoBool
# Default: no
# Example: auto, y/n, yes/no, t/f, true/false, 1/0indent-spaces
# This option specifies if Tidy should indent block-level tags. If set to
# "auto", this option causes Tidy to decide whether or not to indent the
# content of tags such as TITLE, H1-H6, LI, TD, TD, or P depending on
# whether or not the content includes a block-level element. You are advised
# to avoid setting indent to yes as this can expose layout bugs in some
# browsers.

indent: yes

# indent-attributes: .....
# Type: Boolean
# Default: no

```

```

# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should begin each attribute on a new line.

# indent-attributes: no

# indent-spaces: .....
# Type: Integer
# Default: 2
# Example: 0, 1, 2, ...indent
# This option specifies the number of spaces Tidy uses to indent content,
# when indentation is enabled.

# indent-spaces: 2

# literal-attributes: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should ensure that whitespace characters
# within attribute values are passed through unchanged.

# literal-attributes: no

# markup: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should generate a pretty printed version of
# the markup. Note that Tidy won't generate a pretty printed version if it
# finds significant errors (see force-output).

# markup: no

# tab-size: .....
# Type: Integer
# Default: 4
# Example: 0, 1, 2, ...
# This option specifies the number of columns that Tidy uses between
# successive tab stops. It is used to map tabs to spaces when reading the
# input. Tidy never outputs tabs.

tab-size: 2

# wrap: .....
# Type: Integer
# Default: 68
# Example: 0 (no wrapping), 1, 2, ...
# This option specifies the right margin Tidy uses for line wrapping. Tidy
# tries to wrap lines so that they do not exceed this length. Set wrap to
# zero if you want to disable line wrapping.

wrap: 78

# wrap-asp: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should line wrap text contained within ASP
# pseudo elements, which look like: <% ... %>.

# wrap-asp: yes

# wrap-attributes: .....
# Type: Boolean

# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0wrap-script-literals
# This option specifies if Tidy should line wrap attribute values, for
# easier editing. This option can be set independently of
# wrap-script-literals.

# wrap-attributes: yes

# wrap-jste: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should line wrap text contained within JSTE
# pseudo elements, which look like: <# ... #>.

# wrap-jste: yes

# wrap-php: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should line wrap text contained within PHP
# pseudo elements, which look like: <?php ... ?>.

# wrap-php: yes

# wrap-script-literals: .....
# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0wrap-attributes
# This option specifies if Tidy should line wrap string literals that appear
# in script attributes. Tidy wraps long script string literals by inserting
# a backslash character before the line break.

# wrap-script-literals: no

# wrap-sections: .....
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should line wrap text contained within <![
# ... ]> section tags.

# wrap-sections: yes

# Character Encoding Options: =====
# ascii-chars:
# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# Can be used to modify behavior of -c (--clean yes) option. Defaults to
# "yes" when using -c. Set to "no" to prevent converting &ndash; &rdquo;,
# and other named character entities to their ascii equivalents.

# ascii-chars: no

# char-encoding: .....
# Type: Encoding
# Default: ascii
# Example: ascii, latin1, raw, utf8, iso2022, mac, win1252, utf16, utf16le,
# utf16be, big5, shiftjis
# This option specifies the character encoding Tidy uses for both the input
# and output. For ascii, Tidy will accept Latin-1 (ISO-8859-1) character
# values, but will use entities for all characters whose value > 127. For

```



```

# raw, Tidy will output values above 127 without translating them into
# entities. For latin1, characters above 255 will be written as entities.
# For utf8, Tidy assumes that both input and output is encoded as UTF-8. You
# can use iso2022 for files encoded using the ISO-2022 family of encodings
# e.g. ISO-2022-JP. For mac and win1252, Tidy will accept vendor specific
# character values, but will use entities for all characters whose value >
# 127.

# char-encoding: win1252

# input-encoding: .....

# Type: Encoding
# Default: latin1
# Example: ascii, latin1, raw, utf8, iso2022, mac, win1252, utf16, utf16le,
# utf16be, big5, shiftjis
# This option specifies the character encoding Tidy uses for the input. See
# char-encoding for more info.

# input-encoding: win1252

# language: .....

# Type: Language
# Default: -none-
# Example: en
# Currently not used, but this option specifies the language Tidy uses.

# output-bom: .....

# Type: AutoBool
# Default: auto
# Example: auto, y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should write a Unicode Byte Order Mark
# character (BOM; also known as Zero Width No-Break Space; has value of
# U+FEFF) to the beginning of the output; only for UTF-8 and UTF-16 output
# encodings. If set to "auto", this option causes Tidy to write a BOM to the
# output only if a BOM was present at the beginning of the input. A BOM is
# always written for XML/XHTML output using UTF-16 output encodings.

# output-bom: auto

# output-encoding: .....

# Type: Encoding
# Default: ascii
# Example: ascii, latin1, raw, utf8, iso2022, mac, win1252, utf16, utf16le,
# utf16be, big5, shiftjis
# This option specifies the character encoding Tidy uses for the output. See
# char-encoding for more info. May only be different from input-encoding for
# Latin encodings (ascii, latin1, mac, win1252).

# output-encoding: win1252

# newline: .....

# Type: Enum
# Default: Platform Dependent
# Example: LF, CRLF, CR
# The default is appropriate to the current platform: CRLF on Windows and
# OS/2, CR on the Mac and LF everywhere else (Unix and Linux).

# newline: CRLF

# Miscellaneous Options: =====

# fix-backslash:

# Type: Boolean
# Default: yes

# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should replace backslash characters "\" in
# URLs by forward slashes "/".

# fix-backslash: yes

# output-file: .....

# Type: String
# Default: -none-
# error-file
# This option specifies the output file Tidy uses for markup. Normally
# markup is written to "stdout".

# error-file: .....

# Type: String
# Default: -none-
# output-file
# This option specifies the error file Tidy uses for errors and warnings.
# Normally errors and warnings are output to "stderr".

# force-output: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should produce output even if errors are
# encountered. Use this option with care - if Tidy reports an error, this
# means Tidy was not able to, or is not sure how to, fix the error, so the
# resulting output may not reflect your intention.

# force-output: no

# keep-time: .....

# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should alter the last modified time for
# files it writes back to. The default is no, which allows you to tidy files
# without affecting which ones will be uploaded to a Web server when using a
# tool such as 'SiteCopy'. Note that this feature may not work on some
# platforms.

# keep-time: yes

# write-back: .....

# Type: Boolean
# Default: no
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should write back the tidied markup to the
# same file it read from. You are advised to keep copies of important files
# before tidying them, as on rare occasions the result may not be what you
# expect.

# write-back: no

# tidy-mark: .....

# Type: Boolean
# Default: yes
# Example: y/n, yes/no, t/f, true/false, 1/0
# This option specifies if Tidy should add a meta element to the document
# head to indicate that the document has been tidied. Tidy won't add a meta
# element if one is already present.

tidy-mark: no

```

typeset.cfg

```
% typeset.cfg
% =====

% User-editable typesetting parameters.
% This file is part of Gerolf Markup Shredder (www.Gerolf.org)
% =====

\hyphenmessage

\message {- User-editable typesetting parameters,
as described by Donald Ervin Knuth}
\message {in "The TeXbook", pages 272 - 275, 348 - 349 and 451.}

% 1. Numbers: -----

% Logging:

\showboxbreadth = 5 % plain.tex
\showboxbreadth = 0 % GMS
\showboxdepth = 3 % plain.tex
\showboxdepth = 0 % GMS

\errorcontextlines = 5 % plain.tex / Knuth p. 34
\errorcontextlines = 0 % GMS
\maxdeadcycles = 25 % plain.tex

% Tracing:

\tracinglostchars = 1 % plain.tex
\tracingstats = 1 % GMS
\tracingcommands = 3 % GMS
\tracingmacros = 3 % GMS

% Box badness:

\hbadness = 1000 % plain.tex
\vbadness = 1000 % plain.tex

% Hyphenation tolerance:

\pretolerance = 100 % plain.tex
\pretolerance = 50 % GMS
\tolerance = 200 % plain.tex
\tolerance = 300 % GMS

% Hyphenation:

\lefthyphenmin = 2 % plain.tex
\righthyphenmin = 3 % plain.tex
\righthyphenmin = 2 % GMS & most pattern files

\uchyph = 1 % plain.tex

% Penalties:

\linepenalty = 10 % plain.tex
\hyphenpenalty = 50 % plain.tex
\hyphenpenalty = 10 % GMS
\exhyphenpenalty = 50 % plain.tex
\exhyphenpenalty = 10 % GMS

\binoppenalty = 700 % plain.tex
\relpenalty = 500 % plain.tex

\clubpenalty = 150 % plain.tex
\clubpenalty = 10000 % GMS

\widowpenalty = 150 % plain.tex
\widowpenalty = 10000 % GMS

\displaywidowpenalty = 50 % plain.tex
\brokenpenalty = 100 % plain.tex
\predisplaypenalty = 10000 % plain.tex

% Demerits:

\doublehyphendemerits = 10000 % plain.tex
\doublehyphendemerits = 10 % GMS
\finalhyphendemerits = 5000 % plain.tex
\finalhyphendemerits = 10 % GMS
\adjdemerits = 10000 % plain.tex
\adjdemerits = 150 % GMS

% Characters:

\defaultshyphenchar = \- % plain.tex
\defaultskewchar = -1 % plain.tex
\newlinechar = -1 % plain.tex

% Additional GMS parameters:

\pdfoutput = 1 % output (0: DVI, 1: PDF)
% Caution: KERNING = 2 may lead to hangup if font-size is reduced:
\KERNING = 1 % GMS (=pdfprotrudechars)
\FSIZEbase = 3 % 1 to 7, like deprecated <basefont size="n">
\CELLdefaultno = 3 % default number of table columns
\IMAGEResolution = 72 % dpi default

% Colors (RGB):

\COLORanchorex = {0.1 0 0.9} % hyperlinks to the internet (blue)
\COLORanchorint = {0 0.9 0.1} % hyperlinks to internal bookmarks (green)
\COLORulldot = {0.5 0.5 0.5} % unordered list dot (grey)

% Mathmode:

\delimiterfactor = 901 % plain.tex

% 2. Dimensions: -----

% Debugging:

\overfullrule = 5.0pt % plain.tex
\overfullrule = 0.0mm % GMS

% Maxima:

% The largest dimension TeX can handle is 16383.99999pt:

\boxmaxdepth = 16383.99999pt % plain.tex
\splitmaxdepth = 16383.99999pt % plain.tex

\maxdepth = 4.0pt % plain.tex

% Fuzz:

\hfuzz = 0.1pt % plain.tex
\hfuzz = 1.0pt % GMS
\vfuzz = 0.1pt % plain.tex

\emergencystretch = 1.0cm

% Delimiters:
```

```

\delimitershortfall = 5.0pt % plain.tex
\nulldelimiterspace = 1.2pt % plain.tex

\scriptspace = 0.5pt % plain.tex

% Sizes:

%\parindent = 20.0pt % plain.tex
\TEXTindent = 8.0mm % GMS (= \parindent)

% Implicit margins: 1.0in % TeX, see Knuth, page 251

%\hoffset = 0.0in % plain.tex
%\voffset = 0.0in % plain.tex
\hoffset = -0.4mm % GMS, for margins = 2.5cm
\voffset = -0.4mm % GMS

%\hsize = 6.5in % plain.tex (letter paper)
%\vsize = 8.9in % plain.tex
\hsize = 16.0cm % GMS ( A4 paper)
\vsize = 24.7cm % GMS

% Additional GMS parameters:

\PAGEwidth = 21.0cm % GMS ( A4 paper)
\PAGEheight = 29.7cm % GMS

\PAGEmargintop = 2.5cm % GMS
\PAGEmarginright = 2.5cm % GMS
\PAGEmarginbottom = 2.5cm % GMS
\PAGEmarginleft = 2.5cm % GMS

% Skips:

%\baselineskip = 12.0pt % plain.tex
\LINEheight = 4.0mm % GMS (= \baselineskip)
\lineskip = 0.0pt % plain.tex
\lineskiplimit = 0.0pt % plain.tex

\topskip = 10.0pt % plain.tex

\splittopskip = 10.0pt % plain.tex

%\parskip = 0pt plus 1pt % plain.tex
\parskip = 0pt % GMS
\parfillskip = 0pt plus 1fil % plain.tex

% Mathmode:

% Mathmode is not yet supported by Markup Shredder.
% A markup definition file for MathML must be written for this.

\abovedisplayskip = 12pt plus 3pt minus 9pt % plain.tex
\abovedisplayshortskip = 0pt plus 3pt % plain.tex

\belowdisplayskip = 12pt plus 3pt minus 9pt % plain.tex
\belowdisplayshortskip = 7pt plus 3pt minus 4pt % plain.tex

\thinmuskip = 3mu % plain.tex
\medmuskip = 4mu plus 2mu minus 4mu % plain.tex
\thickmuskip = 5mu plus 5mu % plain.tex

% Additional GMS parameter:

\FSIZE = 3.5mm

% 3. Strings: -----

% Additional GMS parameter:

% Default fonts:
\FONTfamily = {Times}
\FONTmono = {CourierNarrow}

% Fallback fonts:
\FONTmonospace = {CourierNew}
\FONTsansserif = {Arial}
\FONTserif = {TimesNewRoman}

\endinput

```

texmf.cnf

```

% texmf.cnf
% =====

% This file was adapted from kpathsea for use with Gerolf Markup Shredder.

% =====

% Non-path options:

% If a dynamic file creation fails, log the command to this file, in
% either the current directory or TEXMFOUTPUT. Set to the
% empty string or 0 to avoid logging.
MISSFONT_LOG = missfont.log

% Set to a colon-separated list of words specifying warnings to suppress.
% To suppress everything, use TEX_HUSH = all; this is equivalent to
% TEX_HUSH = checksum:lostchar:readable:special
TEX_HUSH = none

% Enable system commands via \write18{...}?
shell_escape = t

% Allow TeX \openin, \openout, or \input on filenames starting with `.'
% (e.g., .rhosts) or outside the current tree (e.g., /etc/passwd)?
% a (any)      : any file can be opened.
% r (restricted) : disallow opening "dotfiles".
% p (paranoid)  : as 'r' and disallow going to parent directories, and
%               : restrict absolute paths to be under $TEXMFOUTPUT.
openout_any = a
openin_any = a

% Allow TeX, MF, and MP to parse the first line of an input file for
% the %&format construct.
parse_first_line = f

% Enable the mktex... scripts by default? These must be set to 0 or 1.

MKTEXTEX = 0
MKTEXPK = 0
MKTEXMF = 0
MKTEXTFM = 0
MKOCP = 0
MKOPM = 0

% What MetaPost runs to make MPX files:
MPXCOMMAND = 0

% Array and other sizes for TeX: -----

% If you want to change some of these sizes only for a certain TeX
% variant, the usual dot notation works, e.g.,
% main_memory.hugetex = 20000000
%
% If a change here appears to be ignored, try redumping the format file.

% Memory. Must be less than 8,000,000 total.
%
% main_memory is relevant only to initex, extra_mem_* only to non-ini.
% Thus, have to redump the .fmt file after changing main_memory; to add
% to existing fmt files, increase extra_mem_*. (To get an idea of how
% much, try \tracingstats=2 in your TeX source file;
% web2c/tests/memtest.tex might also be interesting.)
%
% To increase space for boxes (as might be needed by, e.g., PiCTeX),
% increase extra_mem_bot.
%
% For some xy-pic samples, you may need as much as 700000 words of memory.
% For the vast majority of documents, 60000 or less will do.

%
main_memory = 1000000 % words of inmemory available; also applies to inimf&mp
extra_mem_top = 1000 % extra high memory for chars, tokens, etc.
extra_mem_bot = 1000 % extra low memory for boxes, glue, breakpoints, etc.

% Words of font info for TeX (total size of all TFM files, approximately).
font_mem_size = 400000

% Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
font_max = 1000

% Extra space for the hash table of control sequences:
%hash_extra = 15000
hash_extra = 65535

% Max number of characters in all strings, including all error messages,
% help texts, font names, control sequences.
% pool_size = 250000
pool_size = 500000
% Minimum pool space after TeX's own strings; must be at least
% 25000 less than pool_size, but doesn't need to be nearly that large.
string_vacancies = 25000
% Maximum number of strings.
max_strings = 30000
% min pool space left after loading .fmt
pool_free = 5000

% Hyphenation trie. As distributed, the maximum is 65535; this should
% work unless 'unsigned short' is not supported or is smaller than 16
% bits. This value should suffice for UK English, US English, French,
% and German (for example). To increase, you must change
% 'ssup_trie_opcode' and 'ssup_trie_size' in tex.ch (and rebuild TeX);
% the trie will then consume four bytes per entry, instead of two.
% US English: 10000.
%
trie_size = 196000

% Buffer size. TeX uses the buffer to contain input lines, but macro
% expansion works by writing material into the buffer and reparsing the
% line. As a consequence, certain constructs require the buffer to be
% very large. As distributed, the size is 50000; most documents can be
% handled within a tenth of this size.
buf_size = 50000

hyph_size = 1000 % number of hyphenation exceptions, >610 and <32767.
%nest_size = 100 % simultaneous semantic levels (e.g., groups)
nest_size = 255 % simultaneous semantic levels (e.g., groups)
% max_in_open = 15 % simultaneous input files and error insertions
max_in_open = 100
param_size = 4000 % simultaneous macro parameters
%save_size = 4000 % for saving values outside current group
save_size = 20000 % for saving values outside current group
stack_size = 4000 % simultaneous input sources

% These work best if they are the same as the I/O buffer size, but it
% doesn't matter much. Must be a multiple of 8.
dvi_buf_size = 16384 % TeX

% It's probably inadvisable to change these. At any rate, we must have:
% 45 < error_line < 255;
% 30 < half_error_line < error_line - 15;
% 60 <= max_print_line;
% error_line = 79
error_line = 128
%half_error_line = 50
half_error_line = 40
% max_print_line = 79
%max_print_line = 128
max_print_line = 79

```

Log Files

gerolf.log

```
This is pdfTeX, Version 3.141592-1.11a-2.1 (Web2c 7.5.2) (INITEX) 5 JAN 2008
18:33
entering extended mode
\write18 enabled.
***gerolf
(f:/wamp/www/tex/gerolf/gerolf.tex{f:/wamp/www/etc/pdftex.cfg}
(f:/wamp/www/etc/prologue.cfg (f:/wamp/www/tex/plain/plain.tex
Preloading the plain format: codes, registers, parameters, fonts, more fonts,
macros, math definitions, output routines, hyphenation
(f:/wamp/www/tex/plain/hyphen.tex ... skipped.))
<GMS>-----
This is Gerolf Markup Shredder, Version 0.08a, written by G. D. Brettschneider
--<initialize>-----
1 -<interface>-----
codepages, win-entities.
- Recognized attributes:
align, alt, background, bgcolor, border, cellpadding, cellspacing, class,
color, colspan, compact, content, data, dir, face, height, href, http-equiv,
id, lang, language, media, name, rel, rev, rowspan, size, span, src, start,
style, summary, text, title, valign, value, vlink, width, ...
- Pre-defined classes:
break, breakifeven, breakifodd, kerning, nofloat, noindent, nokerning, nolink,
nopagenumbers, noprint, noscreen, pagenumbers, twocolumns.
- Block-level elements:
<col /> <img /> <input />
- Inline-level elements:
- Unrendered elements:
<form> <link /> <meta />
- Recognized style properties:
background-color, background-color, background-image, background-repeat,
border-color, border-style, border-width, color, content, direction,
direction, display, float, font-family, font-size, font-size-adjust,
font-style, font-weight, height, line-height, margin, margin-bottom,
margin-left, margin-right, margin-top, orphans, padding-bottom, padding-left,
padding-top, padding-right, page-break-before, page-break-inside, size,
text-align, text-decoration, text-indent, vertical-align, widows, width, ...
- Gedcom XML elements:
-- a -<typeset>-----
(f:/wamp/www/etc/typeset.cfg -
- User-editable typesetting parameters, as described by Donald Ervin Knuth
in "The TeXbook", pages 272 - 275, 348 - 349 and 451.)
-----</typeset>-----
-- b -<alias>-----
(f:/wamp/www/etc/alias.cfg -
- User-defined font names (typeface aliases):
Arabic, Cyrillic, Greek, Hebrew, Indian, Latin, Thai, Unicode, Vietnamese.
Family aliases, family-specific encodings, encoding aliases.)
-----</alias>-----
fallback fonts: cmr (default), (monospace), (sans-serif), (serif);
---</interface>-----
(f:/wamp/www/etc/plugin.cfg - Module loader.
2 -<modules>-----
-- a -<font>-----
- (font.map - These names can be used as font face or
font-family style names in markup files:
\openout4 = `font.lst'.
system(sort < font.lst > sortfont.lst)...executed.
AgencyFB. Aharoni. Algerian. Andalus. AngsanaNew. AngsanaUPC. Antiqua.
ArabicTransparent. Arial. ArialArabic. ArialBlack. ArialCyrillic. ArialGreek.
ArialHebrew. ArialMT. ArialNarrow. ArialRoundedMTBold. ArialVietnam. Augie.
Avantgarde. AYummyApology. BaskOldFace. Bauhaus93. Beautiful. BeautifulES.
BellMT. BellMTBold. BellMTItalic. BerlinSansFB. BerlinSansFBDemi. BernardMT.
BlackadderITC. BodoniMT. BodoniMTBlack. BodoniMTCondensed.
BodoniMTPosterCompressed. BookAntiqua. Bookman. BookmanOldStyle.
BradleyHandITC. BritannicBold. Broadway. BrowalliaNew. BrowalliaUPC.
BrushScript. BrushScriptMT. CACChampagne. CalifornianFB. CalisMTBol.
CalistoMT. Cardo. CardoGreek. Castellar. Centaur. Century. CenturyGothic.
CenturySchl. CenturySchoolbook. Champagne. Charter. CharterBT. Chiller.
CMSuperCaps. CMSuperCapsCyrillic. CMSuperDunhill. CMSuperDunhillCyrillic.
CMSuperFibonacci. CMSuperFibonacciCyrillic. CMSuperFunny.
CMSuperFunnyCyrillic. CMSuperRoman. CMSuperRomanCyrillic. CMSuperSans.
CMSuperSansCyrillic. CMSuperTypewriter. CMSuperTypewriterCyrillic.
CMSuperVariable. CMSuperVariableCyrillic. ColonnaMT. ComicSans.
ComicSansCyrillic. ComicSansGreek. ComicSansMS. CooperBlack.
CopperplateGothic. CordiaNew. CordiaUPC. Courier. CourierArabic.
CourierCyrillic. CourierGreek. CourierHebrew. CourierMonoThai. CourierNarrow.
CourierNew. CourierNewPSMT. CourierProportionalThai. CourierSlim.
CourierVietnam. CourMonoThai. CourPropThai. CurlzMT. DataGlyph. David.
DavidTransparent. DayRoman. DayRomanExpert. DilleniaUPC. DilleniaUPCBold.
DilleniaUPCBoldItalic. DilleniaUPCItalic. EdwardianScriptITC. Elephant.
EngraversMT. ErasITC. EstrangeloEdessa. EucrosiaUPC. EucrosiaUPCBold.
EucrosiaUPCBoldItalic. EucrosiaUPCItalic. FelixTitlingMT.
FixedMiriamTransparent. FootlightMTLight. ForteMT. FranklinGothic.
FranklinGothicCyrillic. FranklinGothicGreek. FrankRuehl. FreesiaUPC.
FreesiaUPCBold. FreesiaUPCBoldItalic. FreestyleScript. FrenchScriptMT.
Garamond. GaramondCyrillic. GaramondGreek. Gautami. Gentium. GentiumGreek.
Georgia. GeorgiaCyrillic. GeorgiaGreek. Gigi. GillSans. GillSansMT.
GloucesterMT. GoodCityModern. GothicCyrillic. GothicGreek. GoudyOldStyleT.
GoudyStout. Grotesk. Gutenberg. Haettenschweiler. HarlowSolid. Harrington.
Helvetica. HighTowerText. IBMCourier. Impact. ImpactCyrillic. ImpactGreek.
ImprintMT. InformalRoman. IrisUPC. IrisUPCBold. IrisUPCBoldItalic.
IrisUPCItalic. JasmineUPC. Jokerman. JuiceITC. KodchiangUPC. KristenITC.
KunstlerScript. Latha. LatinWide. Levenim. LilyUPC. LilyUPCBold.
LilyUPCBoldItalic. LilyUPCItalic. LucidaBright. LucidaCalligraphy.
LucidaConsole. LucidaConsoleGreek. LucidaFax. LucidaHandwriting. LucidaSans.
LucidaSansGreek. LucidaSansUnicode. LuxiMono. LuxiMonoNarrow. Magneto.
MaiandraGD. Mangal. MaturaMTScriptCapitals. MicrosoftSansSerif. Miriam.
MiriamFixed. MiriamTransparent. Mistral. Modern. MonotypeCorsiva. MVBoli.
Narkisim. National. NationalFirst. Palatino. Raavi. Rod. RodTransparent.
Sanity. SanityWide. SanityWideBold. SFDH1000. SFFB1000. SFFF1000. SFFI1000.
Shree. ShreeAss001. ShreeAss002. ShreeBan001. ShreeBan002. ShreeDev001.
ShreeDev002. ShreeGuj001. ShreeGuj002. ShreeKan001. ShreeKan002. ShreeMal001.
ShreeMal002. ShreeOri002. ShreePun001. ShreePun002. ShreeTam001. ShreeTam002.
ShreeTel001. ShreeTel002. ShruTi. Shusha. SimplifiedArabic.
SimplifiedArabicFixed. Sylfaen. SylfaenGreek. Symbol. SymbolMT. Tahoma.
TahomaArabic. TahomaCyrillic. TahomaGreek. TahomaHebrew. TahomaVietnam. Times.
TimesArabic. TimesCyrillic. TimesGreek. TimesHebrew. TimesNewRoman.
TimesVietnam. TraditionalArabic. Trebuchet. TrebuchetCyrillic. TrebuchetGreek.
Tunga. Unknown. URWAntiquaT. URWBookmanL. URWPalladioL. Utopia. Verdana.
VerdanaCyrillic. VerdanaGreek. VerdanaHebrew. VerdanaVietnam. Webdings.
Wingdings. ZapfChancery. ZapfDingbats. )
-----</font>-----
-- b -<kerning>-----
(phv.krn) - HELVETICA. (ptm.krn) - TIMES.
-----</kerning>-----
-- c -<language>-----
(f:/wamp/www/tex/hyphen/ca.tex
Catalan ..... iso-8859-1/windows-1252 ..... [ca]
(f:/wamp/www/tex/hyphen/cs.tex
Czech ..... iso-8859-2/windows-1250 ..... [cs]
(f:/wamp/www/tex/hyphen/da.tex
Danish ..... iso-8859-1/windows-1252 ..... [da]
(f:/wamp/www/tex/hyphen/de-rf.tex
German, reformed (1999-2004) ..... iso-8859-1/windows-1252 ..... [de-rf]
(f:/wamp/www/tex/hyphen/de.tex
```

```

German, traditional ..... iso-8859-1/windows-1252 ..... [de]
(f:/wamp/www/tex/hyphen/el.tex
Modern Greek ..... windows-1253 ..... [el]
(f:/wamp/www/tex/hyphen/en-UK.tex
British English ..... ASCII ..... [en-UK]
(f:/wamp/www/tex/hyphen/en-US.tex
American English ..... ASCII ..... [en-US]
(f:/wamp/www/tex/hyphen/es.tex
Spanish (Español, Castilian) ..... iso-8859-1/windows-1252 ..... [es]
(f:/wamp/www/tex/hyphen/et.tex
Estonian ..... iso-8859-1/windows-1252 ..... [et]
(f:/wamp/www/tex/hyphen/eu.tex
Basque ..... ASCII ..... [eu]
(f:/wamp/www/tex/hyphen/fi.tex
Finnish ..... iso-8859-1/windows-1252 ..... [fi]
(f:/wamp/www/tex/hyphen/fr.tex
French ..... iso-8859-1/windows-1252 ..... [fr]
(f:/wamp/www/tex/hyphen/ga.tex
Irish (Gaeilge) ..... iso-8859-1/windows-1252 ..... [ga]
(f:/wamp/www/tex/hyphen/hr.tex
Croatian ..... iso-8859-2/windows-1250 ..... [hr]
(f:/wamp/www/tex/hyphen/hu.tex
Hungarian ..... iso-8859-2/windows-1250 ..... [hu]
(f:/wamp/www/tex/hyphen/ia.tex
Interlingua ..... ASCII ..... [ia]
(f:/wamp/www/tex/hyphen/id.tex
Indonesian ..... ASCII ..... [id]
(f:/wamp/www/tex/hyphen/is.tex
Icelandic ..... iso-8859-1/windows-1252 ..... [is]
(f:/wamp/www/tex/hyphen/it.tex
Italian ..... ASCII ..... [it]
(f:/wamp/www/tex/hyphen/la.tex
Latin ..... iso-8859-1/windows-1252 ..... [la]
(f:/wamp/www/tex/hyphen/nl.tex
Dutch, reformed ..... iso-8859-1/windows-1252 ..... [nl]
(f:/wamp/www/tex/hyphen/no.tex
Norwegian ..... iso-8859-1/windows-1252 ..... [no]
(f:/wamp/www/tex/hyphen/pl.tex
Polish ..... iso-8859-2/windows-1250 ..... [pl]
(f:/wamp/www/tex/hyphen/pt.tex
Portuguese ..... iso-8859-1/windows-1252 ..... [pt]
(f:/wamp/www/tex/hyphen/ru.tex
Russian ..... windows-1251 ..... [ru]
(f:/wamp/www/tex/hyphen/sr.tex
Serbian ..... windows-1251 ..... [sr]
(f:/wamp/www/tex/hyphen/sv.tex
Swedish ..... iso-8859-1/windows-1252 ..... [sv]
(f:/wamp/www/tex/hyphen/tr.tex
Turkish ..... ASCII ..... [tr]
(f:/wamp/www/tex/hyphen/uk.tex
Ukrainian ..... windows-1251 ..... [uk]
(f:/wamp/www/tex/hyphen/wen.tex
Sorbian (Wendish) ..... iso-8859-2/windows-1250 ..... [wen]
More pattern files are available at http://www.ctan.org/tex-archive/language//
-----</language>-----
.. d <unicode>-----
- Installed rows of the Unicode character space: - 1) [u0000.row]
Controls-0, Basic Latin (ISO-646/ASCII),
Controls-1, Latin-1 Supplement (ISO-8859-1) - 2) [u0100.row] Latin Extended-A,
Latin Extended-B/1 - 3) [u0200.row] Latin Extended-B/2, International Phonetic
Alphabet Extensions, Spacing Modifier Letters - 4) [u0300.row]
Combining Diacritical Marks, Greek - 5) [u0400.row] Cyrillic - 6) [u0500.row]
Cyrillic Supplementary, Armenian, Hebrew - 7) [u0600.row] Arabic - 8)
[u0900.row] Devanagari, Bengali - 9) [u0e00.row] Thai, Lao - 10) [u1e00.row]
Latin Extended Additional - 11) [u2000.row]
General Punctuation, Super- and Subscripts, Currency
Symbols, Combining Diacritical Marks for Symbols - 12) [u2100.row]
Letterlike Symbols, Number Forms, Arrows - 13) [u2200.row]
Mathematical Operators - 14) [u2300.row] Miscellaneous Technical - 15)
[u2400.row] Control Pictures, Optical Character Recognition,
Enclosed Alphanumerics - 16) [u2500.row] Box Drawing, Block Elements,
Geometric Shapes - 17) [u2600.row] Miscellaneous Symbols
-----</unicode>-----
.. e <entity>-----
\openout6 = `entity.lst'.

- (f:/wamp/www/data/ent/HTMLlat1.ent) - (f:/wamp/www/data/ent/HTMLspec.ent) -
(f:/wamp/www/data/ent/HTMLsymb.ent)
system(sort < entity.lst > sortent.lst)...executed.

&Aacute; &aacute; &acirc; &Acirc; &acute;
&aelig; &Aelig; &Agrave; &agrave; &alefsym; &alpha; &Alpha; &amp; &and; &ang;
&Aring; &aring; &asymp; &Atilde; &atilde; &Auml; &auml; &bdquo; &beta; &Beta;
&brvbar; &bull; &cap; &Ccedil; &ccedil; &cedil; &cent; &chi; &Chi; &circ;
&clubs; &cong; &copy; &crarr; &cup; &curren; &dagger; &Dagger; &dArr; &darr;
&deg; &delta; &Delta; &diam; &divide; &Eacute; &eacute; &Ecirc; &ecirc;
&egrave; &Egrave; &empty; &emsp; &ensp; &epsilon; &Epsilon; &equiv; &Eta;
&eta; &eth; &ETH; &Euml; &euml; &exist; &fnof; &forall; &frac12; &frac14;
&frac34; &frasl; &gamma; &Gamma; &ge; &gt; &harr; &hArr; &hearts; &hellip;
&Iacute; &iacute; &Icirc; &icirc; &iexcl; &igrave; &Igrave; &image; &infin;
&int; &iota; &iota; &iquest; &isin; &iUml; &iuml; &kappa; &Kappa; &lambda;
&Lambda; &lang; &laquo; &larr; &LArr; &lceil; &lDquo; &le; &lfloor; &lowast;
&lrm; &lrm; &lsaquo; &lsquo; &lt; &lt; &macr; &mdash; &micro; &middot; &minus; &mu;
&Mu; &nbla; &nbsp; &ndash; &ne; &ni; &not; &notin; &nsb; &ntilde; &Ntilde;
&Nu; &nu; &oacute; &Oacute; &Ocirc; &ocirc; &OElig; &oelig; &Ograve; &ograve;
&oline; &Omega; &omega; &Omicron; &omicron; &oplus; &or; &ordf; &ordm;
&oslash; &Oslash; &otilde; &Otilde; &otimes; &Ouml; &ouml; &para; &part;
&permil; &perp; &phi; &Phi; &pi; &Pi; &pi; &pi; &plusmn; &pound; &prime; &Prime;
&prod; &prop; &psi; &Psi; &quot; &radic; &rang; &raquo; &rArr; &rarr; &rceil;
&rdquo; &real; &reg; &rfloor; &Rho; &rho; &rlm; &rsaquo; &rsquo; &sbquo;
&scaron; &Scaron; &sdot; &sect; &shy; &Sigma; &sigma; &sigmaf; &sim; &spades;
&sub; &sube; &sum; &sup; &sup1; &sup2; &sup3; &supe; &szlig; &tau; &Tau;
&there4; &Theta; &theta; &thetasym; &thinsp; &THORN; &thorn; &times; &trade;
&uacute; &Uacute; &uArr; &uarr; &Ucirc; &ucirc; &Ugrave; &ugrave; &uml;
&upsih; &Upsilon; &upsilon; &Uuml; &uuml; &weierp; &xi; &Xi; &yacute; &Yacute;
&yen; &yuml; &Yuml; &zeta; &Zeta; &zwj; &zwnj;
-----</entity>-----
.. f <glyph>-----
(dingbats.gly). ! Undefined unicode row 2700. ! Undefined unicode row F800.
(gms.gly). ! Undefined unicode row A00. ! Undefined unicode row 3000.
! Undefined unicode row 3100. ! Undefined unicode row 3200.
! Undefined unicode row 3300. ! Undefined unicode row F600.
! Undefined unicode row F700. ! Undefined unicode row FB00.
! Undefined unicode row FE00. ! Undefined unicode row FF00.
-----</glyph>-----
.. g <codepage>-----
- 1. cp1250.txt - 2. cp1251.txt - 3. cp1252.txt - 4. cp1253.txt - 5.
cp1254.txt - 6. cp1255.txt - 7. cp1256.txt - 8. cp1257.txt - 9. cp1258.txt -
10. cp437.txt - 11. cp850.txt - 12. cp860.txt - 13. cp863.txt - 14. cp865.txt
- 15. cp874.txt - 16. i8859-1.txt - 17. i8859-10.txt - 18. i8859-11.txt - 19.
i8859-13.txt - 20. i8859-14.txt - 21. i8859-15.txt - 22. i8859-2.txt - 23.
i8859-3.txt - 24. i8859-4.txt - 25. i8859-5.txt - 26. i8859-6.txt - 27.
i8859-7.txt - 28. i8859-8.txt - 29. i8859-9.txt - 30. iscii.txt - 31.
iso646.txt - 32. us-ascii.txt - 33. viscii.txt - Writing encoding file(s):
\openout3 = `cp1250.enc'.

cp1250.enc
\openout3 = `cp1251.enc'.

cp1251.enc
\openout3 = `cp1252.enc'.

cp1252.enc
\openout3 = `cp1253.enc'.

cp1253.enc
\openout3 = `cp1254.enc'.

```

cp1254.enc
\openout3 = `cp1255.enc'.

cp1255.enc
\openout3 = `cp1256.enc'.

cp1256.enc
\openout3 = `cp1257.enc'.

cp1257.enc
\openout3 = `cp1258.enc'.

cp1258.enc
\openout3 = `cp437.enc'.

cp437.enc
\openout3 = `cp850.enc'.

cp850.enc
\openout3 = `cp860.enc'.

cp860.enc
\openout3 = `cp863.enc'.

cp863.enc
\openout3 = `cp865.enc'.

cp865.enc
\openout3 = `cp874.enc'.

cp874.enc
\openout3 = `i8859-1.enc'.

i8859-1.enc
\openout3 = `i8859-10.enc'.

i8859-10.enc
\openout3 = `i8859-11.enc'.

i8859-11.enc
\openout3 = `i8859-13.enc'.

i8859-13.enc
\openout3 = `i8859-14.enc'.

i8859-14.enc
\openout3 = `i8859-15.enc'.

i8859-15.enc
\openout3 = `i8859-2.enc'.

i8859-2.enc
\openout3 = `i8859-3.enc'.

i8859-3.enc
\openout3 = `i8859-4.enc'.

i8859-4.enc
\openout3 = `i8859-5.enc'.

i8859-5.enc
\openout3 = `i8859-6.enc'.

i8859-6.enc
\openout3 = `i8859-7.enc'.

i8859-7.enc
\openout3 = `i8859-8.enc'.

i8859-8.enc
\openout3 = `i8859-9.enc'.

i8859-9.enc
\openout3 = `iscii.enc'.

iscii.enc
\openout3 = `iso646.enc'.

iso646.enc
\openout3 = `us-ascii.enc'.

us-ascii.enc
\openout3 = `viscii.enc'.

viscii.enc
-----</codepage>-----
----</modules>-----
--</initialize>-----
Release: 20080107 | Domain: www.Gerolf.org | Mail: MarkupShredder(a)Gerolf.org
</GMS>-----
(f:/wamp/www/etc/epilogue.cfg - Writing encoding file(s):
\openout3 = `cp1250.enc'.

cp1250.enc
\openout3 = `cp1251.enc'.

cp1251.enc
\openout3 = `cp1252.enc'.

cp1252.enc
\openout3 = `cp1253.enc'.

cp1253.enc
\openout3 = `cp1254.enc'.

cp1254.enc
\openout3 = `cp1255.enc'.

cp1255.enc
\openout3 = `cp1256.enc'.

cp1256.enc
\openout3 = `cp1257.enc'.

cp1257.enc
\openout3 = `cp1258.enc'.

cp1258.enc
\openout3 = `cp437.enc'.

cp437.enc
\openout3 = `cp850.enc'.

cp850.enc
\openout3 = `cp860.enc'.

cp860.enc
\openout3 = `cp863.enc'.

cp863.enc
\openout3 = `cp865.enc'.

cp865.enc

\openout3 = `cp874.enc'.
cp874.enc
\openout3 = `i8859-1.enc'.

i8859-1.enc
\openout3 = `i8859-10.enc'.

i8859-10.enc
\openout3 = `i8859-11.enc'.

i8859-11.enc
\openout3 = `i8859-13.enc'.

i8859-13.enc
\openout3 = `i8859-14.enc'.

i8859-14.enc
\openout3 = `i8859-15.enc'.

i8859-15.enc
\openout3 = `i8859-2.enc'.

i8859-2.enc
\openout3 = `i8859-3.enc'.

i8859-3.enc
\openout3 = `i8859-4.enc'.

i8859-4.enc
\openout3 = `i8859-5.enc'.

i8859-5.enc
\openout3 = `i8859-6.enc'.

i8859-6.enc
\openout3 = `i8859-7.enc'.

i8859-7.enc
\openout3 = `i8859-8.enc'.

i8859-8.enc
\openout3 = `i8859-9.enc'.

i8859-9.enc
\openout3 = `iscii.enc'.

iscii.enc
\openout3 = `iso646.enc'.

iso646.enc
\openout3 = `us-ascii.enc'.

us-ascii.enc
\openout3 = `viscii.enc'.

viscii.enc
\openout3 = `u0000.enc'.

u0000.
\openout3 = `g0000.enc'.

g0000.enc
\openout3 = `u0100.enc'.

u0100.

\openout3 = `g0100.enc'.
g0100.enc
\openout3 = `u0200.enc'.

u0200.
\openout3 = `g0200.enc'.

g0200.enc
\openout3 = `u0300.enc'.

u0300.
\openout3 = `g0300.enc'.

g0300.enc
\openout3 = `u0400.enc'.

u0400.
\openout3 = `g0400.enc'.

g0400.enc
\openout3 = `u0500.enc'.

u0500.
\openout3 = `g0500.enc'.

g0500.enc
\openout3 = `u0600.enc'.

u0600.
\openout3 = `g0600.enc'.

g0600.enc
\openout3 = `u0900.enc'.

u0900.
\openout3 = `g0900.enc'.

g0900.enc
\openout3 = `u0E00.enc'.

u0E00.
\openout3 = `g0E00.enc'.

g0E00.enc
\openout3 = `u1E00.enc'.

u1E00.
\openout3 = `g1E00.enc'.

g1E00.enc
\openout3 = `u2000.enc'.

u2000.
\openout3 = `g2000.enc'.

g2000.enc
\openout3 = `u2100.enc'.

u2100.
\openout3 = `g2100.enc'.

g2100.enc
\openout3 = `u2200.enc'.

u2200.
\openout3 = `g2200.enc'.

```

g2200.enc
\openout3 = `u2300.enc'.

u2300.
\openout3 = `g2300.enc'.

g2300.enc
\openout3 = `u2400.enc'.

u2400.
\openout3 = `g2400.enc'.

g2400.enc
\openout3 = `u2500.enc'.

u2500.
\openout3 = `g2500.enc'.

g2500.enc
\openout3 = `u2600.enc'.

u2600.
\openout3 = `g2600.enc'.

g2600.enc))
*\dump
Beginning to dump on file gerolf.efmt
(format=gerolf 2008.1.5)
11023 strings of total length 151649
301281 memory locations dumped; current usage is 38153&240080
8819 multiletter control sequences
\font\nullfont=nullfont
\font\tenrm=cmr10
\font\sevenrm=cmr7
\font\fiverm=cmr5
\font\teni=cmmi10
\font\seveni=cmmi7
\font\fivei=cmmi5
\font\tensy=cmsy10
\font\sevensy=cmsy7
\font\fivesy=cmsy5

```

```

\font\tenex=cmex10
\font\tenbf=cmbx10
\font\sevenbf=cmbx7
\font\fivebf=cmbx5
\font\tentt=cmtt10
\font\tensl=cmsl10
\font\tenit=cmti10
4739 words of font info for 16 preloaded fonts
581 hyphenation exceptions
Hyphenation trie of length 119578 has 3201 ops out of 35111
110 for language 31
82 for language 30
10 for language 29
127 for language 28
62 for language 27
112 for language 26
7 for language 25
194 for language 24
224 for language 23
265 for language 22
29 for language 21
30 for language 20
148 for language 19
21 for language 18
40 for language 17
117 for language 16
21 for language 15
229 for language 14
86 for language 13
31 for language 12
5 for language 11
113 for language 10
76 for language 9
229 for language 8
224 for language 7
19 for language 6
207 for language 5
235 for language 4
60 for language 3
63 for language 2
25 for language 1
No pages of output.

```

gmssetup.log

gmssetup.log

////////////////////////////////////

This is the setup log file for Gerolf Markup Shredder.
To write the run time debug log file, set GMS_DEBUG=Z.
Format: called batch (parameter 1) ... (parameter 9),
followed by values of important environment variables.
shredder.bat reads user input and calls batch modules.
l_type modules only contribute if their compiled equi-
valents are absent.

-shell () () () () () () () ()

GMS_VERSION=0.07a

GMS_CODEPAGE=CP1252

GMS_SETTING=F:\GMS 007a\etc

GMS_ROOT=F:\GMS 007a

folder (called_by) (gmssetup) () () () () ()

GMS_PATH=E:\WINDOWS\system32;E:\WINDOWS;E:\WINDOWS\System32\Wbem;E:\Programme\Gemeinsame Dateien\Adaptec Shared\System

PATH=F:\GMS 007a\batch;F:\GMS 007a\bin\win;E:\WINDOWS\system32;E:\WINDOWS;E:\WINDOWS\System32\Wbem;E:\Programme\Gemeinsame Dateien\Adaptec Shared\System

GMS_DRIVE=F:

see 'F:\GMS 007a\batch\compiler.log' for syntax check results

Font Maps

cm.map

```
% cm.map
% ////////////////////////////////////////////////////

cmb10 CMB10 <cmb10.pfb
cmbsy10 CMBSY10 <cmbsy10.pfb
cmbx10 CMBX10 <cmbx10.pfb
cmbx12 CMBX12 <cmbx12.pfb
cmbx5 CMBX5 <cmbx5.pfb
cmbx6 CMBX6 <cmbx6.pfb
cmbx7 CMBX7 <cmbx7.pfb
cmbx8 CMBX8 <cmbx8.pfb
cmbx9 CMBX9 <cmbx9.pfb
cmbxsl10 CMBXSL10 <cmbxsl10.pfb
cmbxti10 CMBXTI10 <cmbxti10.pfb
cmcsc10 CMCSCL10 <cmcsc10.pfb
cmdunh10 CMDUNH10 <cmdunh10.pfb
cmex10 CMEX10 <cmex10.pfb
cmff10 CMFF10 <cmff10.pfb
cmfi10 CMFI10 <cmfi10.pfb
cmfib8 CMFIB8 <cmfib8.pfb
cminch CMINCH <cminch.pfb
cmitt10 CMITT10 <cmitt10.pfb
cmmi10 CMMI10 <cmmi10.pfb
cmmi12 CMMI12 <cmmi12.pfb
cmmi5 CMMI5 <cmmi5.pfb
cmmi6 CMMI6 <cmmi6.pfb
cmmi7 CMMI7 <cmmi7.pfb
cmmi8 CMMI8 <cmmi8.pfb
cmmi9 CMMI9 <cmmi9.pfb
cmmib10 CMMIB10 <cmmib10.pfb
cmr10 CMR10 <cmr10.pfb
cmr12 CMR12 <cmr12.pfb
cmr17 CMR17 <cmr17.pfb
cmr5 CMR5 <cmr5.pfb
cmr6 CMR6 <cmr6.pfb
cmr7 CMR7 <cmr7.pfb
cmr8 CMR8 <cmr8.pfb
cmr9 CMR9 <cmr9.pfb
cmsl10 CMSL10 <cmsl10.pfb
cmsl12 CMSL12 <cmsl12.pfb
cmsl8 CMSL8 <cmsl8.pfb
cmsl9 CMSL9 <cmsl9.pfb
cmsltt10 CMSLTT10 <cmsltt10.pfb
cmss10 CMSS10 <cmss10.pfb
cmss12 CMSS12 <cmss12.pfb
cmss17 CMSS17 <cmss17.pfb
cmss8 CMSS8 <cmss8.pfb
cmss9 CMSS9 <cmss9.pfb
cmssbx10 CMSSBX10 <cmssbx10.pfb
cmssdc10 CMSSDC10 <cmssdc10.pfb
cmssi10 CMSSI10 <cmssi10.pfb

cmssi12 CMSSI12 <cmssi12.pfb
cmssi17 CMSSI17 <cmssi17.pfb
cmssi8 CMSSI8 <cmssi8.pfb
cmssi9 CMSSI9 <cmssi9.pfb
cmssq8 CMSSQ8 <cmssq8.pfb
cmssqi8 CMSSQI8 <cmssqi8.pfb
cmsy10 CMSY10 <cmsy10.pfb
cmsy5 CMSY5 <cmsy5.pfb
cmsy6 CMSY6 <cmsy6.pfb
cmsy7 CMSY7 <cmsy7.pfb
cmsy8 CMSY8 <cmsy8.pfb
cmsy9 CMSY9 <cmsy9.pfb
cmtcsc10 CMTCSCL10 <cmtcsc10.pfb
cmtex10 CMTEX10 <cmtex10.pfb
cmtex8 CMTEX8 <cmtex8.pfb
cmtex9 CMTEX9 <cmtex9.pfb
cmti10 CMTI10 <cmti10.pfb
cmti12 CMTI12 <cmti12.pfb
cmti7 CMTI7 <cmti7.pfb
cmti8 CMTI8 <cmti8.pfb
cmti9 CMTI9 <cmti9.pfb
cmtt10 CMTT10 <cmtt10.pfb
cmtt12 CMTT12 <cmtt12.pfb
cmtt8 CMTT8 <cmtt8.pfb
cmtt9 CMTT9 <cmtt9.pfb
cmu10 CMU10 <cmu10.pfb
cmvtt10 CMVTT10 <cmvtt10.pfb
lasy10 LASY10 <lasy10.pfb
lasy5 LASY5 <lasy5.pfb
lasy6 LASY6 <lasy6.pfb
lasy7 LASY7 <lasy7.pfb
lasy8 LASY8 <lasy8.pfb
lasy9 LASY9 <lasy9.pfb
lasyb10 LASYB10 <lasyb10.pfb
lcircle10 LCIRCLE10 <lcircle10.pfb
lcirclew10 LCIRCLEW10 <lcirclew10.pfb
lcms8 LCMSS8 <lcms8.pfb
lcmsb8 LCMSSB8 <lcmsb8.pfb
lcmsi8 LCMSSI8 <lcmsi8.pfb
line10 LINE10 <line10.pfb
linew10 LINEW10 <linew10.pfb
logo10 LOGO10 <logo10.pfb
logo8 LOGO8 <logo8.pfb
logo9 LOGO9 <logo9.pfb
logobf10 LOGOBF10 <logobf10.pfb
logosi10 LOGOSI10 <logosi10.pfb

% ////////////////////////////////////////////////////
% http://www.ctan.org/tex-archive/fonts/cm/[tfm, ps-type1/bluesky]
```

font.map

% font.map - F:/wamp/www/fonts/[font-type]/[supplier]/[font-family]

% Generated by Gerolf Markup Shredder (www.Gerolf.org)

% on Fri, 2006/12/08, 5:06:31 CET

% Default codepage: cp1252

% [1] Base name of font files
% [2] PostScript font face name
% " [3] ReEncodeFont " PostScript encoding name
% <[4] Encoding file [* .enc]
% <[5] Glyph file [*.pf*, *.ttf]

_AGENCYB AgencyFB-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <AGENCYB.TTF
_AGENCYB3 AgencyFB-Bold " cp1253-encoding ReEncodeFont " <cp1253.enc <AGENCYB.TTF
_AGENCYR AgencyFB-Reg " cp1252-encoding ReEncodeFont " <cp1252.enc <AGENCYR.TTF
_AGENCYR3 AgencyFB-Reg " cp1253-encoding ReEncodeFont " <cp1253.enc <AGENCYR.TTF
_ALGER Algerian " cp1252-encoding ReEncodeFont " <cp1252.enc <ALGER.TTF
_ANTQUAB BookAntiqua-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <ANTQUAB.TTF
_ANTQUABI BookAntiqua-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <ANTQUABI.TTF
_ANTQUAI BookAntiqua-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <ANTQUAI.TTF
ARBLI Arial-BlackItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <ARBLI.TTF
_ARIALN ArialNarrow " cp1252-encoding ReEncodeFont " <cp1252.enc <ARIALN.TTF
_ARIALN00 ArialNarrow " g0000-Encoding ReEncodeFont " <g0000.enc <ARIALN.TTF
_ARIALN01 ArialNarrow " g0100-Encoding ReEncodeFont " <g0100.enc <ARIALN.TTF
_ARIALN02 ArialNarrow " g0200-Encoding ReEncodeFont " <g0200.enc <ARIALN.TTF
_ARIALN03 ArialNarrow " g0300-Encoding ReEncodeFont " <g0300.enc <ARIALN.TTF
_ARIALN04 ArialNarrow " g0400-Encoding ReEncodeFont " <g0400.enc <ARIALN.TTF
_ARIALN1E ArialNarrow " g1E00-Encoding ReEncodeFont " <g1e00.enc <ARIALN.TTF
_ARIALN20 ArialNarrow " g2000-Encoding ReEncodeFont " <g2000.enc <ARIALN.TTF
_ARIALN21 ArialNarrow " g2100-Encoding ReEncodeFont " <g2100.enc <ARIALN.TTF
_ARIALN22 ArialNarrow " g2200-Encoding ReEncodeFont " <g2200.enc <ARIALN.TTF
_ARIALN23 ArialNarrow " g2300-Encoding ReEncodeFont " <g2300.enc <ARIALN.TTF
_ARIALN25 ArialNarrow " g2500-Encoding ReEncodeFont " <g2500.enc <ARIALN.TTF
_ARIALN26 ArialNarrow " g2600-Encoding ReEncodeFont " <g2600.enc <ARIALN.TTF
_ARIALNB ArialNarrow-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <ARIALNB.TTF
_ARIALNB00 ArialNarrow-Bold " g0000-Encoding ReEncodeFont " <g0000.enc <ARIALNB.TTF
_ARIALNB01 ArialNarrow-Bold " g0100-Encoding ReEncodeFont " <g0100.enc <ARIALNB.TTF
_ARIALNB02 ArialNarrow-Bold " g0200-Encoding ReEncodeFont " <g0200.enc <ARIALNB.TTF
_ARIALNB03 ArialNarrow-Bold " g0300-Encoding ReEncodeFont " <g0300.enc <ARIALNB.TTF
_ARIALNB04 ArialNarrow-Bold " g0400-Encoding ReEncodeFont " <g0400.enc <ARIALNB.TTF
_ARIALNB1E ArialNarrow-Bold " g1E00-Encoding ReEncodeFont " <g1e00.enc <ARIALNB.TTF
_ARIALNB20 ArialNarrow-Bold " g2000-Encoding ReEncodeFont " <g2000.enc <ARIALNB.TTF
_ARIALNB21 ArialNarrow-Bold " g2100-Encoding ReEncodeFont " <g2100.enc <ARIALNB.TTF
_ARIALNB22 ArialNarrow-Bold " g2200-Encoding ReEncodeFont " <g2200.enc <ARIALNB.TTF
_ARIALNB23 ArialNarrow-Bold " g2300-Encoding ReEncodeFont " <g2300.enc <ARIALNB.TTF
_ARIALNB25 ArialNarrow-Bold " g2500-Encoding ReEncodeFont " <g2500.enc <ARIALNB.TTF
_ARIALNB26 ArialNarrow-Bold " g2600-Encoding ReEncodeFont " <g2600.enc <ARIALNB.TTF
_ARIALNBI ArialNarrow-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <ARIALNBI.TTF
_ARIALNBI00 ArialNarrow-BoldItalic " g0000-Encoding ReEncodeFont " <g0000.enc <ARIALNBI.TTF
_ARIALNBI01 ArialNarrow-BoldItalic " g0100-Encoding ReEncodeFont " <g0100.enc <ARIALNBI.TTF
_ARIALNBI02 ArialNarrow-BoldItalic " g0200-Encoding ReEncodeFont " <g0200.enc <ARIALNBI.TTF
_ARIALNBI03 ArialNarrow-BoldItalic " g0300-Encoding ReEncodeFont " <g0300.enc <ARIALNBI.TTF
_ARIALNBI04 ArialNarrow-BoldItalic " g0400-Encoding ReEncodeFont " <g0400.enc <ARIALNBI.TTF
_ARIALNBI1E ArialNarrow-BoldItalic " g1E00-Encoding ReEncodeFont " <g1e00.enc <ARIALNBI.TTF
_ARIALNBI20 ArialNarrow-BoldItalic " g2000-Encoding ReEncodeFont " <g2000.enc <ARIALNBI.TTF
_ARIALNBI21 ArialNarrow-BoldItalic " g2100-Encoding ReEncodeFont " <g2100.enc <ARIALNBI.TTF
_ARIALNBI22 ArialNarrow-BoldItalic " g2200-Encoding ReEncodeFont " <g2200.enc <ARIALNBI.TTF
_ARIALNBI23 ArialNarrow-BoldItalic " g2300-Encoding ReEncodeFont " <g2300.enc <ARIALNBI.TTF
_ARIALNBI25 ArialNarrow-BoldItalic " g2500-Encoding ReEncodeFont " <g2500.enc <ARIALNBI.TTF
_ARIALNBI26 ArialNarrow-BoldItalic " g2600-Encoding ReEncodeFont " <g2600.enc <ARIALNBI.TTF
_ARIALNBIVI ArialNarrow-BoldItalic " viscii-encoding ReEncodeFont " <viscii.enc <ARIALNBI.TTF
_ARIALNBWI1 ArialNarrow-BoldItalic " cp1251-encoding ReEncodeFont " <cp1251.enc <ARIALNBI.TTF
_ARIALNBW3 ArialNarrow-BoldItalic " cp1253-encoding ReEncodeFont " <cp1253.enc <ARIALNBI.TTF
_ARIALNBW5 ArialNarrow-BoldItalic " cp1255-encoding ReEncodeFont " <cp1255.enc <ARIALNBI.TTF
_ARIALNBW6 ArialNarrow-BoldItalic " cp1256-encoding ReEncodeFont " <cp1256.enc <ARIALNBI.TTF
_ARIALNBVI ArialNarrow-Bold " viscii-encoding ReEncodeFont " <viscii.enc <ARIALNB.TTF
_ARIALNBW1 ArialNarrow-Bold " cp1251-encoding ReEncodeFont " <cp1251.enc <ARIALNB.TTF
_ARIALNBW3 ArialNarrow-Bold " cp1253-encoding ReEncodeFont " <cp1253.enc <ARIALNB.TTF

_ARIALNBW5 ArialNarrow-Bold " cp1255-encoding ReEncodeFont " <cp1255.enc <ARIALNB.TTF
 _ARIALNBW6 ArialNarrow-Bold " cp1256-encoding ReEncodeFont " <cp1256.enc <ARIALNB.TTF
 _ARIALNI ArialNarrow-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <ARIALNI.TTF
 _ARIALNI00 ArialNarrow-Italic " g0000-Encoding ReEncodeFont " <g0000.enc <ARIALNI.TTF
 _ARIALNI01 ArialNarrow-Italic " g0100-Encoding ReEncodeFont " <g0100.enc <ARIALNI.TTF
 _ARIALNI02 ArialNarrow-Italic " g0200-Encoding ReEncodeFont " <g0200.enc <ARIALNI.TTF
 _ARIALNI03 ArialNarrow-Italic " g0300-Encoding ReEncodeFont " <g0300.enc <ARIALNI.TTF
 _ARIALNI04 ArialNarrow-Italic " g0400-Encoding ReEncodeFont " <g0400.enc <ARIALNI.TTF
 _ARIALNI1E ArialNarrow-Italic " g1E00-Encoding ReEncodeFont " <g1e00.enc <ARIALNI.TTF
 _ARIALNI20 ArialNarrow-Italic " g2000-Encoding ReEncodeFont " <g2000.enc <ARIALNI.TTF
 _ARIALNI21 ArialNarrow-Italic " g2100-Encoding ReEncodeFont " <g2100.enc <ARIALNI.TTF
 _ARIALNI22 ArialNarrow-Italic " g2200-Encoding ReEncodeFont " <g2200.enc <ARIALNI.TTF
 _ARIALNI23 ArialNarrow-Italic " g2300-Encoding ReEncodeFont " <g2300.enc <ARIALNI.TTF
 _ARIALNI25 ArialNarrow-Italic " g2500-Encoding ReEncodeFont " <g2500.enc <ARIALNI.TTF
 _ARIALNI26 ArialNarrow-Italic " g2600-Encoding ReEncodeFont " <g2600.enc <ARIALNI.TTF
 _ARIALNIVI ArialNarrow-Italic " viscii-encoding ReEncodeFont " <viscii.enc <ARIALNI.TTF
 _ARIALNIW1 ArialNarrow-Italic " cp1251-encoding ReEncodeFont " <cp1251.enc <ARIALNI.TTF
 _ARIALNIW3 ArialNarrow-Italic " cp1253-encoding ReEncodeFont " <cp1253.enc <ARIALNI.TTF
 _ARIALNIW5 ArialNarrow-Italic " cp1255-encoding ReEncodeFont " <cp1255.enc <ARIALNI.TTF
 _ARIALNIW6 ArialNarrow-Italic " cp1256-encoding ReEncodeFont " <cp1256.enc <ARIALNI.TTF
 _ARIALNVI ArialNarrow " viscii-encoding ReEncodeFont " <viscii.enc <ARIALN.TTF
 _ARIALNW1 ArialNarrow " cp1251-encoding ReEncodeFont " <cp1251.enc <ARIALN.TTF
 _ARIALNW3 ArialNarrow " cp1253-encoding ReEncodeFont " <cp1253.enc <ARIALN.TTF
 _ARIALNW5 ArialNarrow " cp1255-encoding ReEncodeFont " <cp1255.enc <ARIALN.TTF
 _ARIALNW6 ArialNarrow " cp1256-encoding ReEncodeFont " <cp1256.enc <ARIALN.TTF
 _ARLRDBD ArialRoundedMTBold " cp1252-encoding ReEncodeFont " <cp1252.enc <ARLRDBD.TTF
 _BASKVILL BaskoldFace " cp1252-encoding ReEncodeFont " <cp1252.enc <BASKVILL.TTF
 _BAUHS93 Bauhaus93 " cp1252-encoding ReEncodeFont " <cp1252.enc <BAUHS93.TTF
 _BELL BellMT " cp1252-encoding ReEncodeFont " <cp1252.enc <BELL.TTF
 _BELLB BellMTBold " cp1252-encoding ReEncodeFont " <cp1252.enc <BELLB.TTF
 _BELLI BellMTItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <BELLI.TTF
 _BERNHC BernardMT-Condensed " cp1252-encoding ReEncodeFont " <cp1252.enc <BERNHC.TTF
 _BKANT BookAntiqua " cp1252-encoding ReEncodeFont " <cp1252.enc <BKANT.TTF
 _BOD_B BodoniMT-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_B.TTF
 _BOD_BI BodoniMT-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_BI.TTF
 _BOD_BIAI BodoniMTBlack-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_BIAI.TTF
 _BOD_BIAR BodoniMTBlack " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_BIAR.TTF
 _BOD_CB BodoniMTCondensed-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_CB.TTF
 _BOD_CBI BodoniMTCondensed-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_CBI.TTF
 _BOD_CI BodoniMTCondensed-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_CI.TTF
 _BOD_CR BodoniMTCondensed " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_CR.TTF
 _BOD_I BodoniMT-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_I.TTF
 _BOD_PSTC BodoniMTPosterCompressed " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_PSTC.TTF
 _BOD_R BodoniMT " cp1252-encoding ReEncodeFont " <cp1252.enc <BOD_R.TTF
 _BOOKOS BookmanOldStyle " cp1252-encoding ReEncodeFont " <cp1252.enc <BOOKOS.TTF
 _BOOKOSB BookmanOldStyle-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <BOOKOSB.TTF
 _BOOKOSBI BookmanOldStyle-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <BOOKOSBI.TTF
 _BOOKOSI BookmanOldStyle-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <BOOKOSI.TTF
 _BRADHITC BradleyHandITC " cp1252-encoding ReEncodeFont " <cp1252.enc <BRADHITC.TTF
 _BRITANIC BritanicBold " cp1252-encoding ReEncodeFont " <cp1252.enc <BRITANIC.TTF
 _BRLNSB BerlinSansFB-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <BRLNSB.TTF
 _BRLNSDB BerlinSansFB Demi-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <BRLNSDB.TTF
 _BRLNSR BerlinSansFB-Reg " cp1252-encoding ReEncodeFont " <cp1252.enc <BRLNSR.TTF
 _BROADW Broadway " cp1252-encoding ReEncodeFont " <cp1252.enc <BROADW.TTF
 _BRUSHSCI BrushScriptMT " cp1252-encoding ReEncodeFont " <cp1252.enc <BRUSHSCI.TTF
 _CALIFB CalifornianFB-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <CALIFB.TTF
 _CALIFI CalifornianFB-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <CALIFI.TTF
 _CALIFR CalifornianFB-Reg " cp1252-encoding ReEncodeFont " <cp1252.enc <CALIFR.TTF
 _CALIST CalistoMT " cp1252-encoding ReEncodeFont " <cp1252.enc <CALIST.TTF
 _CALISTB CalisMTBol " cp1252-encoding ReEncodeFont " <cp1252.enc <CALISTB.TTF
 _CALISTBI CalistoMT-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <CALISTBI.TTF
 _CALISTI CalistoMT-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <CALISTI.TTF
 _CASTELAR Castellar " cp1252-encoding ReEncodeFont " <cp1252.enc <CASTELAR.TTF
 _CENSGBK CenturySchoolbook " cp1252-encoding ReEncodeFont " <cp1252.enc <CENSGBK.TTF
 _CENTAUR Centaur " cp1252-encoding ReEncodeFont " <cp1252.enc <CENTAUR.TTF
 _CENTURY Century " cp1252-encoding ReEncodeFont " <cp1252.enc <CENTURY.TTF
 _CHILLER Chiller-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <CHILLER.TTF
 _COLONNA ColonnaMT " cp1252-encoding ReEncodeFont " <cp1252.enc <COLONNA.TTF
 _COOPBL CooperBlack " cp1252-encoding ReEncodeFont " <cp1252.enc <COOPBL.TTF
 _COPRGTB CopperplateGothic-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <COPRGTB.TTF
 _COPRGTL CopperplateGothic-Light " cp1252-encoding ReEncodeFont " <cp1252.enc <COPRGTL.TTF

_CURLZ__ CurlzMT " cp1252-encoding ReEncodeFont " <cp1252.enc <CURLZ__.TTF
 _DATAGLY DataGlyph-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <DATAGLY.TTF
 _ELEPHNT Elephant-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <ELEPHNT.TTF
 _ELEPHNTI Elephant-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <ELEPHNTI.TTF
 _ENGR EngraversMT " cp1252-encoding ReEncodeFont " <cp1252.enc <ENGR.TTF
 _ERASBD ErasITC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <ERASBD.TTF
 _ERASDEMI ErasITC-Demi " cp1252-encoding ReEncodeFont " <cp1252.enc <ERASDEMI.TTF
 _ERASLGHT ErasITC-Light " cp1252-encoding ReEncodeFont " <cp1252.enc <ERASLGHT.TTF
 _ERASMD ErasITC-Medium " cp1252-encoding ReEncodeFont " <cp1252.enc <ERASMD.TTF
 _FELIXTI FelixTitlingMT " cp1252-encoding ReEncodeFont " <cp1252.enc <FELIXTI.TTF
 _FORTE ForteMT " cp1252-encoding ReEncodeFont " <cp1252.enc <FORTE.TTF
 _FRABK FranklinGothic-Book " cp1252-encoding ReEncodeFont " <cp1252.enc <FRABK.TTF
 _FRABKIT FranklinGothic-BookItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <FRABKIT.TTF
 _FRADM FranklinGothic-Demi " cp1252-encoding ReEncodeFont " <cp1252.enc <FRADM.TTF
 _FRADMCN FranklinGothic-DemiCond " cp1252-encoding ReEncodeFont " <cp1252.enc <FRADMCN.TTF
 _FRADMIT FranklinGothic-DemiItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <FRADMIT.TTF
 _FRAHV FranklinGothic-Heavy " cp1252-encoding ReEncodeFont " <cp1252.enc <FRAHV.TTF
 _FRAHVIT FranklinGothic-HeavyItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <FRAHVIT.TTF
 _FRAMDCN FranklinGothic-MediumCond " cp1252-encoding ReEncodeFont " <cp1252.enc <FRAMDCN.TTF
 _FRAMDCNW1 FranklinGothic-MediumCond " cp1251-encoding ReEncodeFont " <cp1251.enc <FRAMDCN.TTF
 _FRAMDCNW3 FranklinGothic-MediumCond " cp1253-encoding ReEncodeFont " <cp1253.enc <FRAMDCN.TTF
 _FREESCPT FreestyleScript-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <FREESCPT.TTF
 _FRSCRIPT FrenchScriptMT " cp1252-encoding ReEncodeFont " <cp1252.enc <FRSCRIPT.TTF
 _FTLTLT FootlightMTLight " cp1252-encoding ReEncodeFont " <cp1252.enc <FTLTLT.TTF
 _GARA Garamond " cp1252-encoding ReEncodeFont " <cp1252.enc <GARA.TTF
 _GARABD Garamond-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <GARABD.TTF
 _GARABDW1 Garamond-Bold " cp1251-encoding ReEncodeFont " <cp1251.enc <GARABD.TTF
 _GARABDW3 Garamond-Bold " cp1253-encoding ReEncodeFont " <cp1253.enc <GARABD.TTF
 _GARAIT Garamond-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <GARAIT.TTF
 _GARAITW1 Garamond-Italic " cp1251-encoding ReEncodeFont " <cp1251.enc <GARAIT.TTF
 _GARAITW3 Garamond-Italic " cp1253-encoding ReEncodeFont " <cp1253.enc <GARAIT.TTF
 _GARAW1 Garamond " cp1251-encoding ReEncodeFont " <cp1251.enc <GARA.TTF
 _GARAW3 Garamond " cp1253-encoding ReEncodeFont " <cp1253.enc <GARA.TTF
 _GIGI Gigi-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <GIGI.TTF
 _GILBI__ GillSansMT-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <GILBI__.TTF
 _GILB__ GillSansMT-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <GILB__.TTF
 _GILC__ GillSansMT-Condensed " cp1252-encoding ReEncodeFont " <cp1252.enc <GILC__.TTF
 _GILI__ GillSansMT-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <GILI__.TTF
 _GILLUBCD GillSans-UltraBoldCondensed " cp1252-encoding ReEncodeFont " <cp1252.enc <GILLUBCD.TTF
 _GILSANUB GillSans-UltraBold " cp1252-encoding ReEncodeFont " <cp1252.enc <GILSANUB.TTF
 _GIL__ GillSansMT " cp1252-encoding ReEncodeFont " <cp1252.enc <GIL__.TTF
 _GLECB GloucesterMT-ExtraCondensed " cp1252-encoding ReEncodeFont " <cp1252.enc <GLECB.TTF
 _GLSNECB GillSansMT-ExtraCondensedBold " cp1252-encoding ReEncodeFont " <cp1252.enc <GLSNECB.TTF
 _GOTHIC CenturyGothic " cp1252-encoding ReEncodeFont " <cp1252.enc <GOTHIC.TTF
 _GOTHICB CenturyGothic-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <GOTHICB.TTF
 _GOTHICBI CenturyGothic-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <GOTHICBI.TTF
 _GOTHICBIW1 CenturyGothic-BoldItalic " cp1251-encoding ReEncodeFont " <cp1251.enc <GOTHICBI.TTF
 _GOTHICBIW3 CenturyGothic-BoldItalic " cp1253-encoding ReEncodeFont " <cp1253.enc <GOTHICBI.TTF
 _GOTHICBW1 CenturyGothic-Bold " cp1251-encoding ReEncodeFont " <cp1251.enc <GOTHICB.TTF
 _GOTHICBW3 CenturyGothic-Bold " cp1253-encoding ReEncodeFont " <cp1253.enc <GOTHICB.TTF
 _GOTHICI CenturyGothic-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <GOTHICI.TTF
 _GOTHICIW1 CenturyGothic-Italic " cp1251-encoding ReEncodeFont " <cp1251.enc <GOTHICI.TTF
 _GOTHICIW3 CenturyGothic-Italic " cp1253-encoding ReEncodeFont " <cp1253.enc <GOTHICI.TTF
 _GOTHICW1 CenturyGothic " cp1251-encoding ReEncodeFont " <cp1251.enc <GOTHIC.TTF
 _GOTHICW3 CenturyGothic " cp1253-encoding ReEncodeFont " <cp1253.enc <GOTHIC.TTF
 _GOUDOS GoudyOldStyleT-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <GOUDOS.TTF
 _GOUDOSB GoudyOldStyleT-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <GOUDOSB.TTF
 _GOUDOSI GoudyOldStyleT-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <GOUDOSI.TTF
 _GOUDYSTO GoudyStout " cp1252-encoding ReEncodeFont " <cp1252.enc <GOUDYSTO.TTF
 _HARLOWSI HarlowSolid " cp1252-encoding ReEncodeFont " <cp1252.enc <HARLOWSI.TTF
 _HARNGTON Harrington " cp1252-encoding ReEncodeFont " <cp1252.enc <HARNGTON.TTF
 _HATTEN Haettenschweiler " cp1252-encoding ReEncodeFont " <cp1252.enc <HATTEN.TTF
 _HTOWERT HighTowerText-Reg " cp1252-encoding ReEncodeFont " <cp1252.enc <HTOWERT.TTF
 _HTOWERTI HighTowerText-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <HTOWERTI.TTF
 _IMPRIISHA ImprintMT-Shadow " cp1252-encoding ReEncodeFont " <cp1252.enc <IMPRIISHA.TTF
 _INFROMAN InformalRoman-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <INFROMAN.TTF
 _ITCBLKAD BlackadderITC-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <ITCBLKAD.TTF
 _ITCEDSCR EdwardianScriptITC " cp1252-encoding ReEncodeFont " <cp1252.enc <ITCEDSCR.TTF
 _ITCKRIST KristenITC-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <ITCKRIST.TTF
 _JOKERMAN Jokerman-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <JOKERMAN.TTF
 _JUICE__ JuiceITC-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <JUICE__.TTF

_KUNSTLER KunstlerScript " cp1252-encoding ReEncodeFont " <cp1252.enc <KUNSTLER.TTF
_LATINWD LatinWide " cp1252-encoding ReEncodeFont " <cp1252.enc <LATINWD.TTF
_LBRITTE LucidaBright " cp1252-encoding ReEncodeFont " <cp1252.enc <LBRITTE.TTF
_LBRITTED LucidaBright-Demi " cp1252-encoding ReEncodeFont " <cp1252.enc <LBRITTED.TTF
_LBRITTEDI LucidaBright-DemiItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <LBRITTEDI.TTF
_LBRITTEI LucidaBright-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <LBRITTEI.TTF
_LCALLIG LucidaCalligraphy-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <LCALLIG.TTF
_LFAX LucidaFax " cp1252-encoding ReEncodeFont " <cp1252.enc <LFAX.TTF
_LFAXD LucidaFax-Demi " cp1252-encoding ReEncodeFont " <cp1252.enc <LFAXD.TTF
_LFAXDI LucidaFax-DemiItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <LFAXDI.TTF
_LFAXI LucidaFax-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <LFAXI.TTF
_LHANDW LucidaHandwriting-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <LHANDW.TTF
_LSANS LucidaSans " cp1252-encoding ReEncodeFont " <cp1252.enc <LSANS.TTF
_LSANSD LucidaSans-Demi " cp1252-encoding ReEncodeFont " <cp1252.enc <LSANS.D.TTF
_LSANSDI LucidaSans-DemiItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <LSANS.DI.TTF
_LSANSI LucidaSans-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <LSANSI.TTF
_LTYPE LucidaSans-Typewriter " cp1252-encoding ReEncodeFont " <cp1252.enc <LTYPE.TTF
_LTYPEB LucidaSans-TypewriterBold " cp1252-encoding ReEncodeFont " <cp1252.enc <LTYPEB.TTF
_LTYPEBO LucidaSans-TypewriterBoldOblique " cp1252-encoding ReEncodeFont " <cp1252.enc <LTYPEBO.TTF
_LTYPEO LucidaSans-TypewriterOblique " cp1252-encoding ReEncodeFont " <cp1252.enc <LTYPEO.TTF
_MAGNETOB Magneto-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <MAGNETOB.TTF
_MAIAN MaianraGD-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <MAIAN.TTF
_MATURASC MaturaMTScriptCapitals " cp1252-encoding ReEncodeFont " <cp1252.enc <MATURASC.TTF
_MISTRAL Mistral " cp1252-encoding ReEncodeFont " <cp1252.enc <MISTRAL.TTF
_MOD20 Modern-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <MOD20.TTF
_MTCORSVA MonotypeCorsiva " cp1252-encoding ReEncodeFont " <cp1252.enc <MTCORSVA.TTF
_ahronbd AharoniBold " cp1255-encoding ReEncodeFont " <cp1255.enc <ahronbd.ttf
_andlso Andalus " ArabicMT-Encoding ReEncodeFont " <arabicmt.enc <andlso.ttf
_angsa AngsanaNew " cp1252-encoding ReEncodeFont " <cp1252.enc <angsa.ttf
_angsab AngsanaNew-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <angsab.ttf
_angsai AngsanaNew-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <angsai.ttf
_angsau AngsanaUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <angsau.ttf
_angsaub AngsanaUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <angsaub.ttf
_angsau AngsanaUPC-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <angsau.ttf
_angsaz AngsanaUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <angsauz.ttf
_angsaz AngsanaNew-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <angsaaz.ttf
_arial ArialMT " cp1252-encoding ReEncodeFont " <cp1252.enc <arial.ttf
_arial00 ArialMT " g0000-Encoding ReEncodeFont " <g0000.enc <arial.ttf
_arial01 ArialMT " g0100-Encoding ReEncodeFont " <g0100.enc <arial.ttf
_arial02 ArialMT " g0200-Encoding ReEncodeFont " <g0200.enc <arial.ttf
_arial03 ArialMT " g0300-Encoding ReEncodeFont " <g0300.enc <arial.ttf
_arial04 ArialMT " g0400-Encoding ReEncodeFont " <g0400.enc <arial.ttf
_arial05 ArialMT " g0500-Encoding ReEncodeFont " <g0500.enc <arial.ttf
_arial06 ArialMT " g0600-Encoding ReEncodeFont " <g0600.enc <arial.ttf
_arial1E ArialMT " g1E00-Encoding ReEncodeFont " <g1e00.enc <arial.ttf
_arial20 ArialMT " g2000-Encoding ReEncodeFont " <g2000.enc <arial.ttf
_arial21 ArialMT " g2100-Encoding ReEncodeFont " <g2100.enc <arial.ttf
_arial22 ArialMT " g2200-Encoding ReEncodeFont " <g2200.enc <arial.ttf
_arial23 ArialMT " g2300-Encoding ReEncodeFont " <g2300.enc <arial.ttf
_arial25 ArialMT " g2500-Encoding ReEncodeFont " <g2500.enc <arial.ttf
_arial26 ArialMT " g2600-Encoding ReEncodeFont " <g2600.enc <arial.ttf
_arialVI ArialMT " viscii-encoding ReEncodeFont " <viscii.enc <arial.ttf
_arialW1 ArialMT " cp1251-encoding ReEncodeFont " <cp1251.enc <arial.ttf
_arialW3 ArialMT " cp1253-encoding ReEncodeFont " <cp1253.enc <arial.ttf
_arialW5 ArialMT " cp1255-encoding ReEncodeFont " <cp1255.enc <arial.ttf
_arialW6 ArialMT " cp1256-encoding ReEncodeFont " <cp1256.enc <arial.ttf
_arial_0 ArialMT " cp1252-encoding ReEncodeFont " <cp1252.enc <arial_0.ttf
_arial__000 ArialMT " g0000-Encoding ReEncodeFont " <g0000.enc <arial_0.ttf
_arial__001 ArialMT " g0100-Encoding ReEncodeFont " <g0100.enc <arial_0.ttf
_arial__002 ArialMT " g0200-Encoding ReEncodeFont " <g0200.enc <arial_0.ttf
_arial__003 ArialMT " g0300-Encoding ReEncodeFont " <g0300.enc <arial_0.ttf
_arial__004 ArialMT " g0400-Encoding ReEncodeFont " <g0400.enc <arial_0.ttf
_arial__005 ArialMT " g0500-Encoding ReEncodeFont " <g0500.enc <arial_0.ttf
_arial__006 ArialMT " g0600-Encoding ReEncodeFont " <g0600.enc <arial_0.ttf
_arial__01E ArialMT " g1E00-Encoding ReEncodeFont " <g1e00.enc <arial_0.ttf
_arial__020 ArialMT " g2000-Encoding ReEncodeFont " <g2000.enc <arial_0.ttf
_arial__021 ArialMT " g2100-Encoding ReEncodeFont " <g2100.enc <arial_0.ttf
_arial__022 ArialMT " g2200-Encoding ReEncodeFont " <g2200.enc <arial_0.ttf
_arial__023 ArialMT " g2300-Encoding ReEncodeFont " <g2300.enc <arial_0.ttf
_arial__025 ArialMT " g2500-Encoding ReEncodeFont " <g2500.enc <arial_0.ttf
_arial__026 ArialMT " g2600-Encoding ReEncodeFont " <g2600.enc <arial_0.ttf

_arial_OVI ArialMT " viscii-encoding ReEncodeFont " <viscii.enc <arial_0.ttf
_arial_OW1 ArialMT " cp1251-encoding ReEncodeFont " <cp1251.enc <arial_0.ttf
_arial_OW3 ArialMT " cp1253-encoding ReEncodeFont " <cp1253.enc <arial_0.ttf
_arial_OW5 ArialMT " cp1255-encoding ReEncodeFont " <cp1255.enc <arial_0.ttf
_arial_OW6 ArialMT " cp1256-encoding ReEncodeFont " <cp1256.enc <arial_0.ttf
_arialbd Arial-BoldMT " cp1252-encoding ReEncodeFont " <cp1252.enc <arialbd.ttf
_arialbd00 Arial-BoldMT " g0000-Encoding ReEncodeFont " <g0000.enc <arialbd.ttf
_arialbd01 Arial-BoldMT " g0100-Encoding ReEncodeFont " <g0100.enc <arialbd.ttf
_arialbd02 Arial-BoldMT " g0200-Encoding ReEncodeFont " <g0200.enc <arialbd.ttf
_arialbd03 Arial-BoldMT " g0300-Encoding ReEncodeFont " <g0300.enc <arialbd.ttf
_arialbd04 Arial-BoldMT " g0400-Encoding ReEncodeFont " <g0400.enc <arialbd.ttf
_arialbd05 Arial-BoldMT " g0500-Encoding ReEncodeFont " <g0500.enc <arialbd.ttf
_arialbd06 Arial-BoldMT " g0600-Encoding ReEncodeFont " <g0600.enc <arialbd.ttf
_arialbd1E Arial-BoldMT " g1E00-Encoding ReEncodeFont " <g1e00.enc <arialbd.ttf
_arialbd20 Arial-BoldMT " g2000-Encoding ReEncodeFont " <g2000.enc <arialbd.ttf
_arialbd21 Arial-BoldMT " g2100-Encoding ReEncodeFont " <g2100.enc <arialbd.ttf
_arialbd22 Arial-BoldMT " g2200-Encoding ReEncodeFont " <g2200.enc <arialbd.ttf
_arialbd23 Arial-BoldMT " g2300-Encoding ReEncodeFont " <g2300.enc <arialbd.ttf
_arialbd25 Arial-BoldMT " g2500-Encoding ReEncodeFont " <g2500.enc <arialbd.ttf
_arialbd26 Arial-BoldMT " g2600-Encoding ReEncodeFont " <g2600.enc <arialbd.ttf
_arialbdVI Arial-BoldMT " viscii-encoding ReEncodeFont " <viscii.enc <arialbd.ttf
_arialbdW1 Arial-BoldMT " cp1251-encoding ReEncodeFont " <cp1251.enc <arialbd.ttf
_arialbdW3 Arial-BoldMT " cp1253-encoding ReEncodeFont " <cp1253.enc <arialbd.ttf
_arialbdW5 Arial-BoldMT " cp1255-encoding ReEncodeFont " <cp1255.enc <arialbd.ttf
_arialbdW6 Arial-BoldMT " cp1256-encoding ReEncodeFont " <cp1256.enc <arialbd.ttf
_arialbi Arial-BoldItalicMT " cp1252-encoding ReEncodeFont " <cp1252.enc <arialbi.ttf
_arialbi00 Arial-BoldItalicMT " g0000-Encoding ReEncodeFont " <g0000.enc <arialbi.ttf
_arialbi01 Arial-BoldItalicMT " g0100-Encoding ReEncodeFont " <g0100.enc <arialbi.ttf
_arialbi02 Arial-BoldItalicMT " g0200-Encoding ReEncodeFont " <g0200.enc <arialbi.ttf
_arialbi03 Arial-BoldItalicMT " g0300-Encoding ReEncodeFont " <g0300.enc <arialbi.ttf
_arialbi04 Arial-BoldItalicMT " g0400-Encoding ReEncodeFont " <g0400.enc <arialbi.ttf
_arialbi05 Arial-BoldItalicMT " g0500-Encoding ReEncodeFont " <g0500.enc <arialbi.ttf
_arialbi1E Arial-BoldItalicMT " g1E00-Encoding ReEncodeFont " <g1e00.enc <arialbi.ttf
_arialbi20 Arial-BoldItalicMT " g2000-Encoding ReEncodeFont " <g2000.enc <arialbi.ttf
_arialbi21 Arial-BoldItalicMT " g2100-Encoding ReEncodeFont " <g2100.enc <arialbi.ttf
_arialbi22 Arial-BoldItalicMT " g2200-Encoding ReEncodeFont " <g2200.enc <arialbi.ttf
_arialbi23 Arial-BoldItalicMT " g2300-Encoding ReEncodeFont " <g2300.enc <arialbi.ttf
_arialbi25 Arial-BoldItalicMT " g2500-Encoding ReEncodeFont " <g2500.enc <arialbi.ttf
_arialbi26 Arial-BoldItalicMT " g2600-Encoding ReEncodeFont " <g2600.enc <arialbi.ttf
_arialbiVI Arial-BoldItalicMT " viscii-encoding ReEncodeFont " <viscii.enc <arialbi.ttf
_arialbiW1 Arial-BoldItalicMT " cp1251-encoding ReEncodeFont " <cp1251.enc <arialbi.ttf
_arialbiW3 Arial-BoldItalicMT " cp1253-encoding ReEncodeFont " <cp1253.enc <arialbi.ttf
_arialbiW5 Arial-BoldItalicMT " cp1255-encoding ReEncodeFont " <cp1255.enc <arialbi.ttf
_arialbiW6 Arial-BoldItalicMT " cp1256-encoding ReEncodeFont " <cp1256.enc <arialbi.ttf
_ariali Arial-ItalicMT " cp1252-encoding ReEncodeFont " <cp1252.enc <ariali.ttf
_ariali00 Arial-ItalicMT " g0000-Encoding ReEncodeFont " <g0000.enc <ariali.ttf
_ariali01 Arial-ItalicMT " g0100-Encoding ReEncodeFont " <g0100.enc <ariali.ttf
_ariali02 Arial-ItalicMT " g0200-Encoding ReEncodeFont " <g0200.enc <ariali.ttf
_ariali03 Arial-ItalicMT " g0300-Encoding ReEncodeFont " <g0300.enc <ariali.ttf
_ariali04 Arial-ItalicMT " g0400-Encoding ReEncodeFont " <g0400.enc <ariali.ttf
_ariali05 Arial-ItalicMT " g0500-Encoding ReEncodeFont " <g0500.enc <ariali.ttf
_ariali1E Arial-ItalicMT " g1E00-Encoding ReEncodeFont " <g1e00.enc <ariali.ttf
_ariali20 Arial-ItalicMT " g2000-Encoding ReEncodeFont " <g2000.enc <ariali.ttf
_ariali21 Arial-ItalicMT " g2100-Encoding ReEncodeFont " <g2100.enc <ariali.ttf
_ariali22 Arial-ItalicMT " g2200-Encoding ReEncodeFont " <g2200.enc <ariali.ttf
_ariali23 Arial-ItalicMT " g2300-Encoding ReEncodeFont " <g2300.enc <ariali.ttf
_ariali25 Arial-ItalicMT " g2500-Encoding ReEncodeFont " <g2500.enc <ariali.ttf
_ariali26 Arial-ItalicMT " g2600-Encoding ReEncodeFont " <g2600.enc <ariali.ttf
_arialiVI Arial-ItalicMT " viscii-encoding ReEncodeFont " <viscii.enc <ariali.ttf
_arialiW1 Arial-ItalicMT " cp1251-encoding ReEncodeFont " <cp1251.enc <ariali.ttf
_arialiW3 Arial-ItalicMT " cp1253-encoding ReEncodeFont " <cp1253.enc <ariali.ttf
_arialiW5 Arial-ItalicMT " cp1255-encoding ReEncodeFont " <cp1255.enc <ariali.ttf
_arialiW6 Arial-ItalicMT " cp1256-encoding ReEncodeFont " <cp1256.enc <ariali.ttf
_ariblk Arial-Black " cp1252-encoding ReEncodeFont " <cp1252.enc <ariblk.ttf
_ariblk_0 Arial-Black " cp1252-encoding ReEncodeFont " <cp1252.enc <ariblk_0.ttf
_artrbdo ArabicTransparent-Bold " ArabicMT-Encoding ReEncodeFont " <arabicmt.enc <artrbdo.ttf
_artro ArabicTransparent " ArabicMT-Encoding ReEncodeFont " <arabicmt.enc <artro.ttf
_ass1 Shree-Ass-001 " cp1252-encoding ReEncodeFont " <cp1252.enc <ass1.ttf
_ass2 Shree-Ass-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <ass2.ttf
_ayummy AYummyApology " cp1252-encoding ReEncodeFont " <cp1252.enc <ayummy.ttf
_ban1 Shree-Ban-001 " cp1252-encoding ReEncodeFont " <cp1252.enc <ban1.ttf

_ban2 Shree-Ban-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <ban2.ttf
_bchb CharterBT-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <bchb.pfb
_bchbi CharterBT-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <bchbi.pfb
_bchr CharterBT-Roman " cp1252-encoding ReEncodeFont " <cp1252.enc <bchr.pfb
_bchri CharterBT-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <bchri.pfb
_beauti BeautifulES " cp1252-encoding ReEncodeFont " <cp1252.enc <beauti.ttf
_browa BrowalliaNew " cp1252-encoding ReEncodeFont " <cp1252.enc <browa.ttf
_browab BrowalliaNew-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <browab.ttf
_browai BrowalliaNew-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <browai.ttf
_browau BrowalliaUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <browau.ttf
_browaub BrowalliaUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <browaub.ttf
_browauai BrowalliaUPC-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <browauai.ttf
_browauz BrowalliaUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <browauz.ttf
_browaz BrowalliaNew-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <browaz.ttf
_bzcb ZapfChanceryITCbyBT-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <bzcb.ttf
_bzcm ZapfChanceryITCbyBT-Medium " cp1252-encoding ReEncodeFont " <cp1252.enc <bzcm.ttf
_cardo Cardo " cp1252-encoding ReEncodeFont " <cp1252.enc <cardo.ttf
_cardow3 Cardo " cp1253-encoding ReEncodeFont " <cp1253.enc <cardo.ttf
_champ CACChampagne " cp1252-encoding ReEncodeFont " <cp1252.enc <champ.ttf
_comic ComicSansMS " cp1252-encoding ReEncodeFont " <cp1252.enc <comic.ttf
_comicw1 ComicSansMS " cp1251-encoding ReEncodeFont " <cp1251.enc <comic.ttf
_comicw3 ComicSansMS " cp1253-encoding ReEncodeFont " <cp1253.enc <comic.ttf
_comic_0 ComicSansMS " cp1252-encoding ReEncodeFont " <cp1252.enc <comic_0.ttf
_comic_0w1 ComicSansMS " cp1251-encoding ReEncodeFont " <cp1251.enc <comic_0.ttf
_comic_0w3 ComicSansMS " cp1253-encoding ReEncodeFont " <cp1253.enc <comic_0.ttf
_comicbd ComicSansMS-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <comicbd.ttf
_comicbdw1 ComicSansMS-Bold " cp1251-encoding ReEncodeFont " <cp1251.enc <comicbd.ttf
_comicbdw3 ComicSansMS-Bold " cp1253-encoding ReEncodeFont " <cp1253.enc <comicbd.ttf
_cordia CordiaNew " cp1252-encoding ReEncodeFont " <cp1252.enc <cordia.ttf
_cordiaab CordiaNew-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <cordiab.ttf
_cordiaai CordiaNew-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <cordiai.ttf
_cordiau CordiaUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <cordiau.ttf
_cordiaub CordiaUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <cordiaub.ttf
_cordiaui CordiaUPC-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <cordiaui.ttf
_cordiauz CordiaUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <cordiauz.ttf
_cordiaz CordiaNew-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <cordiaz.ttf
_cour CourierNewPSMT " cp1252-encoding ReEncodeFont " <cp1252.enc <cour.ttf
_cour00 CourierNewPSMT " g0000-Encoding ReEncodeFont " <g0000.enc <cour.ttf
_cour01 CourierNewPSMT " g0100-Encoding ReEncodeFont " <g0100.enc <cour.ttf
_cour02 CourierNewPSMT " g0200-Encoding ReEncodeFont " <g0200.enc <cour.ttf
_cour03 CourierNewPSMT " g0300-Encoding ReEncodeFont " <g0300.enc <cour.ttf
_cour04 CourierNewPSMT " g0400-Encoding ReEncodeFont " <g0400.enc <cour.ttf
_cour05 CourierNewPSMT " g0500-Encoding ReEncodeFont " <g0500.enc <cour.ttf
_cour06 CourierNewPSMT " g0600-Encoding ReEncodeFont " <g0600.enc <cour.ttf
_cour1E CourierNewPSMT " g1E00-Encoding ReEncodeFont " <g1E00.enc <cour.ttf
_cour20 CourierNewPSMT " g2000-Encoding ReEncodeFont " <g2000.enc <cour.ttf
_cour21 CourierNewPSMT " g2100-Encoding ReEncodeFont " <g2100.enc <cour.ttf
_cour22 CourierNewPSMT " g2200-Encoding ReEncodeFont " <g2200.enc <cour.ttf
_cour23 CourierNewPSMT " g2300-Encoding ReEncodeFont " <g2300.enc <cour.ttf
_cour25 CourierNewPSMT " g2500-Encoding ReEncodeFont " <g2500.enc <cour.ttf
_cour26 CourierNewPSMT " g2600-Encoding ReEncodeFont " <g2600.enc <cour.ttf
_courVI CourierNewPSMT " viscii-encoding ReEncodeFont " <viscii.enc <cour.ttf
_courw1 CourierNewPSMT " cp1251-encoding ReEncodeFont " <cp1251.enc <cour.ttf
_courw3 CourierNewPSMT " cp1253-encoding ReEncodeFont " <cp1253.enc <cour.ttf
_courw5 CourierNewPSMT " cp1255-encoding ReEncodeFont " <cp1255.enc <cour.ttf
_courw6 CourierNewPSMT " cp1256-encoding ReEncodeFont " <cp1256.enc <cour.ttf
_cour__0 CourierNewPSMT " cp1252-encoding ReEncodeFont " <cp1252.enc <cour__0.ttf
_cour__000 CourierNewPSMT " g0000-Encoding ReEncodeFont " <g0000.enc <cour__0.ttf
_cour__001 CourierNewPSMT " g0100-Encoding ReEncodeFont " <g0100.enc <cour__0.ttf
_cour__002 CourierNewPSMT " g0200-Encoding ReEncodeFont " <g0200.enc <cour__0.ttf
_cour__003 CourierNewPSMT " g0300-Encoding ReEncodeFont " <g0300.enc <cour__0.ttf
_cour__004 CourierNewPSMT " g0400-Encoding ReEncodeFont " <g0400.enc <cour__0.ttf
_cour__005 CourierNewPSMT " g0500-Encoding ReEncodeFont " <g0500.enc <cour__0.ttf
_cour__006 CourierNewPSMT " g0600-Encoding ReEncodeFont " <g0600.enc <cour__0.ttf
_cour__01E CourierNewPSMT " g1E00-Encoding ReEncodeFont " <g1E00.enc <cour__0.ttf
_cour__020 CourierNewPSMT " g2000-Encoding ReEncodeFont " <g2000.enc <cour__0.ttf
_cour__021 CourierNewPSMT " g2100-Encoding ReEncodeFont " <g2100.enc <cour__0.ttf
_cour__022 CourierNewPSMT " g2200-Encoding ReEncodeFont " <g2200.enc <cour__0.ttf
_cour__023 CourierNewPSMT " g2300-Encoding ReEncodeFont " <g2300.enc <cour__0.ttf
_cour__025 CourierNewPSMT " g2500-Encoding ReEncodeFont " <g2500.enc <cour__0.ttf
_cour__026 CourierNewPSMT " g2600-Encoding ReEncodeFont " <g2600.enc <cour__0.ttf

_dev2 Shree-Dev-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <dev2.ttf
 _estre EstrangeloEdessa " cp1252-encoding ReEncodeFont " <cp1252.enc <estre.ttf
 _faum Augie " cp1252-encoding ReEncodeFont " <cp1252.enc <faum.pfb
 _faumn Augie " 0.1 SlantFont 0.9 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <faum.pfb
 _faumno Augie " 0.5 SlantFont 0.9 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <faum.pfb
 _fbsm BrushScript-Italic " -0.4 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fbsmi.pfa
 _fbsmi BrushScript-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <fbsmi.pfa
 _fcmbl SFBI1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmbl.pfb
 _fcmblW1 SFBI1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmbl.pfb
 _fcmblx SFBX1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmblx.pfb
 _fcmblxW1 SFBX1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmblx.pfb
 _fcmcc SFCC1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmcc.pfb
 _fcmccW1 SFCC1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmcc.pfb
 _fcmcco SFCC1000 " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmcc.pfb
 _fcmccoW1 SFCC1000 " 0.2 SlantFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmcc.pfb
 _fcmhd SFDH1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmhd.pfb
 _fcmhdW1 SFDH1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmhd.pfb
 _fcmhdW1 SFDH1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmhd.pfb
 _fcmhdW1 SFDH1000 " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmhd.pfb
 _fcmhdW1 SFDH1000 " 0.2 SlantFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmhd.pfb
 _fcmfb SFFB1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmfb.pfb
 _fcmfbW1 SFFB1000 " 0.8 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmfb.pfb
 _fcmfbW1 SFFB1000 " 0.8 ExtendFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmfb.pfb
 _fcmfbc SFFB1000 " 0.2 SlantFont 0.8 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmfb.pfb
 _fcmfbcW1 SFFB1000 " 0.2 SlantFont 0.8 ExtendFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmfb.pfb
 _fcmff SFFF1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmff.pfb
 _fcmffi SFFI1000 " 0.1 SlantFont 1.05 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmfi.pfb
 _fcmffiW1 SFFI1000 " 0.1 SlantFont 1.05 ExtendFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmfi.pfb
 _fcmffu SFFF1000 " 0.1 SlantFont 1.3 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmff.pfb
 _fcmffuW1 SFFF1000 " 0.1 SlantFont 1.3 ExtendFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmff.pfb
 _fcmfi SFFI1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmfi.pfb
 _fcmr SFRM1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmr.pfb
 _fcmrW1 SFRM1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmr.pfb
 _fcmss SFSS1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmss.pfb
 _fcmssW1 SFSS1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmss.pfb
 _fcmssW1 SFSS1000 " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmss.pfb
 _fcmssW1 SFSS1000 " 0.2 SlantFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmss.pfb
 _fcmxs SF SX1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmxs.pfb
 _fcmxsW1 SF SX1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmxs.pfb
 _fcmxso SF SX1000 " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmxs.pfb
 _fcmxsoW1 SF SX1000 " 0.2 SlantFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmxs.pfb
 _fcmti SFTI1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmti.pfb
 _fcmtiW1 SFTI1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmti.pfb
 _fcmtt SFTT1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmtt.pfb
 _fcmttW1 SFTT1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmtt.pfb
 _fcmtto SFTT1000 " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmtt.pfb
 _fcmttoW1 SFTT1000 " 0.2 SlantFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmtt.pfb
 _fcmvt SFVT1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmvt.pfb
 _fcmvtW1 SFVT1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmvt.pfb
 _fcmvto SFVT1000 " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmvt.pfb
 _fcmvtoW1 SFVT1000 " 0.2 SlantFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmvt.pfb
 _fcmxc SFXC1000 " cp1252-encoding ReEncodeFont " <cp1252.enc <fcmxc.pfb
 _fcmxcW1 SFXC1000 " cp1251-encoding ReEncodeFont " <cp1251.enc <fcmxc.pfb
 _fcmxco SFXC1000 " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fcmxc.pfb
 _fcmxcoW1 SFXC1000 " 0.2 SlantFont cp1251-encoding ReEncodeFont " <cp1251.enc <fcmxc.pfb
 _fdrr DayRoman " cp1252-encoding ReEncodeFont " <cp1252.enc <fdrr.pfb
 _fdrr DayRoman " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fdrr.pfb
 _fdxr DayRoman-Expert " cp1252-encoding ReEncodeFont " <cp1252.enc <fdxr.pfb
 _fdxro DayRoman-Expert " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fdxr.pfb
 _fgcm GoodCityModern " cp1252-encoding ReEncodeFont " <cp1252.enc <fgcm.pfb
 _fgcmx GoodCityModern " cp1252-encoding ReEncodeFont " <cp1252.enc <fgcm.pfb
 _fgcmx GoodCityModern " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <fgcm.pfb
 _fgmb Baramond " cp1252-encoding ReEncodeFont " <cp1252.enc <fgmb.ttf
 _fgmbi Baramond " cp1252-encoding ReEncodeFont " <cp1252.enc <fgmbi.ttf
 _fgmr Baramond " cp1252-encoding ReEncodeFont " <cp1252.enc <fgmr.ttf
 _fgmri Baramond " cp1252-encoding ReEncodeFont " <cp1252.enc <fgmri.ttf
 _fnfr National-FirstFont " cp1252-encoding ReEncodeFont " <cp1252.enc <fnfr.pfb
 _fnfr National-FirstFont " 1.2 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fnfr.pfb
 _fnfr National-FirstFont " 0.2 SlantFont 1.2 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fnfr.pfb
 _framd FranklinGothic-Medium " cp1252-encoding ReEncodeFont " <cp1252.enc <framd.ttf
 _framdW1 FranklinGothic-Medium " cp1251-encoding ReEncodeFont " <cp1251.enc <framd.ttf

_framdW3 FranklinGothic-Medium " cp1253-encoding ReEncodeFont " <cp1253.enc <framd.ttf
 _framd_0 FranklinGothic-Medium " cp1252-encoding ReEncodeFont " <cp1252.enc <framd_0.ttf
 _framd_0W1 FranklinGothic-Medium " cp1251-encoding ReEncodeFont " <cp1251.enc <framd_0.ttf
 _framd_0W3 FranklinGothic-Medium " cp1253-encoding ReEncodeFont " <cp1253.enc <framd_0.ttf
 _framdit FranklinGothic-MediumItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <framdit.ttf
 _framditW1 FranklinGothic-MediumItalic " cp1251-encoding ReEncodeFont " <cp1251.enc <framdit.ttf
 _framditW3 FranklinGothic-MediumItalic " cp1253-encoding ReEncodeFont " <cp1253.enc <framdit.ttf
 _frank FrankRuehl " cp1255-encoding ReEncodeFont " <cp1255.enc <frank.ttf
 _fsab SanityWideBold " cp1252-encoding ReEncodeFont " <cp1252.enc <fsab.pfb
 _fsabx SanityWideBold " 1.1 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fsab.pfb
 _fsabxo SanityWideBold " 0.2 SlantFont 1.1 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fsab.pfb
 _fsar Sanity-Wide " cp1252-encoding ReEncodeFont " <cp1252.enc <fsar.pfb
 _fsarx Sanity-Wide " 1.2 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fsar.pfb
 _fsarxo Sanity-Wide " 0.2 SlantFont 1.2 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <fsar.pfb
 _gautami Gautami " cp1252-encoding ReEncodeFont " <cp1252.enc <gautami.ttf
 _geni Gentium-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <geni.ttf
 _geniW3 Gentium-Italic " cp1253-encoding ReEncodeFont " <cp1253.enc <geni.ttf
 _genr Gentium " cp1252-encoding ReEncodeFont " <cp1252.enc <genr.ttf
 _genrW3 Gentium " cp1253-encoding ReEncodeFont " <cp1253.enc <genr.ttf
 _georgia Georgia " cp1252-encoding ReEncodeFont " <cp1252.enc <georgia.ttf
 _georgia0 Georgia " cp1252-encoding ReEncodeFont " <cp1252.enc <georgia0.ttf
 _georgia0W1 Georgia " cp1251-encoding ReEncodeFont " <cp1251.enc <georgia0.ttf
 _georgia0W3 Georgia " cp1253-encoding ReEncodeFont " <cp1253.enc <georgia0.ttf
 _georgiaW1 Georgia " cp1251-encoding ReEncodeFont " <cp1251.enc <georgia.ttf
 _georgiaW3 Georgia " cp1253-encoding ReEncodeFont " <cp1253.enc <georgia.ttf
 _georgiab Georgia-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <georgiab.ttf
 _georgiabW1 Georgia-Bold " cp1251-encoding ReEncodeFont " <cp1251.enc <georgiab.ttf
 _georgiabW3 Georgia-Bold " cp1253-encoding ReEncodeFont " <cp1253.enc <georgiab.ttf
 _georgiai Georgia-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <georgiai.ttf
 _georgiaiW1 Georgia-Italic " cp1251-encoding ReEncodeFont " <cp1251.enc <georgiai.ttf
 _georgiaiW3 Georgia-Italic " cp1253-encoding ReEncodeFont " <cp1253.enc <georgiai.ttf
 _georgiaz Georgia-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <georgiaz.ttf
 _georgiazW1 Georgia-BoldItalic " cp1251-encoding ReEncodeFont " <cp1251.enc <georgiaz.ttf
 _georgiazW3 Georgia-BoldItalic " cp1253-encoding ReEncodeFont " <cp1253.enc <georgiaz.ttf
 _guj1 Shree-Guj-001 " cp1252-encoding ReEncodeFont " <cp1252.enc <guj1.ttf
 _guj2 Shree-Guj-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <guj2.ttf
 _hlmb LuxiMono-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <hlmb.pfb
 _hlmbc LuxiMono-Bold " 0.75 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <hlmb.pfb
 _hlmbco LuxiMono-Bold " 0.25 SlantFont 0.75 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <hlmb.pfb
 _hlmbx LuxiMono-Bold " 0.25 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <hlmb.pfb
 _hlmr LuxiMono " cp1252-encoding ReEncodeFont " <cp1252.enc <hlmr.pfb
 _hlmrc LuxiMono " 0.75 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <hlmr.pfb
 _hlmrco LuxiMono " 0.25 SlantFont 0.75 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <hlmr.pfb
 _hlmro LuxiMono " 0.25 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <hlmr.pfb
 _impact Impact " cp1252-encoding ReEncodeFont " <cp1252.enc <impact.ttf
 _impactW1 Impact " cp1251-encoding ReEncodeFont " <cp1251.enc <impact.ttf
 _impactW3 Impact " cp1253-encoding ReEncodeFont " <cp1253.enc <impact.ttf
 _impact_0 Impact " cp1252-encoding ReEncodeFont " <cp1252.enc <impact_0.ttf
 _kan1 Shree-Kan-001 " cp1252-encoding ReEncodeFont " <cp1252.enc <kan1.ttf
 _kan2 Shree-Kan-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <kan2.ttf
 _kartika Unknown " cp1252-encoding ReEncodeFont " <cp1252.enc <kartika.ttf
 _l_10646 LucidaSansUnicode " cp1252-encoding ReEncodeFont " <cp1252.enc <l_10646.ttf
 _l_106460 LucidaSansUnicode " cp1252-encoding ReEncodeFont " <cp1252.enc <l_106460.ttf
 _l_10646W3 LucidaSansUnicode " cp1253-encoding ReEncodeFont " <cp1253.enc <l_10646.ttf
 _l_10646W5 LucidaSansUnicode " cp1255-encoding ReEncodeFont " <cp1255.enc <l_10646.ttf
 _latha Latha " cp1252-encoding ReEncodeFont " <cp1252.enc <latha.ttf
 _lucon LucidaConsole " cp1252-encoding ReEncodeFont " <cp1252.enc <lucon.ttf
 _luconW3 LucidaConsole " cp1253-encoding ReEncodeFont " <cp1253.enc <lucon.ttf
 _lucon_0 LucidaConsole " cp1252-encoding ReEncodeFont " <cp1252.enc <lucon_0.ttf
 _lvnm LevenimMT " cp1255-encoding ReEncodeFont " <cp1255.enc <lvnm.ttf
 _lvnmbd LevenimMTBold " cp1255-encoding ReEncodeFont " <cp1255.enc <lvnmbd.ttf
 _mal1 Shree-Mal-001 " cp1252-encoding ReEncodeFont " <cp1252.enc <mal1.ttf
 _mal2 Shree-Mal-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <mal2.ttf
 _mangal Mangal-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <mangal.ttf
 _marlett Unknown " cp1252-encoding ReEncodeFont " <cp1252.enc <marlett.ttf
 _marlett0 Unknown " cp1252-encoding ReEncodeFont " <cp1252.enc <marlett0.ttf
 _microssoft MicrosoftSansSerif " cp1252-encoding ReEncodeFont " <cp1252.enc <microssoft.ttf
 _microssoft0 MicrosoftSansSerif " cp1252-encoding ReEncodeFont " <cp1252.enc <microssoft0.ttf
 _mriam Miriam " cp1255-encoding ReEncodeFont " <cp1255.enc <mriam.ttf
 _mriamc MiriamFixed " cp1255-encoding ReEncodeFont " <cp1255.enc <mriamc.ttf
 _mriamfx FixedMiriamTransparent " cp1255-encoding ReEncodeFont " <cp1255.enc <mriamfx.ttf

_mriamtr MiriamTransparent " cp1255-encoding ReEncodeFont " <cp1255.enc <mriamtr.ttf
 _mwboli MVBoli " cp1252-encoding ReEncodeFont " <cp1252.enc <mwboli.ttf
 _ncrb IBMCourier-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <ncrb.pfa
 _ncrbc IBMCourier-Bold " 0.75 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <ncrb.pfa
 _ncrbc0 IBMCourier-Bold " 0.25 SlantFont 0.75 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <ncrb.pfa
 _ncrr IBMCourier " cp1252-encoding ReEncodeFont " <cp1252.enc <ncrr.pfa
 _ncrrc IBMCourier " 0.75 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <ncrr.pfa
 _ncrrco IBMCourier " 0.25 SlantFont 0.75 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <ncrr.pfa
 _nrkis Narkisim " cp1255-encoding ReEncodeFont " <cp1255.enc <nrkis.ttf
 _or1 Shree-Ori-001 " cp1252-encoding ReEncodeFont " <cp1252.enc <or1.ttf
 _ori2 Shree-Ori-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <ori2.ttf
 _pcrb Courier-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc
 _perbo Courier-BoldOblique " cp1252-encoding ReEncodeFont " <cp1252.enc
 _pcrr Courier " cp1252-encoding ReEncodeFont " <cp1252.enc
 _pcrr0 Courier-Oblique " cp1252-encoding ReEncodeFont " <cp1252.enc
 _phvb Helvetica-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc
 _phvbo Helvetica-BoldOblique " cp1252-encoding ReEncodeFont " <cp1252.enc
 _phvr Helvetica " cp1252-encoding ReEncodeFont " <cp1252.enc
 _phvro Helvetica-Oblique " cp1252-encoding ReEncodeFont " <cp1252.enc
 _psyr Symbol " Symbol-Encoding ReEncodeFont " <psy.enc
 _ptmb Times-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc
 _ptmbi Times-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc
 _ptmr Times-Roman " cp1252-encoding ReEncodeFont " <cp1252.enc
 _ptmri Times-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc
 _pun1 Shree-Pun-001 " cp1252-encoding ReEncodeFont " <cp1252.enc <pun1.ttf
 _pun2 Shree-Pun-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <pun2.ttf
 _putb Utopia-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <putb.pfb
 _putbi Utopia-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <putbi.pfb
 _putr Utopia-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <putr.pfb
 _putri Utopia-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <putri.pfb
 _pzd ZapfDingbats " ZapfDingbats-Encoding ReEncodeFont " <pzd.enc
 _raavi Raavi " cp1252-encoding ReEncodeFont " <cp1252.enc <raavi.ttf
 _rod Rod " cp1255-encoding ReEncodeFont " <cp1255.enc <rod.ttf
 _rodtr RodTransparent " cp1255-encoding ReEncodeFont " <cp1255.enc <rodtr.ttf
 _scrbc IBMCourier-Bold " 0.5 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <ncrb.pfa
 _scrbc0 IBMCourier-Bold " 0.25 SlantFont 0.5 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <ncrb.pfa
 _scrrc IBMCourier " 0.5 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <ncrr.pfa
 _scrrco IBMCourier " 0.25 SlantFont 0.5 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <ncrr.pfa
 _shruti Shruti " cp1252-encoding ReEncodeFont " <cp1252.enc <shruti.ttf
 _shusha Shusha " cp1252-encoding ReEncodeFont " <cp1252.enc <shusha.ttf
 _simpbdo SimplifiedArabic-Bold " ArabicMT-Encoding ReEncodeFont " <arabicmt.enc <simpbdo.ttf
 _simpfxo SimplifiedArabicFixed " ArabicMT-Encoding ReEncodeFont " <arabicmt.enc <simpfxo.ttf
 _simpo SimplifiedArabic " ArabicMT-Encoding ReEncodeFont " <arabicmt.enc <simpo.ttf
 _sylfaen Sylfaen " cp1252-encoding ReEncodeFont " <cp1252.enc <sylfaen.ttf
 _sylfaenW1 Sylfaen " cp1251-encoding ReEncodeFont " <cp1251.enc <sylfaen.ttf
 _sylfaenW3 Sylfaen " cp1253-encoding ReEncodeFont " <cp1253.enc <sylfaen.ttf
 _symbol SymbolMT " cp1252-encoding ReEncodeFont " <cp1252.enc <symbol.ttf
 _tahoma Tahoma " cp1252-encoding ReEncodeFont " <cp1252.enc <tahoma.ttf
 _tahomaVI Tahoma " viscii-encoding ReEncodeFont " <viscii.enc <tahoma.ttf
 _tahomaW1 Tahoma " cp1251-encoding ReEncodeFont " <cp1251.enc <tahoma.ttf
 _tahomaW3 Tahoma " cp1253-encoding ReEncodeFont " <cp1253.enc <tahoma.ttf
 _tahomaW5 Tahoma " cp1255-encoding ReEncodeFont " <cp1255.enc <tahoma.ttf
 _tahomaW6 Tahoma " cp1256-encoding ReEncodeFont " <cp1256.enc <tahoma.ttf
 _tahomabd Tahoma-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <tahomabd.ttf
 _tahomabdVI Tahoma-Bold " viscii-encoding ReEncodeFont " <viscii.enc <tahomabd.ttf
 _tahomabdW1 Tahoma-Bold " cp1251-encoding ReEncodeFont " <cp1251.enc <tahomabd.ttf
 _tahomabdW3 Tahoma-Bold " cp1253-encoding ReEncodeFont " <cp1253.enc <tahomabd.ttf
 _tahomabdW5 Tahoma-Bold " cp1255-encoding ReEncodeFont " <cp1255.enc <tahomabd.ttf
 _tahomabdW6 Tahoma-Bold " cp1256-encoding ReEncodeFont " <cp1256.enc <tahomabd.ttf
 _tam1 Shree-Tam-001 " cp1252-encoding ReEncodeFont " <cp1252.enc <tam1.ttf
 _tam2 Shree-Tam-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <tam2.ttf
 _tel1 Shree-Tel-001 " cp1252-encoding ReEncodeFont " <cp1252.enc <tel1.ttf
 _tel2 Shree-Tel-002 " cp1252-encoding ReEncodeFont " <cp1252.enc <tel2.ttf
 _times TimesNewRomanPSMT " cp1252-encoding ReEncodeFont " <cp1252.enc <times.ttf
 _times00 TimesNewRomanPSMT " g0000-Encoding ReEncodeFont " <g0000.enc <times.ttf
 _times01 TimesNewRomanPSMT " g0100-Encoding ReEncodeFont " <g0100.enc <times.ttf
 _times02 TimesNewRomanPSMT " g0200-Encoding ReEncodeFont " <g0200.enc <times.ttf
 _times03 TimesNewRomanPSMT " g0300-Encoding ReEncodeFont " <g0300.enc <times.ttf
 _times04 TimesNewRomanPSMT " g0400-Encoding ReEncodeFont " <g0400.enc <times.ttf
 _times05 TimesNewRomanPSMT " g0500-Encoding ReEncodeFont " <g0500.enc <times.ttf
 _times06 TimesNewRomanPSMT " g0600-Encoding ReEncodeFont " <g0600.enc <times.ttf

_trado TraditionalArabic " ArabicMT-Encoding ReEncodeFont " <arabicmt.enc <trado.ttf
 _trebuc TrebuchetMS " cp1252-encoding ReEncodeFont " <cp1252.enc <trebuc.ttf
 _trebucW1 TrebuchetMS " cp1251-encoding ReEncodeFont " <cp1251.enc <trebuc.ttf
 _trebucW3 TrebuchetMS " cp1253-encoding ReEncodeFont " <cp1253.enc <trebuc.ttf
 _trebuchd TrebuchetMS-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <trebuchd.ttf
 _trebuchdW1 TrebuchetMS-Bold " cp1251-encoding ReEncodeFont " <cp1251.enc <trebuchd.ttf
 _trebuchdW3 TrebuchetMS-Bold " cp1253-encoding ReEncodeFont " <cp1253.enc <trebuchd.ttf
 _trebuchi Trebuchet-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <trebuchi.ttf
 _trebuchiW1 Trebuchet-BoldItalic " cp1251-encoding ReEncodeFont " <cp1251.enc <trebuchi.ttf
 _trebuchiW3 Trebuchet-BoldItalic " cp1253-encoding ReEncodeFont " <cp1253.enc <trebuchi.ttf
 _trebucit TrebuchetMS-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <trebucit.ttf
 _trebucitW1 TrebuchetMS-Italic " cp1251-encoding ReEncodeFont " <cp1251.enc <trebucit.ttf
 _trebucitW3 TrebuchetMS-Italic " cp1253-encoding ReEncodeFont " <cp1253.enc <trebucit.ttf
 _tunga Tunga-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <tunga.ttf
 _uagd URWGothicL-Demi " cp1252-encoding ReEncodeFont " <cp1252.enc <uagd.pfb
 _uagdo URWGothicL-Demi " 0.25 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <uagdo.pfb
 _uagk URWGothicL-Book " cp1252-encoding ReEncodeFont " <cp1252.enc <uagk.pfb
 _uagko URWGothicL-Book " 0.25 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <uagko.pfb
 _uaqr URWAntiquaT-RegularCondensed " 1.1 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <uaqr.pfb
 _uaqrc URWAntiquaT-RegularCondensed " cp1252-encoding ReEncodeFont " <cp1252.enc <uaqrc.pfb
 _uaqro URWAntiquaT-RegularCondensed " 0.2 SlantFont 1.1 ExtendFont cp1252-encoding ReEncodeFont " <cp1252.enc <uaqro.pfb
 _ubkd URWBookmanL-DemiBold " cp1252-encoding ReEncodeFont " <cp1252.enc <ubkd.pfb
 _ubkdi URWBookmanL-DemiBoldItal " cp1252-encoding ReEncodeFont " <cp1252.enc <ubkdi.pfb
 _ubkl URWBookmanL-Ligh " cp1252-encoding ReEncodeFont " <cp1252.enc <ubkl.pfb
 _ubkli URWBookmanL-LighItal " cp1252-encoding ReEncodeFont " <cp1252.enc <ubkli.pfb
 _ugkb URWGroteskT-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <ugkb.pfb
 _ugkbo URWGroteskT-Bold " 0.2 SlantFont cp1252-encoding ReEncodeFont " <cp1252.enc <ugkbo.pfb
 _unbc CenturySchL-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <unbc.pfb
 _unbci CenturySchL-BoldItal " cp1252-encoding ReEncodeFont " <cp1252.enc <unbci.pfb
 _uncr CenturySchL-Roma " cp1252-encoding ReEncodeFont " <cp1252.enc <uncr.pfb
 _uncri CenturySchL-Ital " cp1252-encoding ReEncodeFont " <cp1252.enc <uncri.pfb
 _upcdb DilleniaUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <upcdb.ttf
 _upcdbi DilleniaUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcdbi.ttf
 _upcdi DilleniaUPC-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcdi.ttf
 _upcdl DilleniaUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <upcdl.ttf
 _upceb EucrosiaUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <upceb.ttf
 _upcebi EucrosiaUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcebi.ttf
 _upcei EucrosiaUPC-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcei.ttf
 _upcel EucrosiaUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <upcel.ttf
 _upcfb FreesiaUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <upcfb.ttf
 _upcfbi FreesiaUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcfbi.ttf
 _upcfl FreesiaUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <upcfl.ttf
 _upcib IrisUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <upcib.ttf
 _upcibi IrisUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcibi.ttf
 _upcii IrisUPC-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcii.ttf
 _upcil IrisUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <upcil.ttf
 _upcjb JasmineUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <upcjb.ttf
 _upcjb1 JasmineUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcjb1.ttf
 _upcji JasmineUPC-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcji.ttf
 _upcjl JasmineUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <upcjl.ttf
 _upckb KodchiangUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <upckb.ttf
 _upckbi KodchiangUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <upckbi.ttf
 _upcki KodchiangUPC-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcki.ttf
 _upckl KodchiangUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <upckl.ttf
 _upclb LilyUPC-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <upclb.ttf
 _upclbi LilyUPC-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <upclbi.ttf
 _upcli LilyUPC-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <upcli.ttf
 _upcll LilyUPC " cp1252-encoding ReEncodeFont " <cp1252.enc <upcll.ttf
 _uplb URWPalladioL-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <uplb.pfb
 _uplbi URWPalladioL-BoldItal " cp1252-encoding ReEncodeFont " <cp1252.enc <uplbi.pfb
 _uplri URWPalladioL-Roma " cp1252-encoding ReEncodeFont " <cp1252.enc <uplri.pfb
 _uplri URWPalladioL-Ital " cp1252-encoding ReEncodeFont " <cp1252.enc <uplri.pfb
 _verdana Verdana " cp1252-encoding ReEncodeFont " <cp1252.enc <verdana.ttf
 _verdanaVI Verdana " viscii-encoding ReEncodeFont " <viscii.enc <verdana.ttf
 _verdanaW1 Verdana " cp1251-encoding ReEncodeFont " <cp1251.enc <verdana.ttf
 _verdanaW3 Verdana " cp1253-encoding ReEncodeFont " <cp1253.enc <verdana.ttf
 _verdanaW5 Verdana " cp1255-encoding ReEncodeFont " <cp1255.enc <verdana.ttf
 _verdanaW6 Verdana " cp1256-encoding ReEncodeFont " <cp1256.enc <verdana.ttf
 _verdanab Verdana-Bold " cp1252-encoding ReEncodeFont " <cp1252.enc <verdanab.ttf
 _verdanabVI Verdana-Bold " viscii-encoding ReEncodeFont " <viscii.enc <verdanab.ttf
 _verdanabW1 Verdana-Bold " cp1251-encoding ReEncodeFont " <cp1251.enc <verdanab.ttf

_verdanabW3 Verdana-Bold " cp1253-encoding ReEncodeFont " <cp1253.enc <verdanab.ttf
_verdanabW5 Verdana-Bold " cp1255-encoding ReEncodeFont " <cp1255.enc <verdanab.ttf
_verdanabW6 Verdana-Bold " cp1256-encoding ReEncodeFont " <cp1256.enc <verdanab.ttf
_verdanai Verdana-Italic " cp1252-encoding ReEncodeFont " <cp1252.enc <verdanai.ttf
_verdanaiVI Verdana-Italic " viscii-encoding ReEncodeFont " <viscii.enc <verdanai.ttf
_verdanaiW1 Verdana-Italic " cp1251-encoding ReEncodeFont " <cp1251.enc <verdanai.ttf
_verdanaiW3 Verdana-Italic " cp1253-encoding ReEncodeFont " <cp1253.enc <verdanai.ttf
_verdanaiW5 Verdana-Italic " cp1255-encoding ReEncodeFont " <cp1255.enc <verdanai.ttf
_verdanaiW6 Verdana-Italic " cp1256-encoding ReEncodeFont " <cp1256.enc <verdanai.ttf
_verdanaz Verdana-BoldItalic " cp1252-encoding ReEncodeFont " <cp1252.enc <verdanaz.ttf
_verdanazVI Verdana-BoldItalic " viscii-encoding ReEncodeFont " <viscii.enc <verdanaz.ttf
_verdanazW1 Verdana-BoldItalic " cp1251-encoding ReEncodeFont " <cp1251.enc <verdanaz.ttf
_verdanazW3 Verdana-BoldItalic " cp1253-encoding ReEncodeFont " <cp1253.enc <verdanaz.ttf
_verdanazW5 Verdana-BoldItalic " cp1255-encoding ReEncodeFont " <cp1255.enc <verdanaz.ttf
_verdanazW6 Verdana-BoldItalic " cp1256-encoding ReEncodeFont " <cp1256.enc <verdanaz.ttf
_webdings Webdings " cp1252-encoding ReEncodeFont " <cp1252.enc <webdings.ttf
_wingding Wingdings-Regular " cp1252-encoding ReEncodeFont " <cp1252.enc <wingding.ttf

Dos System Files

drv.sys

```
REM drv.sys  
REM =====
```

```
REM This file must end with just the command "set DRV=", without CR and LF:  
set DRV=
```

pwd.sys

```
REM pwd.sys  
REM =====
```

```
REM This file must end with just the command "set PWD=", without CR and LF:  
set PWD=
```

TeX Exit Files

exit.tex

```
% exit.tex  
% =====
```

```
\message {Exiting ...}  
\end
```

q.tex

```
% q.tex  
% =====
```

```
\message {Exiting ...}  
\end
```

quit.tex

```
% quit.tex  
% =====
```

```
\message {Exiting ...}  
\end
```

x.tex

```
% x.tex  
% =====
```

```
\message {Exiting ...}  
\end
```

[GMS_ROOT]/shell

Shell scripts (Linux)

g_code

```
#!/bin/sh

# g_code
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_CODE=20060927

# Prologue: #####

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_code)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_code $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " g_code ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$?"
fi

# Chapters: #####

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_code

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=3
    export REPLY_SIZE=72
    export REPLY_ITEMS=1

# Remove: -----

elif [ $1 = -remove ]; then
    source "$GMS_SHELL/g_code" -resize called_by g_code -remove
    source "$GMS_SHELL/l_rain" -remove called_by g_code -remove # sic! =>l_code
    source "$GMS_SHELL/l_banner" -lower -folder called_by g_code -remove
# source "$GMS_SHELL/g_menu" -build called_by g_code -remove

# Build: -----

elif [ $1 = -cp_up_build -o $1 = -cp_lo_build ]; then
    BACKUP=$1
    source "$GMS_SHELL/g_select" -resize called_by g_code -build
    source "$GMS_SHELL/l_box" -t11_remove called_by g_code -build
    source "$GMS_SHELL/g_menu" -resize called_by g_code -build
    source "$GMS_SHELL/l_box" -remove called_by g_code -build
    source "$GMS_SHELL/g_code" -resize called_by g_code -build
    if [ $BACKUP = -cp_up_build ]; then
        source "$GMS_SHELL/l_code" -cp_up called_by g_code -build
        source "$GMS_SHELL/l_banner" -lower -cp_up called_by g_code -build
    else
        source "$GMS_SHELL/l_code" -cp_lo
        source "$GMS_SHELL/l_banner" -lower -cp_lo called_by g_code -build
    fi
    BACKUP=
    source "$GMS_SHELL/g_code" -remove called_by g_code -build
    if [ -f "$REPLY_TEMP" ]; then rm "$REPLY_TEMP"; fi
    # Build menu:
    export GMS_COLD=S
    export GMS_HOT=S
    source "$GMS_SHELL/g_menu" -build called_by g_code -build

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action g_code $1
fi
```

g_color

```
#!/bin/sh

# g_color
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_COLOR=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_color)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_color $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_color ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9) >> "Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_color

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=26
    export REPLY_SIZE=20
    export REPLY_ITEMS=8

# Build: -----

elif [ $1 = -build ]; then
    source "$GMS_SHELL/g_color" -resize called_by g_color -build
    source "$GMS_SHELL/l_box" -build called_by g_color -build
    source "$GMS_SHELL/l_banner" -lower -color called_by g_color -build
    source "$GMS_SHELL/l_color" -build called_by g_color -build
    export GMS_HOT=Q
    source "$GMS_SHELL/l_color" -update Q Q called_by g_color -build
    export REPLY_MODULE=g_color
    export REPLY_ACTION=-update
    export GMS_RECEIVE=1
    export GMS_HOT="Q"

# Remove: -----

elif [ $1 = -remove ]; then
    source "$GMS_SHELL/g_rain" -remove called_by g_color -remove
    source "$GMS_SHELL/g_color" -resize called_by g_color -remove
    source "$GMS_SHELL/l_box" -remove called_by g_color -remove
    source "$GMS_SHELL/g_menu" -resize called_by g_color -remove
    source "$GMS_SHELL/l_box" -remove called_by g_color -remove
    source "$GMS_SHELL/l_desk" -remove called_by g_color -remove
    export REPLY_MODULE=g_palet
    export REPLY_ACTION=-build
    export REPLY_OFFSET=0
    export REPLY_SIZE=0
    export REPLY_ITEMS=0

    "$GMS_REPLY" -random 2
    export GMS_RECEIVE=1
    export GMS_RETURN=1

# Update: -----

elif [ $1 = -update ]; then
# Cold:
    if [ "$3" = "" ]; then GMS_COLD=$GMS_HOT; else GMS_COLD=$3; fi
# Hot:
    if [ "$2" = "" ]; then
        source "$GMS_SHELL/l_banner" -no_hot g_color called_by g_color -update
    else
        GMS_HOT="Q"
        case "$2" in
            "T" | "t" | "1" ) GMS_HOT="T";;
            "H" | "h" | "2" ) GMS_HOT="H";;
            "P" | "p" | "3" ) GMS_HOT="P";;
            "B" | "b" | "4" ) GMS_HOT="B";;
            "S" | "s" | "5" ) GMS_HOT="S";;
            "D" | "d" | "6" ) GMS_HOT="D";;
            "L" | "l" | "7" ) GMS_HOT="L";;
            "Q" | "q" | "8" ) GMS_HOT="Q";;
            "R" | "r" | "" ) GMS_HOT="T";; # select first entry
        esac
    fi
# Handle old, update new:
    if [ "$GMS_HOT" = "$GMS_COLD" ]; then
        source "$GMS_SHELL/g_color" -handle $GMS_HOT $GMS_COLD
    else
# Update colors:
        if [ "$4" != "" ]; then
            case "$GMS_COLD" in
                "T" ) export GMS_TEXT=$4;;
                "H" ) export GMS_HOTKEY=$4;;
                "P" ) export GMS_PATTERN=$4;;
                "B" ) export GMS_BANNER=$4;;
                "S" ) export GMS_SHADE=$4;;
                "D" ) export GMS_DESKTOP=$4;;
                "L" ) export GMS_LETTER=$4;;
            esac
        fi
# Update color box:
        source "$GMS_SHELL/l_color" -update $GMS_HOT $GMS_COLD
    fi
    export GMS_RECEIVE=1

# Handle: -----

elif [ $1 = -handle ]; then

# Quit:
    if [ $2 = Q ]; then
        source "$GMS_SHELL/g_palet" -handle \
            $REPLY_DESKTOP $REPLY_PATTERN $REPLY_BANNER $REPLY_TEXT \
            $REPLY_SHADE $REPLY_HOTKEY $REPLY_LETTER
        source "$GMS_SHELL/g_launch" -build called_by g_color -handle
        source "$GMS_SHELL/g_color" -remove called_by g_color -handle
    fi

# Text:
    elif [ $2 = T ]; then
        source "$GMS_SHELL/l_color" -update - T called_by g_color -handle
        setterm -cursor on
        "$GMS_REPLY" -question 8 "$GMS_TEXT" TT 1
        setterm -cursor off

# Hotkey:
    elif [ $2 = H ]; then
```

```

source "$GMS_SHELL/l_color" -update - H called_by g_color -handle
setterm -cursor on
"$GMS_REPLY" -question 10 "$GMS_HOTKEY" HH 2
setterm -cursor off

# Pattern:
elif [ $2 = P ]; then
source "$GMS_SHELL/l_color" -update - P called_by g_color -handle
setterm -cursor on
"$GMS_REPLY" -question 12 "$GMS_PATTERN" PP 3
setterm -cursor off

# Banner:
elif [ $2 = B ]; then
source "$GMS_SHELL/l_color" -update - B called_by g_color -handle
setterm -cursor on
"$GMS_REPLY" -question 14 "$GMS_BANNER" BB 4
setterm -cursor off

# Shadow:
elif [ $2 = S ]; then
source "$GMS_SHELL/l_color" -update - S called_by g_color -handle
setterm -cursor on
"$GMS_REPLY" -question 16 "$GMS_SHADE" SS 5
setterm -cursor off

# Desktop:
elif [ $2 = D ]; then
source "$GMS_SHELL/l_color" -update - D called_by g_color -handle
setterm -cursor on
"$GMS_REPLY" -question 18 "$GMS_DESKTOP" DD 6
setterm -cursor off

# Character:
elif [ $2 = L ]; then
source "$GMS_SHELL/l_color" -update - L called_by g_color -handle
setterm -cursor on
"$GMS_REPLY" -question 20 "$GMS_LETTER" LL 7
setterm -cursor off

# Not found:
else
source "$GMS_SHELL/l_banner" -no_hotkey g_color $2

fi

# Not found: -----
else
source "$GMS_SHELL/l_banner" -no_action g_color $1
fi

```


g_file

```
#!/bin/sh

# g_file
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_FILE=20060927

# Prologue: #####

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_file)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_file $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_file ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: #####

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_file

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=3
    export REPLY_SIZE=44
    export REPLY_ITEMS=13

# Build: -----

elif [ $1 = -build ]; then
    export GMS_STATE=$2
    BACKUP=$3
    source "$GMS_SHELL/g_file" -resize called_by g_file -build
    source "$GMS_SHELL/l_box" -build called_by g_file -build
    source "$GMS_SHELL/l_file" -build "$BACKUP" called_by g_file -build
    export GMS_FOLDER=$PWD
    source "$GMS_SHELL/l_banner" -lower -folder called_by g_file -build
    export REPLY_LIST=0
    source "$GMS_SHELL/g_list" -handle called_by g_file -build
    source "$GMS_SHELL/l_list" -build called_by g_file -build
    export GMS_HOT=N
    if [ "$GMS_STATE" = -folder ]; then export GMS_HOT=H; fi
    source "$GMS_SHELL/l_file" -update $GMS_HOT $GMS_HOT called_by g_file -build
    export REPLY_MODULE=g_file
    export REPLY_ACTION=-update
    BACKUP=
    export GMS_RECEIVE=1

# Remove:-----

elif [ $1 = -remove ]; then
    source "$GMS_SHELL/g_file" -resize called_by g_file -remove
    source "$GMS_SHELL/l_box" -remove called_by g_file -remove
    if [ "$GMS_STATE" = -folder ]; then
        # Selected a folder to save the new file:
        source "$GMS_SHELL/g_save" -build called_by g_file -remove
    elif [ "$GMS_STATE" = -template ]; then
        # Selected a template for the new file:
        source "$GMS_SHELL/g_menu" -resize called_by g_file -remove
        source "$GMS_SHELL/l_menu" -update C Q called_by g_file -remove
        export REPLY_MODULE=g_menu
        export REPLY_ACTION=-update
        export GMS_RECEIVE=1
    else
        # Changed folder and opened markup file:
        source "$GMS_SHELL/g_menu" -resize called_by g_file -remove
        source "$GMS_SHELL/l_menu" -update O Q called_by g_file -remove
        export REPLY_MODULE=g_menu
        export REPLY_ACTION=-update
        export GMS_RECEIVE=1
    fi
fi

# Update: -----

elif [ $1 = -update ]; then
    # Cold:
    if [ "$3" = "" ]; then GMS_COLD=$GMS_HOT; else GMS_COLD=$3; fi
    # Hot:
    if [ "$2" = "" ]; then
        source "$GMS_SHELL/l_banner" -no_hot g_file called_by g_file -update
    else
        GMS_HOT="Q"
        case "$2" in
            "1" ) GMS_HOT="1";;
            "2" ) GMS_HOT="2";;
            "3" ) GMS_HOT="3";;
            "4" ) GMS_HOT="4";;
            "5" ) GMS_HOT="5";;
            "6" ) GMS_HOT="6";;
            "7" ) GMS_HOT="7";;
            "8" ) GMS_HOT="8";;
            "9" ) GMS_HOT="9";;
            "Q" | "q" | "10" ) GMS_HOT="Q";;
            "P" | "p" | "11" ) GMS_HOT="P";;
            "N" | "n" | "12" ) GMS_HOT="N";;
            "H" | "h" | "13" ) GMS_HOT="H";;
        esac
    fi
    # Handle old, update new:
    if [ "$GMS_HOT" = "$GMS_COLD" ]; then
        source "$GMS_SHELL/g_file" -handle $GMS_HOT
    else
        source "$GMS_SHELL/l_file" -update $GMS_HOT $GMS_COLD
    fi
    export GMS_RECEIVE=1
    export GMS_HOT="Q"

# Handle: -----

elif [ $1 = -handle ]; then

# Quit:
if [ $2 = Q ]; then
    source "$GMS_SHELL/g_file" -remove called_by g_file -handle Q

# Higher:
elif [ $2 = H ]; then
    cd ..
    export GMS_FOLDER=$PWD
    export GMS_FILE=
    export REPLY_LIST=0
    export GMS_RECEIVE=1
    source "$GMS_SHELL/g_list" -clear called_by g_file -handle H
```

```

source "$GMS_SHELL/l_banner" -upper -first called_by g_file -handle H
source "$GMS_SHELL/g_list" -handle called_by g_file -handle H
source "$GMS_SHELL/l_list" -build called_by g_file -handle H
source "$GMS_SHELL/l_banner" -lower -folder called_by g_file -handle H
source "$GMS_SHELL/l_file" -update H - called_by g_file -handle H

# Previous:
elif [ $2 = P ]; then
  if [ $REPLY_LIST = 0 -o $REPLY_LIST = 1 -o $REPLY_LIST = 2 \
    -o $REPLY_LIST = 3 -o $REPLY_LIST = 4 -o $REPLY_LIST = 5 \
    -o $REPLY_LIST = 6 -o $REPLY_LIST = 7 -o $REPLY_LIST = 8 ]; then
    # Keep list offset non-negative:
    export REPLY_LIST=0
  else
    # Diminish list offset by 8:
    export REPLY_LIST=${REPLY_LIST - 8}
  fi
  export GMS_RECEIVE=1
  source "$GMS_SHELL/g_list" -clear called_by g_file -handle P
  source "$GMS_SHELL/g_list" -handle called_by g_file -handle P
  source "$GMS_SHELL/l_list" -build called_by g_file -handle P
  source "$GMS_SHELL/l_file" -update P - called_by g_file -handle P

# Next:
elif [ $2 = N ]; then
  if [ "$REPLY9" != "" ]; then
    export REPLY_LIST=${REPLY_LIST + 8}
  fi
  export GMS_RECEIVE=1
  source "$GMS_SHELL/g_list" -clear called_by g_file -handle N
  source "$GMS_SHELL/g_list" -handle called_by g_file -handle N
  source "$GMS_SHELL/l_list" -build called_by g_file -handle N
  source "$GMS_SHELL/l_file" -update N - called_by g_file -handle N

# Selected file no. 1 - 9:
elif [ $2 = 1 ]; then
  export GMS_NEWFILE=$REPLY1
  source "$GMS_SHELL/g_file" -handle -entry called_by g_file -handle 1
elif [ $2 = 2 ]; then
  export GMS_NEWFILE=$REPLY2
  source "$GMS_SHELL/g_file" -handle -entry called_by g_file -handle 2
elif [ $2 = 3 ]; then
  export GMS_NEWFILE=$REPLY3
  source "$GMS_SHELL/g_file" -handle -entry called_by g_file -handle 3
elif [ $2 = 4 ]; then
  export GMS_NEWFILE=$REPLY4
  source "$GMS_SHELL/g_file" -handle -entry called_by g_file -handle 4
elif [ $2 = 5 ]; then
  export GMS_NEWFILE=$REPLY5
  source "$GMS_SHELL/g_file" -handle -entry called_by g_file -handle 5
elif [ $2 = 6 ]; then
  export GMS_NEWFILE=$REPLY6
  source "$GMS_SHELL/g_file" -handle -entry called_by g_file -handle 6
elif [ $2 = 7 ]; then
  export GMS_NEWFILE=$REPLY7
  source "$GMS_SHELL/g_file" -handle -entry called_by g_file -handle 7
elif [ $2 = 8 ]; then
  export GMS_NEWFILE=$REPLY8
  source "$GMS_SHELL/g_file" -handle -entry called_by g_file -handle 8
elif [ $2 = 9 ]; then
  export GMS_NEWFILE=$REPLY9
  source "$GMS_SHELL/g_file" -handle -entry called_by g_file -handle 9

# Handle file or folder:
elif [ $2 = -entry ]; then
  # Check if it is a folder:
  export GMS_FOLDER=$GMS_NEWFILE
  if [ -d "$GMS_FOLDER" ]; then
    cd "$GMS_FOLDER"
    export GMS_FILE=
    source "$GMS_SHELL/g_file" -handle -folder \
      called_by g_file -handle -entry
  else
    # Then it will be a file:
    export GMS_FOLDER=$PWD
    export GMS_FILE=$GMS_NEWFILE
    if [ "$GMS_STATE" = -open ]; then
      source "$GMS_SHELL/g_file" -handle -file \
        called_by g_file -handle -entry
    elif [ "$GMS_STATE" = -template ]; then
      source "$GMS_SHELL/g_file" -handle -template \
        called_by g_file -handle -entry
    fi
  fi

# Handle folder:
elif [ $2 = -folder ]; then
  export GMS_FOLDER=$PWD
  source "$GMS_SHELL/l_banner" -upper -first \
    called_by g_file -handle -folder
  source "$GMS_SHELL/l_banner" -lower -folder \
    called_by g_file -handle -folder
  export REPLY_LIST=0
  source "$GMS_SHELL/g_list" -clear called_by g_file -handle -folder
  source "$GMS_SHELL/g_list" -handle called_by g_file -handle -folder
  source "$GMS_SHELL/l_list" -build called_by g_file -handle -folder
  source "$GMS_SHELL/l_file" -update H 1 called_by g_file -handle -folder
  export REPLY_MODULE=g_file
  export REPLY_ACTION=-update
  export GMS_RECEIVE=1

# Handle file:
elif [ $2 = -file ]; then
  if [ "$GMS_FILE" = "" ]; then
    source "$GMS_SHELL/l_banner" -upper -first \
      called_by g_file -handle -file
  else
    export REPLY_SIZE=50
    export REPLY_OFFSET=0
    "$GMS_REPLY" -banner 1 "$GMS_FILE"
  fi
  source "$GMS_SHELL/g_file" -resize called_by g_file -handle -file
  source "$GMS_SHELL/l_box" -remove called_by g_file -handle -file
  source "$GMS_SHELL/g_menu" -resize called_by g_file -handle -file
  source "$GMS_SHELL/l_menu" -update O Q called_by g_file -handle -file
  source "$GMS_SHELL/g_vars" -write_memo called_by g_file -handle -file
  export REPLY_MODULE=g_menu
  export REPLY_ACTION=-update
  export GMS_RECEIVE=1

# Handle template:
elif [ $2 = -template ]; then
  export GMS_TEMPLATEFILE=$GMS_NEWFILE
  export GMS_TFOLDER=$PWD
  source "$GMS_SHELL/l_banner" -upper -template \
    called_by g_file -handle -template
  export GMS_FOLDER=$GMS_LASTDIR
  export GMS_LASTDIR=

# Select folder:
BACKUP_ANIMATE=$GMS_ANIMATE
GMS_ANIMATE="off"
export GMS_FILEBOX=" 2. Select a folder to save this new copy "
export GMS_FOLDER=$GMS_TEMPLATE
if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
source "$GMS_SHELL/g_file" -build -folder
export GMS_FILEBOX=
GMS_ANIMATE=$BACKUP_ANIMATE
BACKUP_ANIMATE=

# Not found:
else

```

```
source "$GMS_SHELL/l_banner" -no_hot g_file $2
fi
# Not found: .....
else
source "$GMS_SHELL/l_banner" -no_action g_file $1
fi
```

g_font

```
#!/bin/sh

# g_font
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

# GMSdateG_FONT=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_font)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_font $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_font ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Function: -----

# Get lowercase encoding;
function get_lc_enc {
    echo "$1" > enc.txt
    sed "y/ABCDEFGHIJKLMNQRSTUWXYZ/abcdefghijklmnopqrstuvwxyz;\
    s/^/export enc=/ < enc.txt > enc.sh
    if [ -f "enc.sh" ]; then source enc.sh; fi
    if [ -f "enc.sh" ]; then rm enc.sh; fi
    if [ -f "enc.txt" ]; then rm enc.txt; fi; }

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then
    source "$GMS_SHELL/l_banner" -no_action g_font

# Install fonts: -----

# TDS: ../texmf/fonts/fonttype/supplier/typeface
# where fonttype = afm, tfm, type1 (for pfa and pfb), ttf, vf.

# According to the TeX directory structure, afm and pfa / pfb as well as ttf
# files are stored in different trees. - Copies of them must be collected at
# a common place, the tfm directory. The ttf fonts need computing of afm fi-
# les first, before tfm and vf files can be created. After processing of the
# metrics, all vf files are moved to the vf directory, while the pfa/pfb and
# afm copies are deleted. The "core" font metrics get processed during this
# procedure, too.
# The tfm and vf metrics that have been processed algorithmically are *not*
# stored in supplier/typeface subfolders, but in the top level of the tfm or
# vf tree. At the beginning of the metric building process, the top levels
# of the tfm and vf trees are cleared, but files in subfolders remain.

# Handle typeface: -----

elif [ "$1" = "-handle_typeface" ]; then
    # $2 fonttype, $3 supplier, $4 typeface
    t="$GMS_FONTS/tfm"
    if [ "$4" != "_ignore_" -a "$4" != "_IGNORE_" \
        -a "$4" != "cm" -a "$4" != "Cm" -a "$4" != "CM" ]; then
        if [ -d "$GMS_FONTS/$2/$3/$4" ]; then
            cd "$GMS_FONTS/$2/$3/$4"
            if [ "$2" = "type1" ]; then
                for i in *.PFA; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.pfa; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.PFB; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.pfb; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
            elif [ "$2" = "ttf" ]; then
                for i in *.TTF; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.ttf; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
            elif [ "$2" = "afm" ]; then
                for i in *.AFM; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.afm; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
            fi
        fi
    fi; unset t

# Handle supplier: -----

elif [ "$1" = "-handle_supplier" ]; then
    # $2 fonttype, $3 supplier
    t="$GMS_FONTS/tfm"
    if [ "$3" != "_ignore_" -a "$3" != "_IGNORE_" ]; then
        if [ -d "$GMS_FONTS/$2/$3" ]; then
            cd "$GMS_FONTS/$2/$3"
            if [ "$2" = "type1" ]; then
                for i in *.PFA; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.pfa; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.PFB; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.pfb; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
            elif [ "$2" = "ttf" ]; then
                for i in *.TTF; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.ttf; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
            elif [ "$2" = "afm" ]; then
                for i in *.AFM; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.afm; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
            fi
            for i in *; do
                if [ -d "$i" ]; then
                    source "$GMS_SHELL/g_font" -handle_typeface $2 $3 $i
                    if [ -d "$GMS_FONTS/$2/$3" ]; then cd "$GMS_FONTS/$2/$3"; fi
                fi
            done
        fi
    fi; unset t

# Copy font files to tfm top level: -----

elif [ "$1" = "-copy_to_tfm" ]; then
    # $2 fonttype
    t="$GMS_FONTS/tfm"
    if [ "$2" != "_ignore_" -a "$2" != "_IGNORE_" ]; then
        if [ -d "$GMS_FONTS/$2" ]; then
            cd "$GMS_FONTS/$2"
            if [ "$2" = "type1" ]; then
                for i in *.PFA; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.pfa; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.PFB; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.pfb; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
            elif [ "$2" = "ttf" ]; then
                for i in *.TTF; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.ttf; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
            elif [ "$2" = "afm" ]; then
                for i in *.AFM; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
                for i in *.afm; do if [ -f "$i" ]; then cp "$i" "$t"; fi; done;
            fi
        fi
    fi
    for i in *; do
```

```

    if [ -d "$1" ]; then
        source "$GMS_SHELL/g_font" -handle_supplier $2 $i
        if [ -d "$GMS_FONTS/$2" ]; then cd "$GMS_FONTS/$2"; fi
    fi
done
fi
fi; unset t

# Build any font: -----
elif [ "$1" = "-build" ]; then
    # $2: glyph file name
    # $3: encoding
    # $4: core/eroc/embed/corefamily/embedfamily mark
    # $5: extension factor
    # $6: slanting factor
    # $7: new name (or 'none')
    # $8: new suffix (optional)
    enc="$3"
    if [ "$2" != ".*,$4" ]; then
        glyph="$2"
        # Get font face file base name from glyph file name: .....
        echo "$glyph" > font_tmp.txt
        sed "s/\\.*/;/s/^/export GMS_BASE=/ < font_tmp.txt > map_tmp.sh"
        if [ -f "map_tmp.sh" ]; then source map_tmp.sh; fi
        # Modify name: .....
        esab="$GMS_BASE"
        if [ "$7" != "" ]; then
            if [ "$7" = "none" ]; then esab="$esab$8"; else esab="$7"; fi
        fi
        # Change encoding for non-latin fonts: .....
        fo="$GMS_TEMP/fo.sh"
        if [ ! -f "$fo" ]; then
            fj="$GMS_SETTING/encoding.cfg"
            fp="$GMS_TEMP/fp.sh"
            fq="$GMS_TEMP/fq.sh"
            tr -d '\r' < "$fj" > "$fp"
            sed "s/\\#.*//;/^ *$/d;s/^ *//;/s/;/; fi/;" < "$fp" > "$fq"
            sed \
"s/:/\/" ]; then export enc=;/s/=/ *//;s/^/if [ \|$GMS_BASE\| = \/" \
< "$fq" > "$fo"
            if [ -f "$fp" ]; then rm "$fp"; fi
            if [ -f "$fq" ]; then rm "$fq"; fi
            unset fp; unset fq; unset fj
        fi
        oldenc=$enc
        if [ -f "$fo" ]; then source "$fo"; fi
        if [ "$oldenc" != "$enc" -a "$GMS_DEBUG" = "Z" ]; then
            echo " New encoding: $enc" >> "$Z";
        fi
    fi
unset oldenc; unset fo
get_lc_enc $enc
# Run ttf2afm: .....
arg1="$GMS_BINARIES/ttf2afm"
arg2="$GMS_ROOT/data/enc/$enc.enc"
arg3="$GMS_BASE.afm"
if [ -f "$GMS_BASE.ttf" -a "$glyph" = "$GMS_BASE.ttf" ]; then
arg4="$GMS_BASE.ttf"
    echo "\|$arg1\| -e \|$arg2\| -o \|$arg3\| \|$arg4\|" >> fm.log
    "$arg1" "-e" "$arg2" "-o" "$arg3" "$arg4"
elif [ -f "$GMS_BASE.TTF" -a "$glyph" = "$GMS_BASE.TTF" ]; then
arg4="$GMS_BASE.TTF"
    echo "\|$arg1\| -e \|$arg2\| -o \|$arg3\| \|$arg4\|" >> fm.log
    "$arg1" "-e" "$arg2" "-o" "$arg3" "$arg4"
fi
# Define afm2tfm arguments: .....
if [ -f "$GMS_BASE.afm" -o -f "$GMS_BASE.AFM" ]; then
    # A redirection symbol (" < CP1252.enc ") is included in afm2tfm's
    # output, so afm2tfm must be put into its argument variable:
    arg1="$GMS_BINARIES/afm2tfm"
    arg2="./$GMS_BASE.afm"

```

```

arg3="$GMS_ROOT/data/enc/$enc.enc"
arg4="$esab.vpl"
arg5=
arg6=
# Extend typeface: .....
if [ "$5" != "" -a "$5" != 1 -a "$5" != 1.0 -a "$5" != 1.00 -a \
"$5" != 1.000 -a "$5" != 1.0000 -a "$5" != 1.00000 -a \
"$5" != hide ]; then
    arg5="-e $5"
fi
# Slant typeface: .....
if [ "$6" != "" -a "$6" != 0 -a "$6" != 0.0 -a "$6" != 0.00 -a \
"$6" != 0.000 -a "$6" != 0.0000 -a "$6" != 0.00000 ]; then
    arg6="-s $6"
fi
# Display font file base name: .....
export REPLY_OFFSET=0; export REPLY_SIZE=80
"$GMS_BINARIES/reply" -banner 19 ""
arg=""
    echo "$arg$arg"; echo "$arg$arg"
unset arg
export REPLY_OFFSET=43; export REPLY_SIZE=37
"$GMS_BINARIES/reply" -banner 17 "(enc) $GMS_BASE"
echo
# Run afm2tfm, build map item, eventually add glyph: .....
echo \
"\|$arg1\| \|$arg2\| -T \|$arg3\| -v \|$arg4\| $arg5 $arg6 _$esab.tfm" \
>> fm.log
"$arg1" "$arg2" "-T" "$arg3" "-v" "$arg4" $arg5 $arg6 _$esab.tfm" \
> item_tmp.map
    if [ "$4" != "core" -a "$4" != "eroc" ]; then
        # sed "s/\/ \|<glyph/ < item_tmp.map > meti_tmp.map"
        sed "s/\/ \|<glyph/ < item_tmp.map > meti_tmp.map"
        cat meti_tmp.map >> "$GMS_SETTING/unsort.map"
    else
        sed "s/\/ \|<enc/ < item_tmp.map > meti_tmp.map"
        # cat item_tmp.map >> "$GMS_SETTING/unsort.map"
        cat meti_tmp.map >> "$GMS_SETTING/unsort.map"
    fi
export arguments=
# Run vptovf, move processed core afm fonts to _trans_: .....
echo
arg1="$GMS_BINARIES/vptovf"
echo "\|$arg1\| \|$esab.vpl\| \|$esab.vf\| \|$esab.tfm\|" >> fm.log
"$arg1" "$esab.vpl" "$esab.vf" "$esab.tfm"
if [ "$4" = "eroc" ]; then
    if [ -f "$GMS_BASE.afm" -a -d "../_trans_" ]; then
        cp "$GMS_BASE.afm" "../_trans_"; fi
    if [ -f "$GMS_BASE.AFM" -a -d "../_trans_" ]; then
        cp "$GMS_BASE.AFM" "../_trans_"; fi
    if [ -f "$GMS_BASE.afm" ]; then rm "$GMS_BASE.afm"; fi
    if [ -f "$GMS_BASE.AFM" ]; then rm "$GMS_BASE.AFM"; fi
fi
fi; echo >> fm.log
export GMS_BASE=
unset esab
unset glyph
fi; unset enc
# Clean display lines: .....
export REPLY_OFFSET=0; export REPLY_SIZE=66 ###
"$GMS_BINARIES/reply" -banner 20
"$GMS_BINARIES/reply" -banner 21
"$GMS_BINARIES/reply" -banner 22
arg="#####"
echo; echo " $arg$arg"
arg=

# Build specifically encoded fonts: -----
elif [ "$1" = "-build_encode" ]; then

```

```

# $2: encoding
# be="$GMS_SHELL/g_font" -build
e=$2
if [ "$2" != "" ]; then
  for i in $e*.afm; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $e eroc; fi; done
  for i in $e*.AFM; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $e eroc; fi; done
  for i in $e*.pfa; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $e embed; fi; done
  for i in $e*.PFA; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $e embed; fi; done
  for i in $e*.pfb; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $e embed; fi; done
  for i in $e*.PFB; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $e embed; fi; done
  for i in $e*.ttf; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $e embed; fi; done
  for i in $e*.TTF; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $e embed; fi; done
fi; unset e

```

```

# Build transformed fonts: -----

```

```

elif [ "$1" = "-build_trans" ]; then
# $2: base name
# $3: encoding
# $4: core/embed mark
# $5: extension factor
# $6: slanting factor
# $7: new name (or 'none')
# $8: new suffix (optional)
b=$2; e=$3; m=$4
f="$5 $6 $7 $8"; myttf="1.0 0.0 $7 $8"
# Display font file base name: -----
export REPLY_OFFSET=0; export REPLY_SIZE=80
"$GMS_BINARIES/reply" -banner 19 ""
arg=""
echo "$arg$arg"; echo "$arg$arg"
unset arg
export REPLY_OFFSET=43; export REPLY_SIZE=37
get_lc_enc $3
"$GMS_BINARIES/reply" -banner 17 "($enc) $2"
#-----
if [ "$m" != "" ]; then
  if [ "$m" = "core" ]; then
    # This is only possible if a typel glyph file is present:
    if [ -f $b.afm ]; then
      source "$GMS_SHELL/g_font" -build $b.pfa $e $m $f; fi
    fi
  if [ "$m" = "corefamily" ]; then
    for i in $b*.afm; do if [ -f "$i" ]; then
      source "$GMS_SHELL/g_font" -build $i $e embed $f; fi; done
    fi
  if [ "$m" = "embed" ]; then
    if [ -f $b.pfa ]; then
      source "$GMS_SHELL/g_font" -build $b.pfa $e $m $f; fi
    if [ -f $b.pfb ]; then
      source "$GMS_SHELL/g_font" -build $b.pfb $e $m $f; fi
    if [ -f $b.ttf ]; then
      source "$GMS_SHELL/g_font" -build $b.ttf $e $m $myttf; fi
    fi
  if [ "$m" = "embedfamily" ]; then
    for i in $b*.pfa; do if [ -f "$i" ]; then
      source "$GMS_SHELL/g_font" -build $i $e embed $f; fi; done
    for i in $b*.pfb; do if [ -f "$i" ]; then
      source "$GMS_SHELL/g_font" -build $i $e embed $f; fi; done
    for i in $b*.ttf; do if [ -f "$i" ]; then
      source "$GMS_SHELL/g_font" -build $i $e embed $f; fi; done
    fi
  fi

```

```

fi; unset b; unset e; unset f; unset m; unset myttf

```

```

# Build all fonts: -----

```

```

elif [ "$1" = "-build_all" ]; then
export BEFORE=$(date +%s)
export REPLY_BANNER=15; export REPLY_TEXT=0
if [ "$2" != "-quiet" ]; then
  setterm -clear -cursor off
  echo "$GMS_FILE"; cat "$GMS_SETTING/desktop.scn"
  export REPLY_OFFSET=0; export REPLY_SIZE=82
  "$GMS_BINARIES/reply" -banner 1 "font.map"
  "$GMS_BINARIES/reply" -banner 24 "GMS: Running font metric processor ..."
  "$GMS_BINARIES/reply" -banner 14
  "$GMS_BINARIES/reply" -banner 15 "Please remember to re-initialize the"
  export REPLY_OFFSET=38; export REPLY_SIZE=44
  "$GMS_BINARIES/reply" -banner 15 "format file after re-writing the font map"
fi
export REPLY_OFFSET=0; export REPLY_SIZE=82
"$GMS_BINARIES/reply" -banner 22; "$GMS_BINARIES/reply" -banner 16
"$GMS_BINARIES/reply" -banner 21; "$GMS_BINARIES/reply" -banner 17
"$GMS_BINARIES/reply" -banner 20; "$GMS_BINARIES/reply" -banner 18
"$GMS_BINARIES/reply" -banner 19
# Delete old tfm and vf; get afm, ttf, pfa/pfb: -----
export REPLY_OFFSET=0; export REPLY_SIZE=32
echo "----- Collecting fonts:" >> "$2"
"$GMS_BINARIES/reply" -banner 17 "Collecting: Font files ..."
echo; echo
echo " These processes may run faster if there is no file"
echo " manager task to be updated in the background."
if [ -d "$GMS_FONTS/tfm" ]; then cd "$GMS_FONTS/tfm"; fi
for i in *.tfm *.TFM; do if [ -f "$i" ]; then rm "$i"; fi; done
if [ -d "$GMS_FONTS/vf" ]; then cd "$GMS_FONTS/vf"; fi
for i in *.vf *.VF; do if [ -f "$i" ]; then rm "$i"; fi; done
export REPLY_OFFSET=14; export REPLY_SIZE=66
"$GMS_BINARIES/reply" -banner 17 "*.ttf - True Type Fonts ..."
source "$GMS_SHELL/g_font" -copy_to_tfm ttf
"$GMS_BINARIES/reply" -banner 17 "*.pfa* - Postscript Fonts ... Ascii/Binary"
source "$GMS_SHELL/g_font" -copy_to_tfm typel
"$GMS_BINARIES/reply" -banner 17 "*.afm - Adobe Font Metrics ..."
source "$GMS_SHELL/g_font" -copy_to_tfm afm
# Get new tfm and vf: -----
export REPLY_OFFSET=0; export REPLY_SIZE=14
"$GMS_BINARIES/reply" -banner 17 Computing;
export REPLY_OFFSET=14; export REPLY_SIZE=66
"$GMS_BINARIES/reply" -banner 17 "*.tfm - TeX Font Metrics ..."
if [ -f "$GMS_SETTING/font.map" ]; then
  rm "$GMS_SETTING/font.map"; fi
if [ -f "$GMS_SETTING/unsort.map" ]; then
  rm "$GMS_SETTING/unsort.map"; fi
if [ -d "$GMS_FONTS" ]; then cd "$GMS_FONTS"; fi
if [ -d "./_trans_" ]; then rm "./_trans_" -r; fi
if [ ! -d "./_trans_" ]; then mkdir "./_trans_"; fi
# Process font files by encoding: -----
"$GMS_BINARIES/reply" -banner 19 ###
"$GMS_BINARIES/reply" -banner 20 ###
if [ "$GMS_DEBUG" = "Z" ]; then
  echo "----- Process font files by encoding:" >> "$2"
fi
if [ -d "$GMS_ROOT/data/enc" ]; then cd "$GMS_ROOT/data/enc"; fi
for i in *.enc *.ENC; do
  base=$(basename $i .enc)
  if [ -d "$GMS_FONTS/tfm" ]; then cd "$GMS_FONTS/tfm"; fi
  source "$GMS_SHELL/g_font" -build_encode $base
done
unset base
# Process font files by type: -----
if [ "$GMS_DEBUG" = "Z" ]; then
  echo "----- Process font files by type:" >> "$2"
fi
fi

```

```

if [ -d "$GMS_FONTS/tfm" ]; then cd "$GMS_FONTS/tfm"; fi
c=$GMS_CODEPAGE
if [ "$GMS_DEBUG" = "Z" ]; then echo " ..... *.afm:" >> "$Z"; fi
for i in pcr*.afm; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $c eroc; fi; done
for i in phv*.afm; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $c eroc; fi; done
for i in ptm*.afm; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $c eroc; fi; done
if [ "$GMS_DEBUG" = "Z" ]; then echo " ..... *.pfa:" >> "$Z"; fi
for i in *.pfa; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $c embed; fi; done
for i in *.PFA; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $c embed; fi; done
for i in *.pfb; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $c embed; fi; done
for i in *.PFB; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $c embed; fi; done
if [ "$GMS_DEBUG" = "Z" ]; then echo " ..... *.ttf:" >> "$Z"; fi

for i in *.ttf; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $c embed; fi; done

for i in *.TTF; do if [ -f "$i" ]; then
    source "$GMS_SHELL/g_font" -build $i $c embed; fi; done
unset c
# Process transformed files: .....
if [ "$GMS_DEBUG" = "Z" ]; then
    echo " ..... Process transformed fonts files:" >> "$Z"
fi
if [ -d "$GMS_FONTS/_trans_" ]; then cd "$GMS_FONTS/_trans_"; fi
for i in *.afm *.AFM; do if [ -f "$i" ]; then cp $i ../tfm; fi; done
for i in *.afm *.AFM; do if [ -f "$i" ]; then rm $i; fi; done
if [ -d "$GMS_FONTS" ]; then cd "$GMS_FONTS"; fi
if [ -d "$GMS_FONTS/_trans_" ]; then rm "$GMS_FONTS/_trans_" -r; fi
if [ -d "$GMS_SETTING" ]; then cd "$GMS_SETTING"; fi
if [ -f font.cfg ]; then
    tr -d '\r' < font.cfg > font.gfc
    sed "s/%GMS_CODEPAGE%/\$GMS_CODEPAGE/" < font.gfc > tnof.gfc
    sed "s/^\.#.*$/;/^\^ *$/d" < tnof.gfc > snart.gfc
# Fixme:
    sed "s/^\source \"\$GMS_SHELL/g_font\" -build_trans/ \"
    < snart.gfc > trans.sh
    if [ -f font.gfc ]; then rm font.gfc; fi
    if [ -f tnof.gfc ]; then rm tnof.gfc; fi
    if [ -f snart.gfc ]; then rm snart.gfc; fi
    if [ -f trans.sh ]; then cp trans.sh "$GMS_FONTS/tfm"; fi
    if [ -f trans.sh ]; then rm trans.sh; fi
    if [ -d "$GMS_FONTS/tfm" ]; then cd "$GMS_FONTS/tfm"; fi
    if [ -f trans.sh ]; then source trans.sh; fi
    if [ -f trans.sh ]; then rm trans.sh; fi
fi
if [ -d "$GMS_TEMP" ]; then cd "$GMS_TEMP"; fi
if [ -f fo.sh ]; then rm fo.sh; fi
# Install virtual fonts (and clean up the tfm folder): .....
export REPLY_OFFSET=0; export REPLY_SIZE=15
"$GMS_BINARIES/reply" -banner 17 Installing:
export REPLY_OFFSET=14; export REPLY_SIZE=68
"$GMS_BINARIES/reply" -banner 17 " *.v* - Virtual Fonts ... $GMS_CODEPAGE"
arg="
"
echo "$arg$arg"; echo "$arg$arg"; echo "$arg$arg"; echo "$arg$arg"
unset arg
if [ -d "$GMS_FONTS/tfm" ]; then cd "$GMS_FONTS/tfm"; fi
for i in *.afm *.AFM; do if [ -f "$i" ]; then rm $i; fi; done
for i in *.pfa *.PFA; do if [ -f "$i" ]; then rm $i; fi; done

for i in *.pfb *.PFB; do if [ -f "$i" ]; then rm $i; fi; done
for i in *.ttf *.TTF; do if [ -f "$i" ]; then rm $i; fi; done
for i in *.vpl *.VPL; do if [ -f "$i" ]; then rm $i; fi; done
for i in *.vf *.VF; do if [ -f "$i" ]; then
    mv $i "$GMS_FONTS/vf"; fi; done
if [ -f font_tmp.txt ]; then rm font_tmp.txt; fi
if [ -f map_tmp.sh ]; then rm map_tmp.sh; fi
if [ -f item_tmp.map ]; then rm item_tmp.map; fi
if [ -f meti_tmp.map ]; then rm meti_tmp.map; fi
if [ -f short.txt ]; then rm short.txt; fi
if [ -f short.sh ]; then rm short.sh; fi
# Build font map (and log): .....
slashline="/////////////////////////////////////"
if [ -d "$GMS_SETTING" ]; then cd "$GMS_SETTING"; fi
if [ -f encoding.sh ]; then rm encoding.sh; fi
echo "% font.log - Warnings, errors, missing glyphs" > font.log
echo "% $slashline$slashline" >> font.log
arguments="$GMS_FONTS/[font-type]/[supplier]/[font-family]"
echo "% font.map - $arguments" > head.map
echo "% $slashline$slashline$slashline$slashline$slashline$slashline" >> head.map
unset arguments
echo >> head.map
echo "% Generated by Gerolf Markup Shredder (www.Gerolf.org)" \
>> head.map
echo -n "% on " >> head.map
date >> head.map
echo >> head.map
echo "% _[1] Base name of font files" >> head.map
echo "% [2] PostScript font face name" >> head.map
echo "% \" [3] ReEncodeFont \" PostScript encoding name" >> head.map
echo "% % <[4] Encoding file [*.enc]" >> head.map
echo "% % <[5] Glyph file [*.pfa, *.ttf]" >> head.map
echo >> head.map
if [ -f unsort.map ]; then sort < unsort.map > sort.map; fi
if [ -f head.map ]; then
    if [ -f sort.map ]; then
        cat head.map sort.map > font.map
    fi
fi
if [ -f sort.map ]; then rm sort.map; fi
if [ -f head.map ]; then rm head.map; fi
if [ -f unsort.map ]; then rm unsort.map; fi
echo >> font.map
echo "% Warnings, errors, missing glyph list: see font.log" \
>> font.map
get_runtime
echo "% GMS run time: $GMS_RUNTIME" >> font.map
echo >> font.map
echo "% $slashline$slashline$slashline$slashline$slashline$slashline" >> font.map
echo "% $GMS_FOLDER" >> font.map
if [ -d "$GMS_FONTS/tfm" ]; then cd "$GMS_FONTS/tfm"; fi
if [ -f fm.log ]; then mv fm.log "$GMS_SETTING/fontmap.log"; fi
export slashline=
export REPLY_OFFSET=0; export REPLY_SIZE=14
"$GMS_BINARIES/reply" -banner 17 Writing:
export REPLY_OFFSET=14; export REPLY_SIZE=68
"$GMS_BINARIES/reply" -banner 17 " $GMS_SETTING/font.map"
if [ -d "$GMS_SETTING" ]; then cd "$GMS_SETTING"; fi
setterm -cursor on
# Not found: .....
else
    source "$GMS_SHELL/l_banner" -no_action g_font $1
fi

```

g_good

```
#!/bin/sh

# g_good
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_GOOD=20060927

# Prologue: # =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_good)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_good $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " g_good ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_good

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=3
    export REPLY_SIZE=44
    export REPLY_ITEMS=1

# Build: -----

elif [ $1 = -build ]; then
    source "$GMS_SHELL/l_banner" -lower -first called_by g_good -build
    source "$GMS_SHELL/l_banner" -upper -last called_by g_good -build
    source "$GMS_SHELL/g_good" -resize called_by g_good -build
    source "$GMS_SHELL/l_box" -build called_by g_good -build
    source "$GMS_SHELL/l_good" -build called_by g_good -build
    export GMS_HOT=Q
    source "$GMS_SHELL/l_good" -update Q Q
    export REPLY_MODULE=g_good
    export REPLY_ACTION=-remove
    export GMS_RECEIVE=1

# Remove: -----

elif [ $1 = -remove ]; then
    source "$GMS_SHELL/g_good" -resize called_by g_good -remove
    source "$GMS_SHELL/l_good" -update - Q
    source "$GMS_SHELL/g_menu" -resize called_by g_good -remove
    source "$GMS_SHELL/g_menu" -remove called_by g_good -remove
    source "$GMS_SHELL/g_good" -resize called_by g_good -remove
    source "$GMS_SHELL/l_box" -remove called_by g_good -remove
    source "$GMS_SHELL/l_desk" -remove called_by g_good -remove
    setterm -clear
    echo -n " Please support the author of Markup Shredder  -"
    echo "  w w w . G e r o l f . o r g"
    cat "$GMS_SETTING/desktop.scn"
    source gms -q called_by g_good -remove
    export GMS_TEXTMODE=
    export GMS_BREAK=1

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action g_good $1
fi
```


g_launch

```
#!/bin/sh

# g_launch
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_LAUNCH=20060927

# Prologue: =====

line="=====
dash="-----"

# Not running: -----

if [ "$GMS_ROOT" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_launch).
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_launch $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " g_launch ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_launch

# Build 'gerolf' launcher script: -----

# Fixme: check "$HOME" not equal to "/root" ];

elif [ $1 = -build ]; then
    if [ "$HOME" = "/root" ]; then l="gerolf"
    else l="$HOME/gms_user"; fi
    cd "$GMS_SETTING"
    echo "#!/bin/sh" > $1
    echo >> $1
    if [ "$HOME" = "/root" ]; then
        echo "# gerolf" >> $1
        echo "# =====" >> $1
        echo >> $1
        echo "# Launcher script for Gerolf Markup Shredder (Unix)." >> $1
    else
        echo "# gms_user" >> $1
        echo "# =====" >> $1
        echo >> $1
        echo "# User setting script for Gerolf Markup Shredder (Unix)." >> $1
    fi
    echo "# Web: www.Gerolf.org, eMail: MarkupShredder@Gerolf.org" >> $1
    echo -n "# Generated on" >> $1
    date >> $1
    echo >> $1
    echo "# ==$line$line" >> $1
    echo >> $1
    echo "# 0) Start shell: $dash-----" >> $1
    echo >> $1
    echo " if [ \"\$1\" != \"-passive\" ]; then" >> $1
    echo "     if [ \"\$1\" = \"-shell\" ]; then" >> $1
    echo "         shift" >> $1
    echo "     else" >> $1
    echo "         xterm -fg black -bg white -e \"\$0\" -shell \"
        \"\$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9" >> $1
    echo "     fi" >> $1
    echo >> $1
    echo "# 1) General settings: $dash-----" >> $1
    echo >> $1
    if [ "$HOME" = "/root" ]; then
        echo "# Setting directory:" >> $1
        echo "     export GMS_SETTING=\"$GMS_SETTING\"" >> $1
        echo "     if [ \"\$GMS_MODE\" != \"quiet\" ]; then" >> $1
        echo "         setterm -clear -reset; fi" >> $1 ###
        echo "     if [ \"\$1\" != \"-passive\" ]; then" >> $1
        echo "         cat \"\$GMS_SETTING/launch_1.scn\"" >> $1
        echo "     fi" >> $1
        echo >> $1
        echo "# Main directories (can be web server subfolders):" >> $1
        echo "     export GMS_ROOT=\"$GMS_ROOT\"" >> $1
        echo "     export GMS_BODY=\"$GMS_BODY\"" >> $1
        echo >> $1
    fi
    echo "# Debugging mode (0, 1, 2, 3, ... or X, Y, Z)" >> $1
    echo "     export GMS_DEBUG=$GMS_DEBUG" >> $1
    echo >> $1
    echo "# Rewrite gmsdebug log file (do not change):" >> $1
    echo "     export Z=\"$GMS_SETTING/gmsdebug.log\"" >> $1
    echo "     if [ \"\$4\" != \"gmssetup\" ]; then" >> $1
    echo "         if [ \"\$GMS_MODE\" != \"quiet\" ]; then" >> $1
    echo "             if [ -f \"\$Z\" ]; then rm \"\$Z\"" >> $1
    echo "             gmshead=\"$GMS_SETTING/gmshead.txt" >> $1
    echo "             if [ -f \"\$gmshead\" ]; then mv \"\$gmshead\" \"\$Z\"; fi" >> $1
    echo "             unset gmshead" >> $1
    echo -n "         echo \" gerolf (\$1) (\$2) (\$3) (\$4) (\$5)" >> $1
    echo "         (\$6) (\$7) (\$8) (\$9)\" >> \"\$Z\"" >> $1
    echo "         fi" >> $1
    echo "         fi" >> $1
    echo "         fi" >> $1
    echo >> $1
    echo "# Main codepage name:" >> $1
    echo "     export GMS_CODEPAGE=$GMS_CODEPAGE" >> $1
    echo >> $1
    echo "# 2) Programs in search path (binary names): $dash-----" >> $1
    echo >> $1
    echo "# Plain text viewer and editor:" >> $1
    echo "     export GMS_VIEWER=\"$GMS_VIEWER\"" >> $1
    echo "     export GMS_EDITOR=\"$GMS_EDITOR\"" >> $1
    echo "# HTML browser, syntax checker, typesetting engine:" >> $1
    echo "     export GMS_BROWSER=\"$GMS_BROWSER\"" >> $1
    echo "     export GMS_ANALYST=\"$GMS_ANALYST\"" >> $1
    echo "     export GMS_TSETTER=\"$GMS_TSETTER\"" >> $1
    echo "# PDF reader:" >> $1
    echo "     export GMS_READER=\"$GMS_READER\"" >> $1
    echo >> $1
    echo "# 3) Textmode interface variables: $dash-----" >> $1
    echo >> $1
    echo "# Menu animation ('on' or 'off'):" >> $1
    echo "     export GMS_ANIMATE=$GMS_ANIMATE" >> $1
    echo "# Colors and pattern ('R' = 'Random'):" >> $1
    echo "# Foreground colors (0 to 15):" >> $1
    echo "     export GMS_TEXT=$GMS_TEXT" >> $1
    echo "     export GMS_HOTKEY=$GMS_HOTKEY" >> $1
    echo "     export GMS_PATTERN=$GMS_PATTERN" >> $1
    echo "# Background colors (0 to 7):" >> $1
    echo "     export GMS_BANNER=$GMS_BANNER" >> $1
```

```

echo "      export GMS_SHADE=$GMS_SHADE" >> $1
echo "      export GMS_DESKTOP=$GMS_DESKTOP" >> $1
echo " # Fill character (ASCII code 32 to 126)" >> $1
echo "      export GMS_LETTER=$GMS_LETTER" >> $1
echo >> $1
echo " # 4) Browser interface variables: $dash-----" >> $1
echo >> $1
echo " # Maximal size and number of files:" >> $1
echo "      export GMS_MAXSIZE=$GMS_MAXSIZE" >> $1
echo "      export GMS_MAXFILES=$GMS_MAXFILES" >> $1
echo " # Width of text area (join/split):" >> $1
echo "      export GMS_WIDTH_ONE=$GMS_WIDTH_ONE" >> $1
echo "      export GMS_WIDTH_TWO=$GMS_WIDTH_TWO" >> $1
echo " # Height of text area (join/split):" >> $1
echo "      export GMS_HEIGHT_ONE=$GMS_HEIGHT_ONE" >> $1
echo "      export GMS_HEIGHT_TWO=$GMS_HEIGHT_TWO" >> $1
echo " # Font size, in point (join/split):" >> $1
echo "      export GMS_SIZE_ONE=$GMS_SIZE_ONE" >> $1
echo "      export GMS_SIZE_TWO=$GMS_SIZE_TWO" >> $1
echo " # Linking to internal or external target:" >> $1
echo "      export GMS_LINKS=$GMS_LINKS" >> $1
echo >> $1
if [ "$HOME" = "/root" ]; then
echo " # 5) Start program: $dash-----">> $1
echo >> $1
echo " # Do not change the rest of this file:" >> $1
echo >> $1
echo " # Set version number and date:" >> $1
echo "      export GMS_VERSION=$GMS_VERSION" >> $1
echo "      export GMS_DATE=$GMS_DATE" >> $1
echo " # Get folder structure:" >> $1

```

```

echo "      source \"\$GMS_SETTING/folder\" called_by gerolf">> $1
echo "      if [ \"\$GMS_MODE\" != \"quiet\" ]; then" >> $1
echo "          setterm -clear -reset; fi" >> $1 ###
echo "      if [ \"\$1\" != \"-passive\" ]; then" >> $1
echo "          cat \"\$GMS_SETTING/launch_2.scn\" >> $1
echo "          fi" >> $1
echo " # Get work file:" >> $1
echo "      if [ \"\$GMS_MODE\" != \"quiet\" ]; then" >> $1
arg="called_by gerolf \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9"
echo "          source \"\$GMS_SHELL/g_vars\" -ini_file\" >> $1
echo "          $arg" >> $1
unset arg
echo "      fi" >> $1
echo "      if [ \"\$GMS_MODE\" != \"quiet\" ]; then" >> $1
echo "          setterm -clear -reset; fi" >> $1 ###
echo " # Launch textmode interface:" >> $1
echo "      if [ \"\$1\" != \"-passive\" ]; then" >> $1
echo "          cat \"\$GMS_SETTING/launch_3.scn\" >> $1
echo "          source \"\$GMS_SHELL/shredder\" called_by gerolf" >> $1
echo "          fi" >> $1
fi
chmod 755 $1
unset l

# Not found: -----
else
source "$GMS_SHELL/l_banner" -no_action g_launch $1
fi

```

g_list

```
#!/bin/sh

# g_list
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_LIST=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_list)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_list $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_list ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Function: -----

function g_list_handle_names {
    if [ $INDEX -gt $REPLY_LIST -a $INDEX -lt ${REPLY_LIST+10} ]; then
        if [ $INDEX = ${REPLY_LIST+1} ]; then
            export REPLY1="$f"; export GMS1="$str"
        elif [ $INDEX = ${REPLY_LIST+2} ]; then
            export REPLY2="$f"; export GMS2="$str"
        elif [ $INDEX = ${REPLY_LIST+3} ]; then
            export REPLY3="$f"; export GMS3="$str"
        elif [ $INDEX = ${REPLY_LIST+4} ]; then
            export REPLY4="$f"; export GMS4="$str"
        elif [ $INDEX = ${REPLY_LIST+5} ]; then
            export REPLY5="$f"; export GMS5="$str"
        elif [ $INDEX = ${REPLY_LIST+6} ]; then
            export REPLY6="$f"; export GMS6="$str"
        elif [ $INDEX = ${REPLY_LIST+7} ]; then
            export REPLY7="$f"; export GMS7="$str"
        elif [ $INDEX = ${REPLY_LIST+8} ]; then
            export REPLY8="$f"; export GMS8="$str"
        elif [ $INDEX = ${REPLY_LIST+9} ]; then
            export REPLY9="$f"; export GMS9="$str"; break
        fi
    fi; }

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_list

# Clear: -----

elif [ $1 = -clear ]; then
    # File names:
    export REPLY1=
    export REPLY2=
    export REPLY3=
    export REPLY4=
    export REPLY5=
    export REPLY6=
    export REPLY7=
    export REPLY8=
    export REPLY9=
    # Folder marks [//]:
    export GMS1=
    export GMS2=
    export GMS3=
    export GMS4=
    export GMS5=
    export GMS6=
    export GMS7=
    export GMS8=
    export GMS9=

# Handle: -----

elif [ $1 = -handle ]; then
    INDEX=0
    for f in *; do # Folder mark:
        if [ -d "$f" ]; then
            str="//"; INDEX=$((INDEX+1))
            g_list_handle_names
        fi
    done
    for f in *; do
        if [ -f "$f" ]; then
            str=" "; INDEX=$((INDEX+1))
            g_list_handle_names
        fi
    done
    INDEX=
    str=

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action g_list $1
fi
```

g_menu

```

#!/bin/sh

# g_menu
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_MENU=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_menu)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_menu $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_menu ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_menu

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=52
    export REPLY_SIZE=23
    export REPLY_ITEMS=13

# Build: -----

elif [ $1 = -build ]; then
    if [ "$GMS_FILE" = "" ]; then
        source "$GMS_SHELL/l_banner" -upper -first g_menu -build
    else
        source "$GMS_SHELL/l_banner" -upper -file g_menu -build
    fi
    source "$GMS_SHELL/g_menu" -resize
    source "$GMS_SHELL/l_box" -build
    source "$GMS_SHELL/l_menu" -build
    if [ "$GMS_HOT" = "" ]; then GMS_HOT=Q; fi
    source "$GMS_SHELL/l_menu" -update $GMS_HOT $GMS_HOT g_menu -build
    export REPLY_MODULE=g_menu
    export REPLY_ACTION=-update
    export GMS_RECEIVE=1
    export GMS_HOT="Q"

# Remove: -----

elif [ $1 = -remove ]; then
    source "$GMS_SHELL/l_box" -remove

# Update: -----

elif [ $1 = -update ]; then
# Cold:
    if [ "$3" = "" ]; then GMS_COLD=$GMS_HOT; else GMS_COLD=$3; fi

# Hot:
    if [ "$2" = "" ]; then
        source "$GMS_SHELL/l_banner" -no_hot g_menu $2
    else
        GMS_HOT="Q"
        case "$2" in
            "Q" | "q" | "1" ) GMS_HOT="Q" ;;
            "C" | "c" | "2" ) GMS_HOT="C" ;;
            "O" | "o" | "3" ) GMS_HOT="O" ;;
            "V" | "v" | "4" ) GMS_HOT="V" ;;
            "E" | "e" | "5" ) GMS_HOT="E" ;;
            "B" | "b" | "6" ) GMS_HOT="B" ;;
            "A" | "a" | "7" ) GMS_HOT="A" ;;
            "T" | "t" | "8" ) GMS_HOT="T" ;;
            "R" | "r" | "9" ) GMS_HOT="R" ;;
            "L" | "l" | "10" ) GMS_HOT="L" ;;
            "S" | "s" | "11" ) GMS_HOT="S" ;;
            "W" | "w" | "12" ) GMS_HOT="W" ;;
            "I" | "i" | "13" ) GMS_HOT="I" ;;
        esac
    fi
# Handle old, update new:
    if [ "$GMS_HOT" = "$GMS_COLD" ]; then
        source "$GMS_SHELL/g_menu" -handle $GMS_HOT called_by g_menu -update
    else
        source "$GMS_SHELL/l_menu" -update $GMS_HOT $GMS_COLD cd_by g_menu -update
    fi
    export GMS_RECEIVE=1

# Execute: -----

elif [ $1 = -execute ]; then
    backup=$2
    source "$GMS_SHELL/l_box" -remove
    source "$GMS_SHELL/l_desk" -remove
    # Run program:
    if [ "$GMS_FOLDER" != "" -a "$GMS_FILE" != "" ]; then
        source "$GMS_SHELL/gms" $backup
    elif [ "$GMS_HOT" = "W" -o "$GMS_HOT" = "I" ]; then
        source "$GMS_SHELL/gms" $backup
    else
        source "$GMS_SHELL/l_banner" -no_file g_menu
    fi
    export REPLY_OFFSET=0
    export REPLY_SIZE=0
    "$GMS_REPLY" -random 2
    export GMS_RECEIVE=1
    export GMS_TEXTMODE=1
    export REPLY_MODULE=g_palet
    export REPLY_ACTION=-rebuild
    export GMS_RETURN=1
    unset backup

# Handle: -----

elif [ $1 = -handle ]; then

# Analyse:
    if [ $2 = A ]; then
        if [ "$GMS_FILE" = "" ]; then export GMS_FILE="_folder.err"; fi
        source "$GMS_SHELL/g_menu" -execute -A

# View, Edit, Browse, Typeset, Read, Write, Initialize:
    elif [ $2 = V -o $2 = E -o $2 = B -o $2 = T -o \
          $2 = R -o $2 = W -o $2 = I ]; then
        source "$GMS_SHELL/g_menu" -execute $2

# Quit:

```

```

elif [ $2 = Q ]; then
    source "$GMS_SHELL/l_menu" -update - Q
    export REPLY_MODULE=g_good
    export REPLY_ACTION=-build
    export GMS_RECEIVE=1
# Create:
elif [ $2 = C ]; then
    source "$GMS_SHELL/l_menu" -update - C
    source "$GMS_SHELL/l_menu" -update - O
    export GMS_LASTDIR=$PWD
    export GMS_FOLDER=$GMS_TEMPLATE
    export GMS_FILE=
    export GMS_RECEIVE=1
    if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
    source "$GMS_SHELL/l_banner" -upper -first
    source "$GMS_SHELL/l_banner" -lower -folder
    export GMS_FILEBOX=" 1. Select a template to create a copy of "
        source "$GMS_SHELL/g_file" -build -template
    export GMS_FILEBOX=
# Open:
elif [ $2 = 0 ]; then
    source "$GMS_SHELL/l_menu" -update - O
    source "$GMS_SHELL/l_menu" -update - V
    if [ "$GMS_FOLDER" = "$GMS_BINARIES" ]; then
        export GMS_FOLDER="$GMS_SETTING"
        export GMS_FILE=
        source "$GMS_SHELL/l_banner" -upper -first
        source "$GMS_SHELL/l_banner" -lower -folder
    fi
    if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
    export GMS_FILEBOX=" Change folder and open .htm* markup file "
        source "$GMS_SHELL/g_file" -build -open
        export GMS_FILEBOX=
# Learn:
elif [ $2 = L ]; then
    export GMS_FOLDER="$GMS_TEMPLATE/handbook"
    export GMS_FILE="handbook.htm"
    export GMS_BASE="handbook"
    cd "$GMS_FOLDER"
    source "$GMS_SHELL/l_banner" -lower -folder
    source "$GMS_SHELL/l_banner" -upper -handbook
# source "$GMS_SHELL/l_menu" -update B L
    source "$GMS_SHELL/l_menu" -update - L
    export REPLY_MODULE=g_palet
    export REPLY_ACTION=-rebuild
    export GMS_RETURN=1
# Select:
elif [ $2 = S ]; then
    source "$GMS_SHELL/l_menu" -update - S
    source "$GMS_SHELL/l_menu" -update - W
    export REPLY_MODULE=g_select
    export REPLY_ACTION=-build
# Not found:
else
    source "$GMS_SHELL/l_banner" -no_hotkey g_menu $GMS_HOT
fi
# Not found: .....
else
    source "$GMS_SHELL/l_banner" -no_action g_menu $1
fi

```

g_palet

```
#!/bin/sh

# g_palet
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_PALET=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_palet)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_palet $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_palet ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9) >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_palet

# Build: -----

elif [ $1 = -build -o $1 = -rebuild ]; then
    source "$GMS_SHELL/g_palet" -handle \
    $REPLY_DESKTOP $REPLY_PATTERN $REPLY_BANNER \
    $REPLY_TEXT $REPLY_SHADE $REPLY_HOTKEY \
    $REPLY_LETTER
    source "$GMS_SHELL/l_desk" -build
    if [ "$GMS_RETURN" = "" ]; then
        export REPLY_MODULE=g_wel
        export REPLY_ACTION=-remove
        source "$GMS_SHELL/g_wel" -resize
        source "$GMS_SHELL/g_wel" -build
        export GMS_RECEIVE=1
    else
        GMS_RETURN=1
        if [ "$GMS_FILE" = "" ]; then
            source "$GMS_SHELL/l_banner" -upper -first
        else
            source "$GMS_SHELL/l_banner" -upper -file
        fi
        source "$GMS_SHELL/l_banner" -lower -folder
        source "$GMS_SHELL/g_menu" -build
    fi
fi

# Handle: -----

elif [ $1 = -handle ]; then

# Check if color values are acceptable: -----

# 2) Background (desktop) color:
BACKUP=$GMS_DESKTOP
export GMS_DESKTOP="R"
case "$BACKUP" in
    "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" )
        export REPLY_DESKTOP=$BACKUP
        export GMS_DESKTOP=$BACKUP;;
    "R" | "r" ) #export REPLY_DESKTOP=$2;;
        case "$2" in
            "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" )
                export REPLY_DESKTOP=$2;;
        esac
    esac
esac

# 3) Foreground (pattern) color:
BACKUP=$GMS_PATTERN
export GMS_PATTERN="R"
case "$BACKUP" in
    "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | \
    "8" | "9" | "10" | "11" | "12" | "13" | "14" | "15" )
        export REPLY_PATTERN=$BACKUP
        export GMS_PATTERN=$BACKUP;;
    "R" | "r" ) # export REPLY_PATTERN=$3;;
        case "$3" in
            "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | \
            "8" | "9" | "10" | "11" | "12" | "13" | "14" | "15" )
                export REPLY_PATTERN=$3;;
        esac
    esac
esac

# 4) Banner color:
BACKUP=$GMS_BANNER
export GMS_BANNER="R"
case "$BACKUP" in
    "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" )
        export REPLY_BANNER=$BACKUP
        export GMS_BANNER=$BACKUP;;
    "R" | "r" ) # export REPLY_BANNER=$4;;
        case "$4" in
            "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" )
                export REPLY_BANNER=$4;;
        esac
    esac
esac

# 5) Text color:
BACKUP=$GMS_TEXT
export GMS_TEXT="R"
case "$BACKUP" in
    "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | \
    "8" | "9" | "10" | "11" | "12" | "13" | "14" | "15" )
        export REPLY_TEXT=$BACKUP
        export GMS_TEXT=$BACKUP;;
    "R" | "r" ) # export REPLY_TEXT=$5;;
        case "$5" in
            "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | \
            "8" | "9" | "10" | "11" | "12" | "13" | "14" | "15" )
                export REPLY_TEXT=$5;;
        esac
    esac
esac

# 6) Shadow color:
BACKUP=$GMS_SHADE
export GMS_SHADE="R"
case "$BACKUP" in
    "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" )
        export REPLY_SHADE=$BACKUP
        export GMS_SHADE=$BACKUP;;
    "R" | "r" ) # export REPLY_SHADE=$6;;
        case "$6" in
            "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" )
                export REPLY_SHADE=$6;;
        esac
    esac
esac
```

```

esac
# 7) Hotkey color:
BACKUP=$GMS_HOTKEY
export GMS_HOTKEY="R"
case "$BACKUP" in
  "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | \
  "8" | "9" | "10" | "11" | "12" | "13" | "14" | "15" )
    export REPLY_HOTKEY=$BACKUP
    export GMS_HOTKEY=$BACKUP;;
  "R" | "r" ) # export REPLY_HOTKEY=$7;;
    case "$7" in
      "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | \
      "8" | "9" | "10" | "11" | "12" | "13" | "14" | "15" )
        export REPLY_HOTKEY=$7;;
    esac
  esac
esac

# 8) Fill pattern character:
BACKUP=$GMS_LETTER
export GMS_LETTER="R"
case "$BACKUP" in
  "32" | "33" | "34" | "35" | "36" | "37" | "38" | "39" | \
  "40" | "41" | "42" | "43" | "44" | "45" | "46" | "47" | \
  "48" | "49" | "50" | "54" | "52" | "53" | "54" | "55" | \
  "56" | "57" | "58" | "59" | "60" | "61" | "62" | "63" | \
  "64" | "65" | "66" | "67" | "68" | "69" | "70" | "71" | \
  "72" | "73" | "74" | "75" | "76" | "77" | "78" | "79" | \
  "80" | "81" | "82" | "83" | "84" | "85" | "86" | "87" | \
  "88" | "89" | "90" | "91" | "92" | "93" | "94" | "95" | \
  "96" | "97" | "98" | "99" | "100" | "101" | "102" | "103" | \
  "104" | "105" | "106" | "107" | "108" | "109" | "110" | "111" | \
  "112" | "113" | "114" | "115" | "116" | "117" | "118" | "119" | \
  "120" | "121" | "122" | "123" | "124" | "125" | "126" )
    export REPLY_LETTER=$BACKUP
    export GMS_LETTER=$BACKUP;;
  "R" | "r" ) export REPLY_LETTER=$8;;
esac
BACKUP=
export GMS_RECEIVE=1

# Not found: -----
else
  source "$GMS_SHELL/l_banner" -no_action g_palet $1
fi

# Background colors:
# 0 black
# 1 blue
# 2 green
# 3 cyan
# 4 red
# 5 magenta
# 6 brown
# 7 lightgray
# Additional foreground colors:
# 8 darkgray
# 9 lightblue
# 10 lightgreen
# 11 lightcyan
# 12 lightred
# 13 lightmagenta
# 14 yellow
# 15 white

```

g_plug

```
#!/bin/sh

# g_plug
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

# GMSdateG_PLUG=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
  echo "This is Gerolf Markup Shredder (g_plug)."  exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
  source "$GMS_SHELL/l_banner" -debug g_plug $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
  echo "g_plug ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_plug

# Rebuild list of plug-in modules: -----

elif [ "$1" = "-rebuild" ]; then
  plug="$GMS_SETTING/plugin.cfg"
  str="%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"
  # Display info:
  if [ "$2" != "-quiet" ]; then
    export REPLY_BANNER=15
    export REPLY_TEXT=0
    export REPLY_OFFSET=0
    export REPLY_SIZE=16
    "$GMS_BINARIES/reply" -banner 15 "Configuring:"
    export REPLY_OFFSET=14
    export REPLY_SIZE=68
    "$GMS_BINARIES/reply" -banner 15 " $GMS_SETTING/plugin.cfg"
  fi
  # Initial data:
  echo " % plugin.cfg" > "$plug"
  echo " % =====" >> "$plug"
  echo >> "$plug"
  echo " % This TeX file was auto-" >> "$plug"
  echo " % generated by Gerolf Markup Shredder," >> "$plug"
  echo " % written by G. D. Brettschneider (1999-2003)," >> "$plug"
  echo -n " % on ">> "$plug"
  date >> "$plug"
  echo " % All rights reserved." >> "$plug"
  echo " % Send corrections to: MarkupShredder@Gerolf.de" >> "$plug"
  echo " % Subject: GMS TeX macros (module loader)" >> "$plug"
  echo >> "$plug"
  echo " % This file serves for loading all GMS" >> "$plug"
  echo " % modules that could be found in the" >> "$plug"
  echo " % appropriate plug-in directories." >> "$plug"
  echo " % If you have reason to edit this file," >> "$plug"
  echo " % change its name to 'myplugin.cfg'." >> "$plug"
  echo " % The recommended way to add TeX macro files" >> "$plug"

  echo " % to GMS, however, is to \input them in" >> "$plug"
  echo " % 'prologue.cfg' or 'epilogue.cfg'." >> "$plug"
  echo >> "$plug"
  echo " \$str\$str" >> "$plug"
  echo >> "$plug"
  echo " % Initial data:" >> "$plug"
  echo " \def \GMSdate {\$GMS_DATE}" >> "$plug"
  echo " \def \GMSversion {\$GMS_VERSION}" >> "$plug"
  echo " \def \GMScodepage {\$GMS_CODEPAGE}" >> "$plug"
  echo " \def \GMSdebug {\$GMS_DEBUG}" >> "$plug"
  echo >> "$plug"
  echo " \$str\$str" >> "$plug"
  echo >> "$plug"
  # Modules:
  echo " \hyphenmessage" >> "$plug"
  echo " \message {Module loader.}" >> "$plug"
  echo " \fillmessage 42{modules}5" >> "$plug"
  echo >> "$plug"
  # Fonts:
  echo " \fillmessage 6a{font}6" >> "$plug"
  echo " \MAPload" >> "$plug"
  echo " \hyphenmessage" >> "$plug"
  echo " \message {{ font.map}" >> "$plug"
  echo " \hyphenmessage" >> "$plug"
  echo " \message {These names can be used as font face or}" >> "$plug"
  echo " \echo {font-family style names in markup files:}" >> "$plug"
  echo " \MAPfamilyshow" >> "$plug"
  echo " \message {}" >> "$plug"
  echo " \fillmessage 6-{/font}5" >> "$plug"
  echo >> "$plug"
  # Kerning tables:
  echo " \fillmessage 6b{kerning}3" >> "$plug"
  cd "$GMS_ROOT/data/krn"
  for i in *.krn; do
    echo " \KERNINGparseline ($i)\relax" >> "$plug"
  done
  echo " \KERNINGloadtables" >> "$plug"
  echo " \fillmessage 6-{/kerning}2" >> "$plug"
  echo >> "$plug"
  # Hyphenation patterns:
  cd "$GMS_ROOT/tex/hyphen"
  echo " \fillmessage 6c{language}2" >> "$plug"
  for i in *.tex; do
    echo " \LANGUAGEadd $i" >> "$plug"
  done
  echo " \LANGUAGEloadpatterns" >> "$plug"
  echo " \message {More pattern files are available at}" >> "$plug"
  echo " http://www.ctan.org/tex-archive/language/" >> "$plug"
  echo " \fillmessage 6-{/language}1" >> "$plug"
  echo >> "$plug"
  # Unicode rows:
  echo " \fillmessage 6d{unicode}3" >> "$plug"
  cd "$GMS_ROOT/data/row"
  for i in *.row; do
    echo " \UNICODEadd $i" >> "$plug"
  done
  echo " \UNICODEslotsload" >> "$plug"
  echo " \fillmessage 6-{/unicode}2" >> "$plug"
  echo >> "$plug"
  # Entity names:
  echo " \fillmessage 6e{entity}4" >> "$plug"
  cd "$GMS_ROOT/data/ent"
  for i in *.ent; do
    echo " \ENTITYadd $i" >> "$plug"
  done
  echo " \ENTITYnamesload" >> "$plug"
  echo " \ENTITYnamesshow" >> "$plug"
```



```

echo " \fillmessage 6-{\entity}3" >> "$plug"
echo >> "$plug"
# Glyph names:
echo " \fillmessage 6f{glyph}5" >> "$plug"
cd "$GMS_ROOT/data/gly"
for i in *.gly; do
  echo " \GLYPHSadd $i" >> "$plug"
done
echo " \GLYPHSnamesload" >> "$plug"
echo " \fillmessage 6-{\glyph}4" >> "$plug"
echo >> "$plug"
# Codepages:
echo " \fillmessage 6g{codepage}2" >> "$plug"
cd "$GMS_ROOT/data/cp"
for i in *.txt; do
  echo " \CODEPAGEadd $i" >> "$plug"
done
echo " \CODEPAGEloadthem" >> "$plug"
echo " \CODEPAGEencwrite" >> "$plug"
echo " \expandafter \CODEPAGEenable \CODEPAGE \relax" >> "$plug"
## echo " \UNICODEencwrite\relax" >> "$plug"
echo " \fillmessage 6-{\codepage}1" >> "$plug"
echo >> "$plug"
echo " \fillmessage 4-{\modules}3" >> "$plug"
echo " \endinput" >> "$plug"
cd "$GMS_BINARIES"
unset plug

# Build plugin.cfg, but consider myplugin.cfg: -----
elif [ "$1" = "-build" ]; then
  if [ -f "$GMS_SETTING/myplugin.cfg" ]; then
    # If myplugin.cfg exists, copy it to plugin.cfg:
    cp "$GMS_SETTING/myplugin.cfg" "$GMS_SETTING/plugin.cfg"
  else
    # Write new plugin.cfg:
    source "$GMS_SHELL/g_plug" -rebuild g_plug $1
  fi

# Remove plugin.cfg: -----
elif [ "$1" = "-remove" ]; then
  # If myplugin.cfg and plugin.cfg exist, delete plugin.cfg:
  if [ -f "$GMS_SETTING/myplugin.cfg" \
    -a -f "$GMS_SETTING/plugin.cfg" ]; then
    rm "$GMS_SETTING/plugin.cfg"
  fi

# Not found: -----
else
  source "$GMS_SHELL/l_banner" -no_action g_plug $1
fi

```

g_prog

```
#!/bin/sh

# g_prog
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_PROG=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_prog)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_prog $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_prog ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_prog

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=26
    export REPLY_SIZE=20
    export REPLY_ITEMS=7

# Update: -----

elif [ $1 = -update ]; then
# Cold:
    if [ "$3" = "" ]; then GMS_COLD=$GMS_HOT; else GMS_COLD=$3; fi
# Hot:
    if [ "$2" = "" ]; then
        source "$GMS_SHELL/l_banner" -no_hot g_prog called_by g_prog -update
    else
        GMS_HOT="Q"
        case "$2" in
            "v" | "V" | "1" ) GMS_HOT="V";;
            "E" | "e" | "2" ) GMS_HOT="E";;
            "B" | "b" | "3" ) GMS_HOT="B";;
            "A" | "a" | "4" ) GMS_HOT="A";;
            "T" | "t" | "5" ) GMS_HOT="T";;
            "R" | "r" | "6" ) GMS_HOT="R";;
            "Q" | "q" | "7" | "" ) GMS_HOT="Q";;
        esac
    fi
# Handle old, update new:
    if [ "$GMS_HOT" = "$GMS_COLD" ]; then
        source "$GMS_SHELL/g_prog" -handle $GMS_HOT $GMS_COLD
    else
# Set programs:
        source "$GMS_SHELL/l_banner" -lower -select called_by g_prog -update
        if [ "$4" != "" ]; then
            case "$GMS_COLD" in
                "V" ) export GMS_VIEWER=$4;;
                "E" ) export GMS_EDITOR=$4;;
                "B" ) export GMS_BROWSER=$4;;
                "A" ) export GMS_ANALYST=$4;;
                "T" ) export GMS_TSETTER=$4;;
                "R" ) export GMS_READER=$4;;
            esac
        fi
# Normalize writing:
        if [ "$GMS_TSETTER" = "" ]; then export GMS_TSETTER=pdfetex
        elif [ "$GMS_TSETTER" = "PDFTEX" ]; then export GMS_TSETTER=pdfetex
        elif [ "$GMS_TSETTER" = "Pftetex" ]; then export GMS_TSETTER=pdfetex
        elif [ "$GMS_TSETTER" = "pdfTeX" ]; then export GMS_TSETTER=pdfetex
        elif [ "$GMS_TSETTER" = "PDFTEX" ]; then export GMS_TSETTER=pdfTeX
        elif [ "$GMS_TSETTER" = "Pftex" ]; then export GMS_TSETTER=pdfTeX
        elif [ "$GMS_TSETTER" = "pdfTeX" ]; then export GMS_TSETTER=pdfTeX
        elif [ "$GMS_TSETTER" = "ETEX" ]; then export GMS_TSETTER=etex
        elif [ "$GMS_TSETTER" = "Etex" ]; then export GMS_TSETTER=etex
        elif [ "$GMS_TSETTER" = "eTeX" ]; then export GMS_TSETTER=etex
        elif [ "$GMS_TSETTER" = "TEX" ]; then export GMS_TSETTER=TeX
        elif [ "$GMS_TSETTER" = "Tex" ]; then export GMS_TSETTER=TeX
        elif [ "$GMS_TSETTER" = "TeX" ]; then export GMS_TSETTER=TeX
        fi
# Update program box:
        source "$GMS_SHELL/l_prog" -update $GMS_HOT $GMS_COLD # $4
    fi
    export GMS_RECEIVE=1

# Build: -----

elif [ $1 = -build ]; then
    source "$GMS_SHELL/l_banner" -lower -program called_by g_prog -build
    source "$GMS_SHELL/g_prog" -resize called_by g_prog -build
    source "$GMS_SHELL/l_box" -t11_build called_by g_prog -build
    source "$GMS_SHELL/l_prog" -build called_by g_prog -build
    source "$GMS_SHELL/l_prog" -update Q Q called_by g_prog -build
    export REPLY_MODULE=g_prog
    export REPLY_ACTION=-update
    export GMS_RECEIVE=1
    export GMS_HOT="Q"

# Remove: -----

elif [ $1 = -remove ]; then
    source "$GMS_SHELL/g_prog" -resize called_by g_prog -remove
    source "$GMS_SHELL/l_box" -t12_remove called_by g_prog -remove
    source "$GMS_SHELL/g_select" -resize called_by g_prog -remove
    source "$GMS_SHELL/g_select" -update P Q called_by g_prog -remove
    export REPLY_MODULE=g_select
    export REPLY_ACTION=-update
    export GMS_RECEIVE=1

# Handle: -----

elif [ $1 = -handle ]; then

# Quit:
    if [ $2 = Q ]; then
        source "$GMS_SHELL/g_launch" -build called_by g_prog -handle
        source "$GMS_SHELL/g_prog" -remove called_by g_prog -handle

# Viewer:
        elif [ $2 = V ]; then
            export GMS_RECEIVE=1
            source "$GMS_SHELL/l_banner" -lower -viewer called_by g_prog -handle
            source "$GMS_SHELL/l_prog" -update - V called_by g_prog -handle
            setterm -cursor on
            "$GMS_REPLY" -question 8 "$GMS_VIEWER" VV 1
        fi
    fi
fi
```

```

    setterm -cursor off

# Editor:
elif [ $2 = E ]; then
    export GMS_RECEIVE=1
    source $GMS_SHELL/l_banner -lower -editor called_by g_prog -handle
    source $GMS_SHELL/l_prog -update - E called_by g_prog -handle
    setterm -cursor on
    $GMS_REPLY -question 10 "$GMS_EDITOR" EE 2
    setterm -cursor off

# Browser:
elif [ $2 = B ]; then
    export GMS_RECEIVE=1
    source "$GMS_SHELL/l_banner" -lower -browser called_by g_prog -handle
    source "$GMS_SHELL/l_prog" -update - B called_by g_prog -handle
    setterm -cursor on
    "$GMS_REPLY" -question 12 "$GMS_BROWSER" BB 3
    setterm -cursor off

# Analyst:
elif [ $2 = A ]; then
    export GMS_RECEIVE=1
    source "$GMS_SHELL/l_banner" -lower -analyst called_by g_prog -handle
    source "$GMS_SHELL/l_prog" -update - A called_by g_prog -handle
    setterm -cursor on
    "$GMS_REPLY" -question 14 "$GMS_ANALYST" AA 4
    setterm -cursor off

# Typesetter:
elif [ $2 = T ]; then
    export GMS_RECEIVE=1
    source "$GMS_SHELL/l_banner" -lower -tsetter called_by g_prog -handle
    source "$GMS_SHELL/l_prog" -update - T called_by g_prog -handle
    setterm -cursor on
    "$GMS_REPLY" -question 16 "$GMS_TSETTER" TT 5
    setterm -cursor off

# Reader:
elif [ $2 = R ]; then
    export GMS_RECEIVE=1
    source "$GMS_SHELL/l_banner" -lower -reader called_by g_prog -handle
    source "$GMS_SHELL/l_prog" -update - R called_by g_prog -handle
    setterm -cursor on
    "$GMS_REPLY" -question 18 "$GMS_READER" RR 6
    setterm -cursor off

# Not found:
else
    source "$GMS_SHELL/l_banner" -no_hotkey g_prog $2
fi

# Not found: -----
else
    source "$GMS_SHELL/l_banner" -no_action g_prog $1
fi

```

g_rain

```
#!/bin/sh

# g_rain
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_RAIN=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_rain)."
```

exit

```
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_rain $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_rain ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====
```

```
# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_rain

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=3
    export REPLY_SIZE=17
    export REPLY_ITEMS=1

# Build: -----

elif [ $1 = -build ]; then
    source "$GMS_SHELL/g_rain" -resize called_by g_rain -build
    source "$GMS_SHELL/l_rain" -build called_by g_rain -build

# Remove: -----

elif [ $1 = -remove ]; then
    source "$GMS_SHELL/g_rain" -resize called_by g_rain -remove
    source "$GMS_SHELL/l_rain" -remove called_by g_rain -remove

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action g_rain $1
fi
```

g_save

```
#!/bin/sh

# g_save
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_SAVE=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_save)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_save $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_save ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_save

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=3
    export REPLY_SIZE=44
    export REPLY_ITEMS=4

# Build: -----

elif [ $1 = -build ]; then
    source "$GMS_SHELL/l_banner" -lower -folder
    export REPLY_LIST=0
    export GMS_FILE="New$.htm"
    source "$GMS_SHELL/g_save" -resize
    source "$GMS_SHELL/l_box" -t8_build
    GMS_FILEBOX=" 3. Enter a better name for this new file "
    source "$GMS_SHELL/l_save" -build
    GMS_FILEBOX=
# Consider existing file with same name:
    GMS_EXIST=0
    if [ -f "$GMS_FOLDER/$GMS_FILE" ]; then GMS_EXIST=1; fi
    source "$GMS_SHELL/l_save" -update Q Q $GMS_EXIST
    GMS_EXIST=
    export REPLY_MODULE=g_save
    export REPLY_ACTION=-update
    export GMS_RECEIVE=1
    export GMS_HOT="Q"

# Remove: -----

elif [ $1 = -remove ]; then
    source "$GMS_SHELL/g_save" -resize
    source "$GMS_SHELL/l_box" -t8_remove
    source "$GMS_SHELL/g_menu" -resize
    source "$GMS_SHELL/l_menu" -update C Q

export REPLY_MODULE=g_menu
export REPLY_ACTION=-update
export GMS_RECEIVE=1

# Update: -----

elif [ $1 = -update ]; then
# Cold:
    if [ "$3" = "" ]; then GMS_COLD=$GMS_HOT; else GMS_COLD=$3; fi
# Hot:
    if [ "$2" = "" ]; then
        source "$GMS_SHELL/l_banner" -no_hot g_save
    else
        GMS_HOT="Q"
        case "$2" in
            "Q" | "q" | "1" ) GMS_HOT="Q";;
            "C" | "c" | "2" ) GMS_HOT="C";;
            "O" | "o" | "3" ) GMS_HOT="O";;
            "N" | "n" | "4" ) GMS_HOT="N";;
        esac
    fi
# Handle old, update new:
    if [ "$GMS_HOT" = "$GMS_COLD" ]; then
        source "$GMS_SHELL/g_save" -handle $GMS_HOT
# Fixme for spacy names like DOS version
    else
        if [ "$4" != "" ]; then
            export GMS_FILE="$4"
        fi
# Update save box (and consider existing file with same name):
        GMS_EXIST=0
        if [ -f "$GMS_FOLDER/$GMS_FILE" ]; then GMS_EXIST=1; fi
        source "$GMS_SHELL/l_save" -update $GMS_HOT $GMS_COLD $GMS_EXIST
        GMS_EXIST=
    fi
    export GMS_RECEIVE=1

# Handle: -----

elif [ $1 = -handle ]; then

# Overwrite (copy template):
    if [ $2 = "" -o $2 = 0 ]; then
        if [ -f "$GMS_TFOLDER/$GMS_TEMPLATEFILE" ]; then
            cp "$GMS_TFOLDER/$GMS_TEMPLATEFILE" "$GMS_FOLDER/$GMS_FILE"
        fi
        source "$GMS_SHELL/l_banner" -upper -file
        source "$GMS_SHELL/g_save" -remove
        if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi

# Quit:
        elif [ $2 = Q ]; then
            if [ -f "$GMS_FOLDER/$GMS_FILE" ]; then
                source "$GMS_SHELL/l_banner" -upper -file
                source "$GMS_SHELL/g_save" -remove
            else
                source "$GMS_SHELL/g_save" -handle 0
            fi

# Cancel:
        elif [ $2 = C ]; then
            source "$GMS_SHELL/l_banner" -upper -file
            source "$GMS_SHELL/g_save" -remove

# Name:
        elif [ $2 = N ]; then idle=

# Not found:
```

```
else
  source "$GMS_SHELL/1_banner" -no_hotkey g_save $2
fi
```

```
# Not found: -----
else
  source "$GMS_SHELL/1_banner" -no_action g_save $1
fi
```

g_select

```
#!/bin/sh

# g_select
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_select=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_select)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_select $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_select ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then
    source "$GMS_SHELL/l_banner" -no_action g_select
fi

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=3
    export REPLY_SIZE=17
    export REPLY_ITEMS=8
fi

# Update: -----

elif [ $1 = -update ]; then
# Cold:
    if [ "$3" = "" ]; then GMS_COLD=$GMS_HOT; else GMS_COLD=$3; fi
# Hot:
    if [ "$2" = "" ]; then
        source "$GMS_SHELL/l_banner" -no_hot g_select
    else
        GMS_HOT="Q"
        case "$2" in
            "Q" | "q" | "1" ) GMS_HOT="Q";;
            "A" | "a" | "2" ) GMS_HOT="A";;
            "C" | "c" | "3" ) GMS_HOT="C";;
            "P" | "p" | "4" ) GMS_HOT="P";;
            "D" | "d" | "5" ) GMS_HOT="D";;
            "X" | "x" | "0" ) GMS_HOT="D";;
            "Y" | "y" | "o" ) GMS_HOT="D";;
            "E" | "e" | "6" ) GMS_HOT="E";;
            "I" | "i" ) GMS_HOT="E";;
            "U" | "u" | "7" ) GMS_HOT="U";;
            "L" | "l" | "8" ) GMS_HOT="L";;
        esac
    fi
fi

# Handle old, update new:
if [ "$GMS_HOT" = "$GMS_COLD" ]; then
    source "$GMS_SHELL/g_select" -handle $GMS_HOT
else
    # Banner:
    case "$GMS_HOT" in
        "Q" ) source "$GMS_SHELL/l_banner" -lower -first;;
        "A" ) source "$GMS_SHELL/l_banner" -lower -animate;;
        "U" | "L" ) source "$GMS_SHELL/l_banner" -lower -codepage;;
        "C" | "P" | "D" | "E" ) source "$GMS_SHELL/l_banner" -lower -select;;
    esac
    # Debugging:
    if [ "$GMS_COLD" = "D" -a "$4" != "" ]; then
        case "$4" in
            "X" | "x" ) export GMS_DEBUG="X";;
            "Y" | "y" ) export GMS_DEBUG="Y";;
            "Z" | "z" ) export GMS_DEBUG="Z"
        esac
        source "$GMS_SHELL/g_select" -dbg_remove \
            called_by g_select -update;;
        "O" | "o" | "0" ) export GMS_DEBUG="0"
        source "$GMS_SHELL/g_select" -dbg_remove \
            called_by g_select -update;;
    esac
    source "$GMS_SHELL/g_launch" -build called_by g_select -update
fi
# Encoding:
if [ "$GMS_COLD" = "E" -a "$4" != "" ]; then
    export GMS_CODEPAGE=$4
    # Normalize writing:
    if [ "$4" = "ARABIC" ]; then export GMS_CODEPAGE=CP1256; fi
    if [ "$4" = "Arabic" ]; then export GMS_CODEPAGE=CP1256; fi
    if [ "$4" = "arabic" ]; then export GMS_CODEPAGE=CP1256; fi
    if [ "$4" = "ARABICBH" ]; then export GMS_CODEPAGE=CP1256; fi
    if [ "$4" = "ArabicBH" ]; then export GMS_CODEPAGE=CP1256; fi
    if [ "$4" = "arabicbh" ]; then export GMS_CODEPAGE=CP1256; fi
    if [ "$4" = "ARABICMT" ]; then export GMS_CODEPAGE=CP1256; fi
    if [ "$4" = "ArabicMT" ]; then export GMS_CODEPAGE=CP1256; fi
    if [ "$4" = "arabicmt" ]; then export GMS_CODEPAGE=CP1256; fi
    if [ "$4" = "ASCII" ]; then export GMS_CODEPAGE=US-ASCII; fi
    if [ "$4" = "Ascii" ]; then export GMS_CODEPAGE=US-ASCII; fi
    if [ "$4" = "ascii" ]; then export GMS_CODEPAGE=US-ASCII; fi
    if [ "$4" = "BALTIC" ]; then export GMS_CODEPAGE=CP1257; fi
    if [ "$4" = "Baltic" ]; then export GMS_CODEPAGE=CP1257; fi
    if [ "$4" = "baltic" ]; then export GMS_CODEPAGE=CP1257; fi
    if [ "$4" = "CANADA" ]; then export GMS_CODEPAGE=CP863; fi
    if [ "$4" = "Canada" ]; then export GMS_CODEPAGE=CP863; fi
    if [ "$4" = "canada" ]; then export GMS_CODEPAGE=CP863; fi
    if [ "$4" = "CENTRAL" ]; then export GMS_CODEPAGE=CP1250; fi
    if [ "$4" = "Central" ]; then export GMS_CODEPAGE=CP1250; fi
    if [ "$4" = "central" ]; then export GMS_CODEPAGE=CP1250; fi
    if [ "$4" = "Cp437" ]; then export GMS_CODEPAGE=CP437; fi
    if [ "$4" = "cp437" ]; then export GMS_CODEPAGE=CP437; fi
    if [ "$4" = "Cp850" ]; then export GMS_CODEPAGE=CP850; fi
    if [ "$4" = "cp850" ]; then export GMS_CODEPAGE=CP850; fi
    if [ "$4" = "Cp860" ]; then export GMS_CODEPAGE=CP860; fi
    if [ "$4" = "cp860" ]; then export GMS_CODEPAGE=CP860; fi
    if [ "$4" = "Cp863" ]; then export GMS_CODEPAGE=CP863; fi
    if [ "$4" = "cp863" ]; then export GMS_CODEPAGE=CP863; fi
    if [ "$4" = "Cp865" ]; then export GMS_CODEPAGE=CP865; fi
    if [ "$4" = "cp865" ]; then export GMS_CODEPAGE=CP865; fi
    if [ "$4" = "Cp874" ]; then export GMS_CODEPAGE=CP874; fi
    if [ "$4" = "cp874" ]; then export GMS_CODEPAGE=CP874; fi
    if [ "$4" = "Cp1250" ]; then export GMS_CODEPAGE=CP1250; fi
    if [ "$4" = "cp1250" ]; then export GMS_CODEPAGE=CP1250; fi
    if [ "$4" = "Cp1251" ]; then export GMS_CODEPAGE=CP1251; fi
    if [ "$4" = "cp1251" ]; then export GMS_CODEPAGE=CP1251; fi
    if [ "$4" = "Cp1252" ]; then export GMS_CODEPAGE=CP1252; fi
    if [ "$4" = "cp1252" ]; then export GMS_CODEPAGE=CP1252; fi
    if [ "$4" = "Cp1253" ]; then export GMS_CODEPAGE=CP1253; fi
    if [ "$4" = "cp1253" ]; then export GMS_CODEPAGE=CP1253; fi
    if [ "$4" = "Cp1254" ]; then export GMS_CODEPAGE=CP1254; fi

```



```

if [ "$4" = "Usa" ]; then export GMS_CODEPAGE=CP437; fi
if [ "$4" = "usa" ]; then export GMS_CODEPAGE=CP437; fi
if [ "$4" = "US_ASCII" ]; then export GMS_CODEPAGE=US-ASCII; fi
if [ "$4" = "Us_ascii" ]; then export GMS_CODEPAGE=US-ASCII; fi
if [ "$4" = "us_ascii" ]; then export GMS_CODEPAGE=US-ASCII; fi
if [ "$4" = "Us-ascii" ]; then export GMS_CODEPAGE=US-ASCII; fi
if [ "$4" = "us-ascii" ]; then export GMS_CODEPAGE=US-ASCII; fi
if [ "$4" = "VIETNAM" ]; then export GMS_CODEPAGE=VISCII; fi
if [ "$4" = "Vietnam" ]; then export GMS_CODEPAGE=VISCII; fi
if [ "$4" = "vietnam" ]; then export GMS_CODEPAGE=VISCII; fi
if [ "$4" = "VIETNAMESE" ]; then export GMS_CODEPAGE=VISCII; fi
if [ "$4" = "Vietnamese" ]; then export GMS_CODEPAGE=VISCII; fi
if [ "$4" = "vietnamese" ]; then export GMS_CODEPAGE=VISCII; fi
if [ "$4" = "Viscii" ]; then export GMS_CODEPAGE=VISCII; fi
if [ "$4" = "viscii" ]; then export GMS_CODEPAGE=VISCII; fi
if [ "$4" = "WESTERN" ]; then export GMS_CODEPAGE=CP1252; fi
if [ "$4" = "Western" ]; then export GMS_CODEPAGE=CP1252; fi
if [ "$4" = "western" ]; then export GMS_CODEPAGE=CP1252; fi
if [ "$4" = "WINDOWS-1250" ]; then export GMS_CODEPAGE=CP1250; fi
if [ "$4" = "Windows-1250" ]; then export GMS_CODEPAGE=CP1250; fi
if [ "$4" = "windows-1250" ]; then export GMS_CODEPAGE=CP1250; fi
if [ "$4" = "WINDOWS-1251" ]; then export GMS_CODEPAGE=CP1251; fi
if [ "$4" = "Windows-1251" ]; then export GMS_CODEPAGE=CP1251; fi
if [ "$4" = "windows-1251" ]; then export GMS_CODEPAGE=CP1251; fi
if [ "$4" = "WINDOWS-1252" ]; then export GMS_CODEPAGE=CP1252; fi
if [ "$4" = "Windows-1252" ]; then export GMS_CODEPAGE=CP1252; fi
if [ "$4" = "windows-1252" ]; then export GMS_CODEPAGE=CP1252; fi
if [ "$4" = "WINDOWS-1253" ]; then export GMS_CODEPAGE=CP1253; fi
if [ "$4" = "Windows-1253" ]; then export GMS_CODEPAGE=CP1253; fi
if [ "$4" = "windows-1253" ]; then export GMS_CODEPAGE=CP1253; fi
if [ "$4" = "WINDOWS-1254" ]; then export GMS_CODEPAGE=CP1254; fi
if [ "$4" = "Windows-1254" ]; then export GMS_CODEPAGE=CP1254; fi
if [ "$4" = "windows-1254" ]; then export GMS_CODEPAGE=CP1254; fi
if [ "$4" = "windows-1254" ]; then export GMS_CODEPAGE=CP1254; fi
if [ "$4" = "WINDOWS-1255" ]; then export GMS_CODEPAGE=CP1255; fi
if [ "$4" = "Windows-1255" ]; then export GMS_CODEPAGE=CP1255; fi
if [ "$4" = "windows-1255" ]; then export GMS_CODEPAGE=CP1255; fi
if [ "$4" = "WINDOWS-1256" ]; then export GMS_CODEPAGE=CP1256; fi
if [ "$4" = "Windows-1256" ]; then export GMS_CODEPAGE=CP1256; fi
if [ "$4" = "windows-1256" ]; then export GMS_CODEPAGE=CP1256; fi
if [ "$4" = "WINDOWS-1257" ]; then export GMS_CODEPAGE=CP1257; fi
if [ "$4" = "Windows-1257" ]; then export GMS_CODEPAGE=CP1257; fi
if [ "$4" = "windows-1257" ]; then export GMS_CODEPAGE=CP1257; fi
if [ "$4" = "WINDOWS-1258" ]; then export GMS_CODEPAGE=CP1258; fi
if [ "$4" = "Windows-1258" ]; then export GMS_CODEPAGE=CP1258; fi
if [ "$4" = "windows-1258" ]; then export GMS_CODEPAGE=CP1258; fi
source "$GMS_SHELL/g_launch" -build called_by g_select -update
fi
# Select:
if [ "$GMS_ANIMATE" = "on" ]; then
source "$GMS_SHELL/l_select" -update $GMS_HOT $GMS_COLD "off"
else
source "$GMS_SHELL/l_select" -update $GMS_HOT $GMS_COLD "on"
fi
fi; export GMS_RECEIVE=1
# Build: -----
elif [ $1 = -build ]; then
source "$GMS_SHELL/l_banner" -lower -select called_by g_select -build
source "$GMS_SHELL/g_select" -resize called_by g_select -build
source "$GMS_SHELL/l_box" -t11_build called_by g_select -build
if [ "$GMS_ANIMATE" = "on" ]; then
source "$GMS_SHELL/l_select" -build "off" called_by g_select -build
else source "$GMS_SHELL/l_select" -build "on" called_by g_select -build; fi
source "$GMS_SHELL/l_select" -update Q Q called_by g_select -build
export REPLY_MODULE=g_select
export REPLY_ACTION=-update
export GMS_RECEIVE=1
export GMS_HOT="Q"
# Remove: -----
elif [ $1 = -remove ]; then
source "$GMS_SHELL/g_select" -resize called_by g_select -remove
source "$GMS_SHELL/l_box" -t11_remove called_by g_select -remove
source "$GMS_SHELL/g_menu" -resize called_by g_select -remove
source "$GMS_SHELL/l_box" -idle called_by g_select -remove
source "$GMS_SHELL/g_menu" -update S Q called_by g_select -remove
export REPLY_MODULE=g_menu
export REPLY_ACTION=-update
# Remove debug banners: -----
elif [ $1 = -dbg_remove ]; then
source "$GMS_SHELL/l_desk" -resize called_by g_select -dbg_remove
"$GMS_REPLY" -stripe 2 $REPLY_PATTERN $REPLY_LETTER
"$GMS_REPLY" -stripe 23 $REPLY_PATTERN $REPLY_LETTER
source "$GMS_SHELL/g_select" -resize called_by g_select -dbg_remove
# Handle: -----
elif [ $1 = -handle ]; then
# Quit:
if [ $2 = Q ]; then
source "$GMS_SHELL/l_banner" -lower -folder called_by g_select -handle
source "$GMS_SHELL/g_select" -remove called_by g_select -handle
# Animation:
elif [ $2 = A ]; then
if [ "$GMS_ANIMATE" = on ]; then
export GMS_ANIMATE="off"
else export GMS_ANIMATE="on"; fi
source "$GMS_SHELL/g_launch" -build called_by g_select -handle
source "$GMS_SHELL/g_select" -update A Q called_by g_select -handle
export REPLY_MODULE=g_select
export REPLY_ACTION=-update
# Color:
elif [ $2 = C ]; then
source "$GMS_SHELL/l_select" -update - C called_by g_select -handle
source "$GMS_SHELL/l_select" -update - P called_by g_select -handle
source "$GMS_SHELL/g_select" -resize called_by g_select -handle
source "$GMS_SHELL/l_box" -t11_remove called_by g_select -handle
source "$GMS_SHELL/g_rain" -build called_by g_select -handle
source "$GMS_SHELL/g_color" -build called_by g_select -handle
# Programs:
elif [ $2 = P ]; then
source "$GMS_SHELL/l_select" -update - D called_by g_select -handle
source "$GMS_SHELL/l_select" -update - P called_by g_select -handle
source "$GMS_SHELL/g_prog" -build called_by g_select -handle
# Debugging:
elif [ $2 = D ]; then
source "$GMS_SHELL/l_banner" -lower -debug called_by g_select -handle
source "$GMS_SHELL/l_select" -update - D called_by g_select -handle
setterm -cursor on
"$GMS_REPLY" -question 11 "$GMS_DEBUG" DD 5 called_by g_select -handle
setterm -cursor off
# Encoding:
elif [ $2 = E ]; then
source "$GMS_SHELL/l_banner" -lower -encode called_by g_select -handle
source "$GMS_SHELL/l_select" -update - E called_by g_select -handle
setterm -cursor on
"$GMS_REPLY" -question 13 "$GMS_CODEPAGE" EE 6 \
called_by g_select -handle
setterm -cursor off

```

```
# Codepage (upper half):
elif [ $2 = U ]; then
    source "$GMS_SHELL/g_code" -cp_up_build called_by g_select -handle

# Codepage (lower half):
elif [ $2 = L ]; then
    source "$GMS_SHELL/g_code" -cp_lo_build called_by g_select -handle

# Not found:
else
```

```
    source "$GMS_SHELL/l_banner" -no_hotkey g_select $2
fi

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action g_select $1
fi
```

g_vars

```
#!/bin/sh

# g_vars
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_VARS=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_vars)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_vars $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_vars ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_vars

# Initialize file and folders: -----

elif [ "$1" = "-ini_file" ]; then
    arg="##### New work file: #####"
    if [ "$GMS_DEBUG" = "Z" -a "$GMS_MODE" != "quiet" ]; then
        echo " $arg#####" >> "$Z"; fi
    unset arg
    # Dismiss -ini_file and caller(s):
    if [ "$2" = "called_by" ]; then shift; fi
    if [ "$1" = "called_by" ]; then shift; fi
    if [ "$2" = "called_by" ]; then shift; fi
    if [ "$1" = "called_by" ]; then shift; fi
    if [ "$2" = "-passive" ]; then shift; fi
    if [ "$1" = "-passive" ]; then shift; fi
    if [ "$2" = "-a" ]; then shift; fi
    if [ "$2" = "-t" ]; then shift; fi
    shift
    export GMS_RETURN=
    # Now $1 ... $9 can be a file name:
    if [ "$1" != "" ]; then
        # $1 ... $9 may include path, so get filename and pathname: .....
        BACKUP_FOLDER="$GMS_FOLDER"
        export GMS_FILE="$1"
        if [ "$2" != "" ]; then export GMS_FILE="$GMS_FILE $2"; fi
        if [ "$3" != "" ]; then export GMS_FILE="$GMS_FILE $3"; fi
        if [ "$4" != "" ]; then export GMS_FILE="$GMS_FILE $4"; fi
        if [ "$5" != "" ]; then export GMS_FILE="$GMS_FILE $5"; fi
        if [ "$6" != "" ]; then export GMS_FILE="$GMS_FILE $6"; fi
        if [ "$7" != "" ]; then export GMS_FILE="$GMS_FILE $7"; fi
        if [ "$8" != "" ]; then export GMS_FILE="$GMS_FILE $8"; fi
        if [ "$9" != "" ]; then export GMS_FILE="$GMS_FILE $9"; fi
        export GMS_FOLDER="$GMS_FILE"
        # Get file name if $GMS_FILE contains path name:
        export file="$(basename "$GMS_FILE")" # remove dirname
        export GMS_FILE=$file
        # Get path name if $GMS_FOLDER contains file name:
        export path="$(dirname "$GMS_FOLDER")" # remove filename
        export GMS_FOLDER="$BACKUP_FOLDER"
        export GMS_FILE="$BACKUP_FILE"
        unset BACKUP_FOLDER
        unset BACKUP_FILE
        # Check if file is in current folder: .....
        if [ -f "$PWD/$file" ]; then
            if [ "$GMS_DEBUG" = "Z" -a "$GMS_MODE" != "quiet" ]; then
                echo " File is in current folder:" >> "$Z"; fi
            export GMS_RETURN=1
            export GMS_FILE="$file"
            export GMS_FOLDER="$PWD"
            export GMS_MESSAGE="$file"
            cd "$GMS_FOLDER"
            export GMS_FOLDER="$PWD"
            export file=
            export path=
            source "$GMS_SHELL/g_vars" -write_memo called_by g_vars -ini_file
        fi
        # Check if file is in a related folder: .....
        if [ -f "$PWD/$path/$file" ]; then
            if [ "$GMS_DEBUG" = "Z" -a "$GMS_MODE" != "quiet" ]; then
                echo " File is in a related folder:" >> "$Z"; fi
            export GMS_RETURN=1
            export GMS_FILE="$file"
            export GMS_FOLDER="$PWD/$path_"
            export GMS_MESSAGE="$file"
            cd "$GMS_FOLDER"
            export GMS_FOLDER="$PWD"
            export file=
            export path=
            source "$GMS_SHELL/g_vars" -write_memo called_by g_vars -ini_file
        fi
        # Check if file is at a remote place: .....
        if [ -f "$path/$file" ]; then
            if [ "$GMS_DEBUG" = "Z" -a "$GMS_MODE" != "quiet" ]; then
                echo " File is in a remote folder:" >> "$Z"; fi
            export GMS_RETURN=1
            export GMS_FILE="$file"
            export GMS_FOLDER="$path_"
            export GMS_MESSAGE="$file"
            export PWD="$path_"
            if [ "$GMS_DEBUG" = "Z" -a "$GMS_MODE" != "quiet" ]; then
                echo " PWD=$PWD" >> "$Z"; fi
            export file=
            export path=
            source "$GMS_SHELL/g_vars" -write_memo called_by g_vars -ini_file
        fi
        fi
        export file=
        export path=
        # If no file is given, get last one and folder from memo: .....
        if [ "$GMS_FILE" = "" ]; then
            if [ "$GMS_DEBUG" = "Z" ]; then
                echo " Reading file and folder from gms_memo:" >> "$Z"
            fi
            if [ -f "$GMS_SETTING/gms_memo" ]; then
                source "$GMS_SETTING/gms_memo"; fi
            # Fixme: Set GMS_MESSAGE, if file is in a remote folder
            if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
            export GMS_FOLDER="$PWD"
            export GMS_RETURN=
        fi
        # If no folder is present, set to current folder: .....
        if [ "$GMS_FOLDER" = "" ]; then
            if [ "$GMS_DEBUG" = "Z" ]; then
                echo " Reading file and folder from gms_memo:" >> "$Z"
            fi
        fi
    fi
fi
```

```

    echo " File is in current folder:" >> "$Z"
fi
export GMS_FOLDER="$PWD"
export GMS_RETURN=
fi
if [ "$GMS_DEBUG" = "Z" ]; then
    echo " GMS_FILE=$GMS_FILE" >> "$Z"
    echo " GMS_FOLDER=$GMS_FOLDER" >> "$Z"
fi
arg="#####"
if [ "$GMS_DEBUG" = "Z" ]; then echo " $arg#####" >> "$Z"; fi
unset arg

# Write file name and folder to gms_memo.bat startup file: -----

elif [ "$1" = "-write_memo" ]; then
if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
str="$GMS_SETTING/gms_memo"
echo "#!/bin/sh" > "$str"
echo "export GMS_FOLDER=\"$GMS_FOLDER\" >> \"$str"
echo "export GMS_FILE=\"$GMS_FILE\" >> \"$str"
unset str

# Clear environment variables: -----

```

```

elif [ $1 = -clear ]; then
    source "$GMS_SHELL/g_list" -clear called_by g_vars -clear

# Build new gmsdebug log file header:
export Z="$GMS_SETTING/_gmshead.txt"
echo " gmsdebug.log" > "$Z"
arg="#####"
echo " $arg$arg" >> "$Z"
unset arg
echo " This is the debug log file for Gerolf Markup Shredder." >> "$Z"
echo " To disable the run time call tracing, set GMS_DEBUG=0." >> "$Z"
echo " Format: called script (parameter 1) ... (parameter 9)," >> "$Z"
echo " followed by values of important environment variables." >> "$Z"
echo " 'shredder': reads user input and calls script modules." >> "$Z"
echo >> "$Z"
export Z="$GMS_SETTING/gmsdebug.log"

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action g_vars $1
fi

```

g_wel

```
#!/bin/sh

# g_wel
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateG_WEL=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (g_wel)."
```

```
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_wel $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "g_wel ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action g_wel

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=2
    export REPLY_SIZE=25
    export REPLY_ITEMS=1

# Build: -----

elif [ $1 = -build ]; then
    source "$GMS_SHELL/g_wel" -resize called_by g_wel -build
    source "$GMS_SHELL/l_wel" -build called_by g_wel -build
    export GMS_RECEIVE=1

# Remove: -----

elif [ $1 = -remove ]; then
    source "$GMS_SHELL/g_wel" -resize called_by g_wel -remove
    source "$GMS_SHELL/l_wel" -remove called_by g_wel -remove
    source "$GMS_SHELL/l_banner" -upper -first called_by g_wel -remove
    source "$GMS_SHELL/l_banner" -lower -first called_by g_wel -remove
    source "$GMS_SHELL/g_menu" -build called_by g_wel -remove

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action g_wel $1
fi
```

gms

```
#!/bin/sh

# gms
# ===

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateGMS=20060927

# Prologue: =====

# Debug:

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    REPLY_BANNER=2
    REPLY_TEXT=15
    source "$GMS_SHELL/l_banner" -debug g_vars $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" -a "$GMS_MODE" != "quiet" ]; then
    echo " gms ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# 1. Check common variables: =====

export GMS_PWD=$PWD

# If necessary, read launcher script to load environment:

if [ "$GMS_SHELL" = "" ]; then
    source gerolf -passive $1 $2 $3 $4 $5 $6 $7 $8 $9
    if [ "$1" = "-passive" ]; then shift; fi # required for SuSE
    if [ "$GMS_SHELL" = "" ]; then
        echo " GMS error: Could not set environment variables."
        echo " Run 'gmssetup' to create GMS launcher script 'gerolf!'"
        echo
        exit
    fi
fi

# 2. Check chapter variable: =====

# GMS is not yet running:

if [ "$1" = "" ]; then
    source "$GMS_SHELL/l_gms" -welcome called_by gms
    read
    exec "$GMS_SHELL/l_gms" -help called_by gms
fi

# Remove optional "-mime" argument at first position (browser interface): ----

if [ "$1" = "-mime" ]; then
    if [ "$2" = "-read" ]; then
        echo "Content-type: application/pdf"
    else
        echo "Content-type: text/plain"
    fi
    shift
fi

# Get run time: -----

function get_runtime () {
    if [ "$BEFORE" != "" ]; then
        AFTER=$(date +%s)
        if [ "$AFTER" = "$BEFORE" ]; then
            GMS_RUNTIME="<1s"
        else
            GMS_RUNTIME=$(($AFTER - $BEFORE) | sed 's/000$/0000/')s
        fi
    fi
}

# Chapters: =====

# Set file, if $1 is one:

if [ -f "$1" ]; then
    source "$GMS_SHELL/gms" -o "$1"
fi

# Quit: -----

elif [ "$1" = "-g" -o "$1" = "-Q" -o \
"$1" = "-.g" -o "$1" = "-.Q" -o \
"$1" = "/g" -o "$1" = "/Q" -o \
"$1" = "g" -o "$1" = "Q" -o \
"$1" = "gg" -o "$1" = "QQ" ]; then
    if [ "$GMS_MODE" != "quiet" ]; then setterm -clear; fi
    source "$GMS_SHELL/l_gms" -goodbye
# Remove temporary files:
if [ "$1" = "-Q" -o "$1" = "-.Q" -o \
"$1" = "/Q" -o "$1" = "Q" -o \
"$1" = "QQ" ]; then
    if [ -f ./*.log ]; then rm ./*.log 2> ./nul; fi
    if [ -f ./*.ok ]; then rm ./*.ok 2> ./nul; fi
    if [ -f ./*.err ]; then rm ./*.err 2> ./nul; fi
    if [ -f ./nul ]; then rm ./nul; fi
fi
# Return to startup folder:
if [ -d "$GMS_STARTUP" ]; then cd "$GMS_STARTUP"; fi
source "$GMS_SHELL/g_vars" -write_memo
# Restore environment:
source "$GMS_SHELL/g_vars" -clear_called_by gms -g

# Create: -----

elif [ "$1" = "-c" -o "$1" = "-C" -o \
"$1" = "-.c" -o "$1" = "-.C" -o \
"$1" = "/c" -o "$1" = "/C" -o \
"$1" = "c" -o "$1" = "C" -o \
"$1" = "cc" -o "$1" = "CC" ]; then
    # $2: new file name, $3: template file name
    # Get base name of file:
    mybase=$(basename $3 ".htm")
    mybase=$(basename $mybase ".html")
    # Check if there is an old file with that name ($2):
    if [ -f "$GMS_PWD/$2" ]; then
        export GMS_MESSAGE="GMS error: There is already such a file $2"
        source "$GMS_SHELL/l_gms" -help called_by gms -c
        sleep 2
    # Look for template in current directory and copy it:
    elif [ -f "$GMS_PWD/$3" ]; then
        cp "$GMS_PWD/$3" "$GMS_PWD/$2"
    elif [ -f "$GMS_PWD/$mybase.htm" ]; then
        cp "$GMS_PWD/$mybase.htm" "$GMS_PWD/$2"
    elif [ -f "$GMS_PWD/$mybase.html" ]; then
        cp "$GMS_PWD/$mybase.html" "$GMS_PWD/$2"
    # Look for template in template main directory and copy it:
    elif [ -f "$GMS_TEMPLATE/$3" ]; then
        cp "$GMS_TEMPLATE/$3" "$GMS_PWD/$2"
    elif [ -f "$GMS_TEMPLATE/$mybase.htm" ]; then
        cp "$GMS_TEMPLATE/$mybase.htm" "$GMS_PWD/$2"
    elif [ -f "$GMS_TEMPLATE/$mybase.html" ]; then
        cp "$GMS_TEMPLATE/$mybase.html" "$GMS_PWD/$2"
    fi
fi
```

```

cp "$GMS_TEMPLATE/$mybase.html" "$GMS_PWD/$2"
# Look for template in template subdirectory and copy it:
elif [ -f "$GMS_TEMPLATE/$mybase/$3" ]; then
cp "$GMS_TEMPLATE/$mybase/$3" "$GMS_PWD/$2"
elif [ -f "$GMS_TEMPLATE/$mybase/$mybase.htm" ]; then
cp "$GMS_TEMPLATE/$mybase/$mybase.htm" "$GMS_PWD/$2"
elif [ -f "$GMS_TEMPLATE/$mybase/$mybase.html" ]; then
cp "$GMS_TEMPLATE/$mybase/$mybase.html" "$GMS_PWD/$2"
else
export GMS_MESSAGE="GMS: Creating default file - got no template $3"
source "$GMS_SHELL/l_gms" -help called_by gms -c
sleep 2
cp "$GMS_TEMPLATE/default/default.htm" "$GMS_PWD/$2"
fi; unset myfolder
if [ -f "$GMS_PWD/$2" ]; then
export GMS_FILE="$2"; export GMS_FOLDER=$GMS_PWD
source "$GMS_SHELL/g_vars" -write_memo called_by gmssetup
fi

# Open: -----
elif [ "$1" = "-o" -o "$1" = "-O" -o \
"$1" = "--o" -o "$1" = "--O" -o \
"$1" = "/o" -o "$1" = "/O" -o \
"$1" = "o" -o "$1" = "O" -o \
"$1" = "oo" -o "$1" = "OO" ]; then
shift
source "$GMS_SHELL/g_vars" -ini_file $1 $2 $3 $4 $5 $6 $7 $8 $9
source "$GMS_SHELL/l_gms" -help called_by gms -o

# View: -----
elif [ "$1" = "-v" -o "$1" = "-V" -o \
"$1" = "--v" -o "$1" = "--V" -o \
"$1" = "/v" -o "$1" = "/V" -o \
"$1" = "v" -o "$1" = "V" -o \
"$1" = "vv" -o "$1" = "VV" ]; then
export GMS_COMMAND=$GMS_VIEWER
source "$GMS_SHELL/gms" -execute called_by gms -v

# Edit: -----
elif [ "$1" = "-e" -o "$1" = "-E" -o \
"$1" = "--e" -o "$1" = "--E" -o \
"$1" = "/e" -o "$1" = "/E" -o \
"$1" = "e" -o "$1" = "E" -o \
"$1" = "ee" -o "$1" = "EE" ]; then
export GMS_PROGRAM="text editor"
export GMS_COMMAND=$GMS_EDITOR
source "$GMS_SHELL/gms" -execute called_by gms -e

# Browse: -----
elif [ "$1" = "-b" -o "$1" = "-B" -o \
"$1" = "--b" -o "$1" = "--B" -o \
"$1" = "/b" -o "$1" = "/B" -o \
"$1" = "b" -o "$1" = "B" -o \
"$1" = "bb" -o "$1" = "BB" ]; then
export GMS_PROGRAM="markup file browser"
export GMS_COMMAND=$GMS_BROWSER
source "$GMS_SHELL/gms" -execute called_by gms -b

# Analyse file or all files in folder: -----
elif [ "$1" = "-a" -o "$1" = "-A" -o \
"$1" = "--a" -o "$1" = "--A" -o \
"$1" = "/a" -o "$1" = "/A" -o \
"$1" = "a" -o "$1" = "A" -o \
"$1" = "aa" -o "$1" = "AA" ]; then
# Get base name of file:
if [ "$GMS_PERL" != "1" ]; then ###
export GMS_BASE=$(basename $GMS_FILE ".htm")
export GMS_BASE=$(basename $GMS_BASE ".html")
fi
if [ "$GMS_MODE" != "quiet" ]; then
echo " GMS_FOLDER=$GMS_FOLDER" >> "$Z"
echo " GMS_FILE=$GMS_FILE" >> "$Z"
echo " GMS_BASE=$GMS_BASE" >> "$Z"
fi
export GMS_PROGRAM="markup syntax checker"
export GMS_COMMAND=$GMS_VIEWER
if [ "$Z" != "" ]; then shift; fi
slashline="////////////////////////////////////"
# If no file is specified, check folder:
if [ "$GMS_FILE" = "" ]; then export GMS_FILE="_folder.err"; fi
if [ "$GMS_FILE" = "_folder.err" ]; then
# Process folder:
echo " $GMS_FILE - GMS: Folder check. Running $GMS_PROGRAM ..." \
> _folder.err
echo "$slashline$slashline////////////////////////////////////" >> _folder.err
echo >> _folder.err
dashline="-----"
for i in *.htm*; do # Analyse kernel:
"$GMS_BINARIES/$GMS_ANALYST" \
-config "$TEXTMFCNF/tidy.cfg" -f $i.er_ $i > $i.ok
echo " $i"
echo " _____$dashline$dashline"
echo " $i" >> _folder.err
echo " _____$dashline$dashline" >> _folder.err
cat $i.er_
cp _folder.err folder.er_
cat folder.er_ $i.er_ > _folder.err
echo >> _folder.err
if [ -f folder.er_ ]; then rm folder.er_; fi
if [ -f $i.er_ ]; then rm $i.er_; fi
if [ -f $i.ok ]; then rm $i.ok; fi
done; unset dashline
echo " $slashline$slashline////////////////////////////////////" >> _folder.err
echo " $GMS_FOLDER" >> _folder.err
else
export backup_file=$GMS_FILE; export backup_base=$GMS_BASE
if [ "$GMS_MODE" != "quiet" ]; then
source "$GMS_SHELL/l_gms" -desktop -draw called_by gms -a
fi; export GMS_FILE=$backup_file
# Run HTML Tidy:
if [ "$GMS_DEBUG" = "Z" -a "$GMS_MODE" != "quiet" ]; then
echo " GMS_ANALYST=$GMS_ANALYST" >> "$Z"; fi
if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
if [ -f "$GMS_BASE.err" ]; then rm "$GMS_BASE.err"; fi
"$GMS_BINARIES/$GMS_ANALYST" -config "$GMS_SETTING/tidy.cfg" \
-f $GMS_BASE.tmp "$GMS_FILE" > "$GMS_BASE.ok"
export GMS_FILE="$GMS_BASE.err"
if [ "$GMS_MODE" != "quiet" ]; then
echo " $GMS_FILE - GMS: Running $GMS_PROGRAM ..." > errhead.err
echo " $slashline$slashline////////////////////////////////////" >> errhead.err
if [ -f errfoot.err ]; then rm errfoot.err; fi
# Extra empty lines as cosmetics for "no error" results:
for i in 1 2 3 4 5 6 7 8 9 10 11 12; do echo >> errfoot.err; done
echo " $slashline$slashline////////////////////////////////////" >> errfoot.err
echo " $GMS_FOLDER" >> errfoot.err
for i in 1 2 3 4 5 6 7 8 9 10 11 12; do echo >> errfoot.err; done
for i in 1 2 3 4 5 6 7 8 9 10 11 12; do echo >> errfoot.err; done
for i in 1 2 3 4 5 6 7 8 9 10 11 12; do echo >> errfoot.err; done
for i in 1 2 3 4 5 6 7 8 9 10 11 12; do echo >> errfoot.err; done
for i in 1 2 3 4 5 6 7 8 9 10 11 12; do echo >> errfoot.err; done
cat errhead.err "$GMS_BASE.tmp" errfoot.err > "$GMS_BASE.err"
fi; unset slashline
if [ "$GMS_MODE" == "quiet" ]; then mv "$GMS_BASE.tmp" "$GMS_BASE.err"; fi
if [ -f "$GMS_BASE.pmt" ]; then rm "$GMS_BASE.pmt"; fi
if [ -f "$GMS_BASE.tmp" ]; then rm "$GMS_BASE.tmp"; fi
if [ -f errhead.err ]; then rm errhead.err; fi

```

```

if [ -f errfoot.err ]; then rm errfoot.err; fi
fi; source "$GMS_SHELL/gms" -execute called_by gms -a

# Typeset: -----
elif [ "$1" = "-t" -o "$1" = "-T" -o \
"$1" = "--t" -o "$1" = "--T" -o \
"$1" = "/t" -o "$1" = "/T" -o \
"$1" = "t" -o "$1" = "T" -o \
"$1" = "tT" -o "$1" = "TT" ]; then
cd "$GMS_FONTS/tfm"
if [ "$GMS_MODE" != "quiet" ]; then
if [ ! -f *.tfm ]; then
cd -
if [ "$GMS_MODE" != "quiet" ]; then setterm -clear; fi
mymsg="GMS error: TeX font metrics not found. Please 'Write font map'."
echo "$mymsg"
cat "$GMS_SETTING/desktop.scn"
echo -n "$GMS_FONTS/tfm"
if [ "$GMS_DEBUG" = "Z" ]; then
echo "$mymsg ($GMS_FOLDER/$GMS_FILE)." >> "$Z"
fi
unset mymsg
sleep 2
exit
elif [ ! -f "$GMS_SETTING/font.map" ]; then
cd -
if [ "$GMS_MODE" != "quiet" ]; then setterm -clear; fi
mymsg="GMS error: Missing font information. Please 'Write font map'."
echo "$mymsg"
cat "$GMS_SETTING/desktop.scn"
echo -n "$GMS_SETTING"
if [ "$GMS_DEBUG" = "Z" ]; then
echo "$mymsg ($GMS_FOLDER/$GMS_FILE)." >> "$Z"
fi
unset mymsg
sleep 2
exit
fi
fi
cd -
# Get base name of file:
if [ "$GMS_PERL" != "1" ]; then ###
export GMS_BASE=$(basename $GMS_FILE ".htm")
export GMS_BASE=$(basename $GMS_BASE ".html")
fi
export backup_file=$GMS_FILE
export backup_base=$GMS_BASE
export GMS_PROGRAM="$GMS_TSETTER engine"
export GMS_COMMAND="$GMS_VIEWER"
export GMS_FILE="$GMS_BASE.log"
slashline="////////////////////////////////////"
if [ "$GMS_MODE" != "quiet" ]; then
setterm -clear
echo " $GMS_FILE - GMS: Running $GMS_PROGRAM ..."
echo " $slashline$slashline////////////////////////////////////"
fi
export GMS_FILE=$backup_file
export TEXINPUTS="./:$TEXMFCNF:$TEX_USER//"
export WEB2C="$GMS_SETTING"
if [ "$GMS_DEBUG" = "Z" -a "$GMS_MODE" != "quiet" ]; then
echo " GMS_TSETTER=$GMS_TSETTER" >> "$Z"
echo " GMS_FOLDER=$GMS_FOLDER" >> "$Z"
echo " GMS_FILE=$GMS_FILE" >> "$Z"
echo " GMS_BASE=$GMS_BASE" >> "$Z"
echo " TEXINPUTS=$TEXINPUTS" >> "$Z"
fi
if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
export BEFORE=$(date +%s)
"$GMS_BINARIES/$GMS_TSETTER" -prognome=gerolf "$GMS_FILE" 2> gmserr.log
get_runtime

```

```

unset TEXINPUTS; unset WEB2C
export GMS_FILE="$GMS_BASE.log"
if [ "$GMS_MODE" = "quiet" ]; then
cat "$GMS_BASE.log" gmserr.log > "$GMS_BASE.tmp"
if [ -f "$GMS_BASE.tmp" ]; then mv "$GMS_BASE.tmp" "$GMS_FILE"; fi
else
echo " $GMS_FILE - GMS: Running $GMS_PROGRAM ..." > loghd.log
echo " $slashline$slashline////////////////////////////////////" >> loghd.log
if [ -f "readme.txt" ]; then cat "readme.txt" >> loghd.log
elif [ -f "Readme.txt" ]; then cat "Readme.txt" >> loghd.log
elif [ -f "README.TXT" ]; then cat "README.TXT" >> loghd.log; fi
echo "GMS run time: $GMS_RUNTIME" > logft.log
echo " $slashline$slashline////////////////////////////////////" >> logft.log
echo " $GMS_FOLDER" >> logft.log
for i in 1 2 3 4 5 6 7 8 9 10; do echo >> logft.log; done
if [ -f "$GMS_FILE" ]; then mv "$GMS_FILE" "$GMS_BASE.pmt"; fi
# pdfTeX header cosmetics (insert newline):
export AFTER=$(date)
sed "/(format=gerolf.*$/;s/;/XyZzy/;/^$/d;s/< /</g;s/ >/>/g" \
< "$GMS_BASE.pmt" > "$GMS_BASE.pmu"
sed "s/(format=gerolf.*$/;s/;/XyZzy/(format=gerolf) $AFTER/" \
< "$GMS_BASE.pmu" > "$GMS_BASE.tmp"
unset AFTER
cat loghd.log "$GMS_BASE.tmp" logft.log gmserr.log > "$GMS_FILE"
if [ -f "$GMS_BASE.pmt" ]; then rm "$GMS_BASE.pmt"; fi
if [ -f "$GMS_BASE.pmu" ]; then rm "$GMS_BASE.pmu"; fi
if [ -f "$GMS_BASE.tmp" ]; then rm "$GMS_BASE.tmp"; fi
if [ -f loghd.log ]; then rm loghd.log; fi
if [ -f logft.log ]; then rm logft.log; fi
if [ -f gmserr.log ]; then rm gmserr.log; fi
export GMS_BASE=$GMS_LOGBASE
source "$GMS_SHELL/gms" -execute called_by gms -t
fi
unset slashline

# Initialize TeX format: -----
elif [ "$1" = "-i" -o "$1" = "-I" -o \
"$1" = "--i" -o "$1" = "--I" -o \
"$1" = "/i" -o "$1" = "/I" -o \
"$1" = "i" -o "$1" = "I" -o \
"$1" = "iI" -o "$1" = "II" ]; then
mymode=$2
export backup_folder=$GMS_FOLDER
export backup_file=$GMS_FILE
export backup_base=$GMS_BASE
export GMS_FOLDER=$TEXFORMATS
export GMS_FILE=gerolf.log
export GMS_BASE=gerolf
export GMS_SHORT=gerolf.log
export GMS_PROGRAM="$GMS_TSETTER engine"
export GMS_COMMAND=$GMS_VIEWER
export GEROLF="*gerolf"
if [ "$GMS_TSETTER" = "tex" ]; then export GEROLF="gerolf"
elif [ "$GMS_TSETTER" = "pdftex" ]; then export GEROLF="gerolf"; fi
export TEXINPUTS="$GMS_ROOT/data/:$TEX_BASE/:$TEXMFCNF:$GMS_BINARIES"
export WEB2C="$GMS_SETTING"
if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
slashline="////////////////////////////////////"
if [ "$mymode" = "-quiet" ]; then
source "$GMS_SHELL/g Plug" -build -quiet called_by gms -i
else
if [ "$GMS_MODE" != "quiet" ]; then setterm -clear; fi
echo " $GMS_FILE - GMS: Running $GMS_PROGRAM ..."
echo " $slashline$slashline////////////////////////////////////"
source "$GMS_SHELL/g Plug" -build called_by gms -i
fi
echo " $GMS_FILE - GMS: Running $GMS_PROGRAM ..." > gmshd.log
echo " $slashline$slashline////////////////////////////////////" >> gmshd.log
export BEFORE=$(date +%s)

```



```

if [ "$mymode" = "-quiet" ]; then
  echo "\dump" | "$GMS_BINARIES/$GMS_TSETTER" \
    -ini $GEROLF > nil 2> gmserr.log
  if [ -f nil ]; then rm nil; fi
else
  echo "\dump" | "$GMS_BINARIES/$GMS_TSETTER" \
    -ini $GEROLF 2> gmserr.log
fi
get_runtime
echo -n "GMS run time: $GMS_RUNTIME" > gmsft.log
echo >> gmsft.log
echo "$\slashline$\slashline////////////////////////////////////" >> gmsft.log
echo "$GMS_FOLDER" >> gmsft.log
unset slashline
if [ -f gerolf.log ]; then mv gerolf.log gms.pmt; fi
sed "s/^\.openout.*$/;s/^\system.*$/;/^\$/d" < gms.pmt > gms.pmu
# pdfTeX header cosmetics (insert newline):
export AFTER=$(date)
sed "/(INITE\X.*$/;s/;/XyZy/;)" \
  < gms.pmu > gms.pmx
sed "s/(INITE\X.*$/;s/XyZy/(INITE\X) \$AFTER/" \
  < gms.pmx > gms.tmp
unset AFTER
cat gmsd.log gms.tmp gmserr.log gmsft.log > gerolf.log
# Remove temporary files:
if [ -f gmsd.log ]; then rm gmsd.log; fi
if [ -f gms.pmt ]; then rm gms.pmt; fi
if [ -f gms.pmu ]; then rm gms.pmu; fi
if [ -f gms.pmx ]; then rm gms.pmx; fi
if [ -f gms.tmp ]; then rm gms.tmp; fi
if [ -f gmserr.log ]; then rm gmserr.log; fi
if [ -f gmsft.log ]; then rm gmsft.log; fi
if [ -f "$GMS_BINARIES/font.lst" ]; then
  rm "$GMS_BINARIES/font.lst"; fi
if [ -f "$GMS_BINARIES/sortfont.lst" ]; then
  rm "$GMS_BINARIES/sortfont.lst"; fi
if [ -f "$GMS_BINARIES/entity.lst" ]; then
  rm "$GMS_BINARIES/entity.lst"; fi
if [ -f "$GMS_BINARIES/sortent.lst" ]; then
  rm "$GMS_BINARIES/sortent.lst"; fi
# Move .log, .enc and .htm files to appropriate locations:
if [ -f gerolf.log ]; then cp gerolf.log "$GMS_SETTING"; fi
if [ -f gerolf.log ]; then rm gerolf.log; fi
if [ -f register.log ]; then cp register.log "$GMS_SETTING"; fi
if [ -f register.log ]; then rm register.log; fi
for i in *.enc; do if [ -f $i ]; then
  cp $i "$GMS_ROOT/data/enc/"; rm $i; fi; done
unset TEXINPUTS; unset WEB2C
export GMS_FOLDER="$GMS_SETTING"
export GEROLF=
if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
source "$GMS_SHELL/g_plug" -remove called_by gms -i
if [ "$mymode" != "-quiet" ]; then
  source "$GMS_SHELL/gms" -execute called_by gms -w
fi; unset mymode

# Read output file: -----
elif [ "$1" = "-r" -o "$1" = "-R" -o \
"$1" = "-.r" -o "$1" = "-.R" -o \
"$1" = "/r" -o "$1" = "/R" -o \
"$1" = "r" -o "$1" = "R" -o \
"$1" = "rr" -o "$1" = "RR" ]; then
# Get base name of file:
export GMS_BASE=$(basename $GMS_FILE ".htm")
export GMS_BASE=$(basename $GMS_BASE ".html")
export backup_file=$GMS_FILE
export GMS_FILE="$GMS_BASE.pdf"
export GMS_PROGRAM="portable document file reader"
export GMS_COMMAND=$GMS_READER

source "$GMS_SHELL/gms" -execute called_by gms -r

# Learn: -----
elif [ "$1" = "-l" -o "$1" = "-L" -o \
"$1" = "-.l" -o "$1" = "-.L" -o \
"$1" = "/l" -o "$1" = "/L" -o \
"$1" = "l" -o "$1" = "L" -o \
"$1" = "ll" -o "$1" = "LL" ]; then
if [ -f "$GMS_TEMPLATE/handbook/handbook.htm" ]; then
  export GMS_FOLDER="$GMS_TEMPLATE/handbook"
  export GMS_FILE="handbook.htm"
  export GMS_MESSAGE="handbook.htm"
  export GMS_MESSAGE="$GMS_MESSAGE Gerolf Markup Shredder Handbook"
  cd "$GMS_FOLDER"
  source "$GMS_SHELL/g_vars" -write_memo called_by gmssetup
fi; source "$GMS_SHELL/l_gms" -help called_by gms -l

# Select: -----
elif [ "$1" = "-s" -o "$1" = "-S" -o \
"$1" = "-.s" -o "$1" = "-.S" -o \
"$1" = "/s" -o "$1" = "/S" -o \
"$1" = "s" -o "$1" = "S" -o \
"$1" = "ss" -o "$1" = "SS" ]; then
if [ -f "$GMS_SETTING/gerolf" ]; then
  export GMS_FILE="gerolf"; export GMS_FOLDER=$GMS_SETTING
  export GMS_MESSAGE="gerolf: The GMS launcher script. Here"
  export GMS_MESSAGE="$GMS_MESSAGE you can edit the programs"
  export GMS_MESSAGE="$GMS_MESSAGE to use via GMS "
  cd "$GMS_SETTING"
  source "$GMS_SHELL/g_vars" -write_memo called_by gmssetup
fi; source "$GMS_SHELL/l_gms" -help called_by gms -s

# Write font map: -----
elif [ "$1" = "-w" -o "$1" = "-W" -o \
"$1" = "-.w" -o "$1" = "-.W" -o \
"$1" = "/w" -o "$1" = "/W" -o \
"$1" = "w" -o "$1" = "W" -o \
"$1" = "ww" -o "$1" = "WW" ]; then
export backup_folder=$GMS_FOLDER
export backup_file=$GMS_FILE
export backup_base=$GMS_BASE
export GMS_FOLDER=$GMS_SETTING
export GMS_FILE=font.map
export GMS_BASE=font
export GMS_SHORT=font.map
export GMS_COMMAND=$GMS_VIEWER
export GMS_PROGRAM="font installer"
mymode=$2
if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
source "$GMS_SHELL/g_font" -build_all $mymode 2> gmserr.log
if [ -f gmserr.log ]; then
  sed "/^\$/d; s/^\ \$/; s/^\ \$/" < gmserr.log >> font.log
  rm gmserr.log
fi
slashline="////////////////////////////////////"
echo "% $\slashline$\slashline" >> font.log
echo "% $GMS_FOLDER" >> font.log
unset slashline
if [ "$mymode" != "-quiet" ]; then
  source "$GMS_SHELL/gms" -execute called_by gms -w
fi; unset mymode

# Execute program: -----
elif [ $1 = -execute ]; then
# $2: program type, $3: binary name, $4: file name
if [ -f "$GMS_FOLDER/$GMS_FILE" ]; then

```

```

if [ "$GMS_MODE" != "quiet" -a "$?" != "-quiet" ]; then
# Contribute to gmsdebug.log:
if [ "$GMS_DEBUG" = "Z" ]; then
    echo "  GMS_COMMAND=$GMS_COMMAND" >> "$Z"
    echo "  GMS_FOLDER=$GMS_FOLDER" >> "$Z"
    echo "  GMS_FILE=$GMS_FILE" >> "$Z"
fi
# Execute command:
source "$GMS_SHELL/l_gms" -desktop -draw called_by gms -execute
"$GMS_COMMAND" "$GMS_FOLDER/$GMS_FILE"
source "$GMS_SHELL/l_gms" -desktop -draw called_by gms -execute
fi
else
if [ "$GMS_MODE" != "quiet" ]; then setterm -clear; fi
echo " $GMS_FILE - GMS error: File not found"
cat "$GMS_SETTING/desktop.scn"
echo -n " $GMS_FOLDER"
if [ "$GMS_DEBUG" = "Z" ]; then
    mymsg=" ! GMS error: File not found"
    echo "$mymsg ($GMS_FOLDER/$GMS_FILE)." >> "$Z"
    unset mymsg
fi
sleep 2
fi

export GMS_COMMAND=
export GMS_PROGRAM=
if [ "$backup_base" != "" ]; then export GMS_BASE=$backup_base; fi
if [ "$backup_file" != "" ]; then export GMS_FILE=$backup_file; fi
if [ "$backup_folder" != "" ]; then export GMS_FOLDER=$backup_folder; fi
if [ "$backup_folder" != "" ]; then
    if [ -d "$GMS_FOLDER" ]; then cd "$GMS_FOLDER"; fi
fi
export backup_base=
export backup_file=
export backup_folder=
source "$GMS_SHELL/l_gms" -help

# Show error: -----
else
if [ ! -d "$GMS_FOLDER/$GMS_FILE" ]; then
    export GMS_MESSAGE="GMS error: No file or command line option $1"
    source "$GMS_SHELL/l_gms" -help
    sleep 1
    if [ "$GMS_DEBUG" = "Z" ]; then echo " $GMS_MESSAGE" >> "$Z"; fi
fi
exit
fi
fi

```

I_banner

```
#!/bin/sh
# l_banner
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_BANNER=20060927

# Prologue: =====

# Debug: Banners are out of the X/Y-debugging system (which uses banners) ...

# Size:

BACKUP_SIZE=$REPLY_SIZE
REPLY_SIZE=78

BACKUP_OFFSET=$REPLY_OFFSET
REPLY_OFFSET=0

# Chapters: =====

# Empty - show error:

if [ "$1" = "" ]; then
    source ./l_banner -help
    exit
# Debug: -----

elif [ $1 = -debug ]; then
# Upper:
    REPLY_OFFSET=0
    REPLY_SIZE=23
    "$GMS_REPLY" -banner 2 "Module: $2"
    REPLY_OFFSET=20
    REPLY_SIZE=23
    "$GMS_REPLY" -banner 2 "Action: $3"
    REPLY_OFFSET=40
    REPLY_SIZE=23
    "$GMS_REPLY" -banner 2 "Hotkey: $4"
    REPLY_OFFSET=60
    REPLY_SIZE=22
    "$GMS_REPLY" -banner 2 "Coldkey: $5"
# Lower:
    REPLY_OFFSET=0
    REPLY_SIZE=23
    "$GMS_REPLY" -banner 23 "Offset: $BACKUP_OFFSET"
    REPLY_OFFSET=20
    REPLY_SIZE=23
    "$GMS_REPLY" -banner 23 "Size: $BACKUP_SIZE"
    REPLY_OFFSET=40
    REPLY_SIZE=23
    "$GMS_REPLY" -banner 23 "Items: $REPLY_ITEMS"
    REPLY_OFFSET=60
    REPLY_SIZE=22
    "$GMS_REPLY" -banner 23 "Data: $6"
    DEBUG=
    if [ "$GMS_DEBUG" = "Y" ]; then sleep 1; fi

# Diverse: -----

# No module:
elif [ $1 = -no_module ]; then
    "$GMS_REPLY" -banner 1 "GMS error: No module $2"
    REPLY_OFFSET=57

REPLY_SIZE=25
"$GMS_REPLY" -banner 1 "Gerolf Markup Shredder "
sleep 1
break

# No action:
elif [ $1 = -no_action ]; then
    "$GMS_REPLY" -banner 1 "GMS error: No action $3"
    REPLY_OFFSET=57
    REPLY_SIZE=25
    "$GMS_REPLY" -banner 1 "Module: $2 "
    sleep 1

# No hot:
elif [ $1 = -no_hot ]; then
    "$GMS_REPLY" -banner 1 "GMS error: No hotkey $3"
    REPLY_OFFSET=57
    REPLY_SIZE=25
    "$GMS_REPLY" -banner 1 "Module: $2 "
    sleep 1

# No cold:
elif [ $1 = -no_cold ]; then
    "$GMS_REPLY" -banner 1 "GMS error: No coldkey $3"
    REPLY_OFFSET=57
    REPLY_SIZE=25
    "$GMS_REPLY" -banner 1 "Module: $2 "

# No file:
elif [ $1 = -no_file ]; then
    "$GMS_REPLY" -banner 1 "GMS error: No file."
    REPLY_OFFSET=57
    REPLY_SIZE=25
    "$GMS_REPLY" -banner 1 "Module: $2 "

# Upper: -----

elif [ $1 = -upper ]; then

# First upper banner:
if [ $2 = -first ]; then
    "$GMS_REPLY" -banner 1 "Gerolf Markup Shredder $GMS_VERSION"
    REPLY_OFFSET=54
    REPLY_SIZE=28
    "$GMS_REPLY" -banner 1 "MarkupShredder@Gerolf.org"

# Last upper banner:
elif [ $2 = -last ]; then
    "$GMS_REPLY" -banner 1 "MarkupShredder@Gerolf.org"
    REPLY_OFFSET=65
    REPLY_SIZE=16
    "$GMS_REPLY" -banner 1 "www.Gerolf.org"

# Write handbook title on upper banner:
elif [ $2 = -handbook ]; then
    "$GMS_REPLY" -banner 1 "handbook.html"
    REPLY_OFFSET=44
    REPLY_SIZE=38
    "$GMS_REPLY" -banner 1 "The Gerolf Markup Shredder Handbook"

# Write current file on upper banner:
elif [ $2 = -file ]; then
    "$GMS_REPLY" -banner 1 "$GMS_FILE"
    REPLY_OFFSET=57
    REPLY_SIZE=25
    "$GMS_REPLY" -banner 1 "Gerolf Markup Shredder "
```

```

# Write current template on upper banner:
elif [ $2 = -template ]; then
    "$GMS_REPLY" -banner 1 "Template: $GMS_TEMPLATEFILE"
    REPLY_OFFSET=57
    REPLY_SIZE=25
    "$GMS_REPLY" -banner 1 "Gerolf Markup Shredder "
fi

# Lower: .....

elif [ $1 = -lower ]; then

# Write domain on lower banner:
if [ $2 = -domain ]; then
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org "

# First lower banner:
elif [ $2 = -first ]; then
    "$GMS_REPLY" -banner 24 "The typesetting program that uses TeX for"
    REPLY_OFFSET=43
    REPLY_SIZE=39
    "$GMS_REPLY" -banner 24 "document conversion from HTML to PDF "

# Write current folder on lower banner:
elif [ $2 = -folder ]; then
    "$GMS_REPLY" -banner 24 "$GMS_FOLDER"
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org "

# Write warning for select dialog
elif [ $2 = -select ]; then
    "$GMS_REPLY" -banner 24 "Warning: Changes done here will be written"
    REPLY_OFFSET=44
    REPLY_SIZE=38
    "$GMS_REPLY" -banner 24 "to the GMS launcher script 'gerolf' "

# Write help text for menu animation:
elif [ $2 = -animate ]; then
    "$GMS_REPLY" -banner 24 "Menu animation is currently turned $GMS_ANIMATE"
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org "

# Write help text for debugging:
elif [ $2 = -debug ]; then
    "$GMS_REPLY" -banner 24 "X: trace modules, Y: slow motion, Z: write log,"
    REPLY_OFFSET=49
    REPLY_SIZE=32
    "$GMS_REPLY" -banner 24 "0: no debug.   www.Gerolf.org "

# Write help text for encoding:
elif [ $2 = -encode ]; then
    "$GMS_REPLY" -banner 24 "ISO-8859-1 Western, ISO-8859-2 Central,"
    REPLY_OFFSET=41
    REPLY_SIZE=41
    "$GMS_REPLY" -banner 24 "ISO-8859-5 Cyrillic, ISO-8859-6 Arabic "

# Write help text for color dialog:
elif [ $2 = -color ]; then
    "$GMS_REPLY" -banner 24 "Enter a color or ASCII letter number"
    REPLY_OFFSET=38
    REPLY_SIZE=44
    "$GMS_REPLY" -banner 24 "(or 'R' for random changes of appearance) "

# Write help text for codepage:
elif [ $2 = -codepage ]; then
    "$GMS_REPLY" -banner 24 "Show terminal codepage"

REPLY_OFFSET=65
REPLY_SIZE=17
"$GMS_REPLY" -banner 24 "www.Gerolf.org"

# Write help text for codepage (upper half):
elif [ $2 = -cp_up ]; then
    "$GMS_REPLY" -banner 24 "Terminal codepage (upper half) - In HTML,"
    REPLY_OFFSET=43
    REPLY_SIZE=39
    "$GMS_REPLY" -banner 24 "write '&#255;' for character no. 255 " XX

# Write help text for codepage (lower half):
elif [ $2 = -cp_lo ]; then
    "$GMS_REPLY" -banner 24 "American Standard Code for Information Interchange"
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org " XX

# Write help text for programs:
elif [ $2 = -program ]; then
    "$GMS_REPLY" -banner 24 "Select programs to associate with GMS menu items"
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org "

# Write help text for viewer:
elif [ $2 = -viewer ]; then
    "$GMS_REPLY" -banner 24 "Text viewers: more, less, xless, kless, ..."
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org "

# Write help text for editor:
elif [ $2 = -editor ]; then
    "$GMS_REPLY" -banner 24 "Text editors: nedit, kwrite, emacs, gedit, vi, ..."
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org "

# Write help text for browser:
elif [ $2 = -browser ]; then
    "$GMS_REPLY" -banner 24 "HTML browsers: lynx, links, mozilla,"
    REPLY_OFFSET=37
    REPLY_SIZE=45
    "$GMS_REPLY" -banner 24 "netscape, opera, konqueror, amaya, ..."

# Write help text for analyst:
elif [ $2 = -analyst ]; then
    "$GMS_REPLY" -banner 24 "HTML syntax checker: tidy"
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org "

# Write help text for typesetter:
elif [ $2 = -tsetter ]; then
    "$GMS_REPLY" -banner 24 "HTML typesetters: pdftex, pdftex, etex, tex"
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org "

# Write help text for reader:
elif [ $2 = -reader ]; then
    "$GMS_REPLY" -banner 24 "PDF readers: acroread, gv, kpdf, (k)ghostview,
xpdf, ..."
    REPLY_OFFSET=65
    REPLY_SIZE=17
    "$GMS_REPLY" -banner 24 "www.Gerolf.org "
fi

# Error: .....

```

```
else
  echo " This is Gerolf Markup Shredder: No 1_banner $1 $2 "
  sleep 3
fi
```

```
REPLY_OFFSET=$BACKUP_OFFSET
BACKUP_OFFSET=
REPLY_SIZE=$BACKUP_SIZE
BACKUP_SIZE=
```

```
# Epilogue: =====
```

I_box

```
#!/bin/sh

# l_box
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_BOX=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_box)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_box $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " l_box ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Box cycle: -----

function l_box_cycle {
    if [ $1 = -build ]; then
        if [ "$2" != "0" ]; then "$GMS_REPLY" -top $2; fi
        if [ "$3" != "0" ]; then "$GMS_REPLY" -item $3; fi
        if [ "$4" != "0" ]; then "$GMS_REPLY" -shadow $4; fi
        if [ "$5" != "0" ]; then "$GMS_REPLY" -bottom $5; fi
        if [ "$6" != "0" ]; then "$GMS_REPLY" -item $6; fi
    elif [ $1 = -remove ]; then
        if [ "$2" != "0" ]; then "$GMS_REPLY" -bottom $2; fi
        if [ "$3" != "0" ]; then "$GMS_REPLY" -shadow $3; fi
        if [ "$4" != "0" ]; then
            "$GMS_REPLY" -stripe $4 $REPLY_PATTERN $REPLY_LETTER
        fi
        if [ "$5" != "0" ]; then "$GMS_REPLY" -top $5; fi
        if [ "$6" != "0" ]; then
            "$GMS_REPLY" -stripe $6 $REPLY_PATTERN $REPLY_LETTER
        fi
    fi; }

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_box

# Idle: -----

elif [ $1 = -idle ]; then
    "$GMS_REPLY" -item 4 " " " "

# Build: =====

# Box, y-offset = 13:

elif [ $1 = -build ]; then
    if [ "$GMS_ANIMATE" = "on" ]; then
        #
        l_box_cycle -build 12 0 13 0 0
        l_box_cycle -build 11 12 14 13 12
        l_box_cycle -build 10 11 15 14 13
        l_box_cycle -build 9 10 16 15 14
        l_box_cycle -build 8 9 17 16 15
        l_box_cycle -build 7 8 18 17 16
        l_box_cycle -build 6 7 19 18 17
        l_box_cycle -build 5 6 20 19 18
        l_box_cycle -build 4 5 21 20 19
        l_box_cycle -build 3 4 22 21 20
    fi

# Top box, y-offset = 12: -----

elif [ $1 = -t12_build ]; then
    if [ "$GMS_ANIMATE" = "on" ]; then
        #
        l_box_cycle -build 11 0 12 0 0
        l_box_cycle -build 10 11 13 12 11
        l_box_cycle -build 9 10 14 13 12
        l_box_cycle -build 8 9 15 14 13
        l_box_cycle -build 7 8 16 15 14
        l_box_cycle -build 6 7 17 16 15
        l_box_cycle -build 5 6 18 17 16
        l_box_cycle -build 4 5 19 18 17
        l_box_cycle -build 3 4 20 19 18
    fi

# Top box, y-offset = 11: -----

elif [ $1 = -t11_build ]; then
    if [ "$GMS_ANIMATE" = "on" ]; then
        #
        l_box_cycle -build 10 0 11 0 0
        l_box_cycle -build 9 10 12 11 10
        l_box_cycle -build 8 9 13 12 11
        l_box_cycle -build 7 8 14 13 12
        l_box_cycle -build 6 7 15 14 13
        l_box_cycle -build 5 6 16 15 14
        l_box_cycle -build 4 5 17 16 15
        l_box_cycle -build 3 4 18 17 16
    fi

# Top box, y-offset = 8: -----

elif [ $1 = -t8_build ]; then
    if [ "$GMS_ANIMATE" = "on" ]; then
        #
        l_box_cycle -build 7 0 8 0 0
        l_box_cycle -build 6 7 9 8 7
        l_box_cycle -build 5 6 10 9 8
        l_box_cycle -build 4 5 11 10 9
        l_box_cycle -build 3 4 12 11 10
        l_box_cycle -build 0 0 13 12 11
    fi

# Remove: =====

# Box cycle:

elif [ $1 = -cyc_remove ]; then
    if [ "$2" != "0" ]; then "$GMS_REPLY" -bottom $2; fi
    if [ "$3" != "0" ]; then "$GMS_REPLY" -shadow $3; fi
    if [ "$4" != "0" ]; then
        "$GMS_REPLY" -stripe $4 $REPLY_PATTERN $REPLY_LETTER
    fi
    if [ "$5" != "0" ]; then "$GMS_REPLY" -top $5; fi
    if [ "$6" != "0" ]; then
        "$GMS_REPLY" -stripe $6 $REPLY_PATTERN $REPLY_LETTER
    fi
fi
```

```

# Box, y-offset = 13: -----
elif [ $1 = -remove ]; then
  if [ "$GMS_ANIMATE" = "on" ]; then
    # Animation:
    l_box_cycle -remove 20 21 22 4 3
    l_box_cycle -remove 19 20 21 5 4
    l_box_cycle -remove 18 19 20 6 5
    l_box_cycle -remove 17 18 19 7 6
    l_box_cycle -remove 16 17 18 8 7
    l_box_cycle -remove 15 16 17 9 8
    l_box_cycle -remove 14 15 16 10 9
    l_box_cycle -remove 13 14 15 11 10
    l_box_cycle -remove 12 13 14 12 11
    l_box_cycle -remove 0 12 13 0 12
  else # ` ` Offset
    # Erasure:
    for i in 3 22 4 21 5 20; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    for i in 6 19 7 18 8 17; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    for i in 9 16 10 15 11 14; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    for i in 12 13; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    fi
  fi
# Top box, y-offset = 12: -----
elif [ $1 = -t12_remove ]; then
  if [ "$GMS_ANIMATE" = "on" ]; then
    l_box_cycle -remove 18 19 20 4 3
    l_box_cycle -remove 17 18 19 5 4
    l_box_cycle -remove 16 17 18 6 5
    l_box_cycle -remove 15 16 17 7 6
    l_box_cycle -remove 14 15 16 8 7
    l_box_cycle -remove 13 14 15 9 8
    l_box_cycle -remove 12 13 14 10 9
    l_box_cycle -remove 11 12 13 11 10
    l_box_cycle -remove 0 11 12 0 11
  else # ` ` Offset
    # Erasure:
    for i in 3 20 4 19 5 18 ; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    for i in 6 17 7 16 8 15 ; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    for i in 9 14 10 13 11 12 ; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    fi
  fi
fi

# Top box, y-offset = 11: -----
elif [ $1 = -t11_remove ]; then
  if [ "$GMS_ANIMATE" = "on" ]; then
    l_box_cycle -remove 16 17 18 4 3
    l_box_cycle -remove 15 16 17 5 4
    l_box_cycle -remove 14 15 16 6 5
    l_box_cycle -remove 13 14 15 7 6
    l_box_cycle -remove 12 13 14 8 7
    l_box_cycle -remove 11 12 13 9 8
    l_box_cycle -remove 10 11 12 10 9
    l_box_cycle -remove 0 10 11 0 10
  else # ` ` Offset
    # Erasure:
    for i in 3 18 4 17 5 16 ; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    for i in 6 15 7 14 8 13 ; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    for i in 9 12 10 11 ; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    fi
  fi
# Top box, y-offset = 8: -----
elif [ $1 = -t8_remove ]; then
  if [ "$GMS_ANIMATE" = "on" ]; then
    l_box_cycle -remove 13 0 0 0 0
    l_box_cycle -remove 12 13 0 0 0
    l_box_cycle -remove 11 12 13 3 0
    l_box_cycle -remove 10 11 12 4 3
    l_box_cycle -remove 9 10 11 5 4
    l_box_cycle -remove 8 9 10 6 5
    l_box_cycle -remove 7 8 9 7 6
    l_box_cycle -remove 0 7 8 0 7
  else # ` ` Offset
    # Erasure:
    for i in 3 13 4 12 5 11 ; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    for i in 6 10 7 9 8 ; do
      "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
    fi
  fi
# Not found: -----
else
  source "$GMS_SHELL/l_banner" -no_action l_box $1
fi

```

I_code

```
#!/bin/sh

# l_code
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_CODE=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_code)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_code $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " l_code ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Code column: -----

function l_code_code {
# upper half:
    if [ $1 = -up ]; then
        # Character:
        REPLY_SIZE=1
        REPLY_OFFSET=$OFFSET_A
        "$GMS_REPLY" -stripe 5 $REPLY_HOTKEY "$2"
        "$GMS_REPLY" -stripe 6 $REPLY_HOTKEY "$3"
        "$GMS_REPLY" -stripe 7 $REPLY_HOTKEY "$4"
        "$GMS_REPLY" -stripe 8 $REPLY_HOTKEY "$5"
        "$GMS_REPLY" -stripe 9 $REPLY_HOTKEY "$6"
        "$GMS_REPLY" -stripe 10 $REPLY_HOTKEY "$7"
        "$GMS_REPLY" -stripe 11 $REPLY_HOTKEY "$8"
        "$GMS_REPLY" -stripe 12 $REPLY_HOTKEY "$9"
        # Gap:
        REPLY_SIZE=1
        REPLY_OFFSET=$OFFSET_B
        "$GMS_REPLY" -stripe 12 0 32
        "$GMS_REPLY" -stripe 11 0 32
        "$GMS_REPLY" -stripe 10 0 32
        "$GMS_REPLY" -stripe 9 0 32
        "$GMS_REPLY" -stripe 8 0 32
        "$GMS_REPLY" -stripe 7 0 32
        "$GMS_REPLY" -stripe 6 0 32
        "$GMS_REPLY" -stripe 5 0 32
        # Number:
        REPLY_SIZE=7
        REPLY_OFFSET=$OFFSET_C
        "$GMS_REPLY" -banner 5 "$2"
        "$GMS_REPLY" -banner 6 "$3"
        "$GMS_REPLY" -banner 7 "$4"
        "$GMS_REPLY" -banner 8 "$5"
        "$GMS_REPLY" -banner 9 "$6"
        "$GMS_REPLY" -banner 10 "$7"
        "$GMS_REPLY" -banner 11 "$8"
        "$GMS_REPLY" -banner 12 "$9"
        # lower half:
        elif [ $1 = -lo ]; then
            # Character:
            REPLY_SIZE=1
            REPLY_OFFSET=$OFFSET_A
            "$GMS_REPLY" -stripe 20 $REPLY_HOTKEY "$9"
            "$GMS_REPLY" -stripe 19 $REPLY_HOTKEY "$8"
            "$GMS_REPLY" -stripe 18 $REPLY_HOTKEY "$7"
            "$GMS_REPLY" -stripe 17 $REPLY_HOTKEY "$6"
            "$GMS_REPLY" -stripe 16 $REPLY_HOTKEY "$5"
            "$GMS_REPLY" -stripe 15 $REPLY_HOTKEY "$4"
            "$GMS_REPLY" -stripe 14 $REPLY_HOTKEY "$3"
            "$GMS_REPLY" -stripe 13 $REPLY_HOTKEY "$2"
            # Gap:
            REPLY_SIZE=1
            REPLY_OFFSET=$OFFSET_B
            "$GMS_REPLY" -stripe 13 0 32
            "$GMS_REPLY" -stripe 14 0 32
            "$GMS_REPLY" -stripe 15 0 32
            "$GMS_REPLY" -stripe 16 0 32
            "$GMS_REPLY" -stripe 17 0 32
            "$GMS_REPLY" -stripe 18 0 32
            "$GMS_REPLY" -stripe 19 0 32
            "$GMS_REPLY" -stripe 20 0 32
            # Number:
            REPLY_SIZE=7
            REPLY_OFFSET=$OFFSET_C
            "$GMS_REPLY" -banner 20 "$9"
            "$GMS_REPLY" -banner 19 "$8"
            "$GMS_REPLY" -banner 18 "$7"
            "$GMS_REPLY" -banner 17 "$6"
            "$GMS_REPLY" -banner 16 "$5"
            "$GMS_REPLY" -banner 15 "$4"
            "$GMS_REPLY" -banner 14 "$3"
            "$GMS_REPLY" -banner 13 "$2"
            fi; }

# Number column (upper half): -----

function l_code_num {
# upper half:
    if [ $1 = -up ]; then
        REPLY_SIZE=7
        REPLY_OFFSET=$OFFSET_C
        "$GMS_REPLY" -banner 5 "$2"
        "$GMS_REPLY" -banner 6 "$3"
        "$GMS_REPLY" -banner 7 "$4"
        "$GMS_REPLY" -banner 8 "$5"
        "$GMS_REPLY" -banner 9 "$6"
        "$GMS_REPLY" -banner 10 "$7"
        "$GMS_REPLY" -banner 11 "$8"
        "$GMS_REPLY" -banner 12 "$9"
        # lower half:
        elif [ $1 = -lo ]; then
            REPLY_SIZE=7
            REPLY_OFFSET=$OFFSET_C
            "$GMS_REPLY" -banner 20 "$9"
            "$GMS_REPLY" -banner 19 "$8"
            "$GMS_REPLY" -banner 18 "$7"
            "$GMS_REPLY" -banner 17 "$6"
            "$GMS_REPLY" -banner 16 "$5"
            "$GMS_REPLY" -banner 15 "$4"
            "$GMS_REPLY" -banner 14 "$3"
            "$GMS_REPLY" -banner 13 "$2"
            fi; }

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_code
```



```

# Codepage: .....
elif [ $1 = -cp_lo \
-o $1 = -cp_up ]; then
# Clear desktop area for codepage pane:
for i in 12 13 11 14 10 15 ; do "$GMS_REPLY" -stripe $i 0 32; done
for i in 9 16 8 17 7 18 ; do "$GMS_REPLY" -stripe $i 0 32; done
for i in 6 19 5 20 4 21 ; do "$GMS_REPLY" -stripe $i 0 32; done
# Code rows:
PAGE=$1
CODE_UP="l_code_code -up"
CODE_LO="l_code_code -lo"
NUM_UP="l_code_num -up"
NUM_LO="l_code_num -lo"
# Column 5:
OFFSET_A=45
OFFSET_B=44
OFFSET_C=41
if [ $PAGE = -cp_lo ]; then
$CODE_UP " 64" " 65" " 66" " 67" " 68" " 69" " 70" " 71"
$CODE_LO " 72" " 73" " 74" " 75" " 76" " 77" " 78" " 79"
else
$CODE_UP 192 193 194 195 196 197 198 199
$CODE_LO 200 201 202 203 204 205 206 207
fi
# Column 4:
OFFSET_A=36
OFFSET_B=35
OFFSET_C=32
if [ $PAGE = -cp_lo ]; then
$CODE_LO " 56" " 57" " 58" " 59" " 60" " 61" " 62" " 63"
$CODE_UP " 48" " 49" " 50" " 51" " 52" " 53" " 54" " 55"
else
$CODE_LO 184 185 186 187 188 189 190 191
$CODE_UP 176 177 178 179 180 181 182 183
fi
# Column 6:
OFFSET_A=54
OFFSET_B=53
OFFSET_C=50
if [ $PAGE = -cp_lo ]; then
$CODE_UP " 80" " 81" " 82" " 83" " 84" " 85" " 86" " 87"
$CODE_LO " 88" " 89" " 90" " 91" " 92" " 93" " 94" " 95"
else
$CODE_UP 208 209 210 211 212 213 214 215
$CODE_LO 216 217 218 219 220 221 222 223
fi
# Column 3:
OFFSET_A=27
OFFSET_B=26
OFFSET_C=23
if [ $PAGE = -cp_lo ]; then
$CODE_LO " 40" " 41" " 42" " 43" " 44" " 45" " 46" " 47"
$CODE_UP " 32" " 33" " 34" " 35" " 36" " 37" " 38" " 39"
else
$CODE_LO 168 169 170 171 172 173 174 175
$CODE_UP 160 161 162 163 164 165 166 167
fi
# Column 7:
OFFSET_A=63
OFFSET_B=62
OFFSET_C=59
if [ $PAGE = -cp_lo ]; then
$CODE_UP " 96" " 97" " 98" " 99" 100 101 102 103
$CODE_LO 104 105 106 107 108 109 110 111
else
$CODE_UP 224 225 226 227 228 229 230 231
$CODE_LO 232 233 234 235 236 237 238 239
fi
# Column 2:
OFFSET_C=14
if [ $PAGE = -cp_lo ]; then
$NUM_LO " 24" " 25" " 26" " 27" " 28" " 29" " 30" " 31"
$NUM_UP " 16" " 17" " 18" " 19" " 20" " 21" " 22" " 23"
else
$NUM_LO 152 153 154 155 156 157 158 159
$NUM_UP 144 145 146 147 148 149 150 151
fi
# Column 8:
OFFSET_A=72
OFFSET_B=71
OFFSET_C=68
if [ $PAGE = -cp_lo ]; then
$CODE_UP 112 113 114 115 116 117 118 119
$CODE_LO 120 121 122 123 124 125 126 127
else
$CODE_UP 240 241 242 243 244 245 246 247
$CODE_LO 248 249 250 251 252 253 254 255
fi
# Column 1:
OFFSET_C=5
if [ $PAGE = -cp_lo ]; then
$NUM_UP " 0" " 1" " 2" " 3" " 4" " 5" " 6" " 7"
$NUM_LO " 8" " 9" " 10" " 11" " 12" " 13" " 14" " 15"
else
$NUM_UP 128 129 130 131 132 133 134 135
$NUM_LO 136 137 138 139 140 141 142 143
fi
# Unset:
CODE_UP=
CODE_LO=
NUM_UP=
NUM_LO=
# Not found: .....
else
source "$GMS_SHELL/l_banner" -no_action l_code $1
fi

```

I_color

```
#!/bin/sh

# l_color
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_COLOR=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_color)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_color $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " l_color ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9) " >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_color

# Update: -----

elif [ $1 = -update ]; then
# Cold:
    if [ "$3" = "" ]; then source "$GMS_SHELL/l_banner" -no_cold l_color
    elif [ $3 = . ]; then
        "$GMS_REPLY" -item 6 " " " "
    elif [ $3 = T ]; then
        "$GMS_REPLY" -item 7 "Text: 0...15" T 1
        "$GMS_REPLY" -item 8 "$GMS_TEXT"
    elif [ $3 = H ]; then
        "$GMS_REPLY" -item 9 "Hotkey: 0...15" H 2
        "$GMS_REPLY" -item 10 "$GMS_HOTKEY"
    elif [ $3 = P ]; then
        "$GMS_REPLY" -item 11 "Pattern: 0...15" P 3
        "$GMS_REPLY" -item 12 "$GMS_PATTERN"
    elif [ $3 = B ]; then
        "$GMS_REPLY" -item 13 "Banner: 0...7" B 4
        "$GMS_REPLY" -item 14 "$GMS_BANNER"
    elif [ $3 = S ]; then
        "$GMS_REPLY" -item 15 "Shadow: 0...7" S 5
        "$GMS_REPLY" -item 16 "$GMS_SHADE"
        "$GMS_REPLY" -item 17 "Desktop: 0...7" D 6
        "$GMS_REPLY" -item 18 "$GMS_DESKTOP"
        "$GMS_REPLY" -item 19 "Letter: 32..126" L 7
        "$GMS_REPLY" -item 20 "$GMS_LETTER"
        "$GMS_REPLY" -bottom 21
        "$GMS_REPLY" -shadow 22

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_color $1
fi

"$GMS_REPLY" -item 16 "$GMS_SHADE"
elif [ $3 = D ]; then
    "$GMS_REPLY" -item 17 "Desktop: 0...7" D 6
    "$GMS_REPLY" -item 18 "$GMS_DESKTOP"
elif [ $3 = L ]; then
    "$GMS_REPLY" -item 19 "Letter: 32..126" L 7
    "$GMS_REPLY" -item 20 "$GMS_LETTER"
elif [ $3 = Q ]; then
    "$GMS_REPLY" -item 5 "Quit dialog []->" Q 8
else source "$GMS_SHELL/l_banner" -no_cold l_color $3; fi

# Hot:
if [ "$2" = "" ]; then source "$GMS_SHELL/l_banner" -no_hot l_color
elif [ $2 = - ]; then "$GMS_REPLY" -item 6 " " " "
elif [ $2 = T ]; then "$GMS_REPLY" -item 7 "Text: 0...15" TT 1
elif [ $2 = H ]; then "$GMS_REPLY" -item 9 "Hotkey: 0...15" HH 2
elif [ $2 = P ]; then "$GMS_REPLY" -item 11 "Pattern: 0...15" PP 3
elif [ $2 = B ]; then "$GMS_REPLY" -item 13 "Banner: 0...7" BB 4
elif [ $2 = S ]; then "$GMS_REPLY" -item 15 "Shadow: 0...7" SS 5
elif [ $2 = D ]; then "$GMS_REPLY" -item 17 "Desktop: 0...7" DD 6
elif [ $2 = L ]; then "$GMS_REPLY" -item 19 "Letter: 32..126" LL 7
elif [ $2 = Q ]; then "$GMS_REPLY" -item 5 "Quit dialog []->" QQ 8
else source "$GMS_SHELL/l_banner" -no_hot l_color $2; fi

# Build: -----

elif [ $1 = -build ]; then
"$GMS_REPLY" -top 3 "-- Select color --"
"$GMS_REPLY" -item 4 " " " "
"$GMS_REPLY" -item 5 "Quit dialog []->" Q 8
"$GMS_REPLY" -item 6 " " " "
"$GMS_REPLY" -item 7 "Text: 0...15" T 1
"$GMS_REPLY" -item 8 "$GMS_TEXT"
"$GMS_REPLY" -item 9 "Hotkey: 0...15" H 2
"$GMS_REPLY" -item 10 "$GMS_HOTKEY"
"$GMS_REPLY" -item 11 "Pattern: 0...15" P 3
"$GMS_REPLY" -item 12 "$GMS_PATTERN"
"$GMS_REPLY" -item 13 "Banner: 0...7" B 4
"$GMS_REPLY" -item 14 "$GMS_BANNER"
"$GMS_REPLY" -item 15 "Shadow: 0...7" S 5
"$GMS_REPLY" -item 16 "$GMS_SHADE"
"$GMS_REPLY" -item 17 "Desktop: 0...7" D 6
"$GMS_REPLY" -item 18 "$GMS_DESKTOP"
"$GMS_REPLY" -item 19 "Letter: 32..126" L 7
"$GMS_REPLY" -item 20 "$GMS_LETTER"
"$GMS_REPLY" -bottom 21
"$GMS_REPLY" -shadow 22

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_color $1
fi
```

I_desk

```
#!/bin/sh

# l_desk
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_DESK=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_desk)."
    exit
fi

# Debug: -----
if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_desk $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " l_desk ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_desk

# Resize: -----

elif [ $1 = -resize ]; then
    export REPLY_OFFSET=0
    export REPLY_SIZE=78
    export REPLY_ITEMS=0

# Build: -----

elif [ $1 = -build ]; then
    source "$GMS_SHELL/l_desk" -resize
    stty raw -echo
    setterm -reset -cursor off
    if [ "$GMS_ANIMATE" = "on" ]; then
        for i in 1 24 2 23 3 22; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
        for i in 4 21 5 20 6 19; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
        for i in 7 18 8 17 9 16; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
        for i in 10 15 11; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
        for i in 12 13 11; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
        for i in 14 10 15; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
        for i in 9 16 8 17 7 18; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
        for i in 6 19 5 20 4 21; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
        for i in 3 22 2 23 1 24; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
        fi
        for i in 1 24 2 23 3 22; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
        for i in 4 21 5 20 6 19; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
        for i in 7 18 8 17 9 16; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
        for i in 10 15 11 14 12 13; do
            "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
        fi
        # Remove: -----
        elif [ $1 = -remove ]; then
            source "$GMS_SHELL/l_desk" -resize
            if [ "$GMS_ANIMATE" = "on" ]; then
                for i in 12 13 11; do
                    "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
                for i in 14 10 15; do
                    "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
                for i in 9 16 8 17 7 18; do
                    "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
                for i in 6 19 5 20 4 21; do
                    "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
                for i in 3 22 2 23 1 24; do
                    "$GMS_REPLY" -stripe $i $REPLY_PATTERN 32; done
                fi
                for i in 12 13 11 14 10 15; do
                    "$GMS_REPLY" -clear $i $REPLY_PATTERN $REPLY_LETTER; done
                for i in 9 16 8 17 7 18; do
                    "$GMS_REPLY" -clear $i $REPLY_PATTERN $REPLY_LETTER; done
                for i in 6 19 5 20 4 21; do
                    "$GMS_REPLY" -clear $i $REPLY_PATTERN $REPLY_LETTER; done
                for i in 3 22 2 23 1 24; do
                    "$GMS_REPLY" -clear $i $REPLY_PATTERN $REPLY_LETTER; done
                stty -raw echo
                setterm -cursor on
                setterm -clear
            # Not found: -----
            else
                source "$GMS_SHELL/l_banner" -no_action l_desk $1
            fi
        fi
    fi
fi
```

I_file

```
#!/bin/sh

# l_file
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_FILE=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_file)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_file $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " l_file ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_file

# Update: -----

elif [ $1 = -update ]; then

# Cold state:
if [ "$3" = "" ]; then
    source "$GMS_SHELL/l_banner" -no_cold l_file called_by l_file -update
elif [ $3 = 1 ]; then "$GMS_REPLY" -item 11 "1 $GMS1 $REPLY1" 1 1
elif [ $3 = 2 ]; then "$GMS_REPLY" -item 12 "2 $GMS2 $REPLY2" 2 2
elif [ $3 = 3 ]; then "$GMS_REPLY" -item 13 "3 $GMS3 $REPLY3" 3 3
elif [ $3 = 4 ]; then "$GMS_REPLY" -item 14 "4 $GMS4 $REPLY4" 4 4
elif [ $3 = 5 ]; then "$GMS_REPLY" -item 15 "5 $GMS5 $REPLY5" 5 5
elif [ $3 = 6 ]; then "$GMS_REPLY" -item 16 "6 $GMS6 $REPLY6" 6 6
elif [ $3 = 7 ]; then "$GMS_REPLY" -item 17 "7 $GMS7 $REPLY7" 7 7
elif [ $3 = 8 ]; then "$GMS_REPLY" -item 18 "8 $GMS8 $REPLY8" 8 8
elif [ $3 = 9 ]; then "$GMS_REPLY" -item 19 "9 $GMS9 $REPLY9" 9 9
elif [ $3 = - ]; then "$GMS_REPLY" -item 4 " " " "

elif [ $3 = Q ]; then "$GMS_REPLY" -item 5 "Q    Quit dialog" Q 10
elif [ $3 = P ]; then "$GMS_REPLY" -item 7 "P <- Previous files" P 11
elif [ $3 = N ]; then "$GMS_REPLY" -item 8 "N -> Next files" N 12
elif [ $3 = H ]; then "$GMS_REPLY" -item 9 "H .. Higher level" H 13
else
    source "$GMS_SHELL/l_banner" -no_cold l_file $3 called_by l_file -update
fi

# Hot state:
if [ "$2" = "" ]; then
    source "$GMS_SHELL/l_banner" -no_hot l_file called_by l_file -update
elif [ $2 = 1 ]; then "$GMS_REPLY" -item 11 "1 $GMS1 $REPLY1" 11 1
elif [ $2 = 2 ]; then "$GMS_REPLY" -item 12 "2 $GMS2 $REPLY2" 22 2
elif [ $2 = 3 ]; then "$GMS_REPLY" -item 13 "3 $GMS3 $REPLY3" 33 3
elif [ $2 = 4 ]; then "$GMS_REPLY" -item 14 "4 $GMS4 $REPLY4" 44 4
elif [ $2 = 5 ]; then "$GMS_REPLY" -item 15 "5 $GMS5 $REPLY5" 55 5
elif [ $2 = 6 ]; then "$GMS_REPLY" -item 16 "6 $GMS6 $REPLY6" 66 6
elif [ $2 = 7 ]; then "$GMS_REPLY" -item 17 "7 $GMS7 $REPLY7" 77 7
elif [ $2 = 8 ]; then "$GMS_REPLY" -item 18 "8 $GMS8 $REPLY8" 88 8
elif [ $2 = 9 ]; then "$GMS_REPLY" -item 19 "9 $GMS9 $REPLY9" 99 9
elif [ $2 = Q ]; then "$GMS_REPLY" -item 5 "Q    Quit dialog" QQ 10
elif [ $2 = P ]; then "$GMS_REPLY" -item 7 "P <- Previous files" PP 11
elif [ $2 = N ]; then "$GMS_REPLY" -item 8 "N -> Next files" NN 12
elif [ $2 = H ]; then "$GMS_REPLY" -item 9 "H .. Higher level" HH 13
else
    source "$GMS_SHELL/l_banner" -no_hot l_file $2 called_by l_file -update
fi

# Build: -----

elif [ $1 = -build ]; then
    "$GMS_REPLY" -top 3 "$GMS_FILEBOX"
    "$GMS_REPLY" -item 4 " " " "
    "$GMS_REPLY" -item 5 "Q    Quit dialog" Q 10
    "$GMS_REPLY" -item 6 " " " "
    "$GMS_REPLY" -item 7 "P <- Previous files" P 11
    "$GMS_REPLY" -item 8 "N -> Next files" N 12
    "$GMS_REPLY" -item 9 "H .. Higher level" H 13
    for i in 10 11 12 13 14 15; do "$GMS_REPLY" -item $i " " " "; done
    for i in 16 17 18 19 20; do "$GMS_REPLY" -item $i " " " "; done
    "$GMS_REPLY" -bottom 21
    "$GMS_REPLY" -shadow 22

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_file $1
fi
```

l_gms

```
#!/bin/sh

# l_gms
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_GMS=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_gms)."
    exit
fi

# Chapters: =====

# Desktop:

if [ $1 = -desktop ]; then
    setterm -clear
    echo "$GMS_FILE"
    if [ "$2" = "-draw" ]; then
        setterm -clear
        export REPLY_BANNER=15
        export REPLY_TEXT=0
        export REPLY_OFFSET=0
        export REPLY_SIZE=82
        if [ "$GMS_PROGRAM" != "" ]; then
            "$GMS_BINARIES/reply" -banner 1 \
            "$GMS_FILE - GMS: Running $GMS_PROGRAM ..."
        elif [ "$GMS_MESSAGE" != "" ]; then
            "$GMS_BINARIES/reply" -banner 1 "$GMS_MESSAGE"
        else
            "$GMS_BINARIES/reply" -banner 1 "$GMS_FILE"
        fi
        echo
        cat "$GMS_SETTING/desktop.scn"
        if [ "$3" != "-nofolder" ]; then
            "$GMS_BINARIES/reply" -banner 24 "$GMS_FOLDER"
        fi
    else
        for i in 1 2 3 4 5 6 7 8 9 10 11; do echo; done
        for i in 1 2 3 4 5 6 7 8 9 10 11; do echo; done
    fi
fi

# Banner: -----

elif [ $1 = -banner ]; then
    echo "$GMS_FILE"
    echo -n "/////////////////////////////////////"
    echo "/////////////////////////////////////"

# Help: -----

elif [ $1 = -help ]; then
    if [ "$GMS_TEXTMODE" != "1" ]; then
        setterm -clear
        if [ "$GMS_MESSAGE" != "" ]; then
            echo "$GMS_MESSAGE"
        elif [ "$GMS_FILE" = "" ]; then
            echo -n "Command line options          Gerolf"
            echo " Markup Shredder $GMS_VERSION "
        else
            echo "$GMS_FILE"
        fi
        cat "$GMS_SETTING/menu.scn"
        echo
    fi

# Welcome: -----

elif [ $1 = -welcome ]; then
    if [ "$GMS_TEXTMODE" != "1" ]; then
        setterm -clear
        cat "$GMS_SETTING/welcome.scn"
        # Show version number and date:
        export REPLY_BANNER=15
        export REPLY_DESKTOP=15
        export REPLY_SHADE=15
        export REPLY_TEXT=0
        export REPLY_HOTKEY=0
        export REPLY_OFFSET=11
        export REPLY_SIZE=17
        "$GMS_BINARIES/reply" -item 12 "Version $GMS_VERSION" X X
        export REPLY_OFFSET=8
        export REPLY_SIZE=20
        "$GMS_BINARIES/reply" -item 16 "$GMS_DATE" X X
        export REPLY_OFFSET=39
        export REPLY_SIZE=20
        "$GMS_BINARIES/reply" -banner 24 "Press [Enter] ..." X X
        # Set prompt on line 24:
        export REPLY_OFFSET=2
        export REPLY_SIZE=26
        "$GMS_BINARIES/reply" -bottom 23
    fi

# Goodbye: -----

elif [ $1 = -goodbye ]; then
    if [ "$GMS_TEXTMODE" != "1" ]; then
        setterm -clear
        echo ""
        cat "$GMS_SETTING/goodbye.scn"
    fi

# Not found: -----

else
    export GMS_MESSAGE="GMS error (l_gms): No action $1"
    source "$GMS_SHELL/l_gms" -help
fi
```

I_good

```
#!/bin/sh

# l_good
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2007).

GMSdateL_GOOD=20080107

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_good)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_good $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " l_good ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_good

# Update: -----

elif [ $1 = -update ]; then
# Cold:
"$GMS_REPLY" -item 6
# Hot:
if [ "$2" = "" ]; then

        source "$GMS_SHELL/l_banner" -no_hot l_good
    elif [ $2 = . ]; then
        "$GMS_REPLY" -item 5 "          Gerolf Markup Shredder" Q
    elif [ $2 = Q ]; then
        "$GMS_REPLY" -item 5 "          Gerolf Markup Shredder" QQ
    else
        source "$GMS_SHELL/l_banner" -no_hot l_good $2
    fi

# Build: -----

elif [ $1 = -build ]; then
"$GMS_REPLY" -top 3 "----- About -----"
"$GMS_REPLY" -item 4
"$GMS_REPLY" -item 5 "          Gerolf Markup Shredder" Q
"$GMS_REPLY" -item 6
"$GMS_REPLY" -item 7 " Copyright (c) 1999-2008 by G. D. Brett-"
"$GMS_REPLY" -item 8 " schneider, Luchtbergstr. 27, D-28237"
"$GMS_REPLY" -item 9 " Bremen. All rights reserved. This CMS"
"$GMS_REPLY" -item 10 " software comes without ANY warranty."
"$GMS_REPLY" -item 11 " You may freely distribute and use it."
"$GMS_REPLY" -item 12 " "
"$GMS_REPLY" -item 13 " "
"$GMS_REPLY" -item 14 " "
"$GMS_REPLY" -item 15 " "
"$GMS_REPLY" -item 16 " "
"$GMS_REPLY" -item 17 " "
"$GMS_REPLY" -item 18 " "
"$GMS_REPLY" -item 19 " "
"$GMS_REPLY" -item 20 " "
"$GMS_REPLY" -bottom 21
"$GMS_REPLY" -shadow 22

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_good $1
fi
```

l_list

```
#!/bin/sh

# l_list
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_LIST=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_list)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_list $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " l_list ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_list

# Build: -----

elif [ $1 = -build ]; then
    "$GMS_REPLY" -item 11 "1 $GMS1 $REPLY1" 1 1
    "$GMS_REPLY" -item 12 "2 $GMS2 $REPLY2" 2 2
    "$GMS_REPLY" -item 13 "3 $GMS3 $REPLY3" 3 3
    "$GMS_REPLY" -item 14 "4 $GMS4 $REPLY4" 4 4
    "$GMS_REPLY" -item 15 "5 $GMS5 $REPLY5" 5 5
    "$GMS_REPLY" -item 16 "6 $GMS6 $REPLY6" 6 6
    "$GMS_REPLY" -item 17 "7 $GMS7 $REPLY7" 7 7
    "$GMS_REPLY" -item 18 "8 $GMS8 $REPLY8" 8 8
    "$GMS_REPLY" -item 19 "9 $GMS9 $REPLY9" 9 9

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action_list $1
fi
```

I_menu

```
#!/bin/sh

# l_menu
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_MENU=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_menu)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_menu $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "l_menu ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_menu

# Update: -----

elif [ $1 = -update ]; then
    # Cold:
    if [ "$3" = "" ]; then source "$GMS_SHELL/l_banner" -no_cold l_menu
    elif [ $3 = Q ]; then "$GMS_REPLY" -item 5 "Quit GMS program" Q 1
    elif [ $3 = C ]; then "$GMS_REPLY" -item 7 "Create markup file" C 2
    elif [ $3 = O ]; then "$GMS_REPLY" -item 8 "Open folder/file" O 3
    elif [ $3 = V ]; then "$GMS_REPLY" -item 9 "View markup file" V 4
    elif [ $3 = E ]; then "$GMS_REPLY" -item 10 "Edit markup file" E 5
    elif [ $3 = B ]; then "$GMS_REPLY" -item 11 "Browse markup file" B 6
    elif [ $3 = A ]; then "$GMS_REPLY" -item 12 "Analyse folder/file" A 7
    elif [ $3 = T ]; then "$GMS_REPLY" -item 13 "Typeset markup file" T 8
    elif [ $3 = R ]; then "$GMS_REPLY" -item 14 "Read output file" R 9
    elif [ $3 = L ]; then "$GMS_REPLY" -item 16 "Learn GMS tricks" L 10
    elif [ $3 = S ]; then "$GMS_REPLY" -item 17 "Select GMS setting" S 11
    elif [ $3 = W ]; then "$GMS_REPLY" -item 18 "Write GMS fontmap" W 12
    elif [ $3 = I ]; then "$GMS_REPLY" -item 19 "Init GMS format" I 13
    else source "$GMS_SHELL/l_banner" -no_cold l_menu $3; fi
    # Hot:
    if [ "$2" = "" ]; then source "$GMS_SHELL/l_banner" -no_hot l_menu
    elif [ $2 = Q ]; then "$GMS_REPLY" -item 5 "Quit GMS program" QQ 1
    elif [ $2 = . ]; then "$GMS_REPLY" -item 6 " " " "
    elif [ $2 = C ]; then "$GMS_REPLY" -item 7 "Create markup file" CC 2
    elif [ $2 = O ]; then "$GMS_REPLY" -item 8 "Open folder/file" OO 3
    elif [ $2 = V ]; then "$GMS_REPLY" -item 9 "View markup file" VV 4
    elif [ $2 = E ]; then "$GMS_REPLY" -item 10 "Edit markup file" EE 5
    elif [ $2 = B ]; then "$GMS_REPLY" -item 11 "Browse markup file" BB 6
    elif [ $2 = A ]; then "$GMS_REPLY" -item 12 "Analyse folder/file" AA 7
    elif [ $2 = T ]; then "$GMS_REPLY" -item 13 "Typeset markup file" TT 8
    elif [ $2 = R ]; then "$GMS_REPLY" -item 14 "Read output file" RR 9
    elif [ $2 = L ]; then "$GMS_REPLY" -item 16 "Learn G.M.S. tricks" LL 10
    elif [ $2 = S ]; then "$GMS_REPLY" -item 17 "Select GMS setting" SS 11
    elif [ $2 = W ]; then "$GMS_REPLY" -item 18 "Write GMS fontmap" WW 12
    elif [ $2 = I ]; then "$GMS_REPLY" -item 19 "Init GMS format" II 13
    else source "$GMS_SHELL/l_banner" -no_hot l_menu $2; fi

# Build: -----

elif [ $1 = -build ]; then
    "$GMS_REPLY" -top 3 "----- GMS -----"
    "$GMS_REPLY" -item 4 " " " "
    "$GMS_REPLY" -item 5 "Quit GMS program" Q 1
    "$GMS_REPLY" -item 6 " " " "
    "$GMS_REPLY" -item 7 "Create markup file" C 2
    "$GMS_REPLY" -item 8 "Open folder/file" O 3
    "$GMS_REPLY" -item 9 "View markup file" V 4
    "$GMS_REPLY" -item 10 "Edit markup file" E 5
    "$GMS_REPLY" -item 11 "Browse markup file" B 6
    "$GMS_REPLY" -item 12 "Analyse folder/file" A 7
    "$GMS_REPLY" -item 13 "Typeset markup file" T 8
    "$GMS_REPLY" -item 14 "Read output file" R 9
    "$GMS_REPLY" -item 15 " " " "
    "$GMS_REPLY" -item 16 "Learn GMS tricks" L 10
    "$GMS_REPLY" -item 17 "Select GMS setting" S 11
    "$GMS_REPLY" -item 18 "Write GMS fontmap" W 12
    "$GMS_REPLY" -item 19 "Init GMS format" I 13
    "$GMS_REPLY" -item 20 " " " "
    "$GMS_REPLY" -bottom 21
    "$GMS_REPLY" -shadow 22

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_menu $1
fi
```


I_prog

```
#!/bin/sh

# l_prog
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_PROG=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_prog)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_prog $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "l_prog ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_prog

# Update: -----

elif [ $1 = -update ]; then
# Cold state:
    if [ "$3" = "" ]; then
        source "$GMS_SHELL/l_banner" -no_cold l_prog called_by l_prog -update
    elif [ $3 = . ]; then
        "$GMS_REPLY" -item 6 " " " "
    elif [ $3 = V ]; then
        "$GMS_REPLY" -item 7 "Viewer: any text" V 1
        "$GMS_REPLY" -item 8 "$GMS_VIEWER"
    elif [ $3 = E ]; then
        "$GMS_REPLY" -item 9 "Editor: any text" E 2
        "$GMS_REPLY" -item 10 "$GMS_EDITOR"
    elif [ $3 = B ]; then
        "$GMS_REPLY" -item 11 "Browser: HTML" B 3
        "$GMS_REPLY" -item 12 "$GMS_BROWSER"
    elif [ $3 = A ]; then
        "$GMS_REPLY" -item 13 "Analyst: HTML" A 4
        "$GMS_REPLY" -item 14 "$GMS_ANALYST"
    elif [ $3 = T ]; then
        "$GMS_REPLY" -item 15 "Typesetter: HTML" T 5
        "$GMS_REPLY" -item 16 "$GMS_TSETTER"
    elif [ $3 = R ]; then
        "$GMS_REPLY" -item 17 "Reader: PDF" R 6
        "$GMS_REPLY" -item 18 "$GMS_READER"
    elif [ $3 = Q ]; then
        "$GMS_REPLY" -item 5 "Quit dialog" >" Q 7
    else
        source "$GMS_SHELL/l_banner" -no_hot l_prog $2 called_by l_prog -update
    fi

# Build: -----

elif [ $1 = -build ]; then
    "$GMS_REPLY" -top 3 "Select programs"
    "$GMS_REPLY" -item 4 " " " "
    "$GMS_REPLY" -item 5 "Quit dialog" >" Q 7
    "$GMS_REPLY" -item 6 " " " "
    "$GMS_REPLY" -item 7 "Viewer: any text" V 1
    "$GMS_REPLY" -item 8 "$GMS_VIEWER"
    "$GMS_REPLY" -item 9 "Editor: any text" E 2
    "$GMS_REPLY" -item 10 "$GMS_EDITOR"
    "$GMS_REPLY" -item 11 "Browser: HTML" B 3
    "$GMS_REPLY" -item 12 "$GMS_BROWSER"
    "$GMS_REPLY" -item 13 "Analyst: HTML" A 4
    "$GMS_REPLY" -item 14 "$GMS_ANALYST"
    "$GMS_REPLY" -item 15 "Typesetter: HTML" T 5
    "$GMS_REPLY" -item 16 "$GMS_TSETTER"
    "$GMS_REPLY" -item 17 "Reader: PDF" R 6
    "$GMS_REPLY" -item 18 "$GMS_READER"
    "$GMS_REPLY" -bottom 19
    "$GMS_REPLY" -shadow 20

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_prog $1
fi
```

l_rain

```
#!/bin/sh

# l_rain
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_RAIN=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_rain)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_rain $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " l_rain ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Rainbow column: -----

function l_rain_col {
    REPLY_SIZE=1
    REPLY_DESKTOP=$1
    # Get offset and letter number:
    REPLY_OFFSET=$1
    case "$1" in
        "0" ) REPLY_OFFSET=6; LETTER=48;;
        "1" ) REPLY_OFFSET=7; LETTER=49;;
        "2" ) REPLY_OFFSET=8; LETTER=50;;
        "3" ) REPLY_OFFSET=9; LETTER=51;;
        "4" ) REPLY_OFFSET=10; LETTER=52;;
        "5" ) REPLY_OFFSET=11; LETTER=53;;
        "6" ) REPLY_OFFSET=12; LETTER=54;;
        "7" ) REPLY_OFFSET=13; LETTER=55;;
    esac
    # Draw column:
    "$GMS_REPLY" -stripe 5 0 $LETTER
    "$GMS_REPLY" -stripe 6 1 $LETTER
    "$GMS_REPLY" -stripe 7 2 $LETTER
    "$GMS_REPLY" -stripe 8 3 $LETTER
    "$GMS_REPLY" -stripe 9 4 $LETTER
    "$GMS_REPLY" -stripe 10 5 $LETTER
    "$GMS_REPLY" -stripe 11 6 $LETTER
    "$GMS_REPLY" -stripe 12 7 $LETTER
    "$GMS_REPLY" -stripe 13 8 $LETTER
    "$GMS_REPLY" -stripe 14 9 $LETTER
    "$GMS_REPLY" -stripe 15 10 $LETTER
    "$GMS_REPLY" -stripe 16 11 $LETTER
    "$GMS_REPLY" -stripe 17 12 $LETTER
    "$GMS_REPLY" -stripe 18 13 $LETTER
    "$GMS_REPLY" -stripe 19 14 $LETTER
    "$GMS_REPLY" -stripe 20 15 $LETTER
    unset LETTER; }

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_rain

# Rainbow: -----

elif [ $1 = -build ]; then
    # Save colors:
    BACKUP1=$REPLY_DESKTOP
    BACKUP2=$REPLY_PATTERN
    BACKUP3=$REPLY_BANNER
    BACKUP4=$REPLY_TEXT
    # Clear desktop area for pane:
    for i in 12 13 11 14 10 15; do "$GMS_REPLY" -stripe $i 0 32; done
    for i in 9 16 8 17 7 18; do "$GMS_REPLY" -stripe $i 0 32; done
    for i in 6 19 5 20 4 21; do "$GMS_REPLY" -stripe $i 0 32; done
    for i in 3 22; do "$GMS_REPLY" -stripe $i 0 32; done
    # Draw heading:
    REPLY_BANNER=$BACKUP1
    REPLY_TEXT=$REPLY_HOTKEY
    "$GMS_REPLY" -banner 3 " Back- & Fore- "
    "$GMS_REPLY" -banner 22 " -ground colors "
    # Draw columns with background color numbers:
    for i in 0 1 2 3 4 5 6 7; do l_rain_col $i; done
    # Clear overwritten area:
    REPLY_OFFSET=14
    REPLY_SIZE=5
    REPLY_DESKTOP=$BACKUP1
    REPLY_PATTERN=$BACKUP2
    for i in 5 6 7 8 9 10; do "$GMS_REPLY" -stripe $i 0 32; done
    for i in 11 12 13 14 15 16; do "$GMS_REPLY" -stripe $i 0 32; done
    for i in 17 18 19 20; do "$GMS_REPLY" -stripe $i 0 32; done
    # Draw rows with foreground color numbers:
    REPLY_BANNER=$REPLY_DESKTOP
    REPLY_SIZE=3
    # One digit:
    REPLY_OFFSET=18
    REPLY_TEXT=0; "$GMS_REPLY" -banner 5 0 " "
    REPLY_TEXT=1; "$GMS_REPLY" -banner 6 1 " "
    REPLY_TEXT=2; "$GMS_REPLY" -banner 7 2 " "
    REPLY_TEXT=3; "$GMS_REPLY" -banner 8 3 " "
    REPLY_TEXT=4; "$GMS_REPLY" -banner 9 4 " "
    REPLY_TEXT=5; "$GMS_REPLY" -banner 10 5 " "
    REPLY_TEXT=6; "$GMS_REPLY" -banner 11 6 " "
    REPLY_TEXT=7; "$GMS_REPLY" -banner 12 7 " "
    REPLY_TEXT=8; "$GMS_REPLY" -banner 13 8 " "
    REPLY_TEXT=9; "$GMS_REPLY" -banner 14 9 " "
    # Two digits:
    REPLY_OFFSET=17
    REPLY_TEXT=10; "$GMS_REPLY" -banner 15 10 " "
    REPLY_TEXT=11; "$GMS_REPLY" -banner 16 11 " "
    REPLY_TEXT=12; "$GMS_REPLY" -banner 17 12 " "
    REPLY_TEXT=13; "$GMS_REPLY" -banner 18 13 " "
    REPLY_TEXT=14; "$GMS_REPLY" -banner 19 14 " "
    REPLY_TEXT=15; "$GMS_REPLY" -banner 20 15 " "
    # Reset colors:
    REPLY_BANNER=$BACKUP3
    REPLY_TEXT=$BACKUP4
    BACKUP1=
    BACKUP2=
    BACKUP3=
    BACKUP4=

# Rainbow (and codepage) remove: -----

elif [ $1 = -remove ]; then
    # Clear pane:
    if [ "$GMS_ANIMATE" = "on" ]; then
        for i in 4 21 5 20; do "$GMS_REPLY" -stripe $i 0 32; done
        for i in 6 19 7 18 8 17; do "$GMS_REPLY" -stripe $i 0 32; done
        for i in 9 16 10 15 11 14; do "$GMS_REPLY" -stripe $i 0 32; done
    fi
fi
```

```
    for i in 12 13; do "$GMS_REPLY" -stripe $i 0 32 ; done
fi
# Remove pane:
for i in 3 22 4 21 5 20; do
    "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
for i in 6 19 7 18 8 17 ; do
    "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
for i in 9 16 10 15 11 14 ; do
    "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
```

```
    for i in 12 13 ; do
        "$GMS_REPLY" -stripe $i $REPLY_PATTERN $REPLY_LETTER; done
# Not found: -----
else
    source "$GMS_SHELL/l_banner" -no_action l_rain $1
fi
```

I_save

```
#!/bin/sh

# l_save
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_SAVE=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_save)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_save $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "l_save ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_save
# Update: -----

elif [ $1 = -update ]; then
    if [ -f "$GMS_FOLDER/$GMS_FILE" ]; then

# There exists an old file that can be overwritten:
"$GMS_REPLY" -item 8 "Overwrite existing file" O 3
# Cold state:
if [ "$3" = "" ]; then
    source "$GMS_SHELL/l_banner" -no_cold l_save
    elif [ $3 = Q ]; then "$GMS_REPLY" -item 5 "Quit dialog" Q 1
    elif [ $3 = C ]; then "$GMS_REPLY" -item 7 "Cancel file creation" C 2
    elif [ $3 = O ]; then "$GMS_REPLY" -item 8 "Overwrite existing file" O 3
    elif [ $3 = N ]; then "$GMS_REPLY" -item 10 "$GMS_FILE" N 4
    else
        source "$GMS_SHELL/l_banner" -no_cold l_save $3
    fi
# Hot state:
if [ "$2" = "" ]; then
    source "$GMS_SHELL/l_banner" -no_hot l_save
    elif [ $2 = Q ]; then "$GMS_REPLY" -item 5 "Quit dialog" QQ 1
    elif [ $2 = C ]; then "$GMS_REPLY" -item 7 "Cancel file creation" CC 2
    elif [ $2 = O ]; then "$GMS_REPLY" -item 8 "Overwrite existing file" OO 3
    else
        source "$GMS_SHELL/l_banner" -no_hot l_save $2
    fi
fi

# Build: -----

elif [ $1 = -build ]; then
"$GMS_REPLY" -top 3 "$GMS_FILEBOX"
"$GMS_REPLY" -item 4 " " " "
"$GMS_REPLY" -item 5 "Quit dialog" Q 1
"$GMS_REPLY" -item 6 " " " "
"$GMS_REPLY" -item 7 "Cancel file creation" C 2
"$GMS_REPLY" -item 8 "O" O 3
"$GMS_REPLY" -item 8 " " " "
"$GMS_REPLY" -item 9 " " " "
"$GMS_REPLY" -item 10 "$GMS_FILE" N 4
"$GMS_REPLY" -item 11 " " " "
"$GMS_REPLY" -bottom 12
"$GMS_REPLY" -shadow 13

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_save $1
fi

elif [ $2 = N ]; then
    setterm -cursor on
    "$GMS_REPLY" -question 10 "$GMS_FILE" NN 4
    setterm -cursor off
else
    source "$GMS_SHELL/l_banner" -no_hot l_save $2
fi

# There does not exist an old file that can be overwritten:
# Cold state:
if [ "$3" = "" ]; then
    source "$GMS_SHELL/l_banner" -no_cold l_save
    elif [ $3 = Q ]; then "$GMS_REPLY" -item 5 "Quit dialog" Q 1
    elif [ $3 = C ]; then "$GMS_REPLY" -item 7 "Cancel file creation" C 2
    elif [ $3 = O ]; then "$GMS_REPLY" -item 8 " " " " 3
    elif [ $3 = N ]; then "$GMS_REPLY" -item 10 "$GMS_FILE" N 4
    else
        source "$GMS_SHELL/l_banner" -no_cold l_save $3
    fi
# Hot state:
if [ "$2" = "" ]; then
    source "$GMS_SHELL/l_banner" -no_hot l_save
    elif [ $2 = Q ]; then "$GMS_REPLY" -item 5 "Quit dialog" QQ 1
    elif [ $2 = C ]; then "$GMS_REPLY" -item 7 "Cancel file creation" CC 2
    elif [ $2 = O ]; then "$GMS_REPLY" -item 8 " " " OO 3
    elif [ $2 = N ]; then
        setterm -cursor on
        "$GMS_REPLY" -question 10 "$GMS_FILE" NN 4
        setterm -cursor off
    else
        source "$GMS_SHELL/l_banner" -no_hot l_save $2
    fi
fi

# Build: -----

elif [ $1 = -build ]; then
"$GMS_REPLY" -top 3 "$GMS_FILEBOX"
"$GMS_REPLY" -item 4 " " " "
"$GMS_REPLY" -item 5 "Quit dialog" Q 1
"$GMS_REPLY" -item 6 " " " "
"$GMS_REPLY" -item 7 "Cancel file creation" C 2
"$GMS_REPLY" -item 8 "O" O 3
"$GMS_REPLY" -item 8 " " " "
"$GMS_REPLY" -item 9 " " " "
"$GMS_REPLY" -item 10 "$GMS_FILE" N 4
"$GMS_REPLY" -item 11 " " " "
"$GMS_REPLY" -bottom 12
"$GMS_REPLY" -shadow 13

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_save $1
fi
```

l_select

```
#!/bin/sh

# l_select
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_SELECT=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (l_select)."
    exit
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_select $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo "l_select ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then
    source "$GMS_SHELL/l_banner" -no_action l_select
fi

# Update: -----

elif [ $1 = -update ]; then
# Cold:
    if [ "$3" = "" ]; then source "$GMS_SHELL/l_banner" -no_cold l_select
    elif [ $3 = Q ]; then "$GMS_REPLY" -item 5 "Quit dialog >" Q 1
    elif [ $3 = A ]; then "$GMS_REPLY" -item 7 "Animation $4" A 2
    elif [ $3 = C ]; then "$GMS_REPLY" -item 8 "Colors >" C 3
    elif [ $3 = P ]; then "$GMS_REPLY" -item 9 "Programs >" P 4
    elif [ $3 = D ]; then "$GMS_REPLY" -item 10 "Debugging:" D 5
        if [ "$GMS_DEBUG" = "X" ]; then "$GMS_REPLY" -item 11 "$GMS_DEBUG" X
        elif [ "$GMS_DEBUG" = "Y" ]; then "$GMS_REPLY" -item 11 "$GMS_DEBUG" Y
        else "$GMS_REPLY" -item 11 "$GMS_DEBUG" 0; fi
    elif [ $3 = E ]; then "$GMS_REPLY" -item 12 "Encoding:" E 6
    "$GMS_REPLY" -item 13 "$GMS_CODEPAGE" I
    "$GMS_REPLY" -item 14 "Upper half >" U 7
    "$GMS_REPLY" -item 15 "Lower half >" L 8
    "$GMS_REPLY" -item 16 " " " " "
    "$GMS_REPLY" -bottom 17
    "$GMS_REPLY" -shadow 18

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_select $1
fi

"$GMS_REPLY" -item 13 "$GMS_CODEPAGE" I
elif [ $3 = I ]; then "$GMS_REPLY" -item 13 "$GMS_CODEPAGE" I
elif [ $3 = U ]; then "$GMS_REPLY" -item 14 "Upper half >" U 7
elif [ $3 = L ]; then "$GMS_REPLY" -item 15 "Lower half >" L 8
else source "$GMS_SHELL/l_banner" -no_cold l_select $3; fi

# Hot:
if [ "$2" = "" ]; then source "$GMS_SHELL/l_banner" -no_hot l_select $2
elif [ $2 = . ]; then "$GMS_REPLY" -item 4 " " " " "
elif [ $2 = Q ]; then "$GMS_REPLY" -item 5 "Quit dialog >" QQ 1
elif [ $2 = A ]; then "$GMS_REPLY" -item 7 "Animation $4" AA 2
elif [ $2 = C ]; then "$GMS_REPLY" -item 8 "Colors >" CC 3
elif [ $2 = P ]; then "$GMS_REPLY" -item 9 "Programs >" PP 4
elif [ $2 = D ]; then "$GMS_REPLY" -item 10 "Debugging:" DD 5
elif [ $2 = E ]; then "$GMS_REPLY" -item 12 "Encoding:" EE 6
elif [ $2 = U ]; then "$GMS_REPLY" -item 14 "Upper half >" UU 7
elif [ $2 = L ]; then "$GMS_REPLY" -item 15 "Lower half >" LL 8
else source "$GMS_SHELL/l_banner" -no_hot l_select $2; fi

# Build: -----

elif [ $1 = -build ]; then
"$GMS_REPLY" -top 3 "Select / Show"
"$GMS_REPLY" -item 4 " " " " "
"$GMS_REPLY" -item 5 "Quit dialog >" Q 1
"$GMS_REPLY" -item 6 " " " " "
"$GMS_REPLY" -item 7 "Animation $2" A 2
"$GMS_REPLY" -item 8 "Colors >" C 3
"$GMS_REPLY" -item 9 "Programs >" P 4
"$GMS_REPLY" -item 10 "Debugging:" D 5
if [ $GMS_DEBUG = X ]; then "$GMS_REPLY" -item 11 "$GMS_DEBUG" X
elif [ $GMS_DEBUG = Y ]; then "$GMS_REPLY" -item 11 "$GMS_DEBUG" Y
elif [ $GMS_DEBUG = Z ]; then "$GMS_REPLY" -item 11 "$GMS_DEBUG" Z
else "$GMS_REPLY" -item 11 "$GMS_DEBUG" 0; fi
"$GMS_REPLY" -item 12 "Encoding:" E 6
"$GMS_REPLY" -item 13 "$GMS_CODEPAGE" I
"$GMS_REPLY" -item 14 "Upper half >" U 7
"$GMS_REPLY" -item 15 "Lower half >" L 8
"$GMS_REPLY" -item 16 " " " " "
"$GMS_REPLY" -bottom 17
"$GMS_REPLY" -shadow 18

# Not found: -----

else
    source "$GMS_SHELL/l_banner" -no_action l_select $1
fi
```

I_wel

```
#!/bin/sh

# l_wel
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateL_WEL=20060927

# Prologue: =====

# Not running:

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_wel $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " l_wel ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9)" >> "$Z"
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" \
    -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug l_wel $1 $2 $3 $4
fi

# Chapters: =====

# Not defined:

if [ "$1" = "" ]; then source "$GMS_SHELL/l_banner" -no_action l_wel

# Build: -----

elif [ $1 = -build ]; then
# Cycle 1:
"$GMS_REPLY" -shadow 1
REPLY_SIZE=26
"$GMS_REPLY" -shadow 2
"$GMS_REPLY" -shadow 24
# Cycle 2:
"$GMS_REPLY" -top 1
"$GMS_REPLY" -shadow 2
"$GMS_REPLY" -bottom 23
# Cycle 3:
"$GMS_REPLY" -shadow 3
"$GMS_REPLY" -item 2 "<Welcome>"
"$GMS_REPLY" -item 22 "</Welcome>"
# Cycle 4:
"$GMS_REPLY" -shadow 4
REPLY_OFFSET=2
"$GMS_REPLY" -bottom 3
REPLY_OFFSET=5
REPLY_SIZE=23
"$GMS_REPLY" -item 3
REPLY_OFFSET=2
REPLY_SIZE=4
"$GMS_REPLY" -bottom 21
REPLY_OFFSET=5
REPLY_SIZE=23
"$GMS_REPLY" -item 21
# Cycle 5:
"$GMS_REPLY" -shadow 5
"$GMS_REPLY" -item 4 "to"
"$GMS_REPLY" -item 20 "Enjoy!"
# Cycle 6:
"$GMS_REPLY" -shadow 6
"$GMS_REPLY" -item 5
"$GMS_REPLY" -item 19
# Cycle 7:
"$GMS_REPLY" -shadow 7
"$GMS_REPLY" -item 6 "<Gerolf>"
"$GMS_REPLY" -item 18 "</Gerolf>"
# Cycle 8:
"$GMS_REPLY" -shadow 8
REPLY_SIZE=4
"$GMS_REPLY" -bottom 7
REPLY_OFFSET=8
REPLY_SIZE=20
"$GMS_REPLY" -item 7
REPLY_OFFSET=5
REPLY_SIZE=4
"$GMS_REPLY" -bottom 17
REPLY_OFFSET=8
REPLY_SIZE=20
"$GMS_REPLY" -item 17
# Cycle 9:
"$GMS_REPLY" -shadow 9
"$GMS_REPLY" -item 8 "Markup"
"$GMS_REPLY" -item 16 "$GMS_DATE" X X
# Cycle 10:
"$GMS_REPLY" -shadow 10
"$GMS_REPLY" -item 9
"$GMS_REPLY" -item 15
# Cycle 11:
"$GMS_REPLY" -shadow 11
"$GMS_REPLY" -item 10 "<Shredder>"
"$GMS_REPLY" -item 14 "</Shredder>"
# Cycle 12:
"$GMS_REPLY" -shadow 12
REPLY_SIZE=4
"$GMS_REPLY" -bottom 11
REPLY_OFFSET=11
REPLY_SIZE=17
"$GMS_REPLY" -item 11
# Cycle 13:
REPLY_OFFSET=8
REPLY_SIZE=4
"$GMS_REPLY" -bottom 13
REPLY_OFFSET=11
REPLY_SIZE=17
"$GMS_REPLY" -item 13
"$GMS_REPLY" -item 12 "Version $GMS_VERSION" X X
# Wait:
REPLY_OFFSET=2
REPLY_SIZE=26
"$GMS_REPLY" -item 2 "<Welcome>"
"$GMS_REPLY" -top 1 "- XX

# Remove cycle: -----

elif [ $1 = -cyc_remove ]; then
"$GMS_REPLY" -stripe $2 $REPLY_PATTERN $REPLY_LETTER
"$GMS_REPLY" -shadow $3
"$GMS_REPLY" -stripe $4 $REPLY_PATTERN $REPLY_LETTER

# Remove: -----

elif [ $1 = -remove ]; then
# Cycle 1:
REPLY_OFFSET=8
REPLY_SIZE=20
"$GMS_REPLY" -shadow 12
# Cycles 2 - 12:
source "$GMS_SHELL/l_wel" -cyc_remove 13 11 12
```

```
source "$GMS_SHELL/l_wel" -cyc_remove 14 10 11
source "$GMS_SHELL/l_wel" -cyc_remove 15 9 10
REPLY_OFFSET=5
REPLY_SIZE=23
source "$GMS_SHELL/l_wel" -cyc_remove 16 8 9
source "$GMS_SHELL/l_wel" -cyc_remove 17 7 8
source "$GMS_SHELL/l_wel" -cyc_remove 18 6 7
source "$GMS_SHELL/l_wel" -cyc_remove 19 5 6
REPLY_OFFSET=2
REPLY_SIZE=26
source "$GMS_SHELL/l_wel" -cyc_remove 20 4 5
source "$GMS_SHELL/l_wel" -cyc_remove 21 3 4
```

```
source "$GMS_SHELL/l_wel" -cyc_remove 22 2 3
source "$GMS_SHELL/l_wel" -cyc_remove 23 1 2
# Cycle 13:
"$GMS_REPLY" -stripe 24 $REPLY_PATTERN $REPLY_LETTER
"$GMS_REPLY" -stripe 1 $REPLY_PATTERN $REPLY_LETTER

# Not found: -----
else
source "$GMS_SHELL/l_banner" -no_action l_wel $1
fi
```

shredder


```

#!/bin/sh

# shredder
# =====

# This file is part of Gerolf Markup Shredder,
# written by G. D. Brettschneider (1999-2006).

GMSdateSHREDDER=20060927

# Prologue: =====

# Not running:

if [ "$GMS_SHELL" = "" ]; then
    echo "This is Gerolf Markup Shredder (shredder)."
    sleep 1
    exit
fi

# Check if module is given: -----

if [ "$1" = "" ]; then
    if [ -f ../gerolf ]; then
        # Load launcher (if not loaded by launcher):
        source ../gerolf -passive
        export REPLY_MODULE=g_palet
        export REPLY_ACTION=-build
    else
        # Not set up:
        echo
        echo "Please run 'gmssetup' script to initialize Gerolf Markup Shredder."
        exit
    fi
fi

# Help message: -----

if [ $1 = -help \
    -o $1 = --help ]; then
    echo
    echo -n "This is Gerolf Markup Shredder, "
    echo "written by G. D. Brettschneider (www.Gerolf.org) "
    echo -n "Please send bug reports or corrections to "
    echo "MarkupShredder@Gerolf.org - Thank you. "
    exit
else
    # Initialize loop:
    export REPLY_MODULE="$1"
    export REPLY_ACTION="$2"
fi

# Debug: -----

if [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
    source "$GMS_SHELL/l_banner" -debug g_vars $1 $2 $3 $4
elif [ "$GMS_DEBUG" = "Z" ]; then
    echo " shredder ($1) ($2) ($3) ($4) ($5) ($6) ($7) ($8) ($9) >> "$Z"
fi

# Build: -----

export REPLY_TEMP="$GMS_TEMP/replytmp"
if [ -f "$REPLY_TEMP" ]; then rm "$REPLY_TEMP"; fi
export REPLY_MODULE=g_palet
export REPLY_ACTION=-build
export REPLY_OFFSET=0
export REPLY_SIZE=0
export REPLY_ITEMS=0
export GMS_REPLY="$GMS_BINARIES/reply"
"$GMS_REPLY" -random 2

export GMS_RECEIVE=1
export GMS_TEXTMODE=1

# Textmode interface master loop: =====

if [ "$GMS_DEBUG" = "Z" ]; then
    echo " ===== Entering textmode interface master loop" >> "$Z"
fi

while [ "$GMS_BREAK" = "" ]; do

    export REPLY_DATA=

    # Load and remove old temporary file 'replytmp': -----

    if [ "$GMS_RECEIVE" = "1" ]; then
        if [ -f "$REPLY_TEMP" ]; then
            source "$REPLY_TEMP"
            rm "$REPLY_TEMP"
            export GMS_RECEIVE=0
        fi
    fi

    # Debug: -----

    if [ "$GMS_DEBUG" = "Z" ]; then
        echo -n " ----- shredder" >> "$Z"
        echo -n " [$REPLY_MODULE] [$REPLY_ACTION]" >> "$Z"
        echo " [$REPLY_HOT] [$REPLY_COLD] [$REPLY_DATA]" >> "$Z"
    elif [ "$GMS_DEBUG" = "X" -o "$GMS_DEBUG" = "Y" ]; then
        source "$GMS_SHELL/l_banner" -debug \
            $REPLY_MODULE $REPLY_ACTION \
            $REPLY_HOT $REPLY_COLD $REPLY_DATA
    fi

    # Load module: -----

    if [ -f "$GMS_SHELL/$REPLY_MODULE" ]; then
        source "$GMS_SHELL/$REPLY_MODULE" $REPLY_ACTION \
            $REPLY_HOT $REPLY_COLD $REPLY_DATA
    else
        # Not found: -----

        source "$GMS_SHELL/l_banner" -no_module $REPLY_MODULE shredder
    fi

done

# gmsdebug.log footer:

if [ "$GMS_DEBUG" = "Z" ]; then
    echo " ===== Leaving textmode interface master loop" >> "$Z"
    arg="/////////////////////////////////////"
    echo " $arg$arg" >> "$Z"
    unset arg
    echo " $GMS_SETTING" >> "$Z"
fi

# Final screen:

setterm -reset
echo -n " Please support the author of Markup Shredder  -"
echo " www.Gerolf.org"
for i in 1 2 3 4 5 6 7 8 9 10 11; do echo; done
for i in 1 2 3 4 5 6 7 8 9 10 11; do echo; done

# Final unsets:

export GMS_BREAK=
export GMS_DEBUG=
export GMS_SETTING=
export Z=

```

[GMS_ROOT]/tex/gerolf

TeX Macro Scripts

gerolf.tex

```

% gerolf.tex
% =====

% This file is part of Gerolf Markup Shredder,          <!-- ##### -->
% written by G. D. Brettschneider (1999-2008). All rights reserved.
% Send corrections to: MarkupShredder(at)Gerolf.org (www.Gerolf.org)

% TeX programming literature:
% Knuth, Donald Ervin: The TeXbook (1984-1996),
% Addison-Wesley Publishing Comp., ISBN 0-201-13448-9
% Schwarz, Norbert: Einfuehrung in TeX (1991-1993),
% Addison-Wesley Verlag GmbH, ISBN 3-89319-345-6
% Appelt, Wolfgang: TeX fuer Fortgeschrittene (1988),
% Addison-Wesley Verlag GmbH, ISBN 3-89319-115-1

% 1. Basics #####

% Load third - party macros:

\input prologue.cfg

% An "\input plain.tex" command is expected to be found in "prologue.cfg".
% Plain TeX may be replaced by another file that re-defines a few basic macros
% needed by GMS, e. g. \newbox, \newcount, \newdimen, \newtoct, \newrite,
% \newif, \loop etc.
% In Plain TeX, the "\input hyphen" command should be disabled, because GMS
% too has a pattern loading mechanism. Also, font preloading should be
% disabled.

% Character category definitions: -----

% compare Knuth, pages 343-344:
\chardef \CATescape = 0
\chardef \CATopengroup = 1
\chardef \CATclosegroup = 2
\chardef \CATmathmode = 3
\chardef \CATalignment = 4
\chardef \CATendofline = 5
\chardef \CATparameter = 6
\chardef \CATsuperscript = 7
\chardef \CATsubscript = 8
\chardef \CATignore = 9
\chardef \CATwhitespace = 10
\chardef \CATletter = 11
\chardef \CATother = 12
\chardef \CATactive = 13
\chardef \CATcomment = 14
\chardef \CATinvalid = 15

% Basic category code assignments: -----

% This is necessary for defining macros:
\catcode \{ = \CATopengroup
\catcode \} = \CATclosegroup
\catcode \# = \CATparameter

\def \gcatcode {\global \catcode}

\def \CATactiveslessthan {\catcode \< = \CATactive}
\def \CATclosegroupclosebracket {\catcode \} = \CATclosegroup}
\def \CATcommentbar {\catcode \# = \CATcomment}

\def \CATcommenthashmark {\catcode \# = \CATcomment}
\def \CATescapelt {\catcode \< = \CATescape}
\def \CATletterexclamation {\catcode \! = \CATletter}
\def \CATletterhyphen {\catcode \- = \CATletter}
\def \CATothergt {\catcode \> = \CATother}
\def \CATopengroupopenbracket {\catcode \[ = \CATopengroup}
\def \CATotherampersand {\catcode \& = \CATother}
\def \CATotherclosebrace {\catcode \} = \CATother}
\def \CATotherhashmark {\catcode \# = \CATother}
\def \CATotherlessthan {\catcode \< = \CATother}
\def \CATotheropenbrace {\catcode \{ = \CATother}
\def \CATotherpercent {\catcode \% = \CATother}
\def \CATothertilde {\catcode \~ = \CATother}
\def \CATparameterat {\catcode \@ = \CATparameter}
\def \CATparameterdegree {\catcode \^ = \CATparameter}
\def \CATwhitespacegt {\catcode \> = \CATwhitespace}
\def \CATwhitespaceslash {\catcode \\/ = \CATwhitespace}

\def \CATentity {%
  \CATescapelt \CATletterexclamation \CATletterhyphen
  \CATotherampersand \CATotherhashmark \CATothergt \CATparameterat}

\def \CATreference {%
  \CATotherhashmark \CATotherampersand \CATparameterdegree}

\def \CATstylegap {%
  \CATclosegroupclosebracket \CATopengroupopenbracket
  \CATotherclosebrace \CATotheropenbrace}

% This is what initex and "plain.tex" do (Characters not mentioned here
% are assumed to be letters (A-Z, a-z) or other):

\def \endmarkupCAT {%
  \global \catcode 0 = \CATignore % nul
  \global \catcode 1 = \CATsubscript % start of heading
  \global \catcode 9 = \CATwhitespace % horizontal tabulator
  \global \catcode 11 = \CATsuperscript % vertical tabulator
  \global \catcode 12 = \CATactive % form feed
  \global \catcode 13 = \CATendofline % carriage return
  \global \catcode \ = \CATwhitespace % &32#;
  \global \catcode \# = \CATparameter % &35#;
  \global \catcode \$ = \CATmathmode % &36#;
  \global \catcode \% = \CATcomment % &37#;
  \global \catcode \& = \CATalignment % &38#;
  \global \catcode \< = \CATother % &60;
  \global \catcode \| = \CATescape % &92#;
  \global \catcode \^ = \CATsuperscript % &94#;
  \global \catcode \_ = \CATsubscript % &95#;
  \global \catcode \{ = \CATopengroup % &123#;
  \global \catcode \} = \CATclosegroup % &125#;
  \global \catcode \~ = \CATactive % &126#;
  \global \catcode 127 = \CATinvalid % delete}

\endmarkupCAT
\outer \def ^^L{\par} % &#12; form feed
\def \activateshreder {\catcode \< = \CATactive
  \catcode \# = \CATother \catcode \~ = \CATother}
\activateshreder
\def \enableshreder {\let < = \TAGget}
\def \shredder {\activateshreder \enableshreder}

```

```

\endmarkupCAT

% So far, the "plain.tex" assignments will not interfere with Markup Shredder.

\let \CHARampersand \relax
\let \CHARhashmark \relax

\def \preamble {\CHARhashmark \CHARampersand \CHARampersand \CHARhashmark}

\let \CHARampersand = &
\let \CHARhashmark = #

\def \beginhalign {\halign \expandafter \bgroup \preamble \cr}

\def \expandthree {\expandafter \expandafter \expandafter}

% Character category changes for input files using html-style markup: -----

% The less-than sign "<" must be made an active character, because texts using
% "html-style" markup are expected as input. Other TeX macro files must be
% loaded before this is done (via "epilogue.cfg").

% This will be executed by the <html> start tag ("<" is no longer active then
% <!-- in order to allow comments that include <tags> like this one-->):

\def \beginmarkupCAT {%
  \global \catcode 0 = \CATOther % nul
  \global \catcode 1 = \CATOther % start of heading
  \global \catcode 9 = \CATwhitespace % horizontal tabulator
  \global \catcode 11 = \CATOther % vertical tabulator
  \global \catcode 12 = \CATOther % form feed
  \global \catcode 13 = \CATwhitespace % carriage return
  \global \catcode ` \ = \CATwhitespace % &32#;
  \global \catcode `# = \CATOther % &35#;
  \global \catcode `$ = \CATOther % &36#;
  \global \catcode `% = \CATOther % &37#;
  \global \catcode `& = \CATOther % &38#;
  \global \catcode `< = \CATactive % &60;
  \global \catcode `| = \CATOther % &92#;
  \global \catcode `^ = \CATOther % &94#;
  \global \catcode `\_ = \CATOther % &95#;
  \global \catcode `{ = \CATOther % &123#;
  \global \catcode `} = \CATOther % &125#;
  \global \catcode `~ = \CATOther % &126#;
  \global \catcode 127 = \CATOther % delete

% This is used locally, e. g. to skip JavaScripts:

\def \beginscriptCAT {%
  \catcode `# = \CATOther % &35#;
  \catcode `$ = \CATOther % &36#;
  \catcode `% = \CATOther % &37#;
  \catcode `& = \CATOther % &38#;
  \catcode `< = \CATOther % &60;
  \catcode `| = \CATOther % &92#;
  \catcode `^ = \CATOther % &94#;
  \catcode `\_ = \CATOther % &95#;
  \catcode `{ = \CATOther % &123#;
  \catcode `} = \CATOther % &125#;
  \catcode `~ = \CATOther % &126#;

% Information about this software: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\def \GMSformat {gerolf}
\def \GMSname {Gerolf Markup Shredder}
\def \GMSversion {0.08a}% <!-- ##### -->
\def \GMSdate {20080107}% <!-- ##### -->
\def \GMSauthor {G. D. Brettschneider}

\def \GMSaddress {Luchtbergstrasse 27, D-28237 Bremen, Germany}
\def \GMSdomain {www.Gerolf.org}
\def \GMSemail {MarkupShredder(a)Gerolf.org}

\def \GMScopyright {% <!-- ##### -->
  Copyright (c) 1999 - 2008 by
  Gerolf Diethelm Brettschneider,
  Luchtbergstrasse 27, D-28237 Bremen.
  All rights reserved. This GMS
  software comes without ANY warranty.
  You may freely distribute and use it.}

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\tracingstats = 1

% Lines and messages: -----

\def \CONspace { }
\def \CONhyphen {-}
\def \CONhalfline {-----}
\def \CONline {\CONhalfline \CONhalfline}
\def \CONlongline {\CONline \CONline \CONline \CONline \CONline}
\def \CONhalfLINE {=====}
\def \CONLINE {\CONhalfLINE \CONhalfLINE}
\def \CONlongLINE {\CONLINE \CONLINE \CONLINE \CONLINE \CONLINE}

\def \predashes {\ifcase \count 1
  \or -\or --\or ---\or \tempnumber
  \or ----\or --\tempnumber \else ?\fi}
\def \postdashes {\ifcase \count 4
  \or -\or --\or ---\or ----\or -----\or
  \CONhalfline \or \CONhalfline -\or
  \CONhalfline --\or \CONhalfline ---\or
  \CONhalfline ----\or \CONhalfline -----\or
  \CONline \or -\CONline \else ?\fi}

\def \dotmessage {\message {}}
\def \hyphenmessage {\message {-}}
\def \fillmessage #1#2#3#4{%
  \count 1 = #1 \count 4 = #4
  \def \tempsecond {#2}%
  \ifx \tempsecond \CONhyphen \def \tempnumber {----}%
  \else \def \tempnumber {#2 -}\fi
  \message {\predashes <#3>\postdashes \CONlongline}}

\def \GMScodepage {CP1252}
\def \GMSwelcome {\fillmessage 0-{GMS}{13}}
\def \GMSgoodbye {\fillmessage 0-{GMS}{12}}
\def \GMSbeginabout {\message {This is \GMSname,
  Version \GMSversion, written by \GMSauthor}}
\def \GMSendabout {\message {Release: \GMSdate \CONspace |
  Domain: \GMSdomain \CONspace | Mail: \GMSemail}}

% Load interface and modules: -----

\GMSwelcome
\GMSbeginabout
\fillmessage 2-{initialize}4
\fillmessage 41{interface}3

% Fixme: lowercase functions without uppercase prefix should be backed up

\let \mef \mathchardef

\let \neb \newbox
\let \nec \newcount
\let \ned \newdimen
\let \net \newtoks

```

```

\let \new \newif

% Variables: -----

\net \FONTmonospace
\net \FONTsansserif
\net \FONTserif
\let \LINEheight \baselineskip
\ned \LINEheightpageno
\def \LINEupskip {\vskip -\LINEheight}
\ned \TEXTheight
\ned \TEXTwidth

\new \ifCLASSpredefined
\new \ifEND
\new \ifGEDCOM
\new \ifLINEheightgiven
\new \ifOPENbouter
\new \ifOPENcomment
\new \ifOPENdlcompact
\new \ifOPENfloatsuppress
\new \ifOPENfontinner
\new \ifOPENfontouter
\new \ifOPENform
\new \ifOPENh
\new \ifOPENiouter
\new \ifOPENparinner
\new \ifOPENparouter
\new \ifOPENstyle
\new \ifSCRIPTother
\new \ifTEXTindentgiven

\let \HYPHENTables \empty
\let \INDENTinputtype \empty
\let \INDENTinputvalue \empty
\let \NAMESPACE \empty
\let \OBJECTtype \empty
\let \POSTfamilies \empty
\def \REFeuro {80}
\let \SCRIPTtype \empty
\let \TEXTindent \parindent

% Literature: Donald Ervin Knuth, "The TeXbook" (1984-1996), Appendix B and D.

% Abbreviations: -----

\def \gadvance {\global \advance}
\def \glet {\global \let}

\def \ldef {\long \def}
\def \gldef {\global \long \def}

\def \begintrace {\global \tracingmacros = 3 \global \tracingcommands = 1 }
\def \endtrace {\global \tracingmacros = 0 \global \tracingcommands = 0 }

% Indirect definitions: -----

% These definitions can have non-letters in their names (see Knuth, page 375):

\def \Def #1 #2{%
  \csname expandafter\endcsname \def \csname #1\endcsname {#2}}
\def \gDef #1 #2{%
  \csname expandafter\endcsname \gdef \csname #1\endcsname {#2}}
\def \eDef #1 #2{%
  \csname expandafter\endcsname \edef \csname #1\endcsname {#2}}
\ldef \lDef #1 #2{%
  \csname expandafter\endcsname \ldef \csname #1\endcsname {#2}}
\ldef \glDef #1 #2{%
  \csname expandafter\endcsname \gldef \csname #1\endcsname {#2}}

\def \unDef #1{% clear memory
  \csname expandafter\endcsname \let \csname #1\endcsname \undefined}
\def \xDef #1 #2{%
  \csname expandafter\endcsname \xdef \csname #1\endcsname {#2}}

% Search an expandable command (control sequence definition check): -----

% compare Knuth, pages 40 and 308; similar to e-TeX's \ifcsname...\endcsname:

\new \ifDEF

\def \DEFsearch #1{\expandafter
  \ifx \csname #1\endcsname \relax \DEFfalse \else \DEFtrue \fi}

\def \blap #1{\vbox to 0pt {#1\vss}}
\def \tlap #1{\vbox to 0pt {\vss #1}}

\bgroup
  \catcode\^ = \CATsuperscript
  \gdef \^M{\ } % carriage return = space
\egroup % Fixme: move

% System functions: -----

% Log file messages:
\def \echo #1{\immediate \backupwrite 16 {#1}}
\def \noecho {\echo {}}

% System commands:
\def \system #1{\immediate \backupwrite 18 {#1}}

% Count files on Dos:
\def \NOfilemax {1234567}% may be redefined to e.g. 16 on Dos via 'files.cfg'

\def \Message #1{\message {<#1>}}

\ifx \pdfpagewidth \undefined
  \echo {! Unsupported: PDF output (pdfTeX required). Writing DVI instead.}%
\fi

% 2. Functions #####

% ALIGN: =====

% compare Knuth, pages 107, 317, 353 and 356:

\new \ifALGN
\new \ifALGNcenter
\new \ifALGNleft
\new \ifALGNright

\def \ALGNtext #1{\edef \algnZ {#1}%
  \ALGNleftfalse \ALGNcenterfalse \ALGNrightfalse
  \ifx \algnZ \CONleft \algnB \ALGNlefttrue
  \else \ifx \algnZ \CONcenter \algnC \ALGNcentertrue
  \else \ifx \algnZ \CONright \algnA \ALGNrighttrue
  \else \ifx \algnZ \CONmiddle \CHKerryoumeantcenter
    \algnC \ALGNcentertrue \fi \fi \fi \fi}
\def \algnA {\leftskip = 0pt plus 2em \algnD} % left
\def \algnB {\rightskip = 0pt plus 2em \algnD} % right
\def \algnC {\leftskip = 0pt plus 2em
  \rightskip = \leftskip \algnD} % center
\def \algnD {\parindent = 0pt \parfillskip = \parindent
  \ALGNtextrespace \hbadness = 10000 \relax} % any
\def \ALGNtextbyvalue {\VALgetlc \ALGNtext \VALlowercase}
\def \ALGNtextend {\ifALGNcenter \hskip \parfillskip \else
  \ifx \BIDI \CONltr \ifALGNright \else \hfill \fi \fi \fi}
\def \ALGNtextrespace {\spaceskip = 0.3333em \xspaceskip = 0.5em }

```

```

% ANCR: =====
% \Def \CONSTYLE a:link {\CLRsetlink} % Fixme
% \Def \CONSTYLE a:visited {\CLRsetvlink}
% Fixme: line and page breaks, outsource \GAPwrite
% Fixme: This may produce masses of control sequences.
% Define limit of destinations to be checked in typeset.cfg

\def \ANCRadd {
  \DEFsearch {\CONdest \ancrAZ}\ifDEF \CHKerrduplicate \ancrAZ
  \else \pdfdest name {\ancrAZ} fit\relax % fith, fitv, fitb, fitbh, fitbv
  \xDef {\CONdest \ancrAZ} {}fi
  \ifx \pdfdest \undefined \let \ANCRadd \relax \fi
}
\def \ANCRbegin {% begin anchor <a>
  \ifLINKunrendered \else \ifLINKunlinked \else
  \ifx \URLhref \empty \ifx \LINKname \empty
  \ifx \LINKid \empty \CHKerrunanchored \fi
  \else \ANCRbyname \LINKname \fi
  \else %% \ifx \LINKname \empty \else \ANCRbyname \LINKname \fi % Fixme
  \ancrA \URLhref \fi \fi \fi
}
\def \ancrA #1{\edef \ancrAZ {#1}\ifx \ancrAZ \empty \else \ancrC \fi}
\def \ancrB {\ANCRdtrue \hskip 0mm % get into horizontal mode
  \bgroup \pdflinkmargin = 0ex \ANCRdtrue
  \ifANCRheightgiven \ancrBA \else \ancrBB \fi
  \ifx \pdfdest \undefined \let \ancrB \relax \fi
}
\def \ancrBA {% height given
  \pdfstartlink height \ANCRheight \CONspace depth 0pt \CONspace}
\ifx \pdfdest \undefined \let \ancrBA \relax \fi
\def \ancrBB {% no height given
  \pdfstartlink height 0.7\LINEheight %% Fixme: differs from \strut
  \CONspace depth 0.3\LINEheight \CONspace} %%
  \ifx \pdfdest \undefined \let \ancrBB \relax \fi
}
\def \ancrC {\let \ANCRext \empty \let \ANCRint \empty
  \URLinternetfalse \URLget \ancrAZ
  \ifURLinternet \ancrCA \else \ancrD \fi} % href
\def \ancrCA {\ifx \ANCRext \ANCRint \else % href ext.
  \edef \ANCRext {\ANCRext \CONhashmark \ANCRint} \fi \ancrCAA \ANCRext}
\def \ancrCAA #1{% start external destination
  \ancrB attr {/C [the \CLRanchorext] /Border [0 0 1] }%
  user {{Subtype /Link /A << /Type /Action /S /URI /URI (#1) >>}}%
  \egroup}
\def \ancrD {\ifx \ANCRint \empty \ancrDA \else \ancrDB \ANCRint \fi} % int.
\def \ancrDA {\expandafter \CHNcopyrelativepath \ancrAZ \to \ANCRint
  \ancrDB \ANCRint} % internal href empty
\def \ancrDB #1{\ancrB attr {/C [the \CLRanchorint] /Border [0 0 1] }%
  goto name {#1}\egroup} % internal destination
\let \ANCRbegext \ancrCA
\def \ANCRbychain #1{% <link>
  \edef \ancrAZ {#1}\ifx \ancrAZ \empty \else \ANCRadd \fi}
\def \ANCRbyname #1{% <a name = "...", <link id = "...">
  \edef \ancrAZ {#1}\ifx \ancrAZ \empty \else \ANCRadd \fi}
\new \ifANCRed
\def \ANCRend {\ifLINKunrendered \else \ifLINKunlinked \else % </a>
  \ifx \URLhref \empty \else
  \ifmode \leavevmode \fi \ANCRendA \fi % Fixme: not if nameLINK
  \fi \fi \ifANCRed \ANCRendA \fi}
\def \ANCRendA {\GAPwrite \ifANCRed \pdfendlink \fi \ANCRendfalse}
\ifx \pdfdest \undefined \let \ANCRendA \relax \fi

\def \ANCRgedcom {\ANCRinit \ELEMENTvalueget \ANCRbegin
  \global \GAP = {>}\GAPwrite \ANCRend}

\ned \ANCRheight \new \ifANCRheightgiven

\def \ANCRinit {\glet \LINKname \empty \glet \URLhref \empty} % <a>

% ARRAY: =====
\net \ARRAY \net \ARRAYentry
\def \ARRAYafter #1{\ARRAY = \expandafter {#1}%
  \xdef #1{\the \ARRAY \the \ARRAYentry}}
\def \ARRAYbefore #1{\ARRAY = \expandafter {#1}%
  \xdef #1{\the \ARRAYentry \the \ARRAY}}

\ldef \ARRAYone [#1] {%
  \ARRAYentry = {\ \ #1} }
\ldef \ARRAYtwo [#1] [#2] {%
  \ARRAYentry = {\ \ #1} [#2] } }
\ldef \ARRAYthree [#1] [#2] [#3] {%
  \ARRAYentry = {\ \ #1} [#2] [#3] } }
\ldef \ARRAYfour [#1] [#2] [#3] [#4] {%
  \ARRAYentry = {\ \ #1} [#2] [#3] [#4] } }
\ldef \ARRAYfive [#1] [#2] [#3] [#4] [#5] {%
  \ARRAYentry = {\ \ #1} [#2] [#3] [#4] [#5] } }
\ldef \ARRAYsix [#1] [#2] [#3] [#4] [#5] [#6] {%
  \ARRAYentry = {\ \ #1} [#2] [#3] [#4] [#5] [#6] } }
\ldef \ARRAYseven [#1] [#2] [#3] [#4] [#5] [#6] [#7] {%
  \ARRAYentry = {\ \ #1} [#2] [#3] [#4] [#5] [#6] [#7] } }
\ldef \ARRAYeight [#1] [#2] [#3] [#4] [#5] [#6] [#7] [#8] {%
  \ARRAYentry = {\ \ #1} [#2] [#3] [#4] [#5] [#6] [#7] [#8] } }
\ldef \ARRAYnine [#1] [#2] [#3] [#4] [#5] [#6] [#7] [#8] [#9] {%
  \ARRAYentry = {\ \ #1} [#2] [#3] [#4] [#5] [#6] [#7] [#8] [#9] } }

% ATT: =====
\new \ifATT % Section restructured (20070202)

\let \ATTactual \relax

\let \attWX \empty
\let \attWXung \empty

\let \attYtest \empty
\let \attZ \empty

\def \ATTget {\edef \attYtest {\the \ATTlist}%
  \expandafter \attA \attYtest ... = "GMS-STOP"}
\ldef \attA #1#2=#3#4{\lowercase {\edef \attWX {#1#2}}%
  \edef \attY {#3}\edef \attZ {#4}}%
  \ifx \attZ \CONpar \let \attZ \CONthatsall \fi
  \let \ATTactual \attA \let \attAnext \relax \ATTlistmissingquotesfalse
  \ifx \attZ \CONthatsall \ATTlistalltrue \else
  \ifx \attZ \CONstopmark \else \attAA #4 ... = \relax \fi \fi
  \ifx \attY \empty \let \attAnext \attC
  \else \ifx \attY \CONspace \let \attAnext \attC
  \else \ifx \attZ \empty \let \attAnext \attD
  \else \let \attAnext \attB \fi \fi \fi \attAnext
  \ifATTlistall \let \ATTactual \attF \fi \ATTactual}
\def \attAA #1 ... = #2\relax {\VAL = {#1}\edef \attAAZ {#2}%
  \ifx \attAAZ \CONthatsall \edef \attY {#1}\ATTlistalltrue
  \else \ATTlistallfalse \fi} % check
\def \attB {\ifx \attZ \CONstopmark \let \ATTactual \relax
  \let \attAfourthnext \attBB
  \else \let \attAfourthnext \attBA \fi \attAfourthnext} % parse attZ
\def \attBA {\edef \attBtest {%
  \attWX = \attY \CONspace \CONspace GMS-STOP} %
  \expandafter \attE \attBtest \VAL = \expandafter {\attZ} %
  \edef \attBtest {\attYETfirst \CONspace \attZ} %
  \expandafter \attG \attBtest \relax} % run
\def \attBB {\edef \attBtest {%
  \attWX = \attY \CONspace \CONspace GMS-STOP} %
  \expandafter \attE \attBtest} % stop
\def \attC {\ifx \attZ \CONstopmark
  \let \ATTactual \relax \else \attCA \fi} % attY empty
\def \attCA {\ifATTlistall \let \ATTactual \relax
  \else \let \ATTactual \attA \fi}

```

```

\ifATTLlistmissingquotes \else \VAL = \expandafter {\attZ}%
\edef \attSTOPdoprobe {\attWX \CONspace}%
\expandafter \attG \attSTOPdoprobe \relax \fi}
\def \attD {\ATTLlistmissingquotestruer % attY given
\edef \attY {\attY \CONspace} \expandafter \attDA \attY \relax
\edef \attYgiventest {\attWX \CONspace}%
\expandafter \attG \attYgiventest \relax
\ifx \attZ \CONstopmark \let \attYgiventest \attDC
\else \let \attYgiventest \attDB \fi \attYgiventest}
\def \attDA #1 #2 #3 \relax {% parse attY
\ifx \attWX \CONendmarkdots \VAL = {} \let \attYtest \empty
\else \DEFsearch {\CONatt \attWX}%
\ifDEF \attDAA {#1}{#2 #3}%
\else \DEFsearch {\CONtag \the \ELEMENT \CONatt \attWX}%
\ifDEF \attDAA {#1}{#2 #3} \else \attDAA {#2}{#3} \fi \fi \fi
\ifx \attYtest \empty \ATTungfalse
\else \ifx \attYtest \CONendmark \ATTungfalse
\else \ifx \attYtest \CONendmarktwo \ATTungfalse
\else \ifx \attYtest \CONendmarkequal \ATTungfalse
\else \ATTungtrue \fi \fi \fi \fi}
\def \attDAA #1#2{\VAL = \expandafter {#1} \edef \attYtest {#2}}% do
\def \attDB {\edef \test {\attWX = \attY \CONspace \CONspace GMS-STOP}%
\expandafter \attE \test}% run
\def \attDC {\edef \test {\attWX \CONspace \attY \CONspace ...}%
\expandafter \attDCA \test \relax}% stop
\def \attDCA #1 #2 ..#3 \relax {% parse stop
\VAL = {#2} \let \ATTactual \relax \edef \attDCAXY {#1 #2}%
\expandafter \attG \attDCAXY \relax \CHKerrunquot}
\ldef \attE #1=#2#3 {% parse unquoted
\def \attWXung {#1} \edef \attEY {#2}%
\ifx \attEY \CONapostroph \CHKerrapostrophized \attEA #3' \relax
\else \edef \attEYZ {#2#3} \attEB \fi
\ifx \attEY \CONpar \let \attEnext \relax \fi \attEnext}
\def \attEA #1#2 \relax {\edef \attEAY {#2} % apostrophe
\ifx \attEAY \empty %% \let \attEnext \attEAA
\let \attEnext \attEC %%
\else \let \attEnext \attEC \VAL = \expandafter {#1}%
\edef \attEAtest {\attWXung \CONspace #1}%
\expandafter \attG \attEAtest \relax \fi}
%% \ldef \attEAA #1#2GMS-STOP#3 {% Fixme: discarding value
%% \edef \attEAY {#3} \let \ATTactual \relax
%% \ifx \attEAY \CONquotmark \let \attEAY \relax \fi \attEAY}
\def \attEB {\ifx \attEYZ \CONstopmark \attEBB \else \attEBA \fi} % do
\def \attEBA {\VAL = \expandafter {\attEYZ}%
\edef \attEall {\attWXung \CONspace \attEYZ}%
\expandafter \attG \attEall \relax \CHKerrunquot
\ifATTLlistall \let \attEnext \attEC
\else \let \attEnext \attE \fi} % run
\def \attEBB {\ifx \attZ \CONstopmark
\let \attEnext \relax \else \attEBBA \fi} % stop
\def \attEBBA {\VAL = \expandafter {\attZ}%
\edef \attEall {\attWXung \CONspace \attZ} % pass stop
\expandafter \attG \attEall \relax \let \attEnext \relax}
\ldef \attEC #1GMS-STOP{\let \ATTactual \relax % gulp unquoted
\edef \attECZ {#1} % Fixme
\ifx \attECZ \CONspace \let \attECZ \attF
\else \ifx \attECZ \CONattstop \attECA \ATTLlistallfalse
\else \ifx \attECZ \CONattstopspace \attECA \ATTLlistallfalse
\else \attECA \fi \fi \fi \attECZ}
\def \attECA {\let \attECZ \relax}
\ldef \attF #1GMS-STOP#2{% stop unquoted
\edef \attFZ {#2} \let \ATTactual \relax
\ifx \attFZ \CONquotmark \let \attFZ \relax \fi \attFZ}
\let \attYETfirst \empty
\ldef \attG #1 #2 \relax {% get attribute
\lowercase {\edef \attYETfirst {#1}} \attH}
\def \attH {\let \attHnext \relax
\ifx \attYETfirst \CONstyle \PRPtrue \PRPlist = \VAL \else
\ifx \attYETfirst \CONclass \attJ \else \attK \fi \fi \attHnext} % do
\def \attJ {\VALgetlc \let \attYETclassdotest \VALlowercase
\DEFsearch {\CONCLASS \attYETclassdotest}%
\ifDEF \CLASSpredefinedtrue
\csname \CONCLASS \attYETclassdotest \endcsname \fi
\DEFsearch {\CONSTYLE \attYETclassdotest} \ifDEF \attJA \else
\DEFsearch {\CONSTYLE \the \ELEMENT . \attYETclassdotest}%
\ifDEF \attJB \else \ifCLASSpredefined \else
\CHKerrundefclass \attYETclassdotest \fi \fi \fi} % class
\def \attJA {\def \attHnext {\expandafter % property
\PRPlistget \csname \CONSTYLE \VALlowercase \endcsname}}
\def \attJB {\def \attHnext {\expandafter % style
\PRPlistget \csname \CONSTYLE \the \ELEMENT . \VALlowercase \endcsname}}
\def \attK {\DEFsearch {\CONtag \the \ELEMENT \CONatt \attYETfirst}%
\ifDEF \attKA \else \attKB \fi} % element
\def \attKA {\def \attHnext {\csname
\CONtag \the \ELEMENT \CONatt \attYETfirst \endcsname}} % next att.
\def \attKB {\DEFsearch {\CONatt \attYETfirst} % next element
\ifDEF \def \attHnext {\csname \CONatt \attYETfirst \endcsname} \else
\let \attHnext \relax \CHKerrunrecogattprop \attYETfirst \fi}
\net \ATTLlist \new \ifATTLlistall \new \ifATTLlistmissingquotes
\new \ifATTung \new \ifATTungmessage % attribute value unquoted
% BACKGROUND: =====
\new \ifBKGD
\def \BKGDclrsetbody {\edef \BKGDtest {\the \VAL \CONspace}%
\expandafter \BKGDclrsetbodyA \BKGDtest \relax}
\def \BKGDclrsetbodyA #1 #2 \relax {\edef \BKGDclrsetbodyAZ {#2}%
\ifx \BKGDclrsetbodyAZ \empty \BKGDgiventruer \CLRsetbg
\ifCLRgetanyfailed \CLRbgfalse \else
\CLRbgbodytrue \edef \CLRbgbody {\CLRbackground} \fi
\else \CHKerrunuppattprop \CONbackground \fi} % set background color
\def \BKGDgeturl #1{\expandafter \BKGDgeturlA #1 url (none) \relax}
\def \BKGDgeturlA #url1(#2) #3 \relax {\edef \BKGDsource {#2} \BKGDload}% parse
\new \ifBKGDgiven
\ned \BKGDheight
\def \BKGDload {\let \FILENAME \CONnone \let \FILEsource \BKGDsource
\IMGheight = 0mm \IMGwidth = 0mm \FILEget
\ifx \FILENAME \CONdotslashnone.jpg \global \BKGDfalse
\else \FILEsearch \FILENAME \ifFILEexist \BKGDloadA
\else \expandafter \FILEsearchimage \FILENAME .jpg \relax
\ifFILEexist \BKGDloadA \else \CHKerrabsent \FILENAME \fi \fi \fi}
\def \BKGDloadA {\xdef \BKGDimage {\FILENAME}%
\ifx \FILENAME \CONnone \else \chkWLMtotfileadvance
\ifFILEtoomany \else \BKGDloadAA \fi \fi}
\def \BKGDloadAA {\global \BKGDtrue \imgGA
\imgH \empty \empty % Fixme: rename
\setbox \IMGbox = \hbox {\imgC}%
\IMGwidth = \wd \IMGbox \IMGheight = \ht \IMGbox
\global \BKGDheight = \IMGheight \global \BKGDwidth = \IMGwidth}
\def \BKGDmake {\let \FILENAME \BKGDimage
\FILEsearch \FILENAME \ifFILEexist \imgH \BKGDheight \BKGDwidth
\vbox to 0mm {\vskip -\OUTmarginint \hbox to 0mm {\hskip -\OUTmarginint
\ifBKGDnorepx \BKGDmakeA \else \BKGDmakeB \fi \hss } \vss } \fi}}
\def \BKGDmakeA {% no repeat-x
\ifBKGDnorepx \hbox {\imgC} \else \BKGDmakeAA {\hbox {\imgC}} \fi}
\def \BKGDmakeAA #1{\vbox {\loop \ifdim \OUTheight > 0mm
#1 \advance \OUTheight by -\BKGDheight \repeat}}
\def \BKGDmakeB {% repeat-x
\ifBKGDnorepx \BKGDmakeBB \else \BKGDmakeAA \BKGDmakeBB \fi}

```

```

\def \BKGDmakeBB {\hbox {\loop \ifdim \OUTwidth > 0mm
  \imgC \advance \OUTwidth by -\BKGDwidth \repeat}} % Fixme: rename

\new \ifBKGDnorepx
\new \ifBKGDnorepy

\def \BKGDsetrepeat {\edef \BKGDrepeattest {\the \VAL}%
  \ifx \BKGDrepeattest \CONrepeattx
    \global \BKGDmorepfalse \global \BKGDnorepytrue
  \else \ifx \BKGDrepeattest \CONrepeaty
    \global \BKGDnorepxtrue \global \BKGDnorepyfalse
  \else \BKGDsetrepeatA \fi \fi}
\def \BKGDsetrepeatA {% no repeat
  \ifx \BKGDrepeattest \CONnorepeat
    \global \BKGDnorepxtrue \global \BKGDnorepytrue
  \else \global \BKGDmorepfalse \global \BKGDnorepyfalse \fi}

\def \BKGDsource {\edef \BKGDsource {\the \VAL}}

\ned \BKGDwidth

% BKP: =====
% Tags and gap contents must be backed up for analyzing tables (2005/10/21):
% Fixme: always clear memory afterwards

\new \ifBKP
\nec \BKPnest

\nec \BKPno \let \BKPnocols \empty \let \BKPnorows \empty % stack
\nec \BKPnotable \let \BKPnotables \empty % stack

\newif \ifBKPpop

\def \BKPpop {\global \BKPpopfalse \global \BKPno = 0 \bkbB
  \ifnum \BKPno < \BKPstop \bkbA \bkbRepeat \global \BKPno = 0 }
\def \bkbA {\gadvance \BKPno by 1 % recall tag
  \edef \bkbAP {\csname BKPt\the \BKPno \endcsname \CONspace}% probe
  \expandafter \bkbAA \bkbAP \relax
  \edef \bkbAP {\csname BKPg\the \BKPno \endcsname \empty}% recall gap
  \expandafter \bkbAB \bkbAP \empty \empty \empty \relax
  \ifbkbABend \else \expandafter \bkbAC \bkbAP \relax \fi \bkbAD \tagE}
\ldef \bkbAA #1#2 #3 \relax {\lowercase {\ELEMENT = #1#2}}% parse
  \edef \TAGgetCX {#1}\def \TAGgetCY {#1#2}% see \TAGgetC
  \def \TAGgetCY {#2}\def \TAGgetCZ {#3}\ATTLlist = {#3}}
\ldef \bkbAB #1#2#3#4 \relax {\def \bkbABtest {#1#2#3}% comment
  \ifx \bkbABtest \CONendcomment \bkbABendtrue
    \tagAfalse \tagAbegfalse \tagDtrue \xdef \GAPcontent {#4}%
    \GAPchkfilled \GAPchklog \GAPchkobey \GAPset {#4}%
  \else \bkbABendfalse \fi}
\newif \ifbkbABend % comment end
\ldef \bkbAC #1 \relax {% gap, see \GAPget
  \xdef \GAPcontent {#1}\GAPchkfilled \GAPchklog \GAPchkobey \GAPset {#1}}
\def \bkbAD {% clear
  \xDef {BKPt\the \BKPno} \empty \xDef {BKPg\the \BKPno} \empty}
\def \bkbB #1\bkbRepeat {\let \bkbRepeat \fi % loop
  \gdef \bkbBbody {#1}\bkbBA} % global to pass moziller.htm
\def \bkbBA {\bkbBbody % iterate (unnestable, see \loop in plain.tex)
  \let \bkbBnext \bkbBA \else \let \bkbBnext \relax \fi \bkbBnext}

\def \BKPpush #1{\iftagA \ifBKP \bkbC \fi \fi
  \edef \BKPelement {\the \ELEMENT}%
  \ifx \BKPelement \CONtable \bkbD \else \ifx \BKPelement \CONtd \bkbE \else
  \ifx \BKPelement \CONth \bkbE \else \ifx \BKPelement \CONtr \bkbF \else
  \ifx \BKPelement \CONendtable \bkbG \fi \fi \fi \fi \fi
  \ifBKP \let #1 \relax \bkbH \fi \ifBKPpop \let #1 \relax \BKPpop \fi}
\def \bkbC {\ELEMENT = \expandafter {\CONcomment}} % comment

\def \bkbD {\gadvance \BKPnotable by 1 % start
  \expandafter \STACKxpush \the \BKPnotable \to \BKPnotables
  \expandafter \STACKxpush \the \BKPtr \to \BKPnorows
  \expandafter \STACKxpush \the \BKPtd \to \BKPnocols
  \gadvance \BKPnest by 1
  \global \BKPtd = 0 \global \BKPtdmax = 0 \global \BKPtr = 0
  \ifnum \BKPnest = 1 \global \BKPtrue \xdef \BKPstart {\the \BKPno} \fi}
\def \bkbE {\gadvance \BKPtd by 1
  \ifnum \BKPtd > \BKPtdmax \global \BKPtdmax = \BKPtd \fi} % td
\def \bkbF {\global \BKPtd = 0 \gadvance \BKPtr by 1} % tr
\def \bkbG {\gadvance \BKPnest by -1
  \STACKxpopnumber \BKPnotable \from \BKPnotables
  \xDef {TROWS\the \BKPnotable} {\the \BKPtr}%
  \STACKxpopnumber \BKPtr \from \BKPnorows
  \xDef {TCOLS\the \BKPnotable} {\the \BKPtdmax}%
  \STACKxpopnumber \BKPtd \from \BKPnocols \SPANrowclear
  \ifnum \BKPnest = 0 \global \BKPpoptrue \global \BKPfalse
  \bkbH \tagAbegfalse \xdef \BKPstop {\the \BKPno} \fi} % stop
\def \bkbH {% exec: create array of tags and gaps
  \iftagAbeg \iftagD \else \let \BKPelement \CONcomment \fi \fi
  \gadvance \BKPno by 1
  \glet \BKPtaglast \BKPtagnew \glet \BKPtagnew \BKPelement
  \ifx \BKPtaglast \CONcomment \ifx \BKPtagnew \CONcomment \GAP = {} \fi \fi
  \xDef {BKPt\the \BKPno} {\BKPelement \CONspace \TAGgetCZ}%
  \xDef {BKPg\the \BKPno} {\the \GAP}}

\nec \BKPtd \nec \BKPtdmax
\nec \BKPtr

% BIDI: =====
\ifx \TeXeTstate \undefined
  \message {No eTeX: Unidirectional typesetting. -}%
  \csname newcount\endcsname \TeXeTstate \fi

\def \BIDI {ltr}

\ifx \eTeXversion \undefined
  \let \BIDIbegin \empty \let \BIDIbeginone \empty
  \let \BIDIend \empty \let \BIDIendone \empty
  \let \BIDIset \empty \let \BIDIsetone \empty
  \else
  \def \BIDIbegin {\ifnum \TeXeTstate = 2 \beginR \fi}
  \def \BIDIbeginone {\ifnum \TeXeTstate = 1 \beginR \fi}
  \def \BIDIend {\ifnum \TeXeTstate = 2 \endR \fi}
  \def \BIDIendone {\ifnum \TeXeTstate = 1 \endR \fi}
  \def \BIDIset {\VALset \BIDI \ifx \BIDI \CONrtl \TeXeTstate = 2 \fi}
  \def \BIDIsetone {\VALset \BIDI \ifx \BIDI \CONrtl \TeXeTstate = 1 \fi} \fi}

% BORDER: =====
% The visible border width between cells has to be twice border-width, and
% border greyness should be one third (or skip width should equal twice rule
% width), independent from border style. If a <td> cell is to be drawn within
% a <table>, and both are told to have a solid border, first comes the
% <table> rule, then two skips, then the <td> rule.
% If TeX is programmed to make a full width overlap of horizontal and
% vertical rule edges, it turns out that both Acrobat Reader and Ghostview
% sometimes show a small offset caused by rounding errors that looks ugly.
% Therefore GMS does half width overlapping, if possible.

\def \BRDdraw #1#2{\ifx \BRDprofile \empty \let \BRDprofile \CONsolid \fi
  \csname \CONborderstyle \BRDprofile #2\endcsname}
\def \brdAgroovecellbegin {\brdCa \brdBaa} % borders carved into canvas
\def \brdAgroovecellend {\brdCb \brdBab}
\def \brdAgroovetablebegin {\brdCc \brdBab}
\def \brdAgroovetableend {\brdCd \brdBaa}
\def \brdAhidencellbegin {\BRDreset} % borders hidden
\let \brdAhidencellend \brdAhidencellbegin

```

```

\let \brdAhiddentablebegin \brdAhiddencellbegin
\let \brdAhiddentableend \brdAhiddencellbegin
\let \brdAnonecellbegin \brdAhiddencellbegin % borders absent
\let \brdAnonecellend \brdAhiddencellbegin
\let \brdAnonetablebegin \brdAhiddencellbegin
\let \brdAnonetableend \brdAhiddencellbegin
\def \brdAinsetcellbegin {\brdCa \brdBba} % box embedded in canvas
\def \brdAinsetcellend {\brdCb \brdBca}
\def \brdAinsettablebegin {\brdCc \brdBbb}
\def \brdAinsettableend {\brdCd \brdBcb}
\def \brdAoutsetcellbegin {\brdCa \brdBcb} % box coming out of canvas
\def \brdAoutsetcellend {\brdCb \brdBbb}
\def \brdAoutsettablebegin {\brdCc \brdBca}
\def \brdAoutsettableend {\brdCd \brdBba}
\def \brdAridgecellbegin {\brdCa \brdBda} % borders coming out of canvas
\def \brdAridgecellend {\brdCb \brdBdb}
\def \brdAridgetablebegin {\brdCc \brdBdb}
\def \brdAridgetableend {\brdCd \brdBda}
\def \brdAdoublecellbegin {\brdCa \brdBba} % simple double borders
\def \brdAdoublecellend {\brdCb \brdBbb}
\def \brdAdoubletablebegin {\brdCc \brdBbb}
\def \brdAdoubletableend {\brdCd \brdBba}
\def \brdAsolidcellbegin {\brdDc 3\BRDwt \BRDwl \brdDa 3\BRDwl \BRDwt}
\def \brdAsolidcellend {\brdDb 3\BRDwb \BRDwr \brdDd 3\BRDwr \BRDwb}
\def \brdAsolidtablebegin {\brdDa 3\BRDwl \BRDwt \brdDc 3\BRDwt \BRDwl}
\def \brdAsolidtableend {\brdDd 3\BRDwr \BRDwb \brdDb 3\BRDwb \BRDwr}
\def \brdBaa {1221}\def \brdBab {2211} % groove
\def \brdBba {0222}\def \brdBbb {2220} % inset
\def \brdBca {1212}\def \brdBcb {2121} % outset
\def \brdBda {1122}\def \brdBdb {2211} % ridge
\def \brdBba {1221}\def \brdBbb {2220} % simple double
\def \brdCa #1{\expandafter \brdCaA #1\BRDwt \BRDwl}
\def \brdCaA #1#2#3#4#5#6{% skip rule skip rule - cell begin
\brdDc {#1}#5#6\brdDa {#2}#6#5\brdDc {#3}#5#6\brdDa {#4}#6#5}
\def \brdCb #1{\expandafter \brdCbA #1\BRDwb \BRDwr}
\def \brdCbA #1#2#3#4#5#6{% rule skip rule skip - cell end
\brdDb {#1}#5#6\brdDd {#2}#6#5\brdDb {#3}#5#6\brdDd {#4}#6#5}
\def \brdCc #1{\expandafter \brdCcA #1\BRDwl \BRDwt}
\def \brdCcA #1#2#3#4#5#6{% rule skip rule skip - table begin
\brdDa {#1}#5#6\brdDc {#2}#6#5\brdDa {#3}#5#6\brdDc {#4}#6#5}
\def \brdCd #1{\expandafter \brdCdA #1\BRDwr \BRDwb}
\def \brdCdA #1#2#3#4#5#6{% skip rule skip rule - table end
\brdDd {#1}#6#5\brdDb {#2}#5#6\brdDd {#3}#6#5\brdDb {#4}#5#6}
\def \brdDa #1#2#3{\CLRizeborder
\global \brdFdw1 = #1#2\global \brdFdw2 = #1#3%
\ifnum \TABLEnesting < 5 \brdDaA \fi} % drop begin
\def \brdDaA {\brdFdw \brdFdw1 \brdFdw2 \brdDaAA
\ vbox \bgroup \ifTABLEinter \else \brdEb \brdFdw2 \fi
\ hbox \bgroup \brdEa \brdFdw1 \brdDaAA}
\def \brdDaAA {\vbox \bgroup
\ifTABLEinter \else \vskip \brdFdw2 \fi \hbox \bgroup \hskip \brdFdw1}
\def \brdDb #1#2#3{\CLRizeborder
\global \brdFdw3 = #1#2\global \brdFdw4 = #1#3%
\ifnum \TABLEnesting < 5 \brdDbA \fi} % drop end
\def \brdDbA {\brdFdw \brdFdw3 \brdFdw4 \brdDbAA \brdEa \brdFdw3 \egroup
\ifTABLEinter \else \brdEb \brdFdw3 \fi \egroup \brdDbAA}
\def \brdDbAA {\hskip \brdFdw3 \egroup
\ifTABLEinter \else \vskip \brdFdw3 \fi \egroup}
% hbox inside vbox with skips of fraction of border-width:
\def \brdDc #1#2#3{\brdFlw \brdFlwt {#1#2} \brdFlw \brdFlwl {#1#3}%
\ vbox \bgroup \ifTABLEinter \else \brdEd \brdFlwt \fi
\ hbox \bgroup \brdEc \brdFlwl} % lift begin
\def \brdDd #1#2#3{\brdFlw \brdFlwr {#1#2} lift end
\brdFlw \brdFlwb {#1#3} \brdEc \brdFlwr \egroup
\ifTABLEinter \else \brdEd \brdFlwb \fi \egroup} % drop h,v:
\def \brdEa #1{\ifdim #1 >0pt \hskip -#1 \vrule width 2#1 \hskip -#1 \fi}
\def \brdEb #1{\ifdim #1 >0pt \vskip -#1 \hrule height 2#1 \vskip -#1 \fi}
\def \brdEc #1{\ifCLRbgttable \brdEaA #1 \else \hskip #1 \fi}
\def \brdEaA #1{\CLRize \CLRbgttable
\hskip 0.1#1\vrule width 0.8#1\hskip 0.1#1} % lift h
\def \brdEd #1{\ifCLRbgttable \brdEaA #1 \else \vskip #1 \fi}
\def \brdEdA #1{\CLRize \CLRbgttable
\vskip 0.1#1\hrule height 0.8#1\vskip 0.1#1} % lift v
% drop width (get; bottom, left, right, top):
\def \brdFdw #1#2{\global \divide #1 by 12\global \divide #2 by 12}
\def \brdFdwb \ned \brdFdw1 \ned \brdFdw2 \ned \brdFdw3 \ned \brdFdw4
% lift width (get; bottom, left, right, top):
\def \brdFlw #1#2{\global #1 = #2\global \divide #1 by 6}
\def \brdFlwb \ned \brdFlw1 \ned \brdFlw2 \ned \brdFlw3 \ned \brdFlw4}
\def \BRDhandle {\ifBRDrulegiven \BRDhandleA \else \BRDreset \fi}
\def \BRDhandleA {\BRDwb = 0mm \BRDwl = 0mm \BRDwr = 0mm \BRDwt = 0mm}
\let \BRDprofile \empty
\def \BRDreset {\BRDw = 0mm
\BRDwb = 0mm \BRDwl = 0mm \BRDwr = 0mm \BRDwt = 0mm}
\new \ifBRDrulegiven
\def \BRDset {\edef \BRDtest {\the \VAL}\BRDsetA}
\def \BRDsetA {\BRDrulegiventrue
\ifx \BRDtest \CONthin \BRDsetAA {12} % Fixme: check numbers
\else \ifx \BRDtest \CONmedium \BRDsetAA 9
\else \ifx \BRDtest \CONthick \BRDsetAA 6
\else \VALsetdim \BRDw \hsizelength \fi \fi \fi}
% Set border-width to a fraction of line-height, if #1 is a keyword
% string, or to #1, which is otherwise supposed to be a dimension:
\def \BRDsetAA #1{\BRDw = \LINEheight
\ifdim \BRDw = Opt \BRDw = \BOXlineheight \fi \divide \BRDw by #1}
\def \BRDsetone #1{\BRDsetoneA = \BRDw \BRDset #1 = \BRDw \BRDw = \BRDsetoneA}
\def \BRDupdate {\BRDwb = \BRDw \BRDwl = \BRDw \BRDwr = \BRDw \BRDwt = \BRDw}
\ned \BRDw % width
\ned \BRDwb \ned \BRDwl \ned \BRDwr \ned \BRDwt % bottom, left, right, top
% BOX: =====
\def \BOXbegin {\SPANcolgivenfalse \xdef \NOcollast {0} <td>, <th>
\BOXwidthgivenfalse \OPENfontinnerfalse
\leftskip = Opt \rightskip = Opt \hbadness = 10000
\ifBOXvaligngiven \let \BOXfillerright \empty \let \BOXfillerleft \empty
\else \let \BOXfillertop \vfil \let \BOXfillerright \empty
\let \BOXfillerbottom \vfil \let \BOXfillerleft \empty \fi
\BOXwidthdefault = \hsizelength \BOXwidth = \BOXwidthdefault \BRDhandle
\NOParreset \ifSPANrowspangiven \SPANrowpre \BOXwidthdefault \fi
\SPANcolpre \BOXwidthdefault \LNGchangedetect \ELEMENTvalueget
\ifdim \LINEheight > Opt \edef \BOXlineheight {\the \LINEheight}%
\LINEheight = Opt \fi \boxA \FONTdo \CLRizeforeground
\ifdim \LINEheight < Opt \LINEheight = \BOXlineheight \fi \MSGtdlanguage}
\def \SPANrowpre #1{% Fixme: rename
\DEFsearch {attSPANrowspan\the \TABLEnesting .\the \NOcol} \ifDEF
\edef \test {\csname
attSPANrowspan\the \TABLEnesting .\the \NOcol \endcsname} %
\count 7 = \NORow \advance \count 7 by -\test
\edef \test {\the \count 7} %
\ifnum \test < 0
\DEFsearch {\CONcolwidth \the \TABLEnesting .\the \NOcol} \ifDEF
\edef \test {\csname
\CONcolwidth \the \TABLEnesting .\the \NOcol \endcsname} %
\ifSPANrowspangivenhere \else \ifx \test \empty
{\divide #1 by \BKPTdmax % Fixme: colspan gets lost
\multiply #1 by \SPANcollast % Fixme: too simple
\hskip #1} \else \hskip \test \fi
\gadvance \NOcol by 1 \SPANrowpre {#1} \fi}

```



```

    \else {\divide #1 by \BKPTdmax \hskip #1}\fi \fi \fi}% recursion
\def \SPANcolpre #1{\DEFsearch {TCOLS\the \NOTable}\ifDEF % Fixme: rename
\edef \SPANcolpretest {\csname TCOLS\the \NOTable \endcsname}%
\ifx \SPANcolpretest \empty \else
\SPANcolpredefault = \SPANcolpretest \fi \fi
\ifx \SPANcol \empty \divide #1 by \SPANcolpredefault
\else \divide #1 by \SPANcol \fi
\nec \SPANcolpredefault
\def \boxA {\hskip Opt \hbox \bgroup \raise \LINEheight
\vtop \bgroup \vskip \MARGIN \hbox \bgroup \hskip \MARGIN
\ifSPANrowspangivenhere \else \ifdim \BOXheight < \BOXheightold
\BOXheight = \BOXheightold \BOXheightgiventrue \fi \fi
\vsizel = \BOXheight \ifBOXwidthgiven \hsizel = \BOXwidth \else
\hsizel = \BOXwidthdefault \BOXwidthget \fi
\advance \vsizel by -2\MARGIN \advance \hsizel by -2\MARGIN
\ifdim \vfuzz < 1pt \vfuzz = 1pt \fi \ifCLRbg \boxB \fi
\ifdim \BRDw > 0mm \BRDupdate \fi \BRDraw \CONbegin \CONcell
\advance \vsizel by -\BRDwb \advance \vsizel by -\BRDwt
\advance \hsizel by -\BRDwl \advance \hsizel by -\BRDwr \PDGbegin
\advance \vsizel by -\PDGtop \advance \vsizel by -\PDGbottom
\advance \hsizel by -\PDGleft \advance \hsizel by -\PDGright
\ifBOXheightgiven \vbox to \vsizel \bgroup \else \vbox \bgroup \fi
\BOXfillertop \hbox to \hsizel \bgroup \strut \BOXfillerleft \vbox \bgroup}
\def \boxB {\ifTABLEEnter \else \ifBOXheightgiven % background
\ifBOXwidthgiven \boxBA \else \boxBB \fi \else \boxBB \fi \fi}
\def \boxBA {\CLRize \CLRbackground % dimensioned
\vbox {\vskip -\vsizel \hbox %
\vrule height \vsizel width \hsizel \hskip -\hsizel }}
\def \boxBB {\CLRize \CLRbackground % undimensioned
{\advance \hsizel by -\PDGleft \vrule width \hsizel \hskip -\hsizel }}

\let \BOXcellvalign \empty

\def \BOXend {\ifOPEND \boxC \else \ifOPENTh \boxC \fi \fi}
\def \boxC {\egroup \BOXfillerright \strut
\egroup \BOXfillerbottom \egroup \PDGend \BRDDraw \CONend \CONcell
\hskip \MARGIN \egroup \vskip \MARGIN
\ifSPANrowspangivenhere \vskip -\BOXheight \else
\ifdim \BOXheight > \BOXheightold
\global \BOXheightold = \BOXheight \fi \fi \egroup \egroup}

\let \BOXfillerbottom \empty \let \BOXfillerleft \empty
\let \BOXfillerright \empty \let \BOXfillertop \empty

\def \BOXgridtrbegin {\global \BOXwidthcolspan = Opt \BOXheightgivenfalse
\BOXwidthdefault = \hsizel \BOXwidth = \BOXwidthdefault
\BOXheightold = Opt \BOXheight = \LINEheight}
\def \BOXgridtrend {\BOXheightgivenfalse \BOXwidthgivenfalse \OPENTrfalse}

\ned \BOXheight \new \ifBOXheightgiven
\ned \BOXheightold \ned \BOXheightswapped
\def \BOXheightswappedinit {\BOXheightswapped = \BOXheight}

\ned \BOXindent

\def \BOXinit {\BOXheightgivenfalse \BOXwidthgivenfalse
\BOXheight = 0mm \BOXheightswapped = 0mm
\BOXindent = \hsizel \BOXwidth = \hsizel}

\edef \BOXlineheight {\the \LINEheight}
\def \BOXlineheightset %
\ifnum \TABLEnesting = 1 \edef \BOXlineheight {\the \LINEheight}\fi}

\def \BOXrowadvance {\NOcol = 0 \advance \NORow by 1 }

\def \BOXspancolexec {\ifBOXwidthgiven \xDef {\CONcolwidth
\the \TABLEnesting.\the \NOcol} {\the \BOXwidth}\fi}

\new \ifBOXvaligngiven
\def \BOXvalignset {\edef \BOXvalign {\the \VAL}%
\ifx \BOXvalign \CONbottom \BOXvalignsetbottom \BOXvaligngiventrue
\else \ifx \BOXvalign \CONtop \BOXvalignsettop \BOXvaligngiventrue
\else \ifx \BOXvalign \CONcenter \CHKerryoumeantmiddle \fi \fi \fi}
\def \BOXvalignsetbottom %
\let \BOXfillertop \vfill \let \BOXfillerbottom \vfill}
\def \BOXvalignsettop %
\let \BOXfillertop \vfill \let \BOXfillerbottom \vfill}

\ned \BOXwidth \ned \BOXwidthcolspan \ned \BOXwidthdefault
\def \BOXwidthget {\DEFsearch {\CONcolwidth \the \TABLEnesting.\the \NOcol }}
\ifDEF \edef \BOXwidthgettest %
\csname \CONcolwidth \the \TABLEnesting.\the \NOcol \endcsname %
\ifx \BOXwidthgettest \empty \else \hsizel = \BOXwidthgettest \fi \fi}
\new \ifBOXwidthgiven

% CHAIN: =====
% Input file:

\def \CHNcheck {\CHNloadnextfalse \let \CHNhrefnext \empty % </html>
\ifOPENobject \else \ifx \CHNhref \empty \else
\let \CHNhrefnext \CHNhref \ENDIFalse \CHNloadnexttrue \fi \fi}

\def \CHNcopyrelativepath #1#2\to #3{\let \CHNfolderbackup \CHNfoldercurrent
\let \chainNEXTfile \empty \let \chainNEXTlevel \empty
\edef \CHNcopyfirst {#1}%
\ifx \CHNcopyfirst \CONslash \edef #3{#1#2}/% Fixme: use some absolute path
\else \edef #3{./#1#2}/\fi \expandafter \chnC #3\relax
\edef #3{\CHNfoldercurrent /chainNEXTfile}%
\let \CHNfoldercurrent \CHNfolderbackup}

\def \CHNfile #1{% executed by </html>, <link>; indirectly: <object>
\edef \CHNfileZ {#1}\let \URLhref \CHNfileZ % load next
\ifx \CHNfileZ \empty \let \chainNEXTa \relax
\else \let \chainNEXTa \chnA \fi \chainNEXTa}
\def \chnA {\let \chainNEXTfile \empty \let \chainNEXTlevel \empty
\edef \CHNfileZ {./\CHNfileZ}/%
\expandafter \chnC \CHNfileZ \relax
\edef \CHNfileZ {\CHNfoldercurrent /chainNEXTfile}%
\FILEsearch \CHNfileZ \global \CHNloadnextfalse
\ifFILEexist \chnB \else \chnAA \fi}
\def \chnAA {\CHKerrabsent \CHNfileZ % absent
\ifOPENobject \else \ENDIFtrue \CHNloadnextfalse \fi}
\def \chnB {\ifOPENobject \chnBB \else \chnBA \fi
\ENDIFalse \CHNloadnexttrue} % present
\def \chnBA % present message, fixme: move
\advance \NOinput by 1 \noecho \message {\CONhalfline}}
\def \chnBB % present object
\ifOPENiframe \advance \NOinclude by 1 \OPENiframefalse
\else \ifFILEhtml \advance \NOinclude by 1 \fi \fi}
\def \chnC #1/#2% folder: analyse next level of path
\def \chnCY {#1}\def \chnCZ {#2}%
\ifx \chnCZ \CONrelax \chnCB \else \ifx \chnCZ \CONslash
\let \chainNEXT \relax \else \chnCA \fi \fi \chainNEXT}
\def \chnCA {\let \chainNEXT \chnC
\xdef \chainNEXTlevel {\chainNEXTfile \chnCY}%
\xdef \chainNEXTfile {\chnCZ}%
\ifx \chainNEXTlevel \CONdotdot \chnD \else \ifx \chainNEXTlevel \CONdot
\else \STACKpush \chainNEXTlevel \to \CHNfoldercurrent \fi \fi}
\def \chnCB {\let \chainNEXT \relax % relax pushing
\xdef \chainNEXTfile {\chainNEXTfile \chnCY}}
\def \chnD {\STACKpop \chnCdopushlevel \from \CHNfoldercurrent % push level
\ifx \chnCdopushlevel \empty
\STACKpush \chainNEXTlevel \to \CHNfoldercurrent
\else \ifx \chnCdopushlevel \CONdotdot \chnDA
\else \ifx \chnCdopushlevel \CONdotdottwo \chnDA
\else \ifx \chnCdopushlevel \CONdotdotthree \chnDA
\else \ifx \chnCdopushlevel \CONdotdotfour \chnDA

```

```

\else \ifx \chnCdotpushlevel \CONdotdotfive \chnDA
\else \ifx \chnCdotpushlevel \CONdotdotsix \chnDA
\else \ifx \chnCdotpushlevel \CONdotdotseven \chnDA
\else \ifx \chnCdotpushlevel \CONdotdoteight \chnDA
\fi \fi \fi \fi \fi \fi \fi \fi \fi
\def \chnDA {\STACKpush \chnCdotpushlevel \to \CHNfoldercurrent
\STACKpush \chainNEXTlevel \to \CHNfoldercurrent}

\let \CHNfolderbackup \empty
\let \CHNfoldercurrent \empty

\def \CHNget #1{\ifx \CHNhref \empty \let \chainNEXTb \relax
\else \let \chainNEXTb \chnEC \fi
\ifOPENobject \let \chainNEXTb \relax \fi \chainNEXTb
\ifFILEplain \chnK \else \chnI \fi
\ifOPENobject \chnJ \else \chnG #1\fi
\let \CHNgetnext \relax \FILEsearch \CHNgetprobe
\ifFILEexist \chnE \CHNgetprobe \else \chnF #1\fi \chainNEXT
\ifOPENobject \let \chainNEXT \relax \else \chnH #1\fi \chainNEXT}
\def \chnE #1{\chnEB #1}%
\expandafter \chnEA #1\empty \empty \relax \chnEB \chnEnameclean
\chnEB {./\chnEnameclean}\chnEB {./\chnEnameclean}%
\ifx \CHNfoldercurrent \empty \else
\chnEB {\CHNfoldercurrent/\chainNEXTfile}\fi
\chnEB \chainNEXTfile \chnEB {./\chainNEXTfile}} % get file
\def \chnEA #1#2#3\relax {\edef \chnEAXY {#1#2}%
\ifx \chnEAXY \CONdotslash \edef \chnEnameclean {#3}\fi} % clean
\def \chnEB #1{\DEFsearch {\CONdest #1}%
\ifDEF \else \ANCRbychain {#1}\fi} % save
\def \chnEC {\CHNfile \CHNhref \relax \glet \CHNhref \empty} % href
\def \chnF #1{\let \chainNEXT \relax \let \CHNgetnext #1} % no file
\def \chnG #1{% no object
\edef \CHNgetprobe {.\CHNfoldercurrent /\chainNEXTfile}%
\gadvance \chkWLMtotfile by -1 \chkWLMtotfileadvance
\ifFILEtoomany \let \chainNEXT #1\fi}
\def \chnH #1{% no object two
\ifx \CHNgetnext \relax \def \chainNEXT {\CHNget #1}%
\else \let \chainNEXT \CHNgetnext \fi}
\def \chnI {\ifFILETeX \OBJECTtex \fi \let \chainNEXT \chnIA} % no plain
\def \chnIA {\shredder \CHNloadnextfalse
\input \CHNgetprobe \relax \shredder}
\def \chnJ {\edef \CHNgetprobe {./\URLhref}} % object
\def \chnK {\OBJECTplain \let \chainNEXT \chnKA} % plain
\def \chnKA {\chnKAA \CHNgetprobe \shredder}
\def \chnKAA #1{\openin \FILEplaintext = #1\relax
\loop \FILEcheckeof \FILEplaintext
\ifFILEnotended \read \FILEplaintext to \myline \strut \myline
\repeat \closein \FILEplaintext}%

\let \CHNhref \empty

\def \CHNiftable {\ifOPENtable \CHNget \CHKerrwellformedness \fi} % <iframe>

\new \ifCHNloadnext

% Fixme: better done in \IMGnameget with respect to relative paths:
\def \CHNobjectloadurl {% executed by <iframe> and <object>
\CHNobjectloadurlA \URLhref \relax}
\def \CHNobjectloadurlA #1{\edef \CHNobjectfirst {#1}%
\let \CHNfolderbackup \CHNfoldercurrent
\let \CHNfoldercurrent \empty \CHNfile \CHNobjectfirst
\glet \CHNfoldercurrent \CHNfolderbackup}

% CHECK error: =====

\let \chkWLMerrordiff \empty

% Fixme: Mention open elements only if required or doctype xhtml

\def \CHKerrabsent #1{\echo {! Absent: "#1" \CHKerrline.}%
\DEFsearch {\CONABSENT #1}\ifDEF \else \gadvance \chkWLMtotabsent by 1
\xDef {\CONABSENT #1} \fi}

\def \CHKerrapostrophized {\chkWLMtotunquotedadvance
\ifx \CHKerrapostroph \undefined \echo {! Apostrophized \CHKerrline.}%
\glet \CHKerrapostroph \empty \fi}

\def \CHKerrcannotrender {\chkWLMtototheradvance
\echo {! Cannot render \CHKerrline.}}

\def \CHKerrcat {% Fixme: move
\CATotherhashmark \CATotherampersand \CATparameterdegree}

\def \CHKerrcodepagereplaced #1{%
\chkWLMtototheradvance \echo {! Replaced: "#1" \CHKerrline.}}

\def \CHKerrdebugend {% Force end of TeX run (GMS debugging)
\csname TAG/trace\endcsname
\Message {end}\MSGproprietary \vfill \eject \getend}

\def \CHKerrdeprecated #1{\chkWLMmessdiff {#1}{Deprecated}%
\chkWLMdffdepr \chkWLMtotdeprecated \CONdepr}

\def \CHKerrdeprecatedatt #1{\chkWLMmessdiff {#1}{Deprecated: #1}%
\chkWLMdffdepr \chkWLMtotdeprecated \CONdeprap}

\def \CHKerrduplicate #1{\echo {! Duplicate: [ #1 ] \CHKerrline.}%
\gadvance \chkWLMtotduplicate by 1}

\def \CHKerrempty #1{\chkWLMmessdiff {#1}{Missing: #1}%
\chkWLMdffempt \chkWLMtotempty \CONEMPTY}

\bgroup \CHKerrcat
\gdef \CHKerrfallback ^1^2\relax {\def \first {^1}%
\ifx \first \CONquotmark \def \first {x}\fi
\echo {! Fallback: &#\first ^2; \CHKerrline.}%
\gadvance \chkWLMtotfallback by 1 }\egroup

\bgroup \CHKerrcat
\gdef \CHKerrfallbackname ^1{\echo {! Fallback: &^1; \CHKerrline.}%
\gadvance \chkWLMtotfallback by 1 }\egroup

\def \CHKerrfontfalse {%
\echo {! False font \CHKerrline.}\chkWLMtototheradvance}

\def \CHKerrfontreset #1{\echo {! Font reset \CHKerrline.}%
\gadvance \chkWLMtotother by 1} % Fixme: \chkWLMtototheradvance

\def \CHKerrfontstyle #1{%
\echo {! Unsupported font-style: "#1" \CHKerrline.}}

\def \CHKerrfontweight #1{%
\echo {! Unsupported font-weight: "#1" \CHKerrline.}}

\def \CHKerrinvalidfontsize {\chkWLMtototheradvance
\echo {! Invalid font size \CHKerrline.}}

\def \CHKerrinvalidsize #1{\chkWLMtototheradvance
\echo {! Invalid font size "#1" \CHKerrline.}}

\def \CHKerrinvalidunit #1{\chkWLMmessdiff {#1}{Invalid unit: "#1"%
\chkWLMdffinvu \chkWLMtotinvalidunit \CONinvalidunit}

\bgroup \CHKerrcat
\gdef \CHKerrlost ^1^2\relax {\def \first {^1}%
\ifx \first \CONquotmark \def \first {x}\fi
\echo {! Lost: &#\first ^2; \CHKerrline.}%
\gadvance \chkWLMtotlost by 1 }\egroup

```

```

\def \CHKerrline {(line \the \inputlineno)}

\def \CHKerrlostplain #1{\chkWLMtototheradvance
\echo {! Lost: #1; \CHKerrline}}

\def \CHKerrmetanospace #1{\chkWLMtototheradvance
\message {\CONpleasedonotwrite}\message {"#1" before "="}%
\message {\CONthereshouldbe}\message {no space}.}}

\def \CHKerrmetaonespace #1{\chkWLMtototheradvance
\message {\CONpleasedonotwrite}\message {"#1" after ","}%
\message {\CONthereshouldbe}\message {one space}.}}

\def \CHKerrmisformeddate {\chkWLMtototheradvance
\echo {! Misformed: "YYYY-MM-DD" \CHKerrline.}}

\def \CHKerrmisnested {\echo {! Misnested \CHKerrline.}\CHKerrmisnesting}

\def \CHKerrmisnesting {\gadvance \chkWLMtotmisnesting by 1}% for statistic

\def \CHKerrmissingendtag #1{% Fixme: outsource
\ifnum \chkWLMtotunclosed < 2 \echo {! Open: #1 \CHKerrline.}\else
\ifchkDCTPpxhtmlclaim \echo {! Open: #1 \CHKerrline.}\fi \fi
\edef \ELEMENTopen #1\edef \backupelement {\the \ELEMENT}%
\edef \backupATList {\the \ATList}%
\glet \backupifATT \ifATT \glet \backupifPROPERTY \ifPROPERTY
\gadvance \chkWLMtotunclosed by 1
\csname \CONtag /#1\endcsname
\global \let \ifPROPERTY \backupifPROPERTY
\global \let \ifATT \backupifATT
\global \ATList = \expandafter {\backupATList}%
\global \ELEMENT = \expandafter {\backupelement}}

\def \CHKerrmissingheight {\chkWLMtototheradvance
\echo {! Unspecified: height \CHKerrline.}}

\def \CHKerrmissingspace {\chkWLMtototheradvance
\echo {! Missing space before /> \CHKerrline.}}

\def \CHKerrmissingwidth {%
\ifnum \NORow < 2 \ifnum \chkWLMtotmissingwidth < 2 \ifSPANcoltable \else
\gadvance \chkWLMtotmissingwidth by 1
\echo {! Unspecified: width \CHKerrline.}\fi \fi \fi}

\def \CHKerrmissingulol {\chkWLMtototheradvance
\echo {! Missing parent element <dl>, <ol> or <ul> \CHKerrline.}}

\def \CHKerrmisspelled {\chkWLMmessdiff {}{Misspelled}%
\chkWLMdffprpr \chkWLMtotproprietary \CONprop}

\def \CHKerrnocodepage #1{\chkWLMtototheradvance
\echo {! No codepage #1 \CHKerrline.}}

\def \CHKerrnofamilydefined #1{\chkWLMtototheradvance
\echo {! No font-family defined for "#1" \CHKerrline.}}

\def \CHKerrnofontforCP #1{\chkWLMtototheradvance
\echo {! No such font(s) installed for
\CONcodepage \CONspace \CPGhtml:}%
\echo {\CONspace \CONspace #1 \CHKerrline.}}

\def \CHKerrnolineheight {\chkWLMtototheradvance
\echo {! Missing line-height \CHKerrline.}}

\def \CHKerrnostylesheettype {\gadvance \chkWLMtotempty by 1
\echo {! Missing attribute: type = "text/css" \CHKerrline.}}

\def \CHKerrnotstrict {\global \chkDCTPnotstricttrue
\ifnum \chkWLMtotdeprecated = 0 \ifnum \chkWLMclass = 0 \MSGnotstrict
\chkWLMtototheradvance \global \chkWLMclass = 1 \else
\ifnum \chkWLMclass = 2 \MSGnotstrict
\chkWLMtototheradvance \global \chkWLMclass = 3 \fi \fi}

\def \CHKerrnotstrictatt #1{\global \chkDCTPnotstricttrue
\ifnum \chkWLMtotdeprecated = 0 \ifnum \chkWLMclass = 0
\chkWLMtototheradvance \MSGnotstrictatt #1}%
\global \chkWLMclass = 1 \else \ifnum \chkWLMclass = 2
\chkWLMtototheradvance \MSGnotstrictatt #1}%
\global \chkWLMclass = 3 \fi \fi}

\def \CHKerropenp {\gadvance \chkWLMtotunclosed by 1
\echo {! Open: p \CHKerrline.}\endgroup}

\def \CHKerrproprietary #1{\chkWLMmessdiff #1}{Proprietary}%
\chkWLMdffprpr \chkWLMtotproprietary \CONprop}

\def \CHKerrproprietaryatt #1{\chkWLMmessdiff #1}{Proprietary: #1}%
\chkWLMdffprpr \chkWLMtotproprietary \CONpropap}

\def \CHKerrsuperfluous {%
% Rendering unescaped end tags in preformatted code sections:
\ifOPENpre \CONlessthan \the \ELEMENT >\else
\ifnum \chkWLMtotunclosed > 0 \edef \CHKerrtest {\the \ELEMENT}%
\edef \CHKerrtestopen {\the \ELEMENTopen}%
\ifx \CHKerrtest \CHKerrtestopen \echo {! Overlapping \CHKerrline.}%
\gadvance \chkWLMtotunclosed by -1 \gadvance \chkWLMtotoverlap by 1
\else \CHKerrsupermessage \fi \else \CHKerrsupermessage \fi \fi}

\def \CHKerrsupermessage {\gadvance \chkWLMtotsuperfluous by 1
\echo {! Superfluous \CHKerrline.}}

\def \CHKerrtracebegin {%
\global \tracinggroups = 1 \global \tracingmacros = 1}

\def \CHKerrtraceend {\global \tracingcommands = 0
\global \tracinggroups = 0 \global \tracingnesting = 0
\global \tracingifs = 0 \global \tracingmacros = 0}

\def \CHKerrunanchored {%
\chkWLMtototheradvance \echo {! Unanchored \CHKerrline.}}

\def \CHKerruncolored #1{%
\chkWLMmessdiff #1}{Uncolored: "#1"%
\chkWLMdffuclr \chkWLMtotuncolored \CONuncolored}

\def \CHKerrundefclass #1{\chkWLMmessdiff #1}{Undeclared: "#1"%
\chkWLMdffudcl \chkWLMtotundefclass \CONundecl}

\def \CHKerrundefref #1{\RFRundeftrue
\chkWLMmessdiff #1}{Undefined: '#1'%
\chkWLMdffudrf \chkWLMtotundefref \CONundefref}

\def \CHKerrundelimited #1{\gadvance \chkWLMtotundelimited by 1
\echo {! Undelimited: \CONampersand #1; \CHKerrline.}}

\def \CHKerrunloadable #1{\chkWLMtototheradvance
\echo {! Unloadable: "#1" \CHKerrline.}}

\def \CHKerrunquot {\ifnum \chkWLMtotunquoted < 1
\CHKerrunquoted {\attWXung
\CONspace = "\the \VAL".}\else %% Fixme: 'class = break' => 'always'
\ifATTunqmessage \else \ATTunqmessagegettrue
%% \ifnum \chkWLMtotunquoted < 20 \message {! ".}\fi
\fi \fi \chkWLMtotunquotedadvance}

\def \CHKerrunquoted #1{\edef \CHKerrunquotedtest {\the \VAL}%
\ifx \CHKerrunquotedtest \empty \else

```

```

\echo {! Unquoted: #1 \CHKerrline.}\chkWLMtotunquotedadvance \fi}

\def \CHKerrunrecogattprop #1{\ifCLASSspredifined \else
\chkWLMmessdiff {#1}{Unrecognized: '#1'}%
\chkWLMdffurec \chkWLMtotunrecognized \CONunrecap \fi}

\def \CHKerrunrecognized #1{\chkWLMmessdiff {#1}{Unrecognized: <#1>%
\chkWLMdffurec \chkWLMtotunrecognized \CONunrec}

\def \CHKerrunselected #1{\chkWLMtototheradvance
\echo {! Unselected:}\message {#1 \CHKerrline.}}

\def \CHKerrunsupattprop #1{\chkWLMmessdiff {#1}{Unsupported: #1}%
\chkWLMdffuspd \chkWLMtotunsupported \CONunsupap}

\def \CHKerrunsupported #1{\chkWLMmessdiff {#1}{Unsupported}%
\chkWLMdffuspd \chkWLMtotunsupported \CONunsup}

\def \CHKerryumeantcenter {\chkWLMmessdiff {}{Confused: align = "center"}%
\chkWLMdffcnfs \chkWLMtotconfused {\CONconfu center}}

\def \CHKerryumeantmiddle {\chkWLMmessdiff {}{Confused: valign = "middle"}%
\chkWLMdffcnfs \chkWLMtotconfused {\CONconfu middle}}

\def \CHKerrwellformedness {\ifinner \ELEMENTcloseinner \fi
\ifnum \ELEMENTlevel > 0 \ELEMENTcloseouter \fi
\ifchkDCTPnotstrict \def \CHKerrwellformednesstest {0}\else
\def \CHKerrwellformednesstest {1}\fi
\chkWLMdo \chkDCTP \CHKerrwellformednesstest}

% check DOCTYPE: -----
% The doctype declaration check sets chkDCTP to a number from 0 to 5:
% 0: XHTML strict
% 1: XHTML transitional or frameset
% 2: HTML strict
% 3: HTML transitional or frameset
% 4: none (initial)
% 5: unrecognized
\nec \chkDCTP \chkDCTP = 4 % "none"

% Detected errors set chkWLMclass to a number from 0 to 6:
% 0: XHTML 1.0 strict: no unclosed tags, no superfluous tags,
% no unquoted atts, no deprecated atts,
% no proprietary atts, no empty atts,
% no undefined classes, no other problems
% 1: XHTML 1.0 trans.: no unclosed tags, no superfluous tags,
% no unquoted atts, no proprietary atts,
% no empty atts, no other problems
% 2: HTML 4.01 strict: no proprietary atts, no deprecated atts,
% no empty atts, no other problems
% 3: HTML 4.01 trans.: no proprietary atts
% 4: HTML proprietary: proprietary atts
% 5: HTML unknown: unrecognized tags or atts
% 6: HTML irregular: misnested
\nec \chkWLMclass \chkWLMclass = 0

\def \chkDCTPdcl #1{% doctype
\ifcase #1\chkDCTPdclA 0 \or \chkDCTPdclA 1 \or
\chkDCTPdclA 2 \or \chkDCTPdclA 3 \or
\MSGnone \else \MSGunrecognized \fi} % 4 else 5
\def \chkDCTPdclA #1{%
\ifnum \chkWLMclass < #1 \MSGplausible \else
\ifnum \chkWLMclass = #1 \MSGverified \else
\ifnum \chkWLMclass < 6 \MSGunverified \else
\ifnum \chkWLMtotproprietary > 0 \MSGfalsified
\else \MSGunverified \fi \fi \fi \fi}

\def \chkDCTPget #1{%
\ifnum \chkDCTP = 4 \global \chkDCTP = 5 % "unrecognized"

\edef \probe {#1\CONspace}%
\expandafter \chkDCTPgetE \probe XHTML {} Strict//EN\relax
\ifchkDCTPxhtmlclaim \else
\expandafter \chkDCTPgetF \probe XHTML {} Transitional//EN\relax \fi
\ifchkDCTPxhtmlclaim \else
\expandafter \chkDCTPgetD \probe XHTML {} Frameset//EN\relax \fi
\ifchkDCTPxhtmlclaim \else %%
\expandafter \chkDCTPgetG \probe GEDCOM {} \relax \fi %%
\ifchkDCTPxhtmlclaim \else
\expandafter \chkDCTPgetB \probe HTML {} Strict//EN\relax \fi
\ifchkDCTPxhtmlclaim \else
\expandafter \chkDCTPgetC \probe HTML {} Transitional//EN\relax \fi
\ifchkDCTPxhtmlclaim \else
\expandafter \chkDCTPgetA \probe HTML {} Frameset//EN\relax \fi \fi}
\def \chkDCTPgetA #1 HTML #2 Frameset//EN#3\relax {%
\chkDCTPgetXA 3{#2}{#3}}
\def \chkDCTPgetB #1 HTML #2//EN#3\relax {%
\chkDCTPgetXA 2{#2}{#3}}
\def \chkDCTPgetC #1 HTML #2 Transitional//EN#3\relax {%
\chkDCTPgetXA 3{#2}{#3}}
\def \chkDCTPgetD #1 XHTML #2 Frameset//EN#3\relax {%
\global \chkDCTPxhtmlclaimtrue \chkDCTPgetXA 1{#2}{#3}}
\def \chkDCTPgetE #1 XHTML #2 Strict//EN#3\relax {%
\global \chkDCTPxhtmlclaimtrue \chkDCTPgetXA 0{#2}{#3}}
\def \chkDCTPgetF #1 XHTML #2 Transitional//EN#3\relax {%
\global \chkDCTPxhtmlclaimtrue \chkDCTPgetXA 1{#2}{#3}}
\def \chkDCTPgetG #1GEDCOM #2\relax {\def \test {#1}%
\ifx \test \empty \global \chkDCTPxhtmlclaimtrue
\chkDCTPgetXA 0{6.0}{GEDCOM}%
\else \chkDCTPgetXA 5{6.0}{GEDCOM}\fi}
\def \chkDCTPgetXA #1#2#3{\def \second {#2}\def \third {#3}%
\ifx \second \empty \global \chkDCTPxhtmlclaimfalse \else
\ifx \third \empty \global \chkDCTPxhtmlclaimfalse \else
\xdef \chkDCTPversion {#2}\global \chkDCTP = #1\fi \fi}

\new \ifchkDCTPnotstrict

\def \chkDCTPmsg #1{%
\ifGEDCOM \message {\CONgedcomxml}\else
\ifcase #1\message {\CONXHTMLstrict } \or % 0
\message {\CONXHTMLtrans } \or \message {\CONHTMLstrict } \or % 1, 2
\message {\CONHTMLtrans } \or \message {\CONHTMLprop } \or % 3, 4
\message {\CONHTMLunknown } \else \message {\CONHTMLirreg}\fi \fi % 5, 6
\ifcase #1\message {\CONspace (.)} \or % 0
\message {\CONspace (-)} \or \message {\CONspace |.} \or % 1, 2
\message {\CONspace |;} \or \message {\CONspace |.} \or % 3, 4
\message {\CONspace -;} \else \message {\CONspace ()} \fi % 5, 6

\def \chkDCTPversion {unknown} % markup version number not checked

\new \ifchkDCTPxhtmlclaim

% Different cases leading to same error message:
\nec \chkWLMdffcnfs % confused
\nec \chkWLMdffdepr % deprecated
\nec \chkWLMdffempt % empty
\nec \chkWLMdffinvu % invalidunit
\nec \chkWLMdffprpr % proprietary
\nec \chkWLMdffuclr % uncolored
\nec \chkWLMdffudcl % undefclass
\nec \chkWLMdffudrf % undefref
\nec \chkWLMdffurec % unrecognized
\nec \chkWLMdffuspd % unsupported

\nec \chkWLMtotabsent
\nec \chkWLMtotconfused
\nec \chkWLMtotdeprecated
\nec \chkWLMtotduplicate
\nec \chkWLMtotelements

```

```

\nec \chkWLMtotempty
\nec \chkWLMtoterrors
\nec \chkWLMtotfallback
\nec \chkWLMtotfile % opened input or image file
\nec \chkWLMtotinvalidunit
\nec \chkWLMtotlost
\nec \chkWLMtotmissingwidth
\nec \chkWLMtotmisnesting
\nec \chkWLMtotother \chkWLMtotother = 0 % reset warning counter
\nec \chkWLMtotoverlap
\nec \chkWLMtotproprietary
\nec \chkWLMtotsuperfluous
\nec \chkWLMtotunclosed
\nec \chkWLMtotuncolored
\nec \chkWLMtotundefclass
\nec \chkWLMtotundefref
\nec \chkWLMtotundelimited
\nec \chkWLMtotunquoted
\nec \chkWLMtotunrecognized
\nec \chkWLMtotunsupported

\def \chkWLMtotfileadvance {\gadvance \chkWLMtotfile by 2
  \ifnum \chkWLMtotfile > \NOfilemax
    \global \FILEtoomanytrue \message {! Ignored.}\fi
  \gadvance \chkWLMtotfile by -1 } % Fixme: move
\def \chkWLMtototheradvance {\message {! Warning \CHKerrline:}%
  \gadvance \chkWLMtotother by 1 }
\def \chkWLMtotunquotedadvance {\gadvance \chkWLMtotunquoted by 1 }

% check PLURAL: -----
\def \chkPLRabsent #1#2{absent file#1}
\def \chkPLRconfused #1#2{confused attribute value#1 #2}
\def \chkPLRdeprecated #1#2{deprecated element#1 or attribute#1 #2}
\def \chkPLRduplicate #1#2{duplicate anchor name#1 or element id#1}
\def \chkPLRfallback #1#2{fallback font character#1}
\def \chkPLRempty #1#2{missing or empty attribute#1 #2}
\def \chkPLRfile #1#2{ignored file#1}
\def \chkPLRinvalidunit #1#2{invalid length unit#1 #2}
\def \chkPLRlost #1#2{lost character#1}
\def \chkPLRmisnesting #1#2{misnested element#1}
\def \chkPLRmissingwidth #1#2{table#1 or cell#1 without specified width#1}
\def \chkPLRrother #1#2{other warning#1}
\def \chkPLRoverlap #1#2{overlapping element#1}
\def \chkPLRproprietary #1#2{proprietary element#1, attribute#1 or
  \propert\ifx #1\empty y\else ies\fi \CONspace #2}
\def \chkPLRsuperfluous #1#2{superfluous end tag#1}
\def \chkPLRunclosed #1#2{missing end tag#1}
\def \chkPLRuncolored #1#2{false color#1 #2}
\def \chkPLRundefclass #1#2{undeclared class selector#1 #2}
\def \chkPLRundefref #1#2{undefined or undelimited character reference#1 #2}
\def \chkPLRundelimited #1#2{undelimited reference#1}
\def \chkPLRunquoted #1#2{attribute value#1 without quotation marks}
\def \chkPLRunrecognized #1#2{unrecognized element#1, attribute#1 or
  \propert\ifx #1\empty y\else ies\fi \CONspace #2}
\def \chkPLRunsupported #1#2{unsupported element#1, attribute#1 or
  \propert\ifx #1\empty y\else ies\fi \CONspace #2}

% check WELLFORM: -----
\def \chkWLMdo #1#2{%
  \ifOUTgutter \def \WELLFORMprobe {%
    /PageLayout /TwoColumnRight \CONspace }% %% Fixme: bidi
  \else \let \WELLFORMprobe \empty \fi
  \ifx \pdfcatalog \undefined \else
    \pdfcatalog {\WELLFORMprobe} openaction goto page 1 {\Fit}\CONspace \fi
  \echo {\CONlongline \CONline \CONhalfline } \GMSendabout \GMSgoodbye
  \message {Warnings and errors:}\chkWLMokaytrue
  \ifnum #1 > 3 \ifnum \chkWLMclass < 2 % no doctype declaration
    \chkWLMclass = 2 \fi \fi
    \ifx \NAMESPACE \empty \xdef \NAMESPACE {none}%
    \ifnum \chkWLMclass < 2 % no namespace declaration
      \chkWLMclass = 2 \fi \fi
    \chkWLMdoC \chkWLMtotmisnesting \CONmisnesting 6
    \chkWLMdoC \chkWLMtotoverlap \CONoverlap 6
    \chkWLMdoC \chkWLMtotsuperfluous \CONsuperfluous 2
    \chkWLMdoC \chkWLMtotunclosed \CONunclosed 2
    \chkWLMdoC \chkWLMtotundelimited \CONundelimited 2
    \chkWLMdoC \chkWLMtotunquoted \CONunquoted 2
    \chkWLMdoB \chkWLMtotempty {%
      \chkWLMdoE \chkWLMdffempt \chkWLMtotempty \CONempty }1 \chkWLMdoA
    \chkWLMdoB \chkWLMtotconfused {%
      \chkWLMdoE \chkWLMdffcnfs \chkWLMtotconfused \CONconfused }4 \empty
    \chkWLMdoB \chkWLMtotdeprecated {\chkWLMdoE \chkWLMdffdepr
      \chkWLMtotdeprecated \CONdeprecated }1 \chkWLMdoA
    \chkWLMdoB \chkWLMtotproprietary {\chkWLMdoE \chkWLMdffprpr
      \chkWLMtotproprietary \CONproprietary }4 \empty
    \chkWLMdoB \chkWLMtotuncolored {\chkWLMdoE \chkWLMdffuclr
      \chkWLMtotuncolored \CONuncolored }1 \chkWLMdoA
    \chkWLMdoB \chkWLMtotundefclass {\chkWLMdoE \chkWLMdffudcl
      \chkWLMtotundefclass \CONundefclass }0 \empty
    \ifnum \chkWLMtotinvalidunit > 0 \chkWLMdoD \CONmissingwidth \fi
    \chkWLMtotinvalidunit \CONinvalidunit \fi
    \ifnum \chkWLMtotduplicate > 0 \chkWLMdoD \CONduplicate \fi
    \ifnum \chkWLMtotunsupported > 0 \chkWLMdoE \chkWLMdffuspd
      \chkWLMtotunsupported \CONunsupported \fi
    \chkWLMdoB \chkWLMtotunrecognized {\chkWLMdoE \chkWLMdffurec
      \chkWLMtotunrecognized \CONunrecognized }5 \empty
    \ifnum \chkWLMtotmissingwidth > 0 \chkWLMdoD \CONmissingwidth \fi
    \ifnum \chkWLMtotundefref > 0 \chkWLMdoE \chkWLMdffudrf
      \chkWLMtotundefref \CONundefref \fi
    \ifnum \chkWLMtotfallback > 0 \chkWLMdoD \CONfallback \fi
    \ifnum \chkWLMtotlost > 0 \chkWLMdoD \CONlost \fi
    \ifnum \chkWLMtotabsent > 0 \chkWLMdoD \CONabsent \fi
    \ifFILEtoomany \gadvance \chkWLMtotfile by 1
      \gadvance \chkWLMtotfile by -\NOfilemax
      \ifnum \chkWLMtotfile > 0 \chkWLMdoD \CONfilelc \fi \fi
    \chkWLMdoB \chkWLMtotother {\chkWLMdoD \CONother }1 \empty
    \ifnum #2 = 0 \ifnum \chkWLMclass = 2 \global \chkWLMclass = 3 \fi \fi
    \ifchkWLMokay \message {none}\noecho \fi
    % add errors which are relevant for well-formedness:
    \chkWLMtoterrors = \chkWLMtotmisnesting
    \advance \chkWLMtoterrors by \chkWLMtotoverlap
    \advance \chkWLMtoterrors by \chkWLMtotsuperfluous
    \advance \chkWLMtoterrors by \chkWLMtotunclosed
    \advance \chkWLMtoterrors by \chkWLMtotunquoted
    \advance \chkWLMtoterrors by \chkWLMtotundelimited
    \ifnum \NOinput > 1 \edef \test {\the\NOinput}%
      \echo {Chained documents: \test } \fi
    \ifnum \NOinclude > 0 \edef \test {\the\NOinclude}%
      \echo {Included documents: \test } \fi
    \edef \test {\the \chkWLMtotelements}%
    \echo {Rated elements: \test}\noecho
    \message {Well-formedness:} \EXP \chkWLMtoterrors \chkWLMtotelements
    \noecho \message {Namespace declaration: \NAMESPACE}%
    \noecho \message {Doctype declaration: \chkDCTPdcl {#1}}%
    \noecho \message {Markup language: \chkDCTPmsg \chkWLMclass \end}
    \def \chkWLMdoA {\ifnum \chkWLMclass = 2 \chkWLMclass = 3 \fi}%
    \def \chkWLMdoB #1#2#3#4{% evaluate
      \ifnum #1 > 0 #2\ifnum \chkWLMclass < #3%
        \chkWLMclass = #3\else #4\fi \fi}
    \def \chkWLMdoC #1#2#3{% evaluate number
      \chkWLMdoB #1}{\chkWLMdoD {#2}}{#3}{#3}{#3}{#3}{#3}{#3}{#3}{#3}{#3}
    \def \chkWLMdoD #1{\chkWLMokayfalse % number
      \ifnum \csname \CONtot #1\endcsname = 1
        \echo { 1 \csname \CONchkPLR #1\endcsname \empty
          \chkWLMerrordiff } \else

```

```

\echo { \the \csname \CONtot #1\endcsname \CONspace \csname
\CONchkPLR #1\endcsname s\chkWLMerrordiff}\fi}
\def \chkWLMdoE #1#2#3{% difference
\chkWLMdonodifferent {#1}{#2}\chkWLMdoD {#3}}
\def \chkWLMdonodifferent #1#2{%
\ifnum #1 > 1 \ifnum #1 < 2
\edef \chkWLMerrordiff {\the #1\CONspace different}%
\else \let \chkWLMerrordiff \empty \fi
\else \ifnum #2 > 1 \let \chkWLMerrordiff \CONallthesame
\else \let \chkWLMerrordiff \empty \fi \fi}

\def \chkWLMmessdiff #1#2#3#4#5{% message different
\DEFsearch {#5#1}\ifDEF \else \echo {! #2 \CHKerrline.}\xDef {#5#1} }%
\gadvance #3 by 1 \fi \gadvance #4 by 1 }

\new \ifchkWLMokay

% CODEPAGE: =====

\ifx \eTeXversion \undefined
\echo {codepages ! Unsupported (eTeX required);}%
\else \message {codepages,}\fi

% Each Unicode Value is stored in a count register.
% The data associated with that Unicode Value is stored in a token register.
% Codepages are stored in eTeX's extended register set,
% which can hold 127 of them. They are allocated in reverse order,
% opposed to the standard allocation mechanism working from bottom to top.
% The first codepage gets \CPGSnumberof = 1 and occupies registers number
% 32512 (= 127*256) to 32767 (= 128*256-1).
% So \CPGregofindex runs from \CPGgregbase=32512 to 32767.
% Second codepage: \CPGSnumberof = 2,
% \CPGregofindex = (128-2)*256-1 .. (128-1)*256-1

% A codepage is a set of 256 (or less) references to the unicode
% character space. The codepage data file format is: < 1) index,
% two hexadecimal digits; 2) unicode point, four hex. digits;
% 3) unicode character name (optional) >.
% The codepage files like CP1252.txt have been extracted from
% http://www.microsoft.com/typography/unicode/1252.htm etc.

\let \CPG \GMScodepage

\ldef \CPGadd #1.#2 {% add codepage file to CPGs list
\def \CPGaddy #1{%
\ifx \CPGaddy \empty \MSGcodepageerror \CPGaddy \else
\uppercase {\edef \CPGadditem {#1}}%
\expandafter \ARRAYone \CPGadditem \ARRAYafter \CPGS \fi}
\ifx \eTeXversion \undefined \ldef \CPGadd #1.#2 {} \fi}

% Create \Def CPG-<document codepage> {<HTML codepage>, <DOS codepage>}:
\def \CPGaddalias #1 (#2, #3){\edef \CPGaddaliasZ {#3}%
\ifx \CPGaddaliasZ \GMScodepage
\Def {\CONCPG #1} {\CHKerrcodepagereplaced {#1}%
\lowercase {\xdef \CPGhtml {#2}}\uppercase {\xdef \CPG {#3}}}\fi}

\ldef \CPGnbl #1\relax {% enable, #1: codepage name like CP125x
\CPGname = \expandafter {#1}%
\uppercase {\edef \CPGnblfirstuc {#1}}%
\lowercase {\edef \CPGnblfirstlc {#1}}%
\DEFsearch {\CONcodepagenumber \CPGnblfirstuc}%
\ifDEF \CPGnblA \CPGnblfirstuc \else
\DEFsearch {\CONcodepagenumber \CPGnblfirstlc}%
\ifDEF \CPGnblA \CPGnblfirstlc \else
\CHKerrnocodepage {\CPGnblfirstlc | \CPGnblfirstuc}\fi \fi}
\ifx \eTeXversion \undefined \ldef \CPGnbl #1\relax {} \fi}
\def \CPGnblA #1{\CPGnblAD {#1}\CPGnblAE
\ifUCDwriteenc \CPGnblAC #1\fi
\CPGregofindex = \CPGgregbase \count 8 = 0

\loop \ifnum \count 8 > 128
\catcode \count 8 = \CATletter % Fixme: This just enables hyphenation
\lccode \count 8 = \count 8 % but is wrong for uppercase letters
\fi \advance \count 8 by 1
\CPGnblAA {\the \count \CPGregofindex}%
\ifnum \regtopofCPG > \CPGregofindex
\advance \CPGregofindex by 1
\repeat \ifUCDwriteenc \CPGnblAB \fi}
\ldef \CPGnblAA #1{% character (Standard Unicode Value, decimal)
\edef \CPGnblAAZ {#1}%
\ifx \CPGnblAAZ \empty \else \UCDslotset {#1}%
\edef \myslot {\the \UCDslot slot}%
\DEFsearch \myslot \ifDEF \UCDslotpointregset {#1}\UCDget
\ifUCDwriteenc \CPGnblAAA \fi \else \MSGnochar {#1}\fi \fi}
\def \CPGnblAAA {% character line
\edef \testglyphname {\the \UCDglyphname }%
\ifx \testglyphname \empty \def \testglyphname {.notdef}\fi
\HEXinput = \the \count \CPGregofindex \HEXoutput = {} \HEX
\edef \codepoint {%
\CONspace \CONpercent \CONspace \the \HEXoutput}%
\CPGencwriteline {\testglyphname \codepoint}}%
\def \CPGnblAB {\CPGencwriteline {} \def}%
\CPGencwriteline {\CONpercent \CONspace
Generated by \GMSname \CONspace (\GMSdomain)}%
\immediate \closeout 3 } % file close
\lowercase {\def \CPGnblAC #1{%
\immediate \openout 3 #1.enc\message {#1.enc}%
\CPGencwriteline {/#1-Encoding []}} % file open}
\def \CPGnblAD #1{\CPGgregbase = 128
\advance \CPGgregbase by -\csname \CONcodepagenumber#1\endcsname
\multiply \CPGgregbase by 256 } % register base
\def \CPGnblAE {\CPGregofindex = \CPGgregbase
\advance \CPGregofindex by 255
\edef \regtopofCPG {\the \CPGregofindex}} % register top

\def \CPGencwrite {\def \ [##1] {\def \first {##1}%
\ifx \first \empty \else \expandafter \CPGnbl \first \relax \fi}%
\hyphenmessage \message {Writing encoding file(s):}%
\UCDwriteenctrue \CPGS \UCDwriteencfalse}

\ldef \CPGencwriteline #1{\immediate \backupwrite 3{#1}}

\def \CPGget #1#2#3\relax {\edef \CPGgetXY {#1#2}%
\ifx \CPGgetXY \CONcp \xdef \CPG {CP#3}\xdef \CPGhtml {CP#3}%
\ifnum #3 > 1249 \ifnum #3 < 1259 \xdef \CPGhtml {windows-#3}\fi \fi \fi}

\def \CPGgetdefaultname {% get codepage info from markup file
\expandafter \CPGget \CPG \relax
\ifx \CPGhtml \empty \expandafter \CPGgetiso \CPG \relax \fi
\ifx \CPGhtml \empty \LENGTHget \CPG
\ifnum \LENGTH > 8 \expandafter \CPGgetwindows \CPG \relax \fi \fi}

\lowercase {\def \CPGgetiso #1#2#3#4#5\relax {\edef \CPGgetisotest {#1#2#3}%
\ifx \CPGgetisotest \CONiso \xdef \CPGhtml {ISO-#5}\xdef \CPG {I#5}\fi}}

\def \CPGgetwindows #1#2#3#4#5#6#7#8#9\relax {%
\edef \CPGgetwindowstest {#1#2#3#4#5#6#7}%
\ifx \CPGgetwindowstest \CONwindows
\xdef \CPGhtml {windows-#9}\xdef \CPG {CP#9}\fi}

\let \CPGhtml \empty

\nec \CPGindex

\def \CPGload {\def \ [##1] {%
\lowercase {\def \lfirst {##1}}\uppercase {\def \ucfirst {##1}}%
\FILEsearch {\lfirst .txt}%
\ifFILEexist \CPGloadA \lfirst \else \FILEsearch {\ucfirst .txt}%
\ifFILEexist \CPGloadA \ucfirst \fi \fi } \CPGS}

```

```

\ifx \eTeXversion \undefined \let \CPGload \relax \fi
\ldef \CPGloadA #1{% Code page name like CP125x
  \CPGloadAB {#1}\CPGloadAC %% \CPGloadAD
  \openin 1 = #1.txt
  \loop \FILEcheckeof 1 \ifFILEnotended \read 1 to \CPGloadline
    \ifx \CPGloadline \CONpar \else
      \ifx \CPGloadline \empty \else \expandafter
        \CPGloadAA \CPGloadline ; ; \relax \fi \fi \repeat \closein 1}
\ldef \CPGloadAA % line
  #1;% Codepage index (2 uppercase hexadecimal digits)
  #2;% Standard Unicode Value or CUS UV (4 uppercase hexadecimal digits)
  #3 ;% Unicode character names for standard UVs
  #4\relax {% Fixme: does it fail if one line is just missing?
    \edef \CPGloadAlinesecnd {#2}%
    \CPGgregofindex = \CPGgregbase
    \uppercase {\advance \CPGgregofindex by "#1}%
    \ifx \CPGloadAlinesecnd \CONundefined \else
      \uppercase {\count \CPGgregofindex = "#2}%
      \uppercase {\UCDsotset {#2}}%
      \edef \CPGloadAlineslot {\the \UCDsot slot}%
      \DEFsearch \CPGloadAlineslot \ifDEF
        \uppercase {\UCDslotpointregtest {#2}}\UCDget
        \ifx \UCDgettest \empty \else
          \ifx \testcharset \GMScharset
            \uppercase {\UCDdataset
              ; ; ; ; ; ; \testcharset; "#1; ; ; ; \relax } \fi
          \edef \CPGloadAlinecheck {\the \CPGname}%
          \ifx \CPGloadAlinecheck \empty
            \uppercase {\UCDdataset
              ; ; ; ; ; ; \testcharset; "#1; ; ; ; \relax } \fi
          \global \toks \UCDslotpointreg =
            \expandafter {\the \UCDdata } \fi
          \else \MSGnoslot \fi \fi}
\ldef \CPGloadAB #1{\edef \testcharset {#1}% number
  \advance \CPGSnumberof by 1
  \xDef {\CONcodepagenumber#1} {\the \CPGSnumberof}%
  \hyphenmessage \message {\the \CPGSnumberof.} \message {#1.txt}}
\ldef \CPGloadAC {\CPGgregbase = 128 % next register base
  \advance \CPGgregbase by -\CPGSnumberof \multiply \CPGgregbase by 256 }
\ldef \CPGloadAD {% show register number
  \edef \CPGloadAregnumbershowtest {\the \CPGgregbase}%
  \message {[ C\CPGloadAregnumbershowtest \CONspace to}%
  \advance \CPGgregbase by 255
  \edef \CPGloadAregnumbershowtest {\the \CPGgregbase}%
  \message {C\CPGloadAregnumbershowtest \CONspace .}}

\net \CPGname

\new \ifCPGplain

\nec \CPGgregbase

\nec \CPGgregofindex

\nec \CPGSnumberof

\ldef \CPGwincheck {\expandafter \CPGwincheckA \CPG \relax}
\ldef \CPGwincheckA #1#2#3#4#5#6\relax {\edef \CPGwinchecktest {#1#2#3#4#5#6}
  \ifx \CPGwinchecktest \CONwindowsCP \CPGwintrue \else \CPGwinfalse \fi}

\let \CPGS \empty

\new \ifCPGwin

% COLOR: =====

\new \ifCLR

\ldef \CLRactive {\CLRforeground}

```

```

\net \CLRanchorex
\net \CLRanchorint

\def \CLRbackground {1 1 1 rg 1 1 1 RG} % RGB white

\new \ifCLRbg
\new \ifCLRbgbody \let \CLRbgbody \CLRbackground
\new \ifCLRbgspan \let \CLRbgspan \CLRbackground %%
\new \ifCLRbgtable

\new \ifCLRborder \def \CLRborder {0 0 0 rg 0 0 0 RG} % RGB black

\def \CLRforeground {0 0 0 rg 0 0 0 RG} % RGB black
\let \CLRforegroundbackup \CLRforeground

% Get color by name or hex string:
\bgrouper \CATootherhashmark \CATparameterdegree
\gdef \CLRget {\VALgetlc \DEFsearch {\CONrgb \VALlowercase}%
  \ifDEF \clrB \else \clrA \fi}
\gdef \clrA {\VALgetuc \expandafter \clrAA \VALuppercase #000000\relax % hex
  \ifCLRgetanyshort \else
    \expandafter \clrC \VALuppercase #000000#\relax \fi}
\gdef \clrAA #1#2#3#4#5#6#7#8\relax {\edef \clrAW {#5}% short hex
  \ifx \clrAW \CONhashmark \CLRgetanyshorttrue
    \expandafter \clrC #2#2#3#3#4#4#000000#\relax
  \else \CLRgetanyshortfalse \fi}
\gdef \clrB {\edef \clrBprobe {\csname \CONrgb \VALlowercase \endcsname}%
  \expandafter \clrC \clrBprobe #\relax}% name
\gdef \clrC #1#2#3#4#5#6#7#8\relax {% rgb
  \lowercase {\edef \clrCtest {#1}}%
  \ifx \clrCtest \empty
    \uppercase {\clrE \CLRgetallred \CLRgetallRED {"#2#3}}%
    \uppercase {\clrE \CLRgetallgreen \CLRgetallGREEN {"#4#5}}%
    \uppercase {\clrE \CLRgetallblue \CLRgetallBLUE {"#6#7}}%
  \else \clrD \fi } \egroup
\def \clrD {\expandafter \clrDA \clrCtest rgb(,,)\relax}
\def \clrDA #1rgb#2(#3,#4,#5)#6\relax {% parse
  \edef \clrDAtest {#1}\global \CLRgetanyfailedfalse
  \ifx \clrDAtest \empty \clrE \CLRgetallred \CLRgetallRED {#3}%
    \clrE \CLRgetallgreen \CLRgetallGREEN {#4}%
    \clrE \CLRgetallblue \CLRgetallBLUE {#5}\else \clrDAA \fi}
\def \clrDAA {\global \CLRgetanyfailedtrue
  \ifBKGDgiven \CHKerrrunsupattprop \CONbackground
  \else \global \CLRgetanyfailedtrue
  \CHKerruncolored \clrDAtest \fi} % error
\def \clrE % trans
  #1% color count register
  #2% color token register
  #3% brightness ("00..."FF)
  \def \clrEfirst {#1}\def \clrEsecond {#2}\clrEfirst = #3
  \multiply \clrEfirst by 1000 \divide \clrEfirst by 256
  \advance \clrEfirst by 3 % rounding range
  \ifnum \clrEfirst < 4 \clrEsecond = {0}\else
  \ifnum \clrEfirst < 10 \clrEA {00}\else
  \ifnum \clrEfirst < 100 \clrEA {0}\else
  \ifnum \clrEfirst < 996 \clrEA \empty \else
  \clrEsecond = {1}\fi \fi \fi \fi}
\def \clrEA #1{\edef \clrEtest {0.#1\the \clrEfirst}}% set
  \clrEsecond = \expandafter {\clrEtest}}
\def \CLRgetall {\the \CLRgetallRED \CONspace
  \the \CLRgetallGREEN \CONspace \the \CLRgetallBLUE}
\nec \CLRgetallblue \net \CLRgetallBLUE \CLRgetallBLUE = {0}
\nec \CLRgetallgreen \net \CLRgetallGREEN \CLRgetallGREEN = {0}
\nec \CLRgetallred \net \CLRgetallRED \CLRgetallRED = {0}
\new \ifCLRgetanyfailed \new \ifCLRgetanyshort

\def \CLRize #1{\pdfliteral {#1}}
\ifx \pdfliteral \undefined \def \CLRize #1{\special {PDF:#1}}\fi

```

```

\def \CLRizebackground {\CLRize \CLRbgbody
\box to 0mm {\vskip -1in \vskip -\voffset
\hbox to 0mm {\hskip -1in \hskip -\hoffset
\vrule height \OUTheight width \OUTwidth \hss }\vss}%
\CLRize \CLRforeground}
\def \CLRizeborder {\ifCLRborder \CLRize \CLRborder
\else \CLRize \CLRforeground \fi}
\def \CLRizeforeground {\ifCLR \CLRize \CLRforeground \fi}

\def \CLRlistdot {\the \CLRuldot \CONspace rg \the \CLRuldot \CONspace RG }

\ifx \TeXeTstate \undefined
\let \CLRltr \relax
\else \def \CLRltr {\ifnum \TeXeTstate < 1 \CLRizeforeground \fi } \fi

\def \CLRrtl {\ifnum \TeXeTstate > 0 \CLRizeforeground \fi}
\ifx \TeXeTstate \undefined \let \CLRrtl \relax \fi

\def \CLRset #1{\edef #1{\CLRgetall \CONspace rg \CLRgetall \CONspace RG}}
\def \CLRsetbackground #1\relax {\edef \CLRbackground {#1}}
\def \CLRsetbackgroundbody {\CLRsetbg \CLRbgbodytrue
\edef \CLRbgbody {\CLRbackground}}
\def \CLRsetbg {\CLRbgtrue \CLRget \CLRset \probe
\ifCLRgetanyfailed \else \expandafter \CLRsetbackground \probe \relax \fi}
\def \CLRsetbgbody {\CLRsetbg \CLRbgbodytrue \CLRbordertrue
\edef \CLRbgbody {\CLRbackground}}
\def \CLRsetbgspan {\CLRsetbg \CLRbgspantrue
\edef \CLRbgspan {\CLRbackground}}
\def \CLRsetbgtable {\CLRbgtabletrue \CLRsetbg
\edef \CLRbgtable {\CLRbackground}}
\def \CLRsetborder {\CLRbordertrue \CLRget \CLRset \CLRborder}
\def \CLRsetforeground {\CLRtrue \CLRget \CLRset \CLRforeground}
\def \CLRsetlink {\{\CLRget \edef \CLRsetlinktest {\CLRgetall}%
\global \CLRanchorexnt = \expandafter {\CLRsetlinktest}}
\def \CLRsetlistdot {\CLRtrue \CLRget \CLRset \CLRlistdot}
\def \CLRsetvlink {\{\CLRget \edef \CLRsetvlinktest {\CLRgetall}%
\global \CLRanchorint = \expandafter {\CLRsetvlinktest}}

\net \CLRuldot

% CONSTANT: =====

\def \CON {}
\def \CONABSENT {ABSENT}
\def \CONabsent {absent}
\def \CONacceptcharset {accept-charset}
\def \CONadobe {(Adobe)}
\def \CONafter {after}
\def \CONalias {ALIAS-}
\def \CONall {all}
\def \CONallthesame {(all the same)}
\def \CONalways {always}
{\catcode `& = \CATOther \gdef \CONampersand {&}}
\def \CONapostroph {'}
\def \CONArial {Arial}
\def \CONasterisk {*}
\def \CONasteriskslash */}
\def \CONat { at }
\def \CONatt {ATT}
\edef \CONattstop {... = \CONspace \CONspace}
\edef \CONattstopspace {... = \CONspace \CONspace}
\def \CONattstop {... = { }{ }
\def \CONattstopspace {... = { }{ }{ }
\def \CONauthor {author}
\def \CONauto {auto}
\def \CONavoid {avoid}
\def \CONbase {base}
\def \CONbasefont {basefont}
\def \CONbefore {before}

\def \CONbegin {begin}
\def \CONblock {block}
\def \CONbold {bold}
\def \CONborderstyle {brdA}
\def \CONbottom {bottom}
\def \CONBOXcell {BOXcell}
\def \CONbp {bp}
\def \CONcanadian {Unified Canadian Aboriginal Syllabic}
\def \CONcc {cc}
\def \CONcell {cell}
\def \CONcenter {center}
\def \CONchkPLR {chkPLR}
\def \CONck {ck}
\def \CONCLASS {CLASS}
\def \CONclass {class}
\def \CONcm {cm}
\def \CONCMR {CMR}
\def \CONcmr {cmr}
\def \CONcodepage {codepage}
\def \CONcodepagenumber {CPGnumber}
\def \CONcolwidth {COLwidth}
\def \CONcomma {,}
\def \CONcommaquotmark {,"}
\def \CONcomment {comment}
\def \CONcommentbegin {!--}
\def \CONcommentend { --}
\def \CONcompatibility {Compatibility}
\def \CONconfu {CONFU}
\def \CONconfused {confused}
\def \CONcon {CON}
\def \CONcontenttype {content-type}
\def \CONcontext {CONTEXT}
\def \CONcorporateuse {Corporate Use}
\def \CONCOURIER {COURIER}
\def \CONCourier {Courier}
\def \CONCourierNew {CourierNew}
\def \CONCourierNewSpace {CourierNew } % includes trailing space
\def \CONcp {CP}
\def \CONCPG {CPG}
\def \CONcreated {created}
\def \CONdate {date}
\def \CONdblhyph {--}
\def \CONdblhyphspace {-- }
\def \CONdblspchiph { -- }
\def \CONdd {dd}
\def \CONde {de}
\def \CONdescription {description}
\def \CONdest {DEST}
\def \CONdepr {DEPR}
\def \CONdeprap {DEPRAP}
\def \CONdeprecated {deprecated}
\def \CONdirection {direction}
\def \CONdoctype {DOCTYPE}
\def \CONdot {.}
\def \CONdotat {.@}
\def \CONdotdot {...}
\def \CONdotdottwo {../..}
\def \CONdotdottthree {.../...}
\def \CONdotdottfour {.../.../...}
\def \CONdotdottfive {.../.../.../...}
\def \CONdotdotseven {.../.../.../.../...}
\def \CONdotdoteight {.../.../.../.../.../...}
\def \CONdohtml {.htm}
\def \CONdotjpg {.jpg}
\def \CONdotjpeg {.jpeg}
\def \CONdotpdf {.pdf}
\def \CONdotpng {.png}

```



```

\def \CONdotslash {./}
\def \CONdotslashnonejpg {./none.jpg}
\def \CONdummy {dummy}
\def \CONduplicate {duplicate}
\def \CONem {em}
\def \CONex {ex}
\def \CONEMPTY {EMPTY}
\def \CONempty {empty}
\def \CONemsp {emsp}
\def \CONencoding {ENCODING-}
\def \CONend {end}
\def \CONendcomment {-->}
\def \CONendcommentendscript {--></script>}
\def \CONendmark {... = {}}
\def \CONendmarkdots {...{}}
\def \CONendmarktwo {... = { }... = {}}
\def \CONendmarkequal {={}{}}
\def \CONendobject </object>
\def \CONendscript </script>
\def \CONendtable {/table}
\def \CONendtd {/td}
\def \CONendth {/th}
\def \CONendtr {/tr}
\def \CONensp {ensp}
\def \CONentity {ENT}
\def \CONexclamation {!}
\def \CONextended {Extended}
\def \CONextension {Extension}
\def \CONexternalid {externalid} % Gedcom
\def \CONfallback {fallback}
\def \CONff {ff}
\def \CONfile {FILE}
\def \CONfilelc {file}
\def \CONfilter {filter}
\def \CONfontfamily {font-family}
\def \CONfontsize {font-size}
\def \CONfontstyle {font-style}
\def \CONfonthyphen {FONT-}
\def \CONform {form}
\def \CONframespacing {framespacing}
\def \CONftp {ftp}
\def \CONgedcomxml {GEDCOM XML 6.0}
\def \CONglyph {GLY}
\def \CONgms {GMS}
{\catcode `# = \CAToOther \gdef \CONhashmark {#}}
\def \CONHELVETICA {HELVETICA}
\def \CONHelvetica {Helvetica}
\def \CONHTMLirreg {HTML irregular}
\def \CONHTMLprop {HTML proprietary}
\def \CONHTMLstrict {HTML 4.01 strict}
\def \CONHTMLtrans {HTML 4.01 transitional}
\def \CONHTMLunknown {HTML unknown}
\def \CONhttp {http}
\def \CONhttpequiv {http-equiv} % check if needed
\def \CONhttps {https}
\def \CONhyphen {-}
\def \CONhyphgmshyph {-GMS-}
\def \CONideograph {Ideograph}
\def \CONIMG {IMG}
\def \CONin {in}
\def \CONinclinebegin {vvvvvvvvvvvv}
\def \CONinclineend {AAAAAAAAAAAA}
\def \CONincluded {included}
\def \CONinvalidunit {invalidunit}
\def \CONisindex {isindex}
\def \CONiso {ISO}
\def \CONitalic {italic}
\def \CONjpegdot {jpeg.}
\def \CONjpgdot {jpg.}
\def \CONkerning {Kerning}
\def \CONkeywords {keywords}
\def \CONkern {KRN-}
\def \CONlarge {large}
\def \CONlarger {larger}
\def \CONLatin {Latin}
\def \CONleft {left}
\def \CONlineheight {line-height}
{\catcode `# = \CAToOther \gdef \CONlessthan {<}}
\def \CONlevel {level}
\def \CONldothtm {l.htm}
\def \CONll {ll}
\def \CONll {Ll} % Fixme: rename to \CONLl?
\def \CONlm {Lm}
\def \CONlo {Lo}
\def \CONlowsrc {Lowsr}
\def \CONlost {lost}
\def \CONlt {Lt}
\def \CONltr {ltr}
\def \CONlu {Lu}
\def \CONmargin {margin}
\def \CONmath {Mathematic}
\def \CONmedium {medium}
\def \CONmenu {menu}
\def \CONmiddle {middle}
\def \CONmisc {Miscellaneous}
\def \CONmisnesting {misnesting}
\def \CONmissingwidth {missingwidth}
\def \CONmm {mm}
\def \CONmonospace {MONOSPACE}
\def \CONnbsp {nbsp}
\def \CONnest {NEST}
\def \CONnext {next}
\def \CONnn {nn}
\def \CONnone {none}
\def \CONnorepeat {no-repeat}
\def \CONnormal {normal}
\def \CONnumeight {8}
\def \CONnumfive {5}
\def \CONnumfour {4}
\mef \CONnumlarge = 1200 % 4
\mef \CONnummedium = 1000 % 3<-\baseFONTsize, to be scaled by 1.2
\def \CONnumone {-1}
\def \CONnumthree {-3}
\def \CONnumtwo {-2}
\def \CONnumnine {9}
\def \CONnumone {1}
\def \CONnumfour {+4}
\def \CONnumpone {+1}
\def \CONnumthree {+3}
\def \CONnumtwo {+2}
\def \CONnumseven {7}
\def \CONnumsix {6}
\mef \CONnumsmall = 833 % 2
\def \CONnumthree {3}
\def \CONnumtwo {2}
\mef \CONnumxlarge = 1440 % 5
\mef \CONnumxsmall = 694 % 1
\mef \CONnumxxlarge = 1728 % 6
\mef \CONnumxxsmall = 579
\mef \CONnumxxxlarge = 2074 % 7
\mef \CONnumxxxsmall = 482
\mef \CONnumxxxxlarge = 2488
\mef \CONnumxxxxlarge = 2986
\def \CONnumzero {0}
\def \CONoblique {oblique}
\def \CONonblur {onblur}

```

```

\def \CONonchange {onchange}
\def \CONonclick {onclick}
\def \CONoncopy {oncopy}
\def \CONondblclick {ondblclick}
\def \CONonfocus {onfocus}
\def \CONonkeydown {onkeydown}
\def \CONonkeypress {onkeypress}
\def \CONonkeyup {onkeyup}
\def \CONonload {onload}
\def \CONonmousedown {onmousedown}
\def \CONonmousemove {onmousemove}
\def \CONonmouseout {onmouseout}
\def \CONonmouseover {onmouseover}
\def \CONonmouseup {onmouseup}
\def \CONonreset {onreset}
\def \CONonselect {onselect}
\def \CONonsubmit {onsubmit}
\def \CONonunload {onunload}
\def \CONopen {OPEN}
\def \CONother {other}
\def \CONoverlap {overlap}
\def \CONp {p}
\def \CONpadding {padding}
\def \CONpar {\par}
\def \CONparam {param}
\def \CONpc {pc}
\def \CONpdfdot {pdf.}
\def \CONpercent {% \CAToother \gdef \CONpercent {%}
\def \CONplain {PLAIN}
\def \CONpleasedonotwrite {Please do not write}
\def \CONpngdot {png.}
\def \CONpresentation {Presentation Forms}
\def \CONprint {print}
\def \CONprop {PROP}
\def \CONpropap {PROPAP}
\def \CONproppadding {PROPPADDING-}
\def \CONproprietary {proprietary}
\def \CONpt {pt}
\def \CONpx {px}
\def \CONquotation {?} % Fixme: "question"!
\def \CONquotmark {"}
\def \CONref {REF}
\def \CONrelax {\relax}
\def \CONrepeatx {repeat-x}
\def \CONrepeaty {repeat-y}
\def \CONrgb {RGB}
\def \CONright {right}
\def \CONroman {roman}
\def \CONrr {rr}
\def \CONrtl {rtl}
\def \CONsansserif {SANS-SERIF}
\def \CONscreen {screen}
\def \CONsemi {;}
\def \CONserif {SERIF}
\def \CONshade {shade}
\def \CONshy {shy}
\def \CONslash {/}
\def \CONslot {slot}
\def \CONsmall {small}
\def \CONsmaller {smaller}
\def \CONsolid {solid}
\def \CONsp {sp}
\def \CONspecific {SPECIFIC-}
\def \CONstop {STOP}
\def \CONstopmark {GMS-STOP}
\def \CONstopreference {GMS-STOP}
\def \CONstyle {style}
\def \CONSTYLE {STYLE}

\def \CONstylesheet {stylesheet}
\def \CONsummary {summary}
\def \CONsuperfluous {superfluous}
\def \CONsupplement {Supplement}
\def \CONsymbols {Symbols}
\def \CONTABLE {TABLE} % Fixme: unused?
\def \CONtable {table}
\def \CONtag {TAG}
\def \CONtext {text}
\def \CONtextalign {text-align}
\def \CONtextarea {textarea}
\def \CONtextcss {text/css}
\def \CONtexthtml {text/html}
\def \CONtextindent {text-indent}
\def \CONtextplain {text/plain}
\def \CONtextTeX {text/TeX}
\def \CONthatsall { ... = }% with three spaces
\def \CONthereshouldbe {there should be}
\def \CONthick {thick}
\def \CONthin {thin}
\def \CONthinsp {thinsp}
\def \CONtifdot {tif.}
\def \CONtiffdot {tiff.}
\def \CONTIMES {TIMES}
\def \CONTimes {Times}
\def \CONTimesNewRoman {TimesNewRoman}
\def \CONtop {top}
\def \CONtot {chkmLmtot}
\def \CONtrue {true}
\def \CONtt {tt}
\def \CONtwocolumns {twocolumns}
\def \CONunclosed {unclosed}
\def \CONuncolored {uncolored}
\def \CONundecl {UNDECL}
\def \CONundefclass {undefclass}
\def \CONundefined {UNDEFINED} % with trailing space
\def \CONundefref {undefref}
\def \CONundelimited {undelimited}
\def \CONunknown {unknown}
\def \CONunlinked {unlinked}
\def \CONunquoted {unquoted}
\def \CONunrec {UNREC}
\def \CONunrecap {UNRECAP}
\def \CONunrecognized {unrecognized}
\def \CONunsup {UNSUP}
\def \CONunsupap {UNSUPAP}
\def \CONunsupported {unsupported}
\def \CONunrendered {unrendered}
\def \CONupskip {LINEupskip}
\def \CONwbr {wbr}
\def \CONwin {WIN}
\def \CONwincp {WINCP}
\def \CONwindows {windows}
\def \CONwindowsCP {CP125}
\def \CONx {x}
\def \CONX {X}
\def \CONXHTMLstrict {XHTML 1.0 strict}
\def \CONXHTMLtrans {XHTML 1.0 transitional}
\def \CONxlarge {x-large}
\def \CONxml {xml}
\def \CONxmlang {xml:lang}
\def \CONxmlns {xmlns}
\def \CONxmlnshtml {xmlns:html}
\def \CONxmlnsutility {xmlns:utility}
\def \CONxmlspace {xml:space}
\def \CONxsmall {x-small}
\def \CONxxlarge {xx-large}
\def \CONxxsmall {xx-small}

```

```

\def \CONzs {Zs}

% CSS: =====
% Parse style gap (a token sequence of the form 'selectors {declaration}':
\new \ifCSSadded

\bgrouP \CATstylegap
\gdef \CSSget #1{\let \CSSgetuncom \empty
\edef \CSSprobe [#1/*\CONstopmark */}%
\expandafter \CSSgetA \CSSprobe \relax
\expandafter \cssA \CSSgetuncom {} \CSSgetB {} \relax ]\egroup
\ldef \CSSgetA #1/*#2*/{\edef \CSSgetAZ {#2}% comment
\edef \CSSgetuncom {\CSSgetuncom #1}% uncommented
\ifx \CSSgetAZ \CONstopmark \let \CSSnext \relax
\else \let \CSSnext \CSSgetA \fi \CSSnext}
\def \CSSgetB #1\relax {} % gulp
\bgrouP \CATstylegap
\gdef \cssA #1#2{\let \cssAnext \relax
\let \cssAgrpbkp \empty % group backup
\let \cssActxbkp \empty % context backup
\GAPchkitvoid \def \cssAY [#1] selectors
\edef \cssAZ [#2] declarations
\ifx \cssAZ \empty \else \ifx \cssAY \empty
\ifx \cssAZ \CONspace \else \cssAA \fi \fi \fi
\ifx \cssAY \empty \let \cssAZ \empty \else
\ifx \cssAY \CONspace \let \cssAZ \empty \fi \fi
\ifx \cssAZ \empty \else \ifx \cssAY \CONdblspchph \else
\ifx \cssAY \CONdblhyphspace \else \ifx \cssAY \CONdblhyph
\else \cssB \fi \fi \fi \fi \cssAnext]
\gdef \cssAA [\CHKerrunselected {\cssAZ} \let \cssAY \CONunknown ]\egroup
\def \cssB {\expandafter \cssBA \cssAY , \CONstopmark
\let \cssAnext \cssA }% group
\ldef \cssBA #1,#2{\edef \cssBAZ {#2}%
\edef \cssBAtest {\cssAgrpbkp #1 \CONspace }%
\expandafter \cssC \cssBAtest \CONstopmark
\ifx \cssBAZ \CONstopmark \cssBAB \else \cssBAA \fi \cssBANext}
\def \cssBAA {\let \cssAgrpbkp \cssBAZ \let \cssBANext \cssBA }% run
\def \cssBAB {\let \cssBANext \relax }% stop
\def \cssC #1 #2{\edef \cssCY {#1} \edef \cssCZ {#2}%
\ifx \cssCZ \CONstopmark \cssCB \else \cssCA \fi \cssBANext}
\let \cssCcont \empty % context - Fixme: rename
\def \cssCA {\edef \cssCcont {\cssActxbkp \cssCY }% run
\ifx \cssCcont \empty \else \cssD \fi
\let \cssActxbkp \cssCZ \let \cssBANext \cssC}
\def \cssCB {\let \cssBANext \relax \cssD \let \cssActxbkp \empty}
\def \cssD {\edef \cssDtest {\cssActxbkp \cssCY }% selector
\expandafter \cssE \cssDtest \relax}
\def \cssE #1#2\relax {\edef \cssEYZ {#1#2}%
\ifx \cssEYZ \cssCcont \else \def \cssEY {#1}%
\ifx \cssEY \CONdot \lowercase {\edef \cssEZ {#2}}%
\else \ifx \cssEY \CONhashmark \lowercase {\edef \cssEZ {ID-#2}}%
\else \lowercase {\edef \cssEZ {#1#2}}\fi \fi
\ifx \cssCcont \CONasteriskslash \else
\ifx \cssCcont \empty \else \cssEA \fi \fi
\edef \cssElast {\csname \CONSTYLE \cssEZ \endcsname }%
\DEFsearch {\CONSTYLE \cssEZ } \ifDEF \cssEB \else \cssEC \fi \fi}
\def \cssEA {\edef \cssEZ {\cssEZ - \CONcontext - \cssCcont}} % do
\def \cssEB {\global \CSSaddedtrue % make style list resident
\xDef {\CONSTYLE \cssEZ} {\cssElast ; \CONdummy : \CONempty ; \cssAZ}}
\def \cssEC {\xDef {\CONSTYLE \cssEZ} {\cssAZ}} % make list resident

% ELEMENT: =====
\net \ELEMENT

\new \ifELEMENTblock

\def \ELEMENTblockindent {% generic
\ifELEMENTblock \ifhmode \par \fi \FLTget \noindent \GAPspacenowtrue \fi}

\def \ELEMENTblockindentbyno {%
\ifALGNcenter \ELEMENTblockindentbynozero \else
\ifnum \Nopar = 1 \ELEMENTblockindentbynozero \else
\ELEMENTblockindentbynosquare \fi \fi}
\def \ELEMENTblockindentbynosquare {%
\parindent = 2 \LINEheight \advance \parindent by -lex \FLTget \indent}
\def \ELEMENTblockindentbynozero {\parindent = 0mm \FLTget \noindent}

% Markup elements can be nested, and this routine checks for well-formedness;
% compare Knuth, pages 21 and 306:

\def \ELEMENTcloseinner {%
\ifinner \message {! </>} \aftergroup \ELEMENTcloseinner \endgroup \fi}

\def \ELEMENTclosenest #1{\csname if\CONopen #1\endcsname
\CHKerrmissingendtag {\csname \CONcon #1\endcsname } \fi}

\def \ELEMENTcloseouter {\ifnum \ELEMENTlevel > 0
\message {! <...>} \aftergroup \ELEMENTcloseouter \endgroup \fi}

\let \ELEMENTcontent \empty

\def \ELEMENTcount {\gadvance \chkWLMtotelements by 1 } % Fixme: move

\nec \ELEMENTlevel

\let \ELEMENTname \empty

\ldef \ELEMENTnew #1
\A #2\B #3\C % start tag commands
\D #4\E #5\F % end tag commands
\ELEMENTnewopen {#1} \gDef \CONcon #1 {#1}%
\gDef \CONtag #1 % define start tag
\csname \CONnest #1\CONbegin \endcsname #2\CLRtl \ELEMENTcount
\beginngroup \advance \ELEMENTlevel by 1
\csname \CONopen #1\CONtrue \endcsname \def \ELEMENTname {#1} #3%
\gDef \CONtag /#1 % define end tag
\csname \CONnest #1\CONend \endcsname \CLRltr #4%
\def \ELEMENTnewtest {#1}%
\ifx \ELEMENTnewtest \ELEMENTname \else
\CHKerrsuperfluous \beginngroup \fi
\CLRtl \endgroup \CLRltr #5}}

\ldef \ELEMENTnewblock #1 % new 2007/06/15 - 2007/07/13
\A #2\B #3\initstart #4\C % start tag commands
\D #5\E #6\F % end tag commands
\ELEMENTnew #1
\A \gapBnospacetrue \GAPwrite
\ifhmode \ALGNtextend \BIDIend \par \fi #2\B
\BLOCKemptytrue %% check this
\MARGINgiventopfalse \MARGINgivenbottomfalse #3\ELEMENTvalueget
\ifLINKunrendered \else
\ifMARGINgiventop \vskip \MARGINtop \fi \BIDIbegin \fi
\CLRizeforeground \FONTdo #4\C \D \GAPwrite
\ifLINKunrendered \else \ifhmode \ifGAPspacenow \hskip -\spaceskip \fi
\ALGNtextend \BIDIend \par \fi
\ifBLOCKspacer \ifdim \pagetotal > \pagegoal \LINEupskip \else
\ifdim \pagetotal < \LINEheight \LINEupskip \fi \fi \fi
\global \BLOCKspacerfalse %%
\ifMARGINgivenbottom \vskip \MARGINbottom \fi \fi #5\E #6\F}

\def \ELEMENTnewblockh #1 #2 #3 #4 #5 {\ELEMENTnewblock h#1
\A \global \hangindent = 0mm \global \hangafter = 1
\NOh = #1 \gadvance \count #1 by 1 % heading counter
\B \OPENhtrue \OUTnopagecheck \Noparresetglobal \parindent = 0mm
\FONTname = \FONTbd \FONTbdtrue \MARGINgivenfalse \SIZEgivenfalse

```

```

\initstart \ifSIZEgiven \else \advance \SIZEbase by #2 \fi
\ifdim \pagetotal > 0.85\pagegoal
\ifdim \pagetotal < \pagegoal \break \fi \fi %%
\vbox \bgroup \edef \TAGHlineheightold {\the \LINEheight}%
\ifLINEheightgiven \else \LINEheight = #3\LINEheight \fi \GAPsetstrut
\ifMARGINgiven \ifdim \pagetotal = Opt
\ifdim \MARGINTop > Opt \vskip \topskip \fi
\else {\dimen 7 = \pagetotal \advance \dimen 7 by 2\LINEheight
\ifdim \dimen 7 > \pagegoal \vskip \topskip \fi } \fi
\else {\LINEheight = \TAGHlineheightold \vskip #4\LINEheight } \fi
\FONTdo %% Fixme: avoid
\ALGNtextrespace \strut %% check this: why explicitly?
\C \D \ifMARGINgivenbottom \else %
\LINEheight = \TAGHlineheightold \vskip #5\LINEheight } \fi
\egroup \E \nointerlineskip \F}

% Heading Empty Scaling Line- Space Space Space
% Level: lines: factor: height: overall: before: after:
% H1 7 2.48832 3 4 3 1
% H2 6 2.0736 3 3 2.25 0.75
% H3 5 1.728 2 3 2.25 0.75
% H4 4 1.44 2 2 1.5 0.5
% H5 3 1.2 2 1 0.75 0.25
% H6 2 1 1 1 0.75 0.25

\def \ELEMENTnewdefa #1 #2 %
\ELEMENTnew #1 \A \GAPwrite \B \ELEMENTvalueget #2\C \D \GAPwrite \E \F}

\ldef \ELEMENTnewgeneric #1 {% 2007/03/09
\lowercase {\edef \ELEMENTnewgenericfirstspace #1 }}%
\expandafter \ELEMENTnew \ELEMENTnewgenericfirstspace
\A \GAPwrite \B \ELEMENTblockfalse \ELEMENTvalueget
\ELEMENTblockindent \ELEMENTnewgenericpseudo \CONbefore
\FONTdo \CLRize \CLRforeground \C
\D \GAPwrite \ELEMENTnewgenericpseudo \CONafter \E \F}
\def \ELEMENTnewgenericpseudo #1{\DEFsearch {\CONSTYLE \ELEMENTname :#1}%
\ifDEF \bgroup \ELEMENTblockfalse
\ELEMENT = \expandafter {\ELEMENTname :#1}%
\ELEMENTvaluegetstyle \ELEMENTblockindent
\FONTdo \CLRize \CLRforeground
\global \GAP = \expandafter {\ELEMENTcontent } \GAPwrite \egroup \fi}

\def \ELEMENTnewopen #1{%
\expandafter \ELEMENTnewopenstart \csname ifOPEN#1\endcsname}

\def \ELEMENTnewopenstart {\csname newif\endcsname}

\let \ELEMENTopen \empty

\def \ELEMENTvalueget %
\ELEMENTvaluegetstyle \ifATT \ATTget \fi \ifPRP \PRPget \fi}
\def \ELEMENTvaluegetstyle {\DEFsearch {\CONSTYLE \the \ELEMENT}%
\ifDEF \ELEMENTvaluegetstyleA \fi}
\def \ELEMENTvaluegetstyleA %
\expandafter \PRPlistget \csname \CONSTYLE \the \ELEMENT \endcsname}

% ENTITY: =====
\begingroup \CATentity
\gldef <!-- @1@2-->{\relax}
\gldef <!ENTITY @1>{\ENTITYYparse @1}\endgroup

\ldef \ENTITYYadd #1.#2 % add ENTITY names file to the ENTITYYname list
\ldef \ENTITYYaddfirst {#1}%
\ifx \ENTITYYaddfirst \empty \else \long \edef \ENTITYYadditem {{#1.#2} }%
\expandafter \ARRAYone \ENTITYYadditem \ARRAYafter \ENTITYYname \fi}

\let \ENTITYYname \empty

\def \ENTITYYnamesload {% load ENTITY names files found on ENTITYYname list
\immediate \openout 6 entity.lst \CATentity
\def \{ #1 } {\def \ENTITYYnamesloadfirst {#1}%
\ifx \ENTITYYnamesloadfirst \empty \else
\hyphenmessage \input \ENTITYYnamesloadfirst \fi } \ENTITYYname
\immediate \closeout 6 }}

\def \ENTITYYnamesshow %
\immediate \backupwrite 18{sort < entity.lst > sortent.lst}%
\def \ENTITYYnamesshowtest {WIN32}%
\ifx \OS \ENTITYYnamesshowtest \openin 7 = entity.lst
\else \openin 7 = sortent.lst \fi
\loop \FILEcheckeof 7 \ifFILEnotended \read 7 to \ENTITYYnamesshowline
\edef \ENTITYYnamesshowline {\ENTITYYnamesshowline \CONspace}%
\expandafter \MSGentityname \ENTITYYnamesshowline \repeat \closein 7}

\begingroup \CATentity
% @1: HTML character name, @2: HTML character number (dec.),
% @3: Unicode character name, @4 Unicode character number (hex.),
% @5: ISO codepage etc.
\ifx \eTeXversion \undefined % parse ENTITY file double line
\gldef \ENTITYYparse @1 CDATA "&#2;" -- @3, U+@4 @5--{\%
\ifnum @4 > 255 \else \immediate \backupwrite 6{@1}%
\xDef {\CONref @1} {#4}\fi }% set unicode reference
\else \gldef \ENTITYYparse @1 CDATA "&#2;" -- @3, U+@4 @5--{\%
\UCDslotset {"@4}\edef \ENTITYYpareslot {\the \UCDslot slot}%
\DEFsearch \ENTITYYpareslot \ifDEF
\UCDslotpointregset {"@4}\UCDdataset -;.;.;.;.;.;.;.;@1;\relax
\global \toks \UCDslotpointreg = \expandafter {\the \UCDdata}%
\immediate \backupwrite 6{@1}%
\xDef {\CONref @1} {#4}% set unicode reference
\else \ERRmessnoslot \fi}\fi \endgroup

% EXPONENTIAL: =====
% Well-formedness in GMS is defined as inverse exponential of the
% errors per element ratio, i. e. exp (-errors/elements):
% exp (-x) = 1 - x + x^2/2! - x^3/3! + ...
% => 3000 exp (-x) = 3000 - (3000x) + (3000x)^2/2000 - (3000x)^3/6000 + ...

\def \EXP #1#2{% #1: errors, #2: elements (count registers)
\EXPwlm = 3000 % get well-formedness
\EXPratio = #1 \multiply \EXPratio by 3000
\ifnum #2 > 0 \divide \EXPratio by #2 \fi
\ifnum #1 < 65535 \EXPterm = 3000
\EXPA {1}{-}{#1}% 1! = 1
\EXPA {2}{-}{#1}% 2! = 2
\EXPA {6}{-}{#1}% 3! = 6
\EXPA {24}{-}{#1}% 4! = 24
\EXPA {120}{-}{#1}% 5! = 120
\EXPA {720}{-}{#1}% 6! = 720
\EXPA {5040}{-}{#1}% 7! = 5040
\divide \EXPwlm by 3
\ifnum \EXPwlm < 1000 \edef \EXPTest {0.}%
\ifnum \EXPwlm < 100 \edef \EXPTest {\EXPTest 0}%
\ifnum \EXPwlm < 10 \edef \EXPTest {\EXPTest 0}%
\ifnum \EXPwlm < 0 \edef \EXPTest {0}\fi \fi \fi
\edef \EXPTest {\EXPTest \the \EXPwlm}%
\else \edef \EXPTest {1}\fi \message {\EXPTest } \fi}
\def \EXPA #1#2#3{% term, #1: faculty, #2: sign, #3: errors (count register)
\ifnum #3 > 0 \multiply \EXPterm by \EXPratio \divide \EXPterm by 3000
#3 = \EXPterm \divide #3 by #1 \advance \EXPwlm by #2#3 \fi}
\nc \EXPratio \nc \EXPterm \nc \EXPwlm % well-form

% FILE: =====
\def \FILEcheckeof #1{% end of file check
\ifeof #1\relax \FILEnotendedfalse \else \FILEnotendedtrue \fi}

```

```

\new \ifFILEexist
\new \ifFILEhtml

\def \FILEget {\URLinternetfalse
\expandafter \fileA \FILEsource \relax
\expandafter \urlB \FILEsource ://\relax \fileC
\ifURLinternet \let \fileB \relax \fi \fileB}
\def \fileA #1#2\relax {\def \fileAY {#1}% get filename from file FILEsource
\ifx \fileAY \CONslash
\edef \FILENAMEprobe {\FILEsource /}% Fixme: use leading ServerRoot
\else \edef \FILENAMEprobe {../\FILEsource /}\fi}
\def \fileB {\let \CHNfolderbackup \CHNfoldercurrent % get related
\OPENobjecttrue \let \FILENAMEprobe \empty
\expandafter \CHNcopyrelativepath \FILEsource \to \FILENAMEprobe
\OPENobjectfalse \edef \FILENAME {\CHNfoldercurrent /\FILENAME}%
\edef \FILENAMEprobe {../\FILENAMEprobe /}%
\fileC \edef \FILENAME {\FILEpath/\FILENAME}%
\glet \CHNfoldercurrent \CHNfolderbackup}
\def \fileC {\let \FILEsourcelast \empty % get source
\let \FILENAME \empty \let \FILEpath \empty
\expandafter \fileD \FILENAMEprobe \relax
\expandafter \fileCA \FILENAME .\relax
\ldef \fileCA #1.#2\relax {% jpg if extension unsupported
\lowercase {\def \FILEimageextensionsecond {#2}}%
\ifx \FILEimageextensionsecond \CONjpgdot \else
\ifx \FILEimageextensionsecond \CONjpgdot \else
\ifx \FILEimageextensionsecond \CONpdfdot \else
\ifx \FILEimageextensionsecond \CONpngdot \else
\edef \FILENAME {#1.jpg}\fi \fi \fi} % get image source
\def \fileD #1/#2{% parse source to extract leading path
\edef \fileDY {#1}\edef \fileDZ {#2}%
\ifx \fileDZ \CONrelax \fileDB
\else \fileDA \fi \FILEsourcext} % Fixme: used globally
\def \fileDA {\let \FILEsourcext \fileD % no relax
\ifx \FILENAME \empty \let \FILEpath \fileDZ \else \fileDAA \fi
\let \FILENAME \fileDZ}
\def \fileDAA {\ifx \fileDY \empty \else % not empty
\edef \FILEpath {\FILEpath /\FILEsourcelast \fileDY}\fi
\edef \FILEsourcelast {\fileDZ}}
\def \fileDB {\let \FILEsourcext \relax
\edef \FILENAME {\FILENAME \fileDY}} % relax

\let \FILENAME \empty

\new \ifFILEnotended

\new \ifFILEplain \newread \FILEplaintext

\def \FILEsearch #1{% file existence check
\openin \FILEsearched = #1\relax
\ifeof \FILEsearched \FILEexistfalse \else \FILEexisttrue \fi
\closein \FILEsearched}
\newread \FILEsearched
\def \FILEsearchimage #1.jpg#2\relax {\edef \FILEsearchimagetest {#2}%
\ifx \FILEsearchimagetest \empty \else \xdef \FILENAME {#1.png}%
\FILEsearch \FILENAME \ifFILEexist \else \CHKerrabsent \FILENAME \fi \fi}

\new \ifFILETeX

\let \FILEthis \empty

\new \ifFILEtoomany

% FLOAT: =====
% new 2007/07/07; compare Knuth, pages 103, 315, 316:
\def \FLTget #1{\FLTgetA #1\FLTgetnext}
\def \FLTgetA {\ifnum \prevgraf < -\hangafter % previous
\edef \FLTgetnext {\prevgraf = \the \prevgraf}%
\else \let \FLTgetnext \empty
\global \hangafter = 0 \global \hangindent = 0pt \fi}

\def \FLTingbeginother {\hbox to 0mm
\group \hskip -\TEXTindent \hskip -\hangindent
\ vbox to 0mm \group \vskip -1ex}
\def \FLTingbeginright {\global \hangindent = -\hangindent
\hbox to 0mm \group \hskip -\TEXTindent \hskip \hangindent
\hskip \hsize \hskip \LINEheight \hskip -1ex
\ vbox to 0mm \group \vskip -1ex}
\def \FLTingend {\vss \egroup \hss \egroup}

\def \FLTpagebegin #1{% #1: dimen register % new 2007/03/01
\FLTpagefreespace = \pagegoal % Fixme: use for tables too
\advance \FLTpagefreespace by -\pagetotal
\ifnum \widowpenalty = 10000 \advance \FLTpagefreespace by -\LINEheight \fi
\xdef \FLTpageheight {\the #1}%
\ifdim \FLTpagefreespace < \FLTpageheight
\ifdim \FLTpagefreespace > 0pt \OBEYbreakpage \parindent = 0pt \noindent
\global \FLTpageextrabreaktrue \fi \fi}
\def \FLTpageend {\ifFLTpageextrabreak \ifdim \pagetotal > \FLTpageheight
\global \hangafter = 1 \global \hangindent = 0pt
\global \FLTpageextrabreakfalse
\else \FLTpagefreespace = \FLTpageheight
\advance \FLTpagefreespace by -\pagetotal
\FLTsizeheight \FLTpagefreespace \fi \fi}
\new \ifFLTpageextrabreak
\nd \FLTpagefreespace
\let \FLTpageheight \LINEheight

\def \FLTsizeheight #1{\global \hangafter = -#1%
\ifdim \LINEheight > 0mm \global \divide \hangafter by \LINEheight
\gadvance \hangafter by -1 \else \CHKerrnolineheight \fi}
\def \FLTsizewidth #1{\global \hangindent = #1%
\gadvance \hangindent by -1ex \gadvance \hangindent by \LINEheight
\ifdim \hangindent > \hsize \global \hangindent = \hsize \fi}

\def \FLTtable {\ifOPENp \ifnum \TABLEnesting = 1
\gadvance \TABLEheight by \TABLEswapheight \FLTsizeheight \TABLEheight \fi
\fi
\let \TABLEnext \relax \ifOPENp \ifnum \TABLEnesting = 1
\let \TABLEnext \FLTingend \fi \fi \TABLEnext}
\def \FLTtablealign {\ifOPENp \let \TABLEalignnext \fltA
\else \let \TABLEalignnext \fltB \fi
\TABLEalignnext \hsize = \BOXwidth}
\def \fltA {\let \TABLEalignnext \relax % do float
\ifnum \TABLEnesting = 1 \FLTsizewidth \BOXwidth
\ifx \TABLEalign \CONright \let \TABLEalignnext \FLTingbeginright
\else \let \TABLEalignnext \FLTingbeginother \fi \fi
\BOXindent = 0mm \TABLEalignnext}
\def \fltB {\ALGNleftfalse \ALGNcenterfalse \ALGNrightfalse
\ifx \TABLEalign \CONright \advance \BOXindent by -\BOXwidth \else
\ifx \TABLEalign \CONcenter \advance \BOXindent by -\BOXwidth
\divide \BOXindent by 2 \else \BOXindent = 0mm \fi \fi
\advance \BOXindent by -\BRDw} % do not float

% FONT: =====
\def \FONTacrobatscheck {\FONTfam = \expandafter {\FONTfamgettest}%
\ifx \FONTfamgettest \CONarial
\FONTacrobatsset \CONHelvetica \CONHELVETICA
\else \ifx \FONTfamgettest \CONCourierNew
\FONTacrobatsset \CONCourier \CONCOURIER
\else \ifx \FONTfamgettest \CONCourierNewSpace %%
\FONTacrobatsset \CONCourier \CONCOURIER %%
\else \ifx \FONTfamgettest \CONTimesNewRoman
\FONTacrobatsset \CONTimes \CONTIMES % Fixme: check space too
\fi \fi \fi \fi}

```

```

\def \FONTacrobatsset #1#2{\FONTfam = \expandafter {#1}%
\edef \FONTfamuc {#2}\FONTfammatchdo \FONTchooseB}

\net \FONTbs % base
\net \FONTbsit % base italic

\new \ifFONTbd
\net \FONTbd % bold
\net \FONTbdit % bold italic

\def \FONTchangenname {\FONTmodechange \CLRizeforeground}

\new \ifFONTchoose % reset
\def \FONTchoose {\FONTchoosetrue % Fixme: backup old font if reset necessary
\FONTfamget \FONTchooseB
\ifFONTdefault \FONTchooseA \else \FONTchooseC \fi \FONTname = \FONTbs}
\def \FONTchooseA {\CPGplaintrue % default
\def \FONTfamuc {CMR}\FONTfam = {cmr}%
\FONTbs = {cmr10}\FONTbsit = {cmti10}%
\FONTbd = {cmbx10}\FONTbdit = {cmbx10}}
\def \FONTchooseB {% define font-faces and fill-up if some are absent:
\FONTchooseBE 00\FONTchooseBA \FONTchooseBE 01\FONTchooseBB
\FONTchooseBE 10\FONTchooseBC \FONTchooseBE 11\FONTchooseBD}
\def \FONTchooseBA {\FONTbs = {\FONTchooseBF 00}\FONTchoosfalse
\FONTbsit = \FONTbs \FONTbd = \FONTbs \FONTbdit = \FONTbs} % base
\def \FONTchooseBB {\FONTbsit = {\FONTchooseBF 01}\FONTchoosfalse
\FONTbdit = \FONTbsit \FONTbd = \FONTbsit} % base italic
\def \FONTchooseBC {\FONTbd = {\FONTchooseBF 10}\FONTchoosfalse} % bold
\def \FONTchooseBD {\FONTbdit = {\FONTchooseBF 11}\FONTchoosfalse} % bd it
\def \FONTchooseBE #1#2#3{%
\DEFsearch {\CONfontthyphen \CPG -\FONTfamuc -#1-#2}\ifDEF #3\fi}
\def \FONTchooseBF #1#2{%
\csname \CONfontthyphen \CPG -\FONTfamuc -#1-#2\endcsname}
\def \FONTchooseC {\ifFONTchoose \ifCPGplain \else
\CHKerrfontreset \FONTfamuc \FONTchooseA \fi \fi} % reset

% Fixme: Use integer font size (in pt) for CM fonts (DVI output compatibility)

\def \FONTcmtial {\font \FONT = cmti10 at \SIZEcurrent \FONT}
\def \FONTcmmath {\font \FONT = cmmi10 at \SIZEcurrent \FONT}
\def \FONTcmsmall {\ifFONTitalic \FONTcmsmalla \else \FONTcmsmallb \fi \FONT}
\def \FONTcmsmalla {\ifONTbd \font \FONT = cmbx10 at .667\SIZEcurrent
\else \font \FONT = cmti10 at .667\SIZEcurrent \fi}
\def \FONTcmsmallb {\ifONTbd \font \FONT = cmbx10 at .667\SIZEcurrent
\else \font \FONT = cmr10 at .667\SIZEcurrent \fi}
\def \FONTcmsmalltype {\font \FONT = cmtt10 at 0.667\SIZEcurrent \FONT}
\def \FONTcmsymb {\font \FONT = cmsy10 at \SIZEcurrent \FONT}
\def \FONTcmttext {% Fixme: do font-size-adjust?
\ifFONTitalic
\ifONTbd \font \FONT = cmbx10 at \SIZEcurrent \else % not italic
\font \FONT = cmti10 at \SIZEcurrent \fi
\else \ifONTbd \font \FONT = cmbx10 at \SIZEcurrent \else
\font \FONT = cmr10 at \SIZEcurrent \fi \fi \FONT}
\def \FONTcmttype {\font \FONT = cmtt10 at \SIZEcurrent \FONT}

\new \ifFONTdecoration

\new \ifFONTdefault

\def \FONTdo {\ifdim \SIZE > 0mm \SIZEcurrent = \SIZE
\ifFONTsizeadjust \FONTsizeadjust \fi
\SIZEmagnify \FONTnametestget
\ifx \FONTnametest \empty \FONTrescue \fi
\font \FONT = \FONTnametest \CONat \SIZEcurrent \FONT}
\ifnum \KRn > 0 \ifKRndone \else \ifALGNcenter \else
\KRndo \fi \fi \fi
\ifALGNcenter \ALGNtextrespace \else
\ifALGNleft \ALGNtextrespace \else
\ifALGNright \ALGNtextrespace \fi \fi \fi
\else \CHKerrinvalidfontsize \fi}

\def \FONTdoinner {\ifOPENtd \OPENfontinnertrue \else
\ifOPENth \OPENfontinnertrue \else \ifOPENp \OPENfontinnertrue \else
\OPENfontinnerfalse \fi \fi \fi}

\def \FONTdomono {\FONTfam = \FONTmono \FONTchoose \FONTdo}

\def \FONTfbk {\FONTfbkserif} % fallback
\edef \FONTfbkmonospace {\the \FONTmonospace}
\edef \FONTfbksansserif {\the \FONTSansserif}
\edef \FONTfbkserif {\the \FONTserif}
\def \FONTfbkset {\expandafter \FONTfbksetA \FONTfbkmonospace \relax
\expandafter \FONTfbksetB \FONTfbksansserif \relax
\expandafter \FONTfbksetC \FONTfbkserif \relax}
\def \FONTfbksetA #1\relax {\uppercase {\xdef \FONTfbkmonospaceuc {#1}}}
\def \FONTfbksetB #1\relax {\uppercase {\xdef \FONTfbksansserifuc {#1}}}
\def \FONTfbksetC #1\relax {\uppercase {\xdef \FONTfbkserifuc {#1}}}
\def \FONTfbkuc {\FONTfbkserifuc}

\net \FONTfam
\net \FONTfambackup
\def \FONTfamget {\let \FONTfbk \FONTfbkserif
\let \FONTfbkuc \FONTfbkserifuc \let \FONTfammatchcuc \empty
\edef \FONTfamgetT {\the \FONTfam}%
\ifx \FONTfamgetT \empty \else \expandafter \FONTfamgetB \the \FONTfam,,\fi
\ifFONTdefault \let \FONTfammatchcuc \CONCMR \fi
\ifx \FONTfammatchcuc \empty \let \FONTfamgetA \fi
\let \FONTfamuc \FONTfammatchcuc
\ifx \FONTfamuc \empty \let \FONTfamuc \CONCMR \fi}
\def \FONTfamgetA {\FONTchoosetrue \FONTdefaultfalse
\edef \FONTfamuc {\the \FONTfam}%
\ifCPGplain \else \FONTfamgetAA \fi}
\def \FONTfamgetAA {\edef \FONTfamlist {\the \FONTfam}%
\edef \FONTfamgetBprobe {\the \FONTfam \CONSPACE}%
\expandafter \FONTfamgetAAB \FONTfamgetBprobe \relax
\ifx \FONTfamgetBprobe \FONTfamgettest
\expandafter \FONTfamgetAAA \FONTfamgetBprobe ""\relax
\ifx \FONTfamgetBprobe \FONTfamgettest
\expandthree
\FONTnoneforcodepage \expandafter {\FONTfamlist}%
\else \FONTacrobatscheck \fi
\else \FONTacrobatscheck \fi} %%%
\def \FONTfamgetAAA #1"#2"#3\relax {% check quotes
\def \FONTfamgetAAAY {#2}%
\ifx \FONTfamgetAAAY \empty \else \edef \FONTfamgettest {#2}\fi}
\def \FONTfamgetAAB #1 #2\relax {% check spaces
\def \FONTfamgetAABZ {#2}\edef \FONTfamgettest {#1#2}%
\ifx \FONTfamgetAABZ \empty \else \ifx \FONTfamgetAABZ \CONSPACE
\edef \FONTfamgettest {#1}\fi \fi}
\def \FONTfamgetB #1#2, {% parse uppercase font family name from list
\edef \FONTfamgettest {#2}\uppercase {\edef \FONTfamgetBYZuc {#1#2}}%
\ifx \FONTfamgetBYZuc \CONcomma \let \FONTfamgetBE \relax
\else \ifx \FONTfammatchcuc \empty
\let \FONTfamuc \FONTfamgetBYZuc \let \FONTfamgetBE \FONTfamgetB
\ifx \FONTfamuc \CONmonospace \FONTfamgetBB \FONTfam = \FONTmonospace \fi
\ifx \FONTfamuc \CONsansserif \FONTfamgetBC \FONTfam = \FONTSansserif \fi
\ifx \FONTfamuc \CONserif \FONTfamgetBD \FONTfam = \FONTserif \fi
\ifx \FONTfamuc \CONCMR \FONTfamgetBA \FONTfam = {cmr}\fi
\ifx \FONTfammatchcuc \empty \FONTfammatchdo \fi
\ifx \FONTfamgettest \empty \let \FONTfamgetBE \relax \fi \fi \fi
\FONTfamgetBE}
\def \FONTfamgetBA {\let \FONTfbk \CONcmr \let \FONTfbkuc \CONCMR
\ifx \FONTfammatchcuc \empty \FONTdefaulttrue \fi} % reset cmr
\def \FONTfamgetBB {\let \FONTfbk \FONTfbkmonospace
\let \FONTfbkuc \FONTfbkmonospaceuc} % reset monospace
\def \FONTfamgetBC {\let \FONTfbk \FONTfbksansserif
\let \FONTfbkuc \FONTfbksansserifuc} % reset sans serif

```

```

\def \FONTfamgetBD {\let \FONTfbk \FONTfbkserif
\let \FONTfbkuc \FONTfbkserifuc} % reset serif
\new \ifFONTfamgiven
\let \FONTfamlist \empty
\let \FONTfammatchuc \empty
\def \FONTfammatchdo {%
\DEFsearch {\CONfontthyphen \CPG -\FONTfamuc -0-0}\ifDEF
\let \FONTfammatchuc \FONTfamuc \fi}
\def \FONTfamset {\FONTfam = \VAL % Fixme: \FONTsizecheck
\FONTfamgiventruer \FONTchoose \FONTdo}
\let \FONTfamuc \empty

\new \ifFONTitalic

\net \FONTname
\let \FONTnametest \empty
\def \FONTnametestget {\uppercase {\edef \FONTnametest {\the \FONTname}}}

\def \FONTnoneforcodepage #1{\def \FONTnoneforcodepagefirst {#1}%
\ifx \FONTnoneforcodepagefirst \empty else
\CHKerrnofontforCP {#1}% Fixme: move to error.tex, pre-expand
\ifx \CPG \GMScodepage else
\message {! Using font for default \CONcodepage.}\fi \fi
\let \CPG \GMScodepage \CPGgetdefaultname
\let \FONTfbk \FONTfbkserif \let \FONTfbkuc \FONTfbkserifuc
\let \FONTfammatchuc \FONTfbkuc}

\def \FONTmodechange {%
\ifFONTbd \FONTmodechangebold else \FONTmodechangebase \fi
\KRNdonefalse %% Check this: time bandit?
\FONTdo}
\def \FONTmodechangebold {%
\ifFONTitalic \FONTname = \FONTbdit else \FONTname = \FONTbd \fi}
\def \FONTmodechangebase {%
\ifFONTitalic \FONTname = \FONTbsit else \FONTname = \FONTbs \fi}
\def \FONTmodestyle {\VALsetlc \FONTmodestyletest
\ifx \FONTmodestyletest \CONnormal \FONTitalicfalse else
\ifx \FONTmodestyletest \CONitalic \FONTitalictrue else
\ifx \FONTmodestyletest \CONoblique \FONTitalictrue else
\CHKerrfontstyle \fi \fi \fi \FONTmodechange}
\def \FONTmodeweight {\VALsetlc \FONTmodeweighttest
\ifx \FONTmodeweighttest \CONnormal \FONTbdfalse else
\ifx \FONTmodeweighttest \CONbold \FONTbdtrue else
\CHKerrfontweight \fi \fi \FONTmodechange}

\net \FONTmono

\net \FONTpageno

\def \FONTrescue {%
\FONTcmtext \FONTname = {cmr10}\FONTnametestget \CHKerrfontfalse}

% Unless font-size-adjust is set to "none", the font shall be scaled such that
% its x-height becomes half the given font-size value:
% fontxheightadjusted := 0.5 fontsize
%
% fontxheightadjusted = 0.5 fontsize
%
% fontxheightadjusted = fontsize * ----- = -----
% fontxheight 2 fontxheight
% Dimensions are represented by integer numbers (Knuth page 118). In order to
% avoid register overflow, they must be divided by e. g. 1000 before they can
% be multiplied.
% 72
% 1mm = .... pt = 2.8346 * 65536sp = 185771sp
% 25.4
% This should be done if font-family, font-size or line-height have been
% changed.

\new \ifFONTsizeadjust
\def \FONTsizeadjust {\FONTnametestget

\ifx \FONTnametest \empty \FONTrescue \fi
\font \FONT = \FONTnametest \CONat \SIZE \FONT
\FONTxheighttempo = \fontdimen 5 \FONT
\divide \FONTxheighttempo by \CONnummedium
\SIZEtempo = \SIZE \divide \SIZEtempo by \CONnummedium % conversion to sp
\SIZEtempotwo = \SIZEtempo \multiply \SIZEtempo by \SIZEtempotwo
\divide \SIZEtempo by \FONTxheighttempo
\multiply \SIZEtempo by 500 \SIZEcurrent = \SIZEtempo sp }
\def \FONTsizeadjustset {\VALgetlc
\ifx \VALlowercase \CONnone \FONTsizeadjustfalse \fi}

\def \SIZEsmaller {\SIZEbasesmaller \CLRizeforeground \FONTdo}

\def \FONTswitchspan {%
\KRNdonefalse \CLRizeforeground \FONTchoose \FONTmodechange \FONTdo}
\def \FONTswitchsubbegin {%
\hbox \bgroup \lower 0.5ex \hbox \bgroup \SIZEsmaller}
\def \FONTswitchsubend {\egroup \egroup}
\def \FONTswitchsubbegin {%
\hbox \bgroup \raise 0.5ex \hbox \bgroup \SIZEsmaller}
\let \FONTswitchsuspend \FONTswitchsubend

\def \FONTunderline #1{%
\ifFONTdecoration \hbox {\raise 0ex \tlap {
\vskip 2.5ex \hrule height 0.1ex \vss \hbox {#1}}}\else #1\fi}

\nec \FONTxheighttempo

% GAP: =====
\net \GAP

\ldef \GAPchk #1{#1}% check for discardable leading whitespace
\def \GAPchkfilled {%
\ifx \GAPcontent \empty \global \GAPemptytrue \global \GAPfilledfalse
\else \global \GAPemptyfalse \global \GAPfilledtrue \fi
\ifx \GAPcontent \CONspace \global \GAPfilledfalse \fi}
\def \GAPchklog {\ifGAPlog \message {\GAPcontent}\fi}
\def \GAPchkobey {\ifGAPobey \OBEYnormalspacesandlines \fi}
\def \GAPchkkitvoid {\let \GAPchk \empty}
\def \GAPchkkitopener {\GAPchk}% Check this

\def \GAPclear {\global \GAP = {}}

\def \GAPdisplay {\VALgetlc
\ifx \VALlowercase \CONnone \GAPclear \GAPwriteomit \LINKunrenderedtrue
\else \GAPwritereset \ifx \VALlowercase \CONblock \ELEMENTblocktrue \fi \fi}
\def \GAPdisplayy {\VALgetlc \GAPclear
\ifx \VALlowercase \CONnone \GAPwriteomit \LINEupskip
\ifnum \NOpair = 1 \NOpairresetglobal \fi \else \GAPwritereset \fi}

\new \ifGAPempty % GAP is "empty", if it does not even contain one space
\new \ifGAPfilled % GAP is "not filled", if it only contains one space

\bgroup \CATactivelessthan
\gldef \GAPget #1<\xdef \GAPcontent {#1}\GAPchkfilled \ifGAPfilled
\ifx \GAPcontent \CONendcomment \tagAfalse \tagAbegfalse \fi \fi
\tagD \GAPcontent \GAPchklog \GAPchkobey \GAPset {#1}%
\CATotherlessthan \TAGget }\egroup

\new \ifGAPlog
\new \ifGAPnostruts
\new \ifGAPobey
\new \ifGAPparsedstyle

\net \GAPsgm % segment of gap
\ldef \GAPsgmapp #1{\global \GAPsgm = \expandafter {\the \GAPsgm #1}} % append

\ldef \GAPset #1{\global \GAP = {#1}}

```



```

\ifx \IMGalt \CONnone \CHKerrempty \CONalt \let \IMGalt \empty \fi
\ifx \URLhref \empty \else \let \FILEsource \URLhref \fi %%
\ifx \FILEsource \empty \CHKerrempty \CONsrc
  \ifx \IMGalt \empty \else \if \IMGalt \relax \else \{ \IMGalt \} \fi \fi
\else \FILEget \FILEsearch \FILEname
  \if \FILEexist \imgA \else \expandafter
    \FILEsearchimage \FILEname .jpg \relax
  \if \FILEexist \imgA \else \CHKerrabsent \FILEname
    \global \IMGabsenttrue \ifx \IMGalt \empty \else
      \if \IMGalt \relax \else \{ \IMGalt \} \fi \fi \fi \fi
\ifOPENTd \ifdim \IMGheight > \BOXheightswapped
  \global \BOXheightswapped = \IMGheight \fi \fi \NOParresetglobal \fi
\def \imgA {\IMGscale \ifx \IMGalign \CONbottom \let \IMGalign \empty \else
  \ifx \IMGalign \CONmiddle \let \IMGalign \empty
  \CHKerrrunsupattprop {\CONalign \CONspace = "\CONmiddle"}%
  \else \ifx \IMGalign \CONtop \let \IMGalign \empty
  \CHKerrrunsupattprop {\CONalign \CONspace = "\CONtop"} \fi \fi \fi
\ifx \IMGalign \empty
  \ifnum \NOPar > 1 \ifOPENa \else \ifhmode \par \LINEupskip \fi
  \vskip 1ex \fi \fi \hskip -\parindent \ifOPENTd \imgAA \else \imgC \fi
\else \ifOPENTd \imgAA \else % Fixme: doesn't float
  \ifOPENp \imgB \else \ifOPENh \imgC \else
    \ifOPENli \imgB \else \imgAB \fi \fi \fi \fi \fi
\def \imgAA {\hskip -\TEXTindent \hbox {\vbox {%
  \BOXfillertop \imgC \BOXfillerbottom
  \vskip -0.2917\LINEheight }}}% factor see \GAPsetstrut
\def \imgAB {\ifdim \BOXheight > \IMGheight % center
  \hbox to \BOXwidth {\hss \vbox to \BOXheight {\vss \imgC \vss }} \hss }%
  \else \hbox to \BOXwidth {\hss \vbox {\imgC } \hss } \fi
\def \imgB {\advance \IMGheight by \IMGmarginb % float
\advance \IMGheight by \IMGmarginl \FLTPagebegin \IMGheight
\FLTsizewidth \IMGwidth \FLTsizeheight \IMGheight
\advance \IMGheight by -\IMGmarginb \advance \IMGheight by -\IMGmarginl
\ifx \IMGalign \CONright \FLTPingbeginright \else
  \ifx \IMGalign \CONleft \FLTPingbeginother \else
    \FLTPingbeginother \fi \fi \imgC \FLTPingend}
\def \imgC {\ifnum \IMGnum > 0 \ifvmode \imgCA \else
  \vbox {\imgCA } \fi \else \CHKerrunloadable \fi }% put
\def \imgCA {\vskip \IMGmarginl \pdfrefximage \IMGnum \vskip \IMGmarginb}
\ifx \pdfximage \undefined \let \imgCA \relax \fi

\end \IMGheight

\def \IMGinit {\let \IMGalign \empty \let \IMGalt \CONnone
\let \FILEname \empty \let \FILEsource \empty
\let \IMGvalign \empty \IMGheight = 0mm \IMGwidth = 0mm
\IMGmarginb = 0mm \IMGmarginl = 0mm }

\nec \IMGLines

\end \IMGmarginb \end \IMGmarginlh \end \IMGmarginr \end \IMGmarginl

\nec \IMGnum

\nec \IMGresolution

\def \IMGscale {\ifdim \IMGwidth = 0mm \ifdim \IMGheight = 0mm \imgD \imgG
  \else \imgH \IMGheight \empty \fi % don't scale if just height given
  \else \imgD \imgF \fi
\def \imgD #1{\imgK \empty #1 \imgH \IMGheight \IMGwidth}
\def \imgE {\setbox \IMGbox = \hbox {\pdfrefximage \pdflastximage}%
  \IMGwidth = \wd \IMGbox \IMGheight = \ht \IMGbox }% exec
  \ifx \pdfximage \undefined \let \imgE \relax \fi
\def \imgF {% get width
  \ifdim \IMGheight = 0mm \pdfximage width \IMGwidth {\FILEname}%
  \else \pdfximage height \IMGheight width \IMGwidth {\FILEname} \fi
  \imgE \imgFA}
\ifx \pdfximage \undefined \let \imgF \relax \fi
% Make \IMGheight a multiple of \LINEheight (+1ex):
\def \imgFA {\ifOPENTd \else \ifOPENTh \else
  \ifdim \LINEheight > 0mm \imgFAA \fi \fi
\def \imgFAA {\IMGLines = -1 % fit
  \loop \ifdim \IMGheight > 0mm
    \advance \IMGLines by 1 \advance \IMGheight by -\LINEheight
  \repeat \IMGheight = \LINEheight
  \multiply \IMGheight by \IMGLines \advance \IMGheight by 1ex }
\def \imgG {\imgGA \pdfximage {\FILEname} \imgE }% get zero
  \ifx \pdfximage \undefined \let \imgG \relax \fi
\def \imgGA {\imgGAA }% default resolution (72dpi)
  \def \imgGAA {\pdfimageresolution = \IMGresolution}
  \ifx \pdfximage \undefined \let \imgGAA \relax \fi
\def \imgH #1#2{\edef \imgHY {#1} \edef \imgHZ {#2}% width, height
  \imgK \imgHA \imgJ }% put
\def \imgHA {\IMGnum \expandafter=\csname \CONIMG \FILEname \endcsname
  \edef \imgHAA {\the \IMGnum }}% new - dummy expand required
\def \imgJ {\ifx \imgHY \empty \imgJB \else \imgJA \fi
  % old - Fixme: Must images used multiple times have the same size?
  \xdef {\CONIMG \FILEname} {\the \pdflastximage} \IMGnum = \pdflastximage}
\ifx \pdfximage \undefined \let \imgJ \relax \fi
\def \imgJA {\ifx \imgHZ \empty
  \pdfximage height \imgHY {\FILEname }% height given
  \else \pdfximage height \imgHY width \imgHZ {\FILEname} \fi
\def \imgJB {\ifx \imgHZ \empty \pdfximage {\FILEname}%
  \else \pdfximage width \imgHZ {\FILEname} \fi }% no height given
\def \imgK #1#2{\DEFsearch {\CONIMG \FILEname } \ifDEF #1 \else #2 \fi}

\ned \IMGwidth

% KERNING: =====
% Fixme: Whole paragraph in italics should be kerned

\nec \KRN \new \ifKRN

\def \KRNcheck {\pdfprotrudechars = \KRN}
  \ifx \pdfprotrudechars \undefined \let \KRNcheck \relax \fi
\def \KRNcheckback {\pdfprotrudechars = 0 }
  \ifx \pdfprotrudechars \undefined \let \KRNcheckback \relax \fi

\nec \KRNcode

\def \KRNdo {% load the kerning table that corresponds to the current font
\def \{ \{ #1 \} #2 \} #3 \} {%
  \lcode \FONT "#1 = #2 \rcode \FONT "#1 = #3 }%
\ifFONTitalic \let \KRNdovariant \CONitalic
  \else \let \KRNdovariant \CONroman \fi
\DEFsearch {\CONkrn \CPG -\FONTfamuc -\KRNdovariant} \ifDEF
  \csname \CONkrn \CPG -\FONTfamuc -\KRNdovariant \endcsname
  \KRNdonottrue \fi
\new \ifKRNdone

\def \KRNget {% read kerning tables that can be found on the list
\def \{ \{ #1 \} #2 \} #3 \} {\def \KRNgetZ {#1}% \uppercase {\edef \First {#1}}%
  \ifx \KRNgetZ \empty \else
    \FONTfam = {#1}% can be corrected by font-family entry in *.krn file
    \krnA \fi } \KRNtables}
\def \krnA {% read kerning table that corresponds to current font family
  \ifx \lcode \undefined \else \let \krnAprobe \empty
  \message {\the \FONTfam.krn}%
  \openin 2 = \the \FONTfam.krn \loop \krnAA \repeat \closein 2 \fi}
\def \krnAA {\FILEcheckeof 2 \ifFILEnotended \read 2 to \krnAline
  \expandafter \krnB \krnAline ;;; \relax}
\ldef \krnB #1;#2;#3;#4;#5;#6 \relax {% read a line from kerning table
  % #1: ord. number (hex.), #2: rom. lpcode (or font-family or codepage),
  % #3: rom. rpcode, #4: itl. lpcode, #5: itl. rpcode, #6: efcode
  \def \krnBU {#1} \uppercase {\def \krnBV {#2}}%
  \ifx \krnBU \CONpar \else \ifx \krnBU \empty \else
    \ifx \krnBU \CONfontfamily \krnBA \else \krnBB {#1} {#2} {#3} \CONroman

```

```

\kernBB {#1}{#4}{#5}\CONitalic \fi \fi \fi}
\def \kernBA {% correct font family taken from kerning file name
\FONTfam = \expandafter {\kernBV } \message {- \the \FONTfam .}%
\uppercase {\xDef \CONkern \CPG -\the \FONTfam - {%
\noecho \message {\CONkerning }}} % define kerning table marker
\def \kernBB #1#2#3#4{% #1: ord.number, #2: lpcode, #3: rpcode, #4: variant
% add kerning data of roman/italic font variant to kerning factors array
\KRNcode = #2 \advance \KRNcode by #3
\ifnum \KRNcode > 0 \ARRAYthree [#1] [#2] [#3] \ARRAYafter \kernAprobe
\uppercase {\xDef \CONkern \CPG -\the \FONTfam -#4 {%
\the \ARRAY \the \ARRAYentry }}\fi}

\bgroup \CATothertilde
\gdef \KRNparseLine #1(#2.#3)#4\relax {% add data to kerning table list
\def \KRNPARSElineX #2)\ifx \KRNPARSElineX \empty \else
\lowercase {\edef \KRNPARSElineitem {[#2] }}\expandafter
\ARRAYone \KRNPARSElineitem \ARRAYafter \KRNTables \fi }\egroup

\def \KRNshow {% show kerning table marker defined for current font family
\csname \CONkern \CPG -\FONTfamuc -\endcsname}

\let \KRNTables \empty

% LANGUAGE: =====

\def \LNGabbr {en-US} % abbreviation

% Add pattern files, delimited by whitespace, to HYPHtables list:
\def \LNGadd #1.#2 {\def \LNGaddy {#1}%
\ifx \LNGaddy \empty \else \LNGaddparse #1--\relax \advance \language by 1
\lowercase {\edef \LNGadditem {%
\the \language} % 1. language number
[\LNGName] % 2. language name
[\LNGExt] }} % 3. language extension
\expandafter \ARRAYthree \LNGadditem \ARRAYafter \HYPHENTables \fi}
% Get language name abbreviation(s) from hyphenation pattern file name:
\ldef \LNGaddparse #1-#2-#3\relax {\def \LNGName {#1}\def \LNGExt {#2}}

\def \LNGchangedetect #1{\xdef \LNGold {\LNGabbr}#1\xdef \LNGnew {\LNGabbr}}

\def \LNGde {\ifx \LNGabbr \CONde \LNGdeA \fi} % German
\def \LNGdeA {% required for traditional German hyphenation:
\ifx \GAPcontent \CONck \LNGdeAA \empty {k}k{ck}\fi
\ifx \GAPcontent \CONff \LNGdeAA {ff}-f\empty \fi
\ifx \GAPcontent \CONlll \LNGdeAA {ll}-l\empty \fi
\ifx \GAPcontent \CONmm \LNGdeAA {mm}-m\empty \fi
\ifx \GAPcontent \CONnn \LNGdeAA {nn}-n\empty \fi
\ifx \GAPcontent \CONpp \LNGdeAA {pp}-p\empty \fi
\ifx \GAPcontent \CONrr \LNGdeAA {rr}-r\empty \fi
\ifx \GAPcontent \CONtt \LNGdeAA {tt}-t\empty \fi}
\def \LNGdeAA #1#2#3#4{% special, see Knuth, pages 95, 96, 314
\global \GAP = {#1}discretionary {#2}{#3}{#4}}

\def \LNGloadpatterns {{\language = -1
\def \ \ [#1] [#2] [#3] {% language number, name, and extension
\def \LNGloadpatternsthird {#3}\language = #1%
\ifx \LNGloadpatternsthird \empty \input #2.tex
\else \input #2-#3.tex \fi }\HYPHENTables}}

\def \LNGset {\expandafter \LNGsetA \LNGabbr \relax %%
\def \ \ [#1] [#2] [#3] {% Fixme: replace list with direct call
\def \LNGsetsecond {#2}\def \LNGsetsecondthird {#2-#3}%
\ifx \LNGlc \LNGsetsecondthird \language = #1 \else
\ifx \LNGlc \LNGsetsecond \language = #1 \fi \fi }\HYPHENTables}
\def \LNGsetA #1\relax {\lowercase {\edef \LNGlc {#1}}}}

% LENGTH: =====

% Length of string (compare Knuth, page 219):
\kernBB {#1}{#4}{#5}\CONitalic \fi \fi \fi}
\nec \LENGTH

\def \LENGTHget #1{% count nonblank tokens in string #1
\LENGTH = 0 \expandafter \LENGTHgetA #1\end}
\def \LENGTHgetA #1{%
\ifx \end #1\let \LENGTHnext \relax \else \advance \LENGTH by 1
\let \LENGTHnext \LENGTHgetA \fi \LENGTHnext}

% LINK: =====

\def \LINKexec {%
\ifx \LINKrel \CONnext \ifOPENobject \else \glet \CHNhref \URLhref \fi
\else \ifx \LINKrel \CONstylesheet
\ifx \LINKtype \CONTEXTcss \SHTstyleclear \SHTload \else
\ifx \LINKtype \empty \CHKernmostylesheettype \SHTstyleclear \SHTload
\fi \fi \fi \fi \SHTnext}

\let \LINKid \empty

\def \LINKinit {\let \CHNhref \empty \glet \URLhref \empty
\let \LINKmedia \empty \let \LINKrel \empty \let \LINKtype \empty
\let \SHTnext \relax}

\let \LINKname \empty

\new \ifLINKunlinked
\new \ifLINKunrendered

% LIST: =====

\def \LISTleftmarginset {\advance \leftskip by 2\LINEheight
\advance \leftskip by -1ex} % default indent is 2\LINEheight -1ex
\def \LISTleftmarginsetdt {\advance \leftskip by \LINEheight
\advance \leftskip by -0.5ex}

\def \LISTmark {\ifLINKunrendered \else \ifOPENul \LISTmarkB
\else \ifOPENol \LISTmarkA \else \CHKerrmissingulol \fi \fi
\Nopar = 0 \CLRizeforeground \fi}
\def \LISTmarkA {\hskip -\LINEheight \hskip 1ex
\llap {\the \NOLI}rlap .\hskip \LINEheight \hskip -1ex} % <ol>
\def \LISTmarkB {\hskip -\LINEheight
{\CLRize \CLRlistdot \vrule height 1ex depth 0mm width 1ex} %
\hskip \LINEheight \hskip -1ex} % <ul>

\def \LISTnodlastsetglobal {\global \Nodlast = \chkWLMtotelements}
\def \LISTnolastsetglobal {\global \NOLIlast = \chkWLMtotelements}
\def \LISTnoadvance {\advance \NOLI by 1}
\def \LISTnostart {\NOLI = 1}

\def \LISTpenalty {\penalty -500}

% MAP: =====

% Create: \Def ALIAS-<PostScript typeface name> {<TeX typeface file name>}
\def \MAPadd #1 (#2){\edef \MAPaddfirst {#1}\edef \MAPaddsecond {#2}%
\uppercase {\edef \MAPaddsecondc {#2}}%
\ifx \MAPaddfirst \empty \CHKerrnofamilydefined {#2}\else
\ifx \MAPaddsecond \empty \else
\Def {\CONalias \MAPaddsecondc} {#1}\fi \fi}
% Create: \Def ALIAS-<PostScript font-family name> {<HTML font-family name>}
\def \MAPaddalias #1 (#2){\Def \CONalias #2 {#1}}
% Create: \Def ALIAS-<Font-specific encoding> {<official encoding>}
\def \MAPaddencoding #1 (#2){\uppercase {\Def \CONencoding #1 {#2}}}}
\def \MAPaddfamily #1 (#2, #3, #4, #5){\MAPadd #1 (#2)\MAPadd #1-Italic (#3)%
\MAPadd #1-Bold (#4)\MAPadd #1-BoldItalic (#5)}
\def \MAPaddfamilylstd #1 (#2){\MAPadd #1 (#2)\MAPadd #1-Italic (#2i)%
\MAPadd #1-Bold (#2bd)\MAPadd #1-BoldItalic (#2bi)}
\def \MAPaddfamilyuni #1 (#2 #3){\MAPadd #1 (#2#3)\MAPadd #1-Italic (#2i#3)%

```

```

\MAPadd #1-Bold (#2bd#3)\MAPadd #1-BoldItalic (#2bi#3)
\def \MAPaddmates #1 (#2, #3){\MAPadd #1 (#2)\MAPadd #1-Bold (#3)}
\def \MAPaddpair #1 (#2, #3){\MAPadd #1 (#2)\MAPadd #1-Italic (#3)}
% Create: \Def SPECIFIC-<HTML font-family name>
\def \MAPaddspecific #1 {\uppercase {\Def {\CONspecific #1} {}}}

% Append new family to PostScript names list:
\def \MAPfamilyappend #1\relax {%
  \ifx \POSTfamilies \empty \ARRAYone [#1] \ARRAYafter \POSTfamilies
  \else \edef \MAPfamilyappendfirst {#1}%
    \let \MAPfamilyappendmatch \empty
    \def \ [#1] {\edef \MAPfamilyappendtest {#1}%
      \ifx \MAPfamilyappendfirst \MAPfamilyappendtest
        \let \MAPfamilyappendmatch \relax \fi} \POSTfamilies
    \ifx \MAPfamilyappendmatch \empty
      \ARRAYone [#1] \ARRAYafter \POSTfamilies \fi \fi
    \uppercase {\xdef \MAPisfamily {#1}}

\uppercase {\def \MAPfamilyparse #1-#2STOP{%
  \DEFsearch {ALIAS-#1}\ifDEF
    \edef \MAPfamilyalias {\csname ALIAS-#1\endcsname}%
    \expandafter \MAPfamilyappend \MAPfamilyalias \relax \fi
  \MAPfamilyappend #1\relax}}

\def \MAPfamilyshow {\def \ [#1] {\immediate \backupwrite 4{#1}%
  \immediate \openout 4 font.lst \POSTfamilies \immediate \closeout 4
  \immediate \backupwrite 18{sort < font.lst > sortfont.lst}%
  \def \MAPfamilyshowtest {\WIN32}%
  \ifx \OS \MAPfamilyshowtest \openin 5 = font.lst
  \else \openin 5 = sortfont.lst \fi
  \loop \FILEcheckeof 5 \ifFILEnotended \read 5 to \MAPfamilyshowline
    \edef \MAPfamilyshowline {\MAPfamilyshowline \CONspace}%
    \expandafter \MAPfamilyshowA \MAPfamilyshowline \repeat \closein 5 }
  \ldef \MAPfamilyshowA #1#2 {\def \MAPfamilyshowAZ {#2}%
  \ifx \MAPfamilyshowAZ \empty \else \message {#1#2.}\fi}

\def \MAPget {\openin 1 = font.map % evaluate font.map:
  \loop \FILEcheckeof 1 \ifFILEnotended \read 1 to \MAPgetline
  \expandafter \MAPgetA \MAPgetline GMS GMS <nix.enc-STOP
  \repeat \closein 1 \edef \MAPisbold {0}\edef \MAPisital {0}}
  \ldef \MAPgetA #1#2 #3 #4<#5.enc#6STOP{% parse to get font face name
  \uppercase {\def \MAPgetAV {#2}}% Font file
  \def \MAPgetAW {#3}% PostScript font name
  \uppercase {\def \MAPgetAY {#5}}% Encoding file base name
  \ifx \MAPgetAW \CONgms \else % Get font style:
    \edef \MAPisfamily {unknown}%
    \edef \MAPisbold {0}\edef \MAPisital {0}%
    \DEFsearch {\CONalias \MAPgetAV }\ifDEF
      \def \MAPgetAW {\csname \CONalias \MAPgetAV \endcsname }\fi
      \expandafter \MAPfamilyparse \MAPgetAW-GMS-STOP%
      \expandafter \MAPgetAA \MAPgetAW Bold-GMS-STOP%
      \expandafter \MAPgetAB \MAPgetAW Demi-GMS-STOP%
      \expandafter \MAPgetAC \MAPgetAW Ital-GMS-STOP%
      \expandafter \MAPgetAD \MAPgetAW Obl-GMS-STOP%
      \expandafter \MAPgetAE \MAPgetAW Black-GMS-STOP%
      % Define fonts:
      % Access fonts with specific encoding via equivalent encoding:
      \uppercase {\def \MAPgetAuencoding {#5}}%
      \DEFsearch {\CONencoding \MAPgetAuencoding}\ifDEF
        \def \MAPgetAY {\csname \CONencoding \MAPgetAuencoding \endcsname}\fi
        \xdef {FONT-\MAPgetAY-\MAPisfamily-\MAPisbold-\MAPisital} {#2}\fi \fi
      % Handle fonts with specific encodings as if they were \CPG-encoded:
      \uppercase {\def \FONTfamuc {#3}}%
      \DEFsearch {\CONspecific \FONTfamuc}\ifDEF
        \xdef {FONT-\CPG-\MAPisfamily-\MAPisbold-\MAPisital} {#2}\fi \fi}
  \def \MAPgetAA #1Bold#2STOP{\def \MAPgetAAZ{#2}%
    \ifx \MAPgetAAZ\CONhyphgmshyph \else \edef \MAPisbold {1}\fi} % bold
  \def \MAPgetAB #1Demi#2STOP{\def \MAPgetABZ{#2}%
    \ifx \MAPgetABZ \CONhyphgmshyph \else \edef \MAPisbold {1}\fi} % demi
  \def \MAPgetAC #1Ital#2STOP{\def \MAPgetACZ {#2}%
    \ifx \MAPgetACZ \CONhyphgmshyph \else \edef \MAPisital {1}\fi} % ital.
  \def \MAPgetAD #1Obl#2STOP{\def \MAPgetADZ {#2}%
    \ifx \MAPgetADZ \CONhyphgmshyph \else \edef \MAPisital {1}\fi} % obl.
  \def \MAPgetAE #1Black#2STOP{\def \MAPgetAEZ {#2}%
    \ifx \MAPgetAEZ \CONhyphgmshyph \else
      \xdef \MAPisfamily {\MAPisfamily BLACK}\fi} % black
      % Prevents e.g. "Arial" fonts from being covered by "Arial-Black".
      % Otherwise this definition is needed to correct Arial-Black (ariblk):
      % \xdef {FONT-\GMScodepage-ARIAL-0-0} {arial}%

% MARGIN: =====
\ned \MARGIN
\ned \MARGINbottom
\ned \MARGINleft
\ned \MARGINpageno % check this
\ned \MARGINright
\ned \MARGIntop

\new \ifMARGIngiven
\new \ifMARGIngivenbottom
\new \ifMARGIngiventop

% Fixme: same for 'background', 'padding', 'border' properties:
\def \MARGINset {\edef \MARGINsettest {\the \VAL \CONspace}%
  \expandafter \MARGINsetA \MARGINsettest - - - -\relax}
\def \MARGINsetA #1 #2 #3 #4 #5\relax {%
  \def \MARGINsetAV {#1}\def \MARGINsetAW {#2}%
  \def \MARGINsetAX {#3}\def \MARGINsetAY {#4}%
  \ifx \MARGINsetAV \CONhyphen \else
    \VAL = {#1}\VALsetdim \MARGIntop \vsize
    \MARGINright = \MARGIntop \MARGINleft = \MARGIntop
    \MARGINbottom = \MARGIntop \MARGIN = \MARGIntop
  \ifx \MARGINsetAW \CONhyphen \else
    \VAL = {#1}\VALsetdim \MARGIntop \vsize
    \VAL = {#2}\VALsetdim \MARGINright \hsize
    \MARGINbottom = \MARGIntop \MARGINleft = \MARGINright
  \ifx \MARGINsetAX \CONhyphen \else
    \ifx \MARGINsetAY \CONhyphen \else
      \VAL = {#1}\VALsetdim \MARGIntop \vsize
      \VAL = {#2}\VALsetdim \MARGINright \hsize
      \VAL = {#3}\VALsetdim \MARGINbottom \vsize
      \VAL = {#4}\VALsetdim \MARGINleft \hsize
    \fi \fi \fi \fi \MARGIngiventru}

% MESSAGE: =====
\def \MSGcodepageerror #1{\message {! Cannot add codepage #1.}}
\def \MSGcodepagelanguage {%
  \ifx \CPGhtml \undefined \else \message {\CPGhtml} \fi \message {\LNGabbr}}

\def \MSGdefaultfont {\message {Using default font \FONTfbkserifthe}}

\def \MSGdot{\message {}}

\ldef \MSGentityname #1#2 {% display sorted list of entity names
  \def \MSGentitynamesecound {#2}%
  \ifx \MSGentitynamesecound \empty \else \message {\CONampersand#1#2}\fi}

\def \MSGfalsified {\message {falsified}}

\def \MSGfontfamily {\message {\FONTfamuc}}

\def \MSGhtml {\ifOPENobject \else \ifnum \Noinput = 1 \GMSwelcome
  \GMSbeginabout \message {\CONlongline \CONline \CONhalfline}\fi \fi}

\def \MSGhyphen {\message {}}

```

```

\def \MSGkerning {\ifKRN \message {Kerning}\fi}

\bgroup \CATparameterat \CATotherhashmark \CATotherampersand
\gdef \MSGnochar % no character with that unicode number
  @1% Standard Unicode Value, decimal
  \HEXinput = @\HEXoutput = {\HEX
  \edef \HEXslot {\the \HEXoutput}%
  \message {! Unknown &#01;/&#0x\HEXslot; character.}}\egroup

\def \MSGnoglyphdata {\message {! No data for U+\GLYPHparselinesecnd.}}

\def \MSGnone {\message {none}}

\def \MSGnoslot #1{% Standard Unicode Value (slot base), decimal
  \xdef \missingslot {\the \UCDslot missing}%
  \DEFsearch \missingslot \ifDEF \else
  \HEXinput = \UCDslot \HEXoutput = {\HEX
  \edef \HEXslot {\the \HEXoutput}%
  \message {! Undefined unicode row \HEXslot.}
  \fi \xDef \the \UCDslot missing {#1}}%

\def \MSGnotexcharacter #1{\message {! No TeX character #1.}}

\def \MSGnotstrict {\message {! Not strict.}}
\def \MSGnotstrictatt #1{\message {! Not strict: #1.}}

\def \MSGobjecthtml {\message {html}}
\def \MSGobjectplain {\message {plain}}
\def \MSGobjectTeX {\message {TeX}}

\def \MSGplausible {\message {plausible}}

\def \MSGproprietary {\message {! Proprietary.}}

\def \MSGtdlanguage {\ifx \LNGold \LNGnew \else \message {\LNGnew }\fi}

\def \MSGunicodedefined #1{\edef \MSGunicodedefinedZ {#1}%
  \DEFsearch \MSGunicodedefinedZ \ifDEF
  \edef \MSGunicodedefinedZZ {\csname #1\endcsname}%
  \message {\MSGunicodedefinedZZ }\fi}

\def \MSGunrecognized {\message {unrecognized}}

\def \MSGunverified {\message {unverified}}

\def \MSGvalue {\VALset \MSGvaluetest
  \ifx \MSGvaluetest \empty \else \message {"\MSGvaluetest"}\fi}

\def \MSGverified {\message {verified}}

% META: =====

\def \METAget {metaA
  \ifx \metahttpequiv \CONcontenttype
  \expandafter \metaB \METAcont \relax \fi}
\ifx \pdffinfo \undefined \let \METAget \relax \fi
\def \metaA {% info
  \ifx \METAname \CONauthor \pdffinfo {/Author (\METAcont) }\else
  \ifx \METAname \CONcreated \expandafter \metaAA \METAcont ---\relax
  \else \ifx \METAname \CONdate \expandafter \metaAB \METAcont ---\relax
  \else \ifx \METAname \CONdescription \pdffinfo {/Subject (\METAcont) }%
  \else \ifx \METAname \CONkeywords
  \pdffinfo {/Keywords (\METAcont) }\fi \fi \fi \fi \fi}
\ifx \pdffinfo \undefined \let \metaAA \relax \fi
\def \metaAA #1-#2-#3-#4\relax {% creation
  \edef \metaAAX {#1}\edef \metaAAY {#2}\edef \metaAAZ {#3}%
  \ifx \metaAAX \empty \CHKerrmisformeddate \else
  \ifx \metaAAY \empty \CHKerrmisformeddate \else
  \ifx \metaAAZ \empty \CHKerrmisformeddate \else
  \pdffinfo {/CreationDate (D:#1#2#3000000) }\fi \fi \fi}
\def \metaAB #1-#2-#3-#4\relax {% modification
  \edef \metaABW {#1}\edef \metaABX {#2}\edef \metaABY {#3}%
  \ifx \metaABW \empty \CHKerrmisformeddate \else
  \ifx \metaABX \empty \CHKerrmisformeddate \else
  \ifx \metaABY \empty \CHKerrmisformeddate \else
  \pdffinfo {/ModDate (D:#1#2#3000000) }\fi \fi \fi}
\ldef \metaB #1;#2charset#3=#4\relax {% content-type
  \edef \metaBW {#1}\edef \metaBX {#2}\edef \metaBY {#3}%
  \lowercase {\edef \metaBZ {#4}}%
  \ifx \metaBX \CONspace \else \CHKerrmetaonespace \metaBX \fi
  \ifx \metaBY \empty \else \CHKerrmetanospace \metaBY \fi
  \ifx \metaBW \CONtexthtml
  \let \CPG \empty \let \CPGhtml \empty \CPGget #4\relax
  \ifx \CPG \empty \CPGgetiso #4\relax \fi
  \ifx \CPG \empty \LENGTHget {#4}%
  \ifnum \LENGTH > 8 \CPGgetwindows #4\relax \fi \fi}
\DEFsearch {\CONCPG \metaBZ }\ifDEF
  \csname \CONCPG \metaBZ \endcsname \fi % get alias
  \ifx \CPG \empty \xdef \CPG {#4}\xdef \CPGhtml {#4}\fi \fi}

\def \METAinit {\let \METAcont \empty
  \let \METAname \empty \let \metahttpequiv \empty}

\def \METApdfinfocreator {\pdffinfo {%
  /Creator (\GMSname \CONspace \GMSversion \CONspace - \GMSdomain) }}
\ifx \pdffinfo \undefined \let \METApdfinfocreator \relax \fi

\def \METApdfcatalogmenu {\pdfcatalog {/ViewerPreferences <<
  /HideToolbar false /HideMenuBar false /HideWindowUI false
  /FitWindow true /CenterWindow true >> }}
\ifx \pdffinfo \undefined \let \METApdfcatalogmenu \relax \fi

\def \METAtitle {\expandafter \METAtitleA \the \GAP \empty \relax}
\ldef \METAtitleA #1\relax {\edef \mytitle {#1}%
  \pdffinfo {/Title (\mytitle) }}
\ifx \pdffinfo \undefined \let \METAtitle \relax \fi

% NEST: =====

\def \NESTabegin {%
  \ifOPENfont \OPENfontoutertrue \else \OPENfontouterfalse \fi
  \ELEMENTclosenest \CONa}
\def \NESTaend {\ifOPENfontouter \else \ELEMENTclosenest \CONfont \fi}

\def \NESTaddressbegin {\ifOPENobject \else \ELEMENTclosenest \CONp \fi}

\def \NESTblockquotebegin {\ELEMENTclosenest \CONp}
\def \NESTblockquoteend {\ELEMENTclosenest \CONp}

\def \NESTbodybegin {%
  \ELEMENTclosenest \CONhead \ELEMENTclosenest \CONhead} % yes, twice!
\def \NESTbodyend {%
  \ifOPENobject \else
  \ELEMENTclosenest \CONp \ELEMENTclosenest \CONscript
  \ELEMENTclosenest \CONtd \ELEMENTclosenest \CONth
  \ELEMENTclosenest \CONtr \ELEMENTclosenest \CONtbody
  \ELEMENTclosenest \CONtable \ELEMENTclosenest \CONform
  \ifOPENparinner \CHKerrmissingendtag \CONp \fi \fi}

\def \NESTcenterbegin {\ELEMENTclosenest \CONp}

\def \NESTcolgroupbegin {\ELEMENTclosenest \CONcolgroup}

\def \NESTddbegin {\ELEMENTclosenest \CONp
  \ELEMENTclosenest \CONdt \ELEMENTclosenest \CONdd}
\def \NESTddend {\ELEMENTclosenest \CONp}

\def \NESTdivbegin {\ELEMENTclosenest \CONfont}

```

```

\ifOPENobject \else \ELEMENTClosenest \CONp \fi}
\def \NESTdivend {\ELEMENTClosenest \CONfont
\ifOPENp \CHKerrmissingendtag \CONp \GAPfilledfalse \fi
\ELEMENTClosenest \CONlabel}

\def \NESTdbegin {\ELEMENTClosenest \CONp}
\def \NESTdlend {\ELEMENTClosenest \CONp \ELEMENTClosenest \CONdt
\ELEMENTClosenest \CONdd}

\def \NESTdtbegin {\ELEMENTClosenest \CONdt \ELEMENTClosenest \CONdd}

\def \NESTformbegin {\ELEMENTClosenest \CONp \ELEMENTClosenest \CONform}
\def \NESTformend {\ELEMENTClosenest \CONselect}

\def \NESThbegin {\ELEMENTClosenest \CONfont
\ifOPENobject \else \ELEMENTClosenest \CONp \fi
\ifOPENparinner \CHKerrmissingendtag \CONp \fi}
\def \NESThend {\ELEMENTClosenest \CONfont}

\def \NESTheadbegin {%
\ifOPENhead \ifOPENbody \else \CHKerrmissingendtag \CONhead \fi \fi}
\def \NESTheadend {\ifOPENstyle \CHKerrmissingendtag \CONstyle \fi}

\def \NESThtmlend {\ELEMENTClosenest \CONhead}

\def \NESTlibegin {\ELEMENTClosenest \CONli \ELEMENTClosenest \CONp}
\def \NESTliend {\ELEMENTClosenest \CONp}

\def \NESTlinkbegin {\ELEMENTClosenest \CONstyle}

\def \NESTmapend {\ELEMENTClosenest \CONp}

\def \NESTmetabegin {\ELEMENTClosenest \CONstyle}

\def \NESTnobrend {\ELEMENTClosenest \CONfont}

\def \NESTolbegin {\ELEMENTClosenest \CONp}
\def \NESTolend {\ELEMENTClosenest \CONp \ELEMENTClosenest \CONli}

\def \NESToprgroupend {\ELEMENTClosenest \CONoption}

\def \NESToptionbegin {\ELEMENTClosenest \CONoption}

\def \NESTpbegin {%
\ifOPENp \ifOPENobject
\ifOPENparinner \ifOPENparouter \else \CHKerrmissingendtag \CONp \fi \fi
\else \CHKerrmissingendtag \CONp \fi \fi
\ifOPEND \else \ifOPENH \else %%
\ifOPENobject \ifOPENiframe \OPENparinnertrue %%
\else \OPENparinnerfalse \fi \fi \fi \fi} %%
\def \NESTpend {\ELEMENTClosenest \CONb \ELEMENTClosenest \CONem
\ELEMENTClosenest \CONi \ELEMENTClosenest \CONstrong \ELEMENTClosenest \CONa
\ifOPENfontinner \CHKerrmissingendtag \CONfont \fi}

\def \NESTprebegin {\ELEMENTClosenest \CONp}
\def \NESTselectend {\ELEMENTClosenest \CONoption}

\def \NESTtablebegin {%
\ifOPENb \OPENboutertrue \else \OPENbouterfalse \fi
\ifOPENi \OPENioutertrue \else \OPENifalse \fi
\ifOPENfont \OPENfontoutertrue \else \OPENfontouterfalse \fi
\ifOPENp \OPENparoutertrue \else \OPENparouterfalse \fi}
\def \NESTtableend {\ELEMENTClosenest \CONtd \ELEMENTClosenest \CONth
\ELEMENTClosenest \CONtr \ELEMENTClosenest \CONthead
\ELEMENTClosenest \CONtbody \ELEMENTClosenest \CONtfoot
\ifOPENfontouter \else \ELEMENTClosenest \CONfont \fi}
\def \NESTtableendouter {\ifOPENbouter \CHKerrmisnested \fi
\ifOPENiouter \CHKerrmisnested \fi \ifOPENfontouter \CHKerrmisnested \fi
\ifOPENbouter \else \ifOPENiouter \else \ifOPENfontouter \else \fi \fi \fi}

\def \NESTtbodybegin {\ELEMENTClosenest \CONcolgroup
\ELEMENTClosenest \CONthead \ELEMENTClosenest \CONtbody
\ELEMENTClosenest \CONtfoot}
\def \NESTtbodyend {\ELEMENTClosenest \CONtd
\ELEMENTClosenest \CONth \ELEMENTClosenest \CONtr}

\def \NESTtdbegin {\ELEMENTClosenest \CONtd \ELEMENTClosenest \CONth}
\def \NESTtdend {\ifOPENbouter \else \ELEMENTClosenest \CONb \fi
\ELEMENTClosenest \CONem \ifOPENiouter \else \ELEMENTClosenest \CONi \fi
\ELEMENTClosenest \CONstrong}

\def \NESTtdthend {\ifOPENfontinner \CHKerrmissingendtag \CONfont \fi
\ELEMENTClosenest \CONa \ifOPENparinner \CHKerrmissingendtag \CONp \fi} %%

\let \NESTtfootbegin \NESTtbodybegin
\let \NESTtfootend \NESTtbodyend

\let \NESTthbegin \NESTtdbegin
\let \NESTthend \NESTtdend %%

\let \NESTtheadbegin \NESTtbodybegin
\let \NESTtheadend \NESTtbodyend

\def \NESTtitlebegin {\ELEMENTClosenest \CONstyle}

\def \NESTtrbegin {\ELEMENTClosenest \CONtr}
\let \NESTtrend \NESTtdbegin % sic!

\let \NESTulbegin \NESTolbegin
\let \NESTulend \NESTolend

% NEW: =====
\ldef \NEWatt #1 {\message {#1},%
\gDef \CONcon #1 {#1}\gDef \CONatt #1 {%
\CHKerrproprietaryatt {#1}} % must be redefined for tag}
\ldef \NEWattdef #1 #2 {\gDef \CONcon #1 {#1}\gDef \CONatt #1 {#2}}
\ldef \NEWattnotstrictunsupported #1 {%
\gDef \CONatt #1 {\CHKerrnotstrict \CHKerrrunsupattprop {#1}}}
\ldef \NEWatttag #1 #2 #3 {% define \TAG#2ATT#1 (reverse args)
\gDef \CONtag #2\CONatt #1 {#3}}
\ldef \NEWatttagdeprecated #1 #2 #3 {%
\gDef \CONtag #2\CONatt #1 {% reverse args
\CHKerrdeprecatedatt {#1}#3}}
\ldef \NEWatttagdeprecatedunsupported #1 #2 {%
\gDef \CONtag #2\CONatt #1 {% reverse args
\CHKerrdeprecatedatt {#1}\CHKerrrunsupattprop {#1}}}
\ldef \NEWatttagproprietary #1 #2 #3 {%
\gDef \CONtag #2\CONatt #1 {% reverse args
\CHKerrproprietaryatt {#1}#3}}
\ldef \NEWatttagproprietaryunsupported #1 #2 {%
\gDef \CONtag #2\CONatt #1 {% reverse args
\CHKerrproprietaryatt {#1}\CHKerrrunsupattprop {#1}}}
\ldef \NEWatttagnotstrict #1 #2 #3 {%
\gDef \CONtag #2\CONatt #1 {% reverse args
\CHKerrnotstrictatt {#2}#3}}
\ldef \NEWatttagnotstrictunsupported #1 #2 {%
\gDef \CONtag #2\CONatt #1 {% reverse args
\CHKerrnotstrict \CHKerrrunsupattprop {#1}}}

```

```

\ldef \NEWattagtemplate #1 #2 #3 {% use \TAG#3ATT#1 as template; reverse args
  \glDef \CONtag #2\CONatt #1 {\csname \CONtag #3\CONatt #1\endsname}}

\ldef \NEWattagunsupported #1 #2 {%
  \glDef \CONtag #2\CONatt #1 {% reverse args
    \CHKerrunSUPattprop {#1}}}

\ldef \NEWattquiet #1 {% like \NEWatt, but without message in gerolf.log
  \glDef \CONcon #1 {#1}\glDef \CONatt #1 {\CHKerrproprietaryatt {#1}}}

\ldef \NEWattproprietary #1 {% like \NEWatt, but without message in gerolf.log
  \glDef \CONcon #1 {#1}\glDef \CONatt #1 {\CHKerrproprietaryatt {#1}}}

\ldef \NEWattunsupported #1 {\glDef \CONatt #1 {\CHKerrunSUPattprop {#1}}}

\ldef \NEWpropquiet #1 #2{% like \NEWpropnew, without \message in gerolf.log
  \glDef \CONprop #1 {#2}}

\ldef \NEWpropmessage #1 {\message {#1},%
  \glDef \CONprop #1 {\CHKerrunSUPattprop {#1}}}

\ldef \NEWpropnew #1 #2{\message {#1},\glDef \CONprop #1 {#2}}

\ldef \NEWpropnewtag #1 #2 #3{\glDef {\CONtag #1\CONprop #2} {#3}}

\ldef \NEWpropproprietary #1 #2{% like \NEWpropnew, with error message in log
  \glDef \CONprop #1 {\CHKerrproprietaryatt {#1}#2}}

\ldef \NEWproptagunsupported #1 #2 {% %%%
  \glDef \CONtag #1\CONprop #2 {% reverse args
    \CHKerrunSUPattprop {#2}}}

\ldef \NEWpropunsupported #1 {\glDef \CONprop #1 {\CHKerrunSUPattprop {#1}}}

% NO: =====

\nec \NOcell
\nec \NOcol
\def \NOcollast {0}
\nec \NOdlast
\nec \NOh
\nec \NOinclude \NOinclude = 0 % included documents
\nec \NOinput \NOinput = 1 % chained documents
\nec \NOli
\nec \NOlilast

% Paragraph numbering:

\nec \NOPar

\def \NOParadvance {\gadvice \NOPar by 1 }

\nec \NOParlast

\def \NOParlastsetglobal {\global \NOParlast = \chkWLMtotelements}

\def \NOParreset {\NOPar = 0 }

\def \NOParresetglobal {\global \NOPar = 0 }

\nec \NOrow
\nec \NOrowspan
\nec \NOTable

% OBEY: =====

\def \OBEYbreak {\VALset \test
  \ifx \test \CONalways \OBEYbreakpage \else
    \ifx \test \CONavoid \nobreak \else
      \ifx \test \CONleft \CLASSbreakifeven \else
        \ifx \test \CONright \CLASSbreakifodd \fi \fi \fi \fi}
\def \OBEYbreakh {\ifnum \count 1 > 1 \OBEYbreak \fi}
\def \OBEYbreakpage {\vfill \par \break}

\def \OBEYend {\ifOPENli \else \NOParresetglobal \fi \hangindent = 0mm}

\def \OBEYnormalspaces {\catcode \ \CATwhitespace}
\def \OBEYnormallines {\catcode \^M=\CATendofline}
\def \OBEYnormalspacesandlines {\OBEYnormalspaces \OBEYnormallines}

\def \OBEYstrutspace {{\setbox 3 = \hbox {x}\strut \hskip \wd 3 }}

% Init active whitespace, compare Knuth, pages 352, 381.
\def \OBEYspaces {\catcode \ =\CATActive} % active whitespace
\def \OBEYspacesandlines {\OBEYspaces \OBEYlines}
% No whitespace and linebreaks allowed here:
{\OBEYspaces\global\let =\OBEYstrutspace} %% check this
{\catcode \^M=\CATActive%
  \gdef \OBEYlines{\catcode \^M=\CATActive\let^M\par}\global\let^M=\par}
{\catcode \^M=\CATActive%
  \gdef \OBEYspacelines{\catcode \^M=\CATActive\let^M\CONspace}}

% OBJECT: =====

\def \OBJECTdata {exit}

\def \OBJECThandle {%
  \ifLINKunrendered \OPENObjectfalse \else
    \ifx \URLhref \empty \else
      \ifx \OBJECTtype \CONTEXThtml \MSGObjecthtml \FILEhtmltrue \else
        \ifx \OBJECTtype \CONTEXTplain \MSGObjectplain \FILEplaintrue \else
          \ifx \OBJECTtype \CONTEXTTeX \MSGObjectTeX \FILETeXtrue \else
            \LINKunrenderedtrue \fi \fi \fi \fi
          \ifLINKunrendered \else \CHNObjectloadurl \fi
          \ifOPENTable \CHNget \CHKerrwellformedness \fi \fi}

\def \OBJECTinit {\ifOPENp \OPENparoutertrue \fi
  \let \URLhref \empty \let \OBJECTtype \empty \FONTfamgivenfalse}

\def \OBJECTplain {\ifFONTfamgiven \FONTchoose \FONTdo \else \FONTdomono \fi
  \parindent = 0mm \catcode \< = \CATOther \OBEYlines \OBEYspaces}

\def \OBJECTtex {\endmarkupCAT \tenrm \parindent = 20pt}

% OUTPUT: =====

% Fixme: outsource margin stuff

\output = {\outD {\unvbox 255 }} % executed by TeX engine itself

\def \OUTbody #1{% <body>, <gedcom>
  \frenchspacing % Fixme: should depend on language
  \FONTsizeadjusttrue \FONTdecorationtrue \let \Style \empty
  \VAL = {black}\CLRsetforeground \DEFsearch {\CONSTYLE @page}%
  \ifDEF \ELEMENT = {@page}\ELEMENTvaluegetstyle \outM \outF \fi
  \ifOPENiframe \outK \fi \ifOPENObject \else \outK \fi
  \ELEMENT = \expandafter {#1}\ELEMENTvalueget
  \global \NOTable = 0 \global \BKPnotable = 0
  \parindent = 0mm \MARGIN = 0mm \outL \outA
  \GAPsetstrut \KRNdonefalse \CLRizeforeground
  \let \CLRbody \CLRforeground
  \FONTpageno = \FONTfam \FONTchoose \FONTdo
  \KRNcheck \KRNshow \MSGfontfamily
  \CPGwincheck \MSGcodepagelanguage \noecho}
\def \outA {% footline
  \xdef \setLINEheightbody {\LINEheight = \the \LINEheight} % dummy expand
  \xdef \setFONTsizebody {\SIZE = \the \SIZE \SIZEbase = 3 }%

```

```

\global \footline = {\ifOUTnpage \ifOUTnpage dont \hfil \else \outAdo \fi
\else \ifOUTnpage do \outAdo \else \hfil \fi \fi}
\def \outAdo {\setLINEheightbody \setFONTsizebody \strut
{FONTfam = \FONTpageno \FONTchoose \FONTdo
\let \outAdoleft \hss \let \outAdoright \hss
\ifOUTgutter \ifodd \OUTnpage \let \outAdoright \empty
\else \let \outAdoleft \empty \fi \fi \outAdoleft
\ifnum \OUTnpage > 2 \the \OUTnpage \else \ifOUTnpage do
\ifnum \OUTnpage > 0 \the \OUTnpage \fi \fi \fi \outAdoright}}
% Double-column output, compare Knuth, pages 416/417 (manmac.tex):
\def \outB {\hbadness = 5000 % begin double column output
\ifnum \clubpenalty > 9999 \clubpenalty = 9999 \fi
\ifnum \widowpenalty > 9999 \widowpenalty = 9999 \fi
\output = {\global \setbox \OUTpagepart = \vbox {
\unvbox 255 \bigskip }}\eject \output = {\outC}%
\hsize = \TEXTwidth \advance \hsize by -.LINEheight \hsize = 0.5\hsize
\vsiz = \TEXTheight \vsize = 2\vsiz}
\def \outC {% begin double page
\splittopskip = \topskip \splittmaxdepth = \maxdepth
\dimen 0 = \TEXTheight \advance \dimen 0 by -.ht \OUTpagepart
\vfuzz = 1ex \vbadness = 10000
\setbox 0 = \vsplit 255 to \dimen 0 \setbox 2 = \vsplit 255 to \dimen 0
\outD \outJ \unvbox 255 \penalty \outputpenalty }
\def \outD #1{\count 1 = 0 \count 2 = 0 \count 3 = 0 \count 4 = 0
\count 5 = 0 \count 6 = 0 \count 7 = 0 \count 8 = 0 \count 9 = 0
\shipout \vbox {\vfuzz = 1ex \offinterlineskip
\ifOUTdouble \outF \outE \else \outF \outK\fi
\ifCLRbgbody \CLRizebackground \fi \ifBKGD \BKGDmake \fi \makeheadline
%%}%\ifCLR \CLRize \CLRactive \fi % Fixme: may cause wrong color on top
%%% \CLRize {1 0 0 rg 1 0 0 RG}% can show lines with wrong color on top
\let \CLRactive \CLRforeground \vbox to \TEXTheight {\#1\vss}%
\ifCLR \CLRize \CLRbody \fi
\vskip \MARGINpageno \line {\GAPsetstrut \strut \the \footline}%
\boxmaxdepth = \maxdepth}% begin single page
\advancepageno \global \hangindent = 0mm \global \hangafter = 1 }
\def \outE {\hsize = \OUTwidth
\advance \hsize by -.OUTmarginl \advance \hsize by -.OUTmarginr
\global \hoffset = -1in \advance \hoffset by \OUTmarginl
\vsiz = \OUTheight \advance \vsiz by -.OUTmarginl
\advance \vsiz by -.OUTmarginb \voffset = -1in
\advance \voffset by \OUTmarginl}% begin single page locally
\def \outF {\DEFsearch {\CONSTYLE :left-CONTEXT-@page}\ifDEF % gutter
\xdef \STYLEpageleft {\csname \CONSTYLE :left-CONTEXT-@page\endcsname}%
\DEFsearch {\CONSTYLE :right-CONTEXT-@page}\ifDEF
\global \OUTguttertrue \xdef \STYLEpageright {
\csname \CONSTYLE :right-CONTEXT-@page\endcsname}%
\ifodd \OUTnpage \ELEMENT = {pageright}\else \ELEMENT = {pageleft}\fi
\ELEMENT\valuegetstyle \outM \fi \fi}
\def \outG {\output = {\outH}}\eject \pagegoal = \TEXTheight \bigskip}
\def \outH {\setbox 0 = \vbox {\unvbox 255}%
\dimen 0 = \ht 0 \advance \dimen 0 by \topskip
\advance \dimen 0 by -.LINEheight \divide \dimen 0 by 2
\splittopskip = \topskip \vfuzz = 1ex \vbadness = 10000
{\loop \global \setbox 3 = \copy 0
\global \setbox 1 = \vsplit 3 to \dimen 0
\ifdim \ht 3 > \dimen 0 \gadvance \dimen 0 by 1pt \repeat}%
\setbox 0 = \vbox to \dimen 0 {\unvbox 1}%
\setbox 2 = \vbox to \dimen 0 {\unvbox 3 \vfil}\outJ}
\def \outJ {\unvbox \OUTpagepart \hbox to \TEXTwidth {\box 0 \hss \box 2}}
\def \outK {\global \hsize = \OUTwidth % init
\gadvance \hsize by -.OUTmarginl \gadvance \hsize by -.OUTmarginr
\global \hoffset = -1in \gadvance \hoffset by \OUTmarginl
\global \vsiz = \OUTheight \gadvance \vsiz by -.OUTmarginl
\gadvance \vsiz by -.OUTmarginb \global \voffset = -1in
\gadvance \voffset by \OUTmarginl \outL}
\def \outL {\global \TEXTheight = \vsiz \global \TEXTwidth = \hsize
\ifdim \LINEheight > 0mm \global \LINEheightpageno = \LINEheight \fi
{\divide \OUTmarginb by 3 \ifdim \OUTmarginb > \LINEheightpageno
\global \MARGINpageno = 1.5\OUTmarginb
\gadvance \MARGINpageno by -.5\LINEheightpageno
\else \global \MARGINpageno = 2\LINEheightpageno \fi}% make
\def \outM {\OUTmargin = \MARGINtop \OUTmarginr = \MARGINright % set margin
\OUTmarginb = \MARGINbottom \OUTmarginl = \MARGINleft
\MARGINtop = 0mm \MARGINright = 0mm \MARGINbottom = 0mm \MARGINleft = 0mm }
\new \ifOUTdouble
\def \OUTdoublebegin {\ifOUTdouble \LINEupskip \outB \fi}% <center>, <div>
\def \OUTdoubleend {\ifOUTdouble \outG \fi}
\new \ifOUTgutter
\let \OUTheight \pdfpageheight % Fixme: rename
\ifx \pdfpageheight \undefined \csname newdimen\endcsname \OUTheight \fi
\let \OUTwidth \pdfpagewidth % Fixme: rename
\ifx \pdfpagewidth \undefined \csname newdimen\endcsname \OUTwidth \fi
\new \OUTmarginb \new \OUTmarginl \new \OUTmarginr \new \OUTmarginl
\let \OUTnpage \pageno %% \OUTnpage = 0
\new \ifOUTnpage \new \ifOUTnpage do \new \ifOUTnpage dont
\def \OUTnpagecheck {\ifnum \count 1 > 1 \global \OUTnpage true \else
\ifnum \count 2 > 1 \global \OUTnpage true \else
\ifnum \count 3 > 1 \global \OUTnpage true \fi \fi \fi}
\new \OUTpagepart
\def \OUTsize {% <@page size>
\VALset \test \expandafter \OUTsizeparse \test \relax}
\def \OUTsizeparse #1 #2\relax {\VAL = {#1}\VALsetdim \OUTwidth \hsize
\VAL = {#2}\VALsetdim \OUTheight \vsiz}
% PADDING: =====
\new \PDGbottom
\def \PDGbegin {\vbox \bgroup
\ifTABLEinter \else \vskip \PDGtop \fi \hbox \bgroup \hskip \PDGleft}
\def \PDGend {\hskip \PDGright \egroup
\ifTABLEinter \else \vskip \PDGbottom \fi \egroup}
\new \PDGleft
\new \PDGright
\def \PDGset {\edef \PDGtest {\the \VAL \CONspace}%
\expandafter \PDGsetA \PDGtest \relax}
% Check space (Fixme - supporting only one value for all):
\def \PDGsetA #1 #2\relax {\edef \PDGsetAZ {#2}%
\ifx \PDGsetAZ \empty \csname \CONproppadding top\endcsname
\PDGright = \PDGtop \PDGbottom = \PDGtop \PDGleft = \PDGtop
\else \CHKerrrunsupattprop \CONpadding \fi}
\new \PDGtop
% PLAIN: =====
% Plain TeX entities:
\def \PLAINAacute {\'A}
\def \PLAINaacute {\'a}
\def \PLAINAcirc {\AA}
\def \PLAINacirc {\Aa}
\def \PLAINacute {\'\empty}
\let \PLAINAelig \AE
\let \PLAINaelig \ae
\def \PLAINAgrave {\'A}
\def \PLAINagrave {\'a}

```

```

\let \PLAINaring \AA
\let \PLAINaring \aa
\def \PLAINalefsym {\FONTcmsymb \char "40}
\def \PLAINAlpha {\FONTcmmath A}
\def \PLAINalpha {\FONTcmmath \char "0B}
\def \PLAINamp {\&}
\def \PLAINand {\FONTcmsymb \char "5E}
\def \PLAINang {\FONTcmsymb \char "36}\lower 1.1ex \hbox { \char "7B}}
\def \PLAINasym {\FONTcmsymb \char "18}
\def \PLAINatilde {\~A}
\def \PLAINatilde {\~a}
\def \PLAINauuml {"A}
\def \PLAINauuml {"a}
\def \PLAINbdquo {, \kern -.2ex,}
\def \PLAINBeta {\FONTcmmath B}
\def \PLAINbeta {\FONTcmmath \char "0C}
\def \PLAINbrvbar {\FONTcmsymb \char "6A}
\def \PLAINbull {\FONTcmsymb \char "0F}
\def \PLAINcap {\FONTcmsymb \char "5C}
\def \PLAINcedil {\char "18}% \c does not work here
\def \PLAINccedil {C\llap {\char "18 \CONspace}}%
\def \PLAINccedil {c\llap {\char "18}}%
\def \PLAINcent {c\kern .2ex\llap /}
\def \PLAINChi {\FONTcmmath X}
\def \PLAINchi {\FONTcmmath \char "1F}
\def \PLAINcirc {\FONTcmsymb \char "0D}
\def \PLAINclubs {\FONTcmsymb \char "7C}
\def \PLAINcopy {\FONTcmsymb \char "0D}\llap {\raise .1ex
\hbox {c}\hskip .667ex}}
\def \PLAINcarr {\FONTcmsymb \char "20}\kern -.035ex
\raise 0.45ex \hbox {\char "37}}
\def \PLAINcup {\FONTcmsymb \char "5B}
\def \PLAINcurren {\llap {\raise .8ex \hbox {}}}%
\kern -.45ex o\kern -.45ex \llap {\raise .8ex \hbox {}}}
\def \PLAINdagger {\FONTcmsymb \char "7A}
\def \PLAINdagger {\FONTcmsymb \char "79}
\def \PLAINdArr {\FONTcmsymb \char "2B}
\def \PLAINdarr {\FONTcmsymb \char "23}
\def \PLAINdeg {\FONTcmttext \char "17}\kern -.05ex \strut}
\def \PLAINDelta {\FONTcmmath \char "01}
\def \PLAINdelta {\FONTcmmath \char "0E}
\def \PLAINdiam {\FONTcmsymb \char "7D}
\def \PLAINdivide {\FONTcmsymb \char "04}
\def \PLAINEacute {"E}
\def \PLAINEacute {"e}
\def \PLAINEcirc {"AE}
\def \PLAINEcirc {"e}
\def \PLAINcong {=\llap {\raise .6ex \hbox {\FONTcmsymb \char "18}}}}
\def \PLAINgrave {"E}
\def \PLAINgrave {"e}
\let \PLAINempty \O
\def \PLAINepsilon {\FONTcmmath E}
\def \PLAINepsilon {\FONTcmmath \char "0F}
\def \PLAINequiv {\FONTcmsymb \char "11}
\def \PLAINeta {\FONTcmmath H}
\def \PLAINeta {\FONTcmmath \char "11}
\def \PLAINETH {\raise 0.3ex \hbox {-}\kern -.085ex D}
\def \PLAINeth {\CONspace \char "13 \kern -.16ex {\FONTcmmath \char "40}}
\def \PLAINEu {\E}
\def \PLAINeu {"e}
\def \PLAINeuro {C\llap {\raise .2ex \hbox {\FONTcmmath =}}}
\def \PLAINexist {\FONTcmsymb \char "39}
\def \PLAINfnof {\FONTcmmath f}
\def \PLAINforall {\FONTcmsymb \char "38}
\def \PLAINfrac12 {\FONTcmsymb \char "36}\FONTcsmall
\raise 1ex \hbox {1}\kern 0.667ex 2}}
\def \PLAINfrac14 {\FONTcmsymb \char "36}\FONTcsmall
\raise 1ex \hbox {1}\kern 0.5ex 4}}
\def \PLAINfrac34 {\FONTcmsymb \char "36}\FONTcsmall
\raise 1ex \hbox {3}\kern 0.5ex 4}}
\def \PLAINfrac1 {\FONTcmsymb \char "36}\hskip 1ex}
\def \PLAINGamma {\FONTcmmath \char "00}
\def \PLAINgamma {\FONTcmmath \char "0D}
\def \PLAINge {\FONTcmsymb \char "15}
\def \PLAINgt {\FONTcmmath >}
\def \PLAINhArr {\FONTcmsymb \char "2C}
\def \PLAINharr {\FONTcmsymb \char "24}
\def \PLAINhearts {\FONTcmsymb \char "7E}
\def \PLAINhellip {\kern -.1ex.\kern -.1ex.}
\def \PLAINIacute {"I}
\def \PLAINIacute {"i}
\def \PLAINIcirc {"AI}
\def \PLAINIcirc {"Ai}
\def \PLAINIexcl {"!}
\def \PLAINIgrave {"I}
\def \PLAINIgrave {"i}
\def \PLAINimage {\FONTcmsymb \char "3D}
\def \PLAINinfin {\FONTcmsymb \char "31}
\def \PLAINint {\FONTcmsymb \char "73}
\def \PLAINiota {\FONTcmmath I}
\def \PLAINiota {\FONTcmmath \char "13}
\def \PLAINiquest {"?}
\def \PLAINisin {\FONTcmsymb \char "32}
\def \PLAINiuml {"I}
\def \PLAINiuml {"i}
\def \PLAINKappa {\FONTcmmath K}
\def \PLAINkappa {\FONTcmmath \char "14}
\def \PLAINLambda {\FONTcmmath \char "03}
\def \PLAINlambda {\FONTcmmath \char "15}
\def \PLAINlang {\FONTcmsymb \char "68}
\def \PLAINlaquo {\FONTcsmalltype <\kern -.5ex <}
\def \PLAINlArr {\FONTcmsymb \char "28}
\def \PLAINlarr {\FONTcmsymb \char "20}
\def \PLAINlceil {\FONTcmsymb \char "65}
\def \PLAINldquo {"`"}
\def \PLAINle {\FONTcmsymb \char "14}
\def \PLAINlfloor {\FONTcmsymb \char "63}
\def \PLAINlowast {\FONTcmmath \char "3F}
\def \PLAINloz {\FONTcmsymb \char "05}
\def \PLAINlrm {\CHKerrlostplain \CONlrm}
\def \PLAINlsquo {\FONTcsmalltype <}
\def \PLAINltdquo {"`"}
\def \PLAINlt {\FONTcmmath <}
\def \PLAINmacr {\raise .75ex \hbox {\char "7B}}
\def \PLAINmdash {\char "7C}
\def \PLAINmicro {\FONTcmmath \char "16}
\def \PLAINmiddot {\FONTcmsymb \char "01}
\def \PLAINminus {\char "7B}
\def \PLAINmu {\FONTcmmath M}
\def \PLAINmu {\FONTcmmath \char "16}
\def \PLAINnabla {\FONTcmsymb \char "72}
\def \PLAINndash {\char "7B}
\def \PLAINne {=\kern -.3ex\llap /}
\def \PLAINni {\FONTcmsymb \char "33}
\def \PLAINnot {\FONTcmsymb \char "3A}
\def \PLAINnotin {\FONTcmsymb \char "32}\kern -.3ex\llap /}
\def \PLAINnsub {\FONTcmsymb \char "1A}\kern -.3ex\llap /}
\def \PLAINNu {\FONTcmmath N}
\def \PLAINnu {\FONTcmmath \char "17}
\def \PLAINntilde {\~N}
\def \PLAINntilde {\~n}
\def \PLAINOacute {"O}
\def \PLAINOacute {"o}
\def \PLAINOcirc {"AO}
\def \PLAINOcirc {"Ao}
\let \PLAINOelig OE

```



```

\let \PLAINoelig \oe
\def \PLAINograve {\`o}
\def \PLAINograve {\`o}
\def \PLAINoline {\raise lex \hbox {\char "2D}}
\def \PLAINomega {\FONTcmath \char "0A}
\def \PLAINomega {\FONTcmath \char "21}
\def \PLAINomicron {\FONTcmath 0}
\def \PLAINomicron {\FONTcmath o}
\def \PLAINor {\FONTcmsymb \char "5F}
\def \PLAINordf {\FONTcsmall \raise lex \hbox {a}}
\def \PLAINordm {\FONTcsmall \raise lex \hbox {o}}
\let \PLAINoslash \o
\let \PLAINoslash \o
\def \PLAINoplus {\FONTcmsymb \char "08}
\def \PLAINotilde {\~o}
\def \PLAINotilde {\~o}
\def \PLAINotimes {\FONTcmsymb \char "0A}
\def \PLAINouml {"O}
\def \PLAINouml {"o}
\def \PLAINpara {\FONTcmsymb \char "7B}
\def \PLAINpart {\FONTcmath \char "40}
\def \PLAINpermil {\{\FONTcmsymb \char "36}\FONTcsmall
\raise lex \hbox {0}\kern 0.75ex 00}}
\def \PLAINperp {\FONTcmsymb \char "3F}
\def \PLAINphi {\FONTcmath \char "08}
\def \PLAINphi {\FONTcmath \char "1E}
\def \PLAINpi {\FONTcmath \char "05}
\def \PLAINpi {\FONTcmath \char "19}
\def \PLAINpiv {\FONTcmath \char "24}
\def \PLAINplusmn {\FONTcmsymb \char "06}
\def \PLAINpound {\FONTcital \char "24}
\def \PLAINprime {\FONTcmsymb \char "30\char "30}
\def \PLAINprime {\FONTcmsymb \char "30}
\def \PLAINprod {\char "05}
\def \PLAINprop {\CHKerrlostplain \CONprop}
\def \PLAINpsi {\FONTcmath \char "09}
\def \PLAINpsi {\FONTcmath \char "20}
\def \PLAINquot {\FONTcmttype "}
\def \PLAINradic {\raise .75\LINEheight \hbox {\FONTcmsymb \char "70}}
\def \PLAINrang {\FONTcmsymb \char "69}
\def \PLAINraquo {\FONTcsmalltype >\kern -.5ex >}
\def \PLAINrArr {\FONTcmsymb \char "29}
\def \PLAINrarr {\FONTcmsymb \char "21}
\def \PLAINrceil {\FONTcmsymb \char "64}
\def \PLAINrdquo {"'}
\def \PLAINreal {\FONTcmsymb \char "3C}
\def \PLAINreg {\{\FONTcmsymb \char "0D}\llap {\lower 0.15ex
\hbox {R\hskip 0.25ex}}}}
\def \PLAINrfloor {\FONTcmsymb \char "62}
\def \PLAINrho {\FONTcmath P}
\def \PLAINrho {\FONTcmath \char "1A}
\def \PLAINrlm {\CHKerrlostplain \CONrlm}
\def \PLAINrsaquo {\FONTcsmalltype >}
\def \PLAINrsquo {"'}
\def \PLAINsdot {\FONTcmsymb \char "01}
\def \PLAINsect {\FONTcmsymb \char "78}
\def \PLAINsbquo {,}
\def \PLAINScaron {\v S}
\def \PLAINscaron {\v s}
\def \PLAINsigma {\FONTcmath \char "06}
\def \PLAINsigma {\FONTcmath \char "1B}
\def \PLAINsigmaf {\FONTcmath \char "26}
\def \PLAINsim {\FONTcmsymb \char "18}
\def \PLAINspades {\FONTcmsymb \char "7F}
\def \PLAINsub {\FONTcmsymb \char "1A}
\def \PLAINsube {\FONTcmsymb \char "13}
\def \PLAINsum {\char "06}
\def \PLAINsup {\FONTcmsymb \char "1B}
\def \PLAINsup1 {\FONTcsmall \raise lex \hbox {1}}
\def \PLAINsup2 {\FONTcsmall \raise lex \hbox {2}}
\def \PLAINsup3 {\FONTcsmall \raise lex \hbox {3}}
\def \PLAINsupe {\FONTcmsymb \char "13}
\let \PLAINszlig \ss
\def \PLAINtau {\FONTcmath T}
\def \PLAINtau {\FONTcmath \char "1C}
\def \PLAINthere4 {\kern -.25ex {\PLAINmldot }\kern -.25ex .}
\def \PLAINtheta {\FONTcmath \char "02}
\def \PLAINtheta {\FONTcmath \char "12}
\def \PLAINthetasym {\FONTcmath \char "23}
\def \PLAINTHORN {\kern -.0525ex {\raise 0.3ex \hbox {o}}}
\def \PLAINthorn {\lower 0.3ex \hbox {I}\kern -.0525ex o}
\def \PLAINtimes {\FONTcmsymb \char "02}
\def \PLAINtrade {\FONTcsmall \raise lex \hbox {TM}}
\def \PLAINuacute {\`u}
\def \PLAINuacute {\`u}
\def \PLAINuArr {\FONTcmsymb \char "2A}
\def \PLAINuarr {\FONTcmsymb \char "22}
\def \PLAINUcirc {\AU}
\def \PLAINucirc {\Au}
\def \PLAINUgrave {\`U}
\def \PLAINugrave {\`u}
\def \PLAINuml {\char "7F}
\def \PLAINupsih {\FONTcmath \char "07}
\def \PLAINUpsilon {\FONTcmath Y}
\def \PLAINupsilon {\FONTcmath u}
\def \PLAINUuml {"U}
\def \PLAINuuml {"u}
\def \PLAINweierp {\FONTcmsymb P}
\def \PLAINXi {\FONTcmath \char "04}
\def \PLAINxi {\FONTcmath \char "18}
\def \PLAINYacute {\`Y}
\def \PLAINyacute {\`y}
\def \PLAINyen {\Yllap =}
\def \PLAINYuml {"Y}
\def \PLAINyuml {"y}
\def \PLAINZeta {\FONTcmath Z}
\def \PLAINzeta {\FONTcmath \char "10}
\def \PLAINzwnj {\penalty 10000 \strut}
\def \PLAINzwj {\strut}

% PROP: =====
\new \ifPRP
\def \PRPget {\expandafter \prpA \the \PRPlist ;STOP:-GMS-STOP;}
\new \PRPlist
\def \PRPlistget #1{% Fixme: replaceable - set \PRPlist, call \PRPget
\expandafter \prpA #1;STOP:-GMS-STOP;}
\def \prpA #1#2:#3#4;{\lowercase {\edef \PRPgetWX {#1#2}}%
\edef \PRPgetY {#3}\edef \PRPgetZ {#4}%
\ifx \PRPgetY \CONquotmark \let \PRPgetYZ \PRPgetZ
\else \edef \PRPgetYZ {#3#4}\fi % Fixme: check apostrophe too
\let \prpAn \relax % next
\ifx \PRPgetZ \CONstopmark \prpAA \else \prpB \fi \prpAn \prpAvalue}
\def \prpAA {\let \prpAvalue \relax} % stop
\def \prpB {\VAL = \expandafter {\PRPgetYZ}% run
\edef \prpBt {\PRPgetYZ \CONspace}\let \prpBv \empty % test, value
\expandafter \prpC \prpBt \CONstopmark \VAL = \expandafter {\prpBv}%
\DEFsearch {\CONtag \the \ELEMENT \CONprop \PRPgetWX}%
\ifDEF \prpBB \else \prpBA \fi \let \prpAvalue \prpA}
\def \prpBA {\DEFsearch {\CONprop \PRPgetWX} % do
\ifDEF \prpBAA \else \CHKerrunrecogattprop \PRPgetWX \fi}
\def \prpBAA {\def \prpAn {\csname \CONprop \PRPgetWX \endcsname}}% next
\def \prpBB {\def \prpAn {\element
\csname \CONtag \the \ELEMENT \CONprop \PRPgetWX \endcsname}}

```

```

\def \prpC #1 #2{\edef \prpCY {#1}\edef \prpCZ {#2}%
\expandafter \prpCA #1"relax % Fixme: check apostrophe too
\ifx \prpCY \CONquotmark \let \prpCY \empty \fi
\ifx \prpCZ \CONquotmark \let \prpCZ \empty \fi
\ifx \prpCZ \CONstopmark \prpCD
\else \ifx \prpCAZ \CONcommaquotmark \prpCD \let \prpAn \prpCC
\else \prpCB \fi \fi \prpAn} % space
\def \prpCA #1#2\relax {\edef \prpCY {#1}\edef \prpCAZ {#2}}% clear
\def \prpCB {\edef \prpBv {\prpBv \prpCY \CONspace \prpCZ }%
\let \prpAn \prpC} % do
\def \prpCC #1GMS-STOP;\fi} % gulp conditional - Fixme: dirty
\def \prpCD {\edef \prpBv {\prpBv \prpCY } \let \prpAn \relax} % stop

% REFERENCE: =====
\new \ifRFRamp
\def \RFRfinish #1{}

\def \RFRget {{\global \GAPsgm = {}} get special signs in gap content
\edef \RFRgetprobe {\the \GAP \CONampersand}%
\expandafter \rfrA \RFRgetprobe \empty ;\global \GAP = \GAPsgm}}

\let \RFRrelax \relax

\new \ifRFRundef

% rfrA: -----
\bgroup \CATreference
\gdef \rfrA ^1&^2^3;{{% parse gap for delimiters "&" and ";":
\ifOPENpre \let \strut \empty \OBEYspacelines \OBEYnormalspaces \fi
\edef \rfrAX {^1}\edef \rfrAZ {^3}%
\ifx \rfrAX \empty \else \GAPsgmapp {^1}\fi
\ifx \rfrAZ \empty \else \rfrAA ^2^3&\relax \fi}}
\gdef \rfrAA ^1&^2\relax {\edef \rfrAAZ {^2}% ampersand
\ifx \rfrAAZ \CONampersand \rfrAAA ^1 \relax
\else \rfrB ^1;\ifRFRundef \else \aftergroup \rfrA \fi \fi}
\gdef \rfrAAA ^1 ^2\relax {%
\RFRundelimitedtrue \CHKerrundelimited {^1}\rfrB ^1;\GAPsgmapp {^2}}
% rfrB - Get special char by given ref to codepage: -----
\gdef \rfrB ^1^2^3;{% find out type of reference
\def \rfrBX {^1}\def \rfrBY {^2}%
\xdef \rfrBXYZ {^1^2^3}%
\ifx \rfrBX \CONhashmark \uppercase {\edef \rfrBZ {^3}}%
\ifx \rfrBY \CONx \rfrDB \rfrBZ
\else \ifx \rfrBY \CONX \rfrDB \rfrBZ
\else \ifnum ^2^3 > 255 \rfrDB {^2^3}\else
\DEFsearch {\CONspecific \FONTfamuc } \ifDEF
\rfrBA {^2^3}\else \rfrDB {^2^3}\fi \fi \fi \fi
\else \rfrC {^1}{^2^3}\rfrBA \fi } \egroup
\def \rfrBA #1{% get special char by number (<=255) in codepage
\ifnum #1 > -1 \ifnum #1 < 256 \ifnum #1 = 160 \rfrCD
\else \GAPsgmapp {\char #1\RFRrelax } \fi %%
\else \MSGnotexcharacter {#1}\fi \else \MSGnotexcharacter {#1}\fi}}
% rfrC - Reference name: -----
\def \rfrC #1#2#3{\edef \rfrCXY {#1#2}\def \rfrCZ {#3}% number
\ifx \rfrCXY \CONshy \rfrCA \else
\ifx \rfrCXY \CONemsp \rfrCB \else
\ifx \rfrCXY \CONensp \rfrCC \else
\ifx \rfrCXY \CONnbsp \rfrCD \else % Fixme: used globally
\ifx \rfrCXY \CONthinsp \rfrCE \else
\ifCPGplain \rfrCF \else \ifCPGwin \rfrCG \else
\rfrCH \fi \fi \fi \fi \fi \fi \fi}
% References that are not treated as characters:
\def \rfrCA {\GAPsgmapp \cdot} % shy
\def \rfrCB {\GAPsgmapp {\hspace 1em}} % emsp
\def \rfrCC {\GAPsgmapp {\hspace 1ex}} % ensp
\def \rfrCD {\GAPsgmapp {\penalty 10000 \strut \hbox {\ }}} % nbsp
\def \rfrCE {\GAPsgmapp {\hspace .5ex}} % thinsp

\def \rfrCF {\DEFsearch {\CONplain \rfrCXY } \ifDEF \expandthree
\GAPsgmapp \expandafter {\csname \CONplain \rfrCXY \endcsname}%
\else \CHKerrundefref {\CONampersand \rfrCXY ;} \fi} % plain TeX
\def \rfrCG {\DEFsearch {\CONwin \rfrCXY } \ifDEF
\edef \rfrCGtest {"\csname \CONwin \rfrCXY \endcsname } % Windows
\expandthree \rfrCZ \expandafter {\rfrCGtest } \else \rfrCH \fi}
\gdef \rfrCH {\DEFsearch {\CONref \rfrCXY } \ifDEF \rfrD % get named ref.
\else \CHKerrundefref {\CONampersand \rfrCXY ;} \fi}
% rfrD - Other named references: -----
\def \rfrD {\DEFsearch {\CONref \rfrCXY } \ifDEF
\edef \rfrDtest {"\csname \CONref \rfrCXY \endcsname }%
\expandafter \ifnum \rfrDtest < 256 \expandthree \rfrCZ
\expandafter {\rfrDtest} \else \rfrDB \rfrDtest \fi \else \rfrDA \fi}
% Get special char by its name in the codepage:
\def \rfrDA {\DEFsearch {\CONplain \rfrCXY } \ifDEF
\edef \rfrDAq {{\FONTcmtxt \csname \CONplain \rfrCXY \endcsname}}%
\expandthree \GAPsgmapp \expandafter {\rfrDAq}%
\else \CHKerrundefref {\CONampersand \rfrCXY ;} \fi} % plain TeX
\def \rfrDB #1{\ifCPGwin \DEFsearch {WIN#1} \ifDEF % unicode
\csname WIN#1\endcsname \else \rfrE {#1}\fi \else \rfrE {#1}\fi}
% rfrE - Number (>255) in codepage: -----
\def \rfrE #1{% number - Fixme: redundant with \FONTchoose \FONTuse
% Fixme: Set Unicode data: catcode, lcode, ucode
\ifnum #1 = 160 \rfrCD \else \edef \rfrEZ {#1}%
\UCDslotset \rfrEZ \UCDpointindexset \rfrEZ
\HEXinput = \UCDslot \HEXoutput = {} \HEX
\expandthree \LENGTHget \expandafter {\the \HEXoutput }%
\let \leading \empty % Fixme: outsource
\ifnum \LENGTH = 3 \def \leading {0} \else
\ifnum \LENGTH = 2 \def \leading {00} \else
\ifnum \LENGTH = 1 \def \leading {000} \fi \fi \fi \HEXinput = \rfrEZ
\ifnum \HEXinput < "A0 \else \edef \CPG {G\leading \the \HEXoutput} \fi
\ifFONTbd \edef \MAPisbold {1} \else \edef \MAPisbold {0} \fi
\ifFONTitalic \edef \MAPisital {1} \else \edef \MAPisital {0} \fi
\edef \somefontname {\CONfontthyphen \CPG
-\FONTfamuc -\MAPisbold -\MAPisital}%
\DEFsearch \somefontname \ifDEF \glet \next \relax \else \rfrEB \fi
\DEFsearch \somefontname \ifDEF \next \rfrEC
\else \DEFsearch {\CONplain \rfrBXYZ} \ifDEF \rfrEA
\else \expandafter \CHKerrlost \rfrEZ \relax \fi \fi \fi}}
\def \rfrEA {\CHKerrfallbackname \rfrBXYZ \GAPsgmapp {{\FONTcmtxt
\csname \CONplain \rfrBXYZ \endcsname}} % fallback number
\def \rfrEB {\edef \somefontname {\CONfontthyphen \CPG
-\FONTfbkuc -\MAPisbold -\MAPisital} % get fallback number
\gdef \next {\expandafter \CHKerrfallback \rfrEZ \relax}}
\def \rfrEC {\edef \rfrECprobe {{\font \FONT =
\csname \somefontname \endcsname \CONat \SIZEcurrent
\FONT \char \number \UCDpointindex }}} % number font
\expandthree \GAPsgmapp \expandafter {\rfrECprobe}}

% RGB: =====
\def \RGBaliceblue {F0F8FF}
\def \RGBantiquewhite {FAEBD7}
\def \RGBaqua {00FFFF}
\def \RGBaquamarine {7FFFD4}
\def \RGBazure {F0FFFF}
\def \RGBbeige {F5F5DC}
\def \RGBbisque {FFB6C1}
\def \RGBblack {000000}
\def \RGBblanchedalmond {FFB6C1}
\def \RGBblue {0000FF}
\def \RGBblueviolet {6A2BE2}
\def \RGBbrown {A52A2A}
\def \RGBburlywood {DEB887}
\def \RGBcadetblue {5F9EAO}
\def \RGBchartreuse {7FFF00}
\def \RGBchocolate {D2691E}
\def \RGBcoral {FF7F50}

```

```

\def \RGBcornflowerblue {6495ED}
\def \RGBcornsilk {FFF8DC}
\def \RGBcyan {00FFFF}
\def \RGBdarkgoldenrod {B8860B}
\def \RGBdarkgreen {006400}
\def \RGBdarkkhaki {BDB76B}
\def \RGBdarkolivegreen {556B2F}
\def \RGBdarkorange {FF8C00}
\def \RGBdarkorchid {9932CC}
\def \RGBdarksalmon {E9967A}
\def \RGBdarkseagreen {8FBC8F}
\def \RGBdarkslateblue {483D8B}
\def \RGBdarkslategray {2F4F4F}
\def \RGBdarkturquoise {00CED1}
\def \RGBdarkviolet {9400D3}
\def \RGBdeeppink {FF1493}
\def \RGBdeepskyblue {00BFFF}
\def \RGBdimgray {696969}
\def \RGBdodgerblue {1E90FF}
\def \RGBfirebrick {B22222}
\def \RGBfloralwhite {FFFAF0}
\def \RGBforestgreen {228B22}
\def \RGBfuchsia {FF00FF}
\def \RGBgainsboro {DCDCDC}
\def \RGBghostwhite {F8F8FF}
\def \RGBgold {FFD700}
\def \RGBgoldenrod {DAA520}
\def \RGBgray {808080}
\def \RGBgreen {008000}
\def \RGBgreenyellow {ADFF2F}
\def \RGBhoneydew {F0FFF0}
\def \RGBhotpink {FF69B4}
\def \RGBindianred {CD5C5C}
\def \RGBivory {FFFFFF}
\def \RGBkhaki {F0E68C}
\def \RGBlavender {E6E6FA}
\def \RGBlavenderblush {FFF0F5}
\def \RGBlawngreen {7CFC00}
\def \RGBlemonchiffon {FFFACD}
\def \RGBlightblue {ADD8E6}
\def \RGBlightcoral {F08080}
\def \RGBlightcyan {E0FFFF}
\def \RGBlightgoldenrodyellow {FAFAD2}
\def \RGBlightpink {FFB6C1}
\def \RGBlightsalmon {FFA07A}
\def \RGBlightseagreen {20B2AA}
\def \RGBlightskyblue {87CEFA}
\def \RGBlightslategray {778899}
\def \RGBlightsteelblue {B0C4DE}
\def \RGBlightyellow {FFFFE0}
\def \RGBlime {00FF00}
\def \RGBlimegreen {32CD32}
\def \RGBlinen {FAF0E6}
\def \RGBmagenta {FF00FF}
\def \RGBmaroon {800000}
\def \RGBmediumaquamarine {66CDAA}
\def \RGBmediumblue {0000CD}
\def \RGBmediumorchid {BA55D3}
\def \RGBmediumpurple {9370DB}
\def \RGBmediumseagreen {3CB371}
\def \RGBmediumslateblue {7B68EE}
\def \RGBmediumspringgreen {00FA9A}
\def \RGBmediumturquoise {48D1CC}
\def \RGBmidnightblue {191970}
\def \RGBmintcream {F5FFFA}
\def \RGBmistyrose {FFE4E1}
\def \RGBmoccasin {FFE4B5}
\def \RGBnavajowhite {FFDEAD}
\def \RGBnavy {000080}
\def \RGBoldlace {FDF5E6}
\def \RGBolive {808000}
\def \RGBolivedrab {6B8E23}
\def \RGBorange {FFA500}
\def \RGBorangered {FF4500}
\def \RGBorchid {DA70D6}
\def \RGBpalegoldenrod {EEE8AA}
\def \RGBpalegreen {98FB98}
\def \RGBpaleturquoise {AFEEEE}
\def \RGBpalevioletred {DB7093}
\def \RGBpapayawhip {FFEDB5}
\def \RGBpeachpuff {FFDAB9}
\def \RGBperu {CD853F}
\def \RGBpink {FFC0CB}
\def \RGBplum {DDA0DD}
\def \RGBpowderblue {B0E0E6}
\def \RGBpurple {800080}
\def \RGBred {FF0000}
\def \RGBrosybrown {BC8F8F}
\def \RGBroyalblue {4169E1}
\def \RGBsaddlebrown {8B4513}
\def \RGBsalmon {FA8072}
\def \RGBsandybrown {F4A460}
\def \RGBseagreen {2E8B57}
\def \RGBseashell {FFF5EE}
\def \RGBsienna {A0522D}
\def \RGBsilver {C0C0C0}
\def \RGBskyblue {87CEEB}
\def \RGBslateblue {6A5ACD}
\def \RGBslategray {708090}
\def \RGBsnow {FFFAFA}
\def \RGBspringgreen {00FF7F}
\def \RGBsteelblue {4682B4}
\def \RGBtan {D2B48C}
\def \RGBteal {008080}
\def \RGBthistle {D8BFD8}
\def \RGBtomato {FF6347}
\def \RGBturquoise {40E0D0}
\def \RGBviolet {EE82EE}
\def \RGBwheat {F5DEB3}
\def \RGBwhite {FFFFFF}
\def \RGBwhitesmoke {F5F5F5}
\def \RGByellow {FFFF00}
\def \RGByellowgreen {9ACD32}

% SCRIPT: =====

\def \SCRIPTcheck #1#2{\ifx \SCRIPTtest #1\let \SCRIPTnext #2\fi}

\def \SCRIPTend {\csname \CONtag / \CONscript \endcsname \shredder}

\def \SCRIPThandle {% Fixme: does not work within tables due to backup mode
\ifx \SCRIPTtype \CONtextTeX \MSGobjectTeX
\endmarkupCAT \tenrm \parindent = 20pt
\ifx \URLhref \empty \else \FILEsearch \URLhref \ifFILEexist
\group \let \end \relax \input \URLhref \relax \egroup
\else \CHKerrabsent \URLhref \fi \fi
\else \SCRIPTtotherttrue \LINKunrenderedtrue % Fixme: word 'script'
\CATwhitespaceg \CATwhitespaceslash \fi}% fools '</script>' tag

\def \SCRIPTinit {\let \SCRIPTtype \empty \let \URLhref \empty}

\ldef \SCRIPTparse #1 {\lowercase {\edef \SCRIPTtest {#1}}%
\let \SCRIPTnext \SCRIPTparse
\SCRIPTcheck \CONscript \SCRIPTend % fooled by word "script" in scripts
\SCRIPTcheck \CONendscript \SCRIPTend
\SCRIPTcheck \CONendcommentendscript \SCRIPTend
\SCRIPTcheck \CONendcomment \shredder \SCRIPTnext}

```

```

\def \SCRIPTset {\edef \SCRIPTtype {\CONTEXT /\the \VAL}}

% SHEET: =====

% Access CSS style sheet:

\let \SHTlast \empty

\def \SHTload {\ifx \LINKmedia \empty \let \SHTnext \shtA \else
\ifx \LINKmedia \CONprint \let \SHTnext \shtA \fi \fi}
% Fixme: better in \FILEget with respect to relative paths
\def \shtA {\URLinternetfalse
\expandafter \urlB \URLhref ://\relax
\expandafter \urlG \URLhref ://:\relax
\let \CHNfolderbackup \CHNfoldercurrent \let \CHNfoldercurrent \empty
\shtAC \shtAB \shtAA \shtAD \relax % if ".css" missing
\shtAA \shtAarelat \shtAaparse
\glet \CHNfoldercurrent \CHNfolderbackup \global \CSSaddedfalse}
\def \shtAA #1#2{\ifx \SHTloadtest \empty #1%
\else \let \SHTloadnext #2\fi \SHTloadnext} % any
\def \shtAaparse {\CSSget \SHTloadtest}
\def \shtAarelat {\let \SHTloadnext \relax}
\def \shtAB {\ifFILEexist \glet \SHTnew \SHTloadprobe \fi
\let \SHTloadtest \empty \shtAE} % any file
\def \shtAC {\ifURLinternet \let \SHTloadnext \shtACone
\else \let \SHTloadnext \shtACTwo \fi \SHTloadnext} % internet
\def \shtACone {\let \FILENAME \empty
\let \FILEpath \empty \let \FILESourcelast \empty
\edef \SHTloadprobe {..\URLhref /}%
\expandafter \fileD \SHTloadprobe \relax % Fixme: rename
\let \SHTloadprobe \FILENAME} % internet one
\def \shtACTwo {\OPENobjecttrue % internet two
\CHNfile \URLhref \CHNloadnextfalse \OPENobjectfalse
\edef \SHTloadprobe {\CHNfoldercurrent /\chainNEXTfile}}
\def \shtAD {\edef \SHTloadprobe {\SHTloadprobe .css}% \empty
\let \SHTloadnext \shtAE
\gadvance \chkWLMtotfile by -1} % compensation if loaded twice
\def \shtAE {\ifx \SHTlast \SHTnew \else \shtAEA \fi} % file
\def \shtAEA {\glet \SHTlast \SHTnew \chkWLMtotfileadvance
\ifFILEtoomany \else \shtAEA \fi}
\def \shtAEA {message {\SHTloadprobe}}%
\openin \FILEplaintext = \SHTloadprobe \relax
\loop \FILEcheckeof \FILEplaintext
\ifFILEnotended \read \FILEplaintext to \myline
\edef \SHTloadtest {\SHTloadtest \myline}%
\repeat \closein \FILEplaintext}

\let \SHTnext \empty

\def \SHTstylebegin {\let \LINKmedia \empty}
\def \SHTstyleclear {\ifCSSadded \glet \SHTlast \empty \fi}
\def \SHTstyleend {\ifx \LINKmedia \CONscreen \else \SHTstyleclear
\edef \expandedGAP {\the \GAP}\CSSget \expandedGAP \GAP = {} \fi}

% SIZE: =====

\nec \SIZE
\def \SIZEbase
\def \SIZEbaseadvance #1{%
\ifnum \SIZEbase < 7 \advance \SIZEbase by #1\fi}
\def \SIZEbasediminish #1{%
\ifnum \SIZEbase > 1 \advance \SIZEbase by -#1\fi}
\def \SIZEbaselarger {\SIZEbaseadvance 1}
\def \SIZEbaseset {\SIZEbase = \the \VAL}% Fixme: needs lowercase
\def \SIZEbasesmaller {\SIZEbasediminish 1}

\def \SIZEchange #1% Get default font size:

\lowercase {\edef \SIZEchangeZ {#1}}%
\ifx \SIZEchangeZ \CONlarger \SIZEbaselarger \else
\ifx \SIZEchangeZ \CONsmaller \SIZEbasesmaller \else
\ifx \SIZEchangeZ \CONxxlarge \SIZEbase = 6 \else
\ifx \SIZEchangeZ \CONxlarge \SIZEbase = 5 \else
\ifx \SIZEchangeZ \CONlarge \SIZEbase = 4 \else
\ifx \SIZEchangeZ \CONmedium \SIZEbase = 3 \else
\ifx \SIZEchangeZ \CONsmall \SIZEbase = 2 \else
\ifx \SIZEchangeZ \CONxsmall \SIZEbase = 1 \else
\ifx \SIZEchangeZ \CONxxsmall \SIZEbase = 0 \else
\VALsetdim \SIZE \SIZE \fi \fi \fi \fi \fi \fi \fi \fi \fi \fi
\KRNcheck \KRNdonefalse} % Fixme: time bandit

\ned \SIZEcurrent

\new \ifSIZEgiven

% Magnification:
% The font size can be set via <basefont size="n"> (deprecated) with n=1 to 7,
% 3 = medium, using CSS2 magnification steps (scaling factor = 1.2).
% This should be done if <big>, <small>, <basefont size="n"> or
% <font size="n"> are executed:

\def \SIZEmagnify {\divide \SIZEcurrent by \CONnummedium
\ifcase \SIZEbase \SIZEtempo = \CONnumxxsmall \or % 0
\SIZEtempo = \CONnumxsmall \or % 1
\SIZEtempo = \CONnumsmall \or % 2
\SIZEtempo = \CONnummedium \or % 3
\SIZEtempo = \CONnumlarge \or % 4
\SIZEtempo = \CONnumxlarge \or % 5
\SIZEtempo = \CONnumxxlarge \or % 6
\SIZEtempo = \CONnumxxxlarge \or % 7
\SIZEtempo = \CONnumxxxxlarge % 8
\else \SIZEmagnifyA \fi \multiply \SIZEcurrent by \SIZEtempo}
\def \SIZEmagnifyA {\ifnum \SIZEbase < 0 \SIZEtempo = \CONnumxxsmall
\else \SIZEtempo = \CONnumxxxxlarge \fi}

\def \SIZEset {\VALset \SIZEsetnew
\ifx \SIZEsetold \SIZEsetnew \else \let \SIZEsetold \SIZEsetnew
\SIZEchange {\the \VAL}\SIZEgiventruetrue \fi}
\let \SIZEsetnew \empty
\let \SIZEsetold \empty

\def \SIZEswitch #1{\edef \SIZEswitchZ {#1}}%
\ifx \SIZEswitchZ \CONnumone \SIZEbaseset \else
\ifx \SIZEswitchZ \CONnumtwo \SIZEbaseset \else
\ifx \SIZEswitchZ \CONnumthree \SIZEbaseset \else
\ifx \SIZEswitchZ \CONnumfour \SIZEbaseset \else
\ifx \SIZEswitchZ \CONnumfive \SIZEbaseset \else
\ifx \SIZEswitchZ \CONnumsix \SIZEbaseset \else
\ifx \SIZEswitchZ \CONnumseven \SIZEbaseset \else
\ifx \SIZEswitchZ \CONnumpone \SIZEbaseadvance 1\else
\ifx \SIZEswitchZ \CONnumtwo \SIZEbaseadvance 2\else
\ifx \SIZEswitchZ \CONnumthree \SIZEbaseadvance 3\else
\ifx \SIZEswitchZ \CONnumfour \SIZEbaseadvance 4\else
\ifx \SIZEswitchZ \CONnumone \SIZEbasediminish 1\else
\ifx \SIZEswitchZ \CONnumtwo \SIZEbasediminish 2\else
\ifx \SIZEswitchZ \CONnumthree \SIZEbasediminish 3\else
\CHKerrinvalidsize #1}%
\fi \fi \fi \fi \fi \fi \fi \fi \fi \fi \fi \fi \fi \fi \fi \fi

\nec \SIZEtempo
\nec \SIZEtempotwo

% SPAN: =====

% Fixme: "rowspan" attribute must be handled before "class"

% Fixme: If attributes and properties were scanned before "executing" tags,

```

```

% table layout information like row- and colspans would be accessible easier.

\def \SPANcol {\CONumone}
\def \SPANcolnoadvance {\advance \NOCOL by 1}
\def \SPANcolxexecdo {\xDEF {%
\CONcolwidth \the \TABLEnesting.\the \NOCOL} {\the \SPANcolgroupwidth}}
\def \SPANcollast {0}
\new \ifSPANcolgiven
\def \SPANcolgroupexec {\ifx \SPANcol \empty \else \SPANcolgroupexecdo \fi}
\def \SPANcolgroupexecdo {{%
\ifSPANcoltable \ifnum \SPANcol > 0 \divide \hsize by \SPANcol
\xDEF \SPANcolgroupwidth {\the \hsize}\fi \fi}}
\def \SPANcolgroupinit {%
\gLET \SPANcol \empty \gLOBAL \SPANcoltablefalse \NOCOL = 0}
\def \SPANcolgroupset {%
\gLOBAL \SPANcoltabletrue \VALgetint \xDEF \SPANcol {\VALinteger}}
\let \SPANcolgroupwidth \empty
\def \SPANcolpost #1{\VALgetint
\ifnum \VALinteger > 0 \xDEF \NOCOLLAST {\the \NOCOL}%
\gADvANCE \NOCOL by -1 \gADvANCE \NOCOL by \VALinteger
\multiply #1 by \VALinteger \xDEF \SPANcollast {\VALinteger}\fi}
\new \ifSPANcoltable

\def \SPANlayout {\DEFsearch {TCOLS\the \NOTable}\ifDEF % Fixme: flatten
\xDEF \SPANlayoutttest {\csname TCOLS\the \NOTable \endcsname}\fi
\DEFsearch {TROWS\the \NOTable}\ifDEF
\xDEF \SPANlayoutttest {\csname TROWS\the \NOTable \endcsname}\fi}

\def \SPANrowclear {\count 8 = 0
\loop \ifnum \count 8 < \BKPTdmax \advance \count 8 by 1
\xDEF {attSPANrowspan\the \TABLEnesting.\the \count 8} {0}% Fixme: empty
\xDEF {\CONcolwidth \the \TABLEnesting.\the \count 8} \empty \repeat}
\def \SPANrowhfill {\ifSPANrowspangiven \hfill \hfill \fi \par}
\def \SPANrowpost #1#2{\ifnum \NOCOLLAST > 0 \gADvANCE \NOCOL by -1 \fi
\gLOBAL \SPANrowspangiventruE \SPANrowspangivenheretruE
\ifx \SPANcolgroupwidth \empty #1\else \SPANcolxexecdo \fi
{\VALgetint \NOROWSPAN = \VALinteger \advance \NOROWSPAN by \NOROW
\xDEF {attSPANrowspan\the \TABLEnesting.\the \NOCOL} {\the \NOROWSPAN}}%
\let #2\vss}
\def \SPANrowrest {\ifnum \TABLEnesting = 1 \SPANrowspangivenfalse \fi}
\new \ifSPANrowspangiven \new \ifSPANrowspangivenhere

% STACK: =====
% A "stack" is a data structure like {cherry/peach/apple/plum/}. Item "plum"
% is the last one that has been pushed onto it, and it would be the first to
% be popped from it; compare Wolfgang Appelt: TeX fuer Fortgeschrittene, pages
% 52 to 54.

% Type A used for chains (directories):

\lDEF \STACKpop #1\from #2{\xDEF \STACK {#2}%
\ifx \STACK \empty \let #1\empty \else
\DEF \STACKparse ##1/##2\relax {\xDEF #1{##2}\xDEF #2{##1}}%
\expandafter \STACKparse \STACK \relax \fi}

\lDEF \STACKpush #1\to #2{\xDEF #2{#2/#1}}% #1: item, #2: stack

% Type B used for nested tables:

\lDEF \STACKxpop #1\from #2{\xDEF \STACKx {#2}%
\ifx \STACKx \empty \let #1\empty \else
\DEF \STACKxparse ##1/##2\relax {\xDEF #1{##1}\xDEF #2{##2}}%
\expandafter \STACKxparse \STACKx \relax \fi}

\DEF \STACKxpopnumber #1\from #2{% #1: count register
\STACKxpop \STACKxprobe \from #2%
\ifx \STACKxprobe \empty \DEF \STACKxprobe {0}\fi
\gLOBAL #1 = \STACKxprobe

\xDEF \STACKxprobe {\the #1}} % dummy expand

\lDEF \STACKxpush #1\to #2{\xDEF #2{#1/#2}}% #1: item, #2: stack

% TABLE: =====

\DEF \TABLEbegin #1{% <table>
\ifmode \hss \par \LINEupskip \fi % Fixme: may cause overlap
\ vbox \bgroup \moveright #1\hbox \bgroup \raise \LINEheight
\ vtop \bgroup \vskip \MARGIN % align cells of different heights
\hbox \bgroup \hskip \MARGIN \ifdim \BRDw > 0mm \BRDupdate \fi
\BRDdraw \CONbegin \CONtable \PDGbegin \vbox \bgroup}
\DEF \TABLEend {\ifOPENTable \hfill \TABLEendA \fi} % </table>
\DEF \TABLEendA {\egroup \PDGend \BRDdraw \CONend \CONtable
\hskip \MARGIN \egroup \vskip \MARGIN \egroup \egroup \egroup}

\DEF \TABLEinit {% <table>
\SPANrowspangivenfalse \NORow = 0 \gADvANCE \NOTable by 1
\let \TABLEalign \CONcenter \let \TABLEsummary \empty
\ifOPENp \else \TEXTindent = 0mm \fi
\ifnum \TABLEnesting = 1 \gGLOBAL \TABLEswapheight = 0mm \fi
\TABLEheight = 0mm \let \BRDprofile \empty \BRDreset \let \SPANcol \empty}

\new \ifTABLEenter
\new \TABLEheight
\new \TABLEnesting

\DEF \TABLErowbegin #1{% split at <tr>
\ifTABLEenobreak \else \ifSPANrowspangiven \else
\ifOPENThead \else \ifOPENTbody \else \ifOPENTfoot \else
\ifnum \TABLEnesting = 1 \ifnum \NORow > 1 % minimal split rows
\TABLErowbeginA #1\fi \fi \fi \fi \fi \fi}
% split table, imitatang \TABLEendA and \TABLEbegin for page break:
\DEF \TABLErowbeginA #1{% interrow split
\xDEF \interNORow {\the \NORow}\xDEF \interhsize {\the \hsize}%
\egroup \gGLOBAL \TABLEentertrue \PDGend \BRDdraw \CONend \CONtable
\hskip \MARGIN \egroup \egroup \egroup \egroup
\ vbox \bgroup \LINEheight = 0pt \moveright #1\hbox \bgroup
\ vtop \bgroup \hbox \bgroup \hskip \MARGIN
\ ifdim \BRDw > 0mm \BRDupdate \fi
\BRDdraw \CONbegin \CONtable \PDGbegin \gGLOBAL \TABLEenterfalse
\ vbox \bgroup \NORow = \interNORow \hsize = \interhsize}
\DEF \TABLErowend #1#2{% swap at <tr>
\ifOPENp \ifnum \TABLEnesting = 1 #1\else
\ ifdim #2 > \LINEheight \gADvANCE \TABLEheight by #2\fi \fi
\ else \gADvANCE \TABLEswapheight by #2%
\ gADvANCE \TABLEswapheight by \LINEheight \fi \fi}

\new \TABLEswapheight

% TAG: =====
% A document is composed of declarations, elements, comments, references,
% and instructions, all of which are indicated by markup.
% The markup input looks like this: ...<tag x>gap x<tag y>gap y<tag z>...
% Thus, it can be 'shreddered' into a sequence of tags and gaps.
% A tag is thought to be of the form "<element attributelist>".

\new \TAG

\lDEF \TAGget #1>{\TAG = {#1}}\TAGgetC #1 >{\CATactivelessthan
\ifnum \TAGgetA = 0 \TAGgetB \fi \ATTuncmessagefalse
\ifGAPobey \OBEYspacesandlines \fi \tagG}
\new \TAGgetA
\DEF \TAGgetB {\beginmarkupCAT \nonstopmode \gGLOBAL \TAGgetA = 1}
\lDEF \TAGgetC #1#2 #3>{% Fixme: used globally
\lowercase {\ELEMENT = {#1#2}}%

```

```

\edef \TAGgetCX #1\def \TAGgetCXY #1#2%
\def \TAGgetCY #2\def \TAGgetCZ #3\ATTLlist = #3%
\let \TAGgetCnext \tagE
\tagB \TAGgetCXY \TAGgetCZ \TAGgetCnext % three args
\ifOPENobject \ifOPENtable \else \BKppush \TAGgetCnext \fi
\else \BKppush \TAGgetCnext \fi \TAGgetCnext}
\new \iftagA \new \iftagAbeg % tag comment begun
\def \tagB #1#2#3% check comment
\expandafter \tagBA #1\empty \empty \empty \relax
\iftagA \expandafter \tagC #2\relax #3\fi}
\ldef \tagBA #1#2#3#4\relax {\edef \tagAbegintest {#1#2#3} run
\ifx \tagAbegintest \CONcommentbegin
\tagAtrue \tagAbegtrue \tagDfalse \tagBAA #4\relax
\else \tagAfalse \fi} % Fixme: one cannot say this here
\ldef \tagBAA #1-#2\relax {\edef \tagBAAZ {#2}% finish
\ifx \tagBAAZ \CONdblhyph \tagDtrue \tagAbegfalse \fi}
\ldef \tagC #1-#2\relax #3{\edef \tagCZ {#2}% parse
\ifBKP \else \ifBKPop \else \GAPwrite \fi \fi
\ifx \tagCZ \CONcommentend \tagCD #3%
\else \def \tagAfirst {#1}% Fixme: outsource
\ifx \tagCZ \empty \ifx \tagAfirst \empty \iftagD
\else \let #3\tagCB \fi \else \let #3\tagCB \fi
\else \ifOPENstyle \else \ifOPENscript \else \tagCA #3\fi \fi \fi}
\def \tagCA #1% run
\ifx \tagAfirst \empty \iftagD \let #1\tagCC \else \let #1\tagCAA \fi
\else \ifx \tagCZ \empty \let #1\tagCAA \fi \fi}
\ldef \tagCAA #1->{\tagCC} % finish
\ldef \tagCB #1->{\tagCC \GAPget} % dummy for <tag> inside comment
\def \tagCC {\tagAbegfalse
\tagDtrue \tagAfalse \beginmarkupCAT \shredder} % finished
\def \tagCD #1{\tagDtrue \tagAbegfalse
\ifOPENstyle \else \ifBKP \else \let #1\GAPwrite \fi \fi} % stop
\new \iftagD % comment finished
\def \tagD #1{\expandafter \tagDA #1\empty \empty \empty \relax}
\ldef \tagDA #1#2#3#4\relax {\edef \tagDatest {#1#2#3}
\ifx \tagDatest \CONendcomment \tagAfalse \tagAbegfalse \tagDtrue \fi}
\def \tagE {\PRPfalse \ifx \TAGgetCZ \empty \ATTfalse \else \ATTtrue \fi
\DEFsearch {\CONtag \the \ELEMENT}}%
\ifDEF \tagEA \else \expandafter \tagEB \the \ELEMENT
\empty \empty \empty \empty \relax \fi \tagEnext}
\def \tagEA {\def \tagEnext {\csname \CONtag \the \ELEMENT \endcsname}}
% Detect <html:xxx> and <?xml...> tags % 2007/03/08
\lowercase {\def \tagEB #1#2#3#4#5#6\relax {%
\edef \tagEBrest {#2#3#4#5#6}%
\edef \tagEBtwofour {#2#3#4}\edef \tagEBsix {#6}%
\ifx \tagEBtwofour \CONxml \let \tagEBsix \CONxml \fi % skip \empty test
\ifx \tagEBsix \empty \tagF \else \DEFsearch {\CONtag #6}%
\ifDEF \def \tagEnext {\csname \CONtag \tagEBsix \endcsname}%
\ELEMENT = \expandafter {\tagEBsix}%
\else \DEFsearch {\CONtag \tagEBrest}%
\ifDEF \def \tagEnext {\csname \CONtag \tagEBrest \endcsname}%
\ELEMENT = \expandafter {\tagEBrest}\else \tagF \fi \fi}}
\def \tagF {\ifx \TAGgetCX \CONexclamation
\def \tagEnext {\tagFA \TAGgetCY \TAGgetCZ}% <! ...
\else \def \tagEnext {\tagFB \TAGgetCX \tagFC}\fi}
% Do not execute tag if it starts with exclamation or question mark:
\def \tagFA #1#2% <! ...
\let \tagEAnotexclamnext \relax \ifx #1\CONdblhyph \tagFAA #2%
\else \ifx #1\CONdoctype \chkDCTPget #2\fi \fi \tagEAnotexclamnext}
\def \tagFAA #1{\ifOPENstyle \tagFAAA #1\fi} % <!--
\def \tagFAAA #1{\ifx \LINKmedia \CONscreen \else
\def \tagEAnotexclamnext {\CSSget #1}\fi}
\def \tagFB #1#2{\ifx #1\CONquotation \relax % <? ...
\ifOPENbody \GAPwrite \fi \else #2\GAPwrite \fi}
\def \tagFC {\ifx \TAGgetCX \CONslash % unrecognized
\ifx \TAGgetCY \empty \else \Message {\TAGgetCX \TAGgetCY}\fi
\else \CHKerrunrecognized {\TAGgetCX \TAGgetCY} \fi}
\def \tagG {\ifEND \tagGB \else \tagH \fi \tagGnext}% check tag
\def \tagGA {\let \tagGnext \relax} % relax
\def \tagGB {\ifOPENobject \tagGA \else \tagGBA \fi} % object
\def \tagGBA {\let \tagGnext \CHKerrwellformedness} % end
\def \tagH {\ifCHNloadnext \tagHA \else \tagI \fi} % chain
\def \tagHA {\ifOPENobject \tagHAA \else
\ifnum \NOinput < 2 \tagHAA \else \tagGA \fi \fi} % input
\def \tagHAA {\def \tagGnext {\CHNget \CHKerrwellformedness}}
\def \tagI {\ifOPENscript \tagJ \else \tagIA \fi}
\def \tagIA {\CATactivelessthan \let \tagGnext \GAPget} % gap
\def \tagJ {\ifSCRIPTother \tagJA \else \tagJB \fi}
\def \tagJA {\endmarkupCAT \beginscriptCAT
\let \tagGnext \SCRIPTparse} % script
\def \tagJB {\tagGA \ifx \SCRIPTtype \CONtextTeX
\ifx \LINKhref \empty \CATotherlessthan \fi \fi} % none
% UNICODE: =====
% Every known character on earth shall be given a unique code number,
% usually expressed in hexadecimal form, with digits running 0 to 9
% and from A to F. The data of (at most) 256 characters is defined in
% slot files (UXXXX.dat), which are excerpts of the unicode database
% files like UnicodeData.txt.
% - The slot file format is: < 1) unicode point, four hexadecimal
% digits; 2) unicode character name; 3) general category;
% 4) canonical combining classes; 5) bidirectional category;
% 6) character decomposition mapping; 7) decimal digit value;
% 8) digit value; 9) numeric value; 10) mirrored; 11) unicode
% character name, version 1.0; 12) ISO 10646 comment field;
% 13) uppercase mapping; 14) lowercase mapping; 15) titlecase
% mapping >.
% - Only columns 1, 3, 13 and 14 are needed for Markup Shredder. With
% exception of U0000.dat, columns 2, 11 and 12 may be deleted in the
% slot files in order to reduce their sizes by 50 percent. If you
% want to know the character names, for instance, or if you need more
% slots for other alphabets, have a look at the source files
% (http://www.unicode.org/Public/UNIDATA/UnicodeData.txt and others).
\ldef \UCDadd #1#2.#3 % add unicode data slot file to UCDSlots list
\def \UCDaddy #2\ifx \UCDaddy \empty \else
\advance \UCDnumberofslots by 1 % count slots
\uppercase {\UCDslot = #2}% get decimal representation of #2
\UCDslotregbase = 128 % get base register of unicode slot
\advance \UCDslotregbase by -\UCDnumberofslots
\multiply \UCDslotregbase by 256
\uppercase {\edef \UCDadditem {\the \UCDnumberofslots} [#2]
[\the \UCDslot] [\the \UCDslotregbase]}% build item
\expandafter \ARRAYfour \UCDadditem \ARRAYafter \UCDSlots \fi} % and list
\ifx \eTeXversion \undefined \ldef \UCDadd #1#2.#3 {} \fi
\nec \UCDcat
\net \UCDdata \UCDdata = {0;0;0.0pt;0.0pt;0;0;} % "pt" means "protruding"
\ldef \UCDdataget #1;#2;#3;#4;#5;#6;#7;#8;#9\relax %
% #1: catcode, #2: lccode, #3: uccode, #4: lpcode rm/it, #5: rpcode rm/it,
% #6: codepage name, #7: codepage index, #8: entity name, #9: glyph name
\UCDcat = #1 \UCDlc = #2 \UCDuc = #3 \UCDlp = #4 \UCDrp = #5
\CPGname = \expandafter {#6} \CPGindex = #7
\UCDentityname = \expandafter {#8} \UCDglyphname = \expandafter {#9}}
\ldef \UCDdataset #1;#2;#3;#4;#5;#6;#7;#8;#9\relax %
% #1: catcode, #2: lccode, #3: uccode, #4: lpcode rm.it, #5: rpcode rm.it,
% #6: codepage name, #7: codepage index, #8: entity name, #9: glyph name
\expandafter \UCDdataget \the \UCDdata \relax \if #1\else \UCDcat = #1\fi
\if #2\else \UCDlc = #2\fi \if #3\else \UCDuc = #3\fi
\if #4\else \UCDlp = #4\fi \if #5\else \UCDrp = #5\fi
\if #6\else \CPGname = \expandafter {#6} \fi \if #7\else \CPGindex = #7\fi
% \if fails for names like aa... or cc...
\edef \UCDdatasetY {#8}\edef \UCDdatasetZ {#9}%
\ifx \UCDdatasetY \CONhyphen \else \UCDentityname = \expandafter {#8}\fi
\ifx \UCDdatasetZ \CONhyphen \else \UCDglyphname = \expandafter {#9}\fi

```

```

\edef \UCDdatasetA {\the \UCDcat ;\the \UCDlc ;\the \UCDuc ;%
\the \UCDlp ;\the \UCDrp ;\the \CPGname ;\the \CPGindex ;%
\the \UCDentityname ;\the \UCDglyphname }%
\UCDdata = \expandafter {\UCDdatasetA}

\net \UCDentityname

\def \UCDget {\UCDgetA {\the \UCDslotpointreg }%
\UCDdata = \expandafter {\the \toks \UCDslotpointreg }%
\edef \UCDgettest {\the \UCDdata}%
\ifx \UCDgettest \empty \UCDglyphname = {.notdef}%
\else \expandafter \UCDdataget {\the \UCDdata \relax \fi}
\def \UCDgetA #1{% set base register number of unicode slot point
\UCDslotregbase = #1 \divide \UCDslotregbase by 256
\multiply \UCDslotregbase by 256 }

\net \UCDglyphname

\nec \UCDlc \nec \UCDuc \ned \UCDlp \ned \UCDrp

\nec \UCDnumberofslots

\nec \UCDpointindex % 0 ... 255
\def \UCDpointindexset #1{% set index of unicode point (0 ... 255)
\UCDpointindex = #1 \advance \UCDpointindex by -\UCDslot}

\nec \UCDslot % 0 ... 65536
\ldef \UCDslotload #1#2#3#4{% read slot list completely
\bgrouper {count 0 = #4 \advance \count 0 by 255
\edef \UCDslotloadtest {\the \count 0}\message {- #1}}%
\lowercase {\message {[u#2.row]}}%
\ifx \GMSdisplayregs \undefined \else
\message {(T#4 to)\message {T\UCDslotloadtest}}\fi \egroup
\MSGunicodedefined {#2name1}\MSGunicodedefined {#2name2}%
\UCDslotpointreg = #4 \lowercase {\openin 1 = u#2.row}%
\loop \FILEcheckeof 1 \ifFILEnotended read 1 to \myline
\ifx \myline \CONpar \else \expandafter
\ucdA \myline ;;;;;;\relax \fi \repeat \closein 1 }
\ifx \eTeXversion \undefined \let \UCDslotload \relax \fi
\ldef \ucdA #1;#2;#3;#4;#5;#6;#7;#8\relax {% read line from unicode table
% #1: Code point, #2: Character name, #3: General Category,
% #4: Canonical Combining Classes, #5: Bidirectional Category,
% #6: Character Decomposition Mapping, #7: Decimal digit value,
% #8: (second half of data line)
\UCDslotpointregset {"#1}\edef \ucdAS {#1}\edef \ucdAU {#3}%
\let \ucdAcat \CATother
\ifx \ucdAU \CONlu \let \ucdAcat \CATletter \fi
\ifx \ucdAU \CONll \let \ucdAcat \CATletter \fi
\ifx \ucdAU \CONlt \let \ucdAcat \CATletter \fi
\ifx \ucdAU \CONlm \let \ucdAcat \CATletter \fi
\ifx \ucdAU \CONlo \let \ucdAcat \CATletter \fi
\ifx \ucdAU \CONzs \let \ucdAcat \CATwhitespace \fi
\ifx \ucdAS \CONpar \else \uppercase {\toks \UCDslotpointreg = {"#1}}\fi
\ucdB #8;;;;;;\relax}
\ldef \ucdB #1;#2;#3;#4;#5;#6;#7;#8\relax {%
% #1: Digit value, #2: Numeric value, #3: Mirrored,
% #4: Unicode 1.0 Name, #5: 10646 comment field,
% #6: Uppercase Mapping, #7: Lowercase Mapping, #8: Titlecase Mapping
\edef \ucdBS {#1}\ifx \ucdBS \CONpar \else
\expandafter \UCDdataset \ucdAcat;"0#7;"0#6;";";";";";\relax
\global \toks \UCDslotpointreg = \expandafter {\the \UCDdata } \fi}

\nec \UCDslotpointreg
\def \UCDslotpointregset #1{% register number of slot point
\UCDslotpointreg = \csname \the \UCDslot slot \endcsname
\advance \UCDslotpointreg by #1 \advance \UCDslotpointreg by -\UCDslot}
\nec \UCDslotregbase \let \UCDslots \empty
\def \UCDslotset #1{% Standard Unicode Value (decimal): set slot
\UCDslot = #1 \divide \UCDslot by 256 \multiply \UCDslot by 256 }

\def \UCDwrite {\UCDwriteenctruer \UCDwriteslot \UCDwriteencfalse}
\new \ifUCDwriteenc
\ldef \UCDwriteline #1\immediate \backupwrite 3{#1}%
\def \UCDwriteslot {\def \ \ % read all Unicode slots found on UCDSlots list
[#1] % number of UCD slot file (1, 2, 3, ... , 127)
[#2] % UCD slot number, hex. (0000, 0100, 0200, ... , FF00)
[#3] % UCD slot number, dec. (0, 256, 512, ... , 65280)
[#4] {% baseno. of UCD slot regs. (32512, 32256, 32000, ... , 256)
\edef \second {#2}\UCDslot = #3
\ifx \second \empty \else \xDef \the \UCDslot slot {#4}%
\ifUCDwriteenc \ucdC {#1}{#2}{#3}{#4}\else
\UCDslotload {#1}{#2}{#3}{#4}\fi \fi}%
\ifUCDwriteenc \else
\message {- Installed rows of the Unicode character space:}\fi \UCDslots}
\ifx \eTeXversion \undefined \let \UCDwriteslot \relax \fi
\def \ucdC #1#2#3#4{\ucdE {#1}{#2}{#3}{#4}\ucdD {#1}{#2}{#3}{#4}}
\def \ucdD #1#2#3#4{% use Adobe glyph names
\immediate \openout 3 g#2.enc\message {g#2.enc}%
\UCDwriteline {/g#2-Encoding []}\UCDslotregbase = #4
\advance \UCDslotregbase by 256 % now reftopofUCDslot
\UCDslotpointreg = #4 \count 8 = "#2
\loop \ifnum \UCDslotregbase > \UCDslotpointreg
{\UCDget \HEXinput = \the \count 8 \HEXoutput = {} \HEX \ucdLine }%
\advance \UCDslotpointreg by 1 \advance \count 8 by 1
\repeat \UCDwriteline {} \def}%
\CPGencwriteline {\CONpercent \CONspace Generated by
\GMSname \CONspace (\GMSdomain)}\immediate \closeout 3 }
\def \ucdLine {\edef \testglyphname {\the \UCDglyphname}%
\ifx \testglyphname \empty \def \testglyphname {.notdef}\fi
\UCDwriteline {/\testglyphname \CONspace \CONpercent
\CONspace \the \HEXoutput}}
\def \ucdE #1#2#3#4{% use Unicode values
\immediate \openout 3 u#2.enc\message {u#2.%}
\UCDwriteline {/u#2-Encoding []}
\UCDslotregbase = "#2 % misusing register name:
\advance \UCDslotregbase by 256 % now reftopofUCDslot
\UCDslotpointreg = "#2
\loop \ifnum \UCDslotregbase > \UCDslotpointreg
\HEXinput = \UCDslotpointreg \HEXoutput = {} \HEX
{\count 2 = \UCDslotpointreg
\ifnum \count 2 < "10 \def \myprefix {000}\else
\ifnum \count 2 < "100 \def \myprefix {00}\else
\ifnum \count 2 < "1000 \def \myprefix {0}\else
\let \myprefix \empty \fi \fi \fi
\UCDglyphname = {uni\myprefix \the \HEXoutput}\ucdEA }%
\advance \UCDslotpointreg by 1 \repeat \UCDwriteline {} \def}%
\CPGencwriteline {\CONpercent \CONspace Generated by
\GMSname \CONspace (\GMSdomain)}\immediate \closeout 3 }
\def \ucdEA {\edef \testglyphname {\the \UCDglyphname}%
\ifx \testglyphname \empty \def \testglyphname {.notdef}\fi
\UCDwriteline {/\testglyphname}} % line

% Unicode slot names: -----
\Def 0000name1 {\Controls-0, Basic \CONlatin \CONspace (ISO-646/ASCII),}
\Def 0000name2 {\Controls-1, \CONlatin-1 \CONsupplement \CONspace (ISO-8859-1)}
\Def 0100name1 {\CONlatin \CONspace \CONextended -A,}
\Def 0100name2 {\CONlatin \CONspace \CONextended -B/1}
\Def 0200name1 {\CONlatin \CONspace \CONextended -B/2, International Phonetic}
\Def 0200name2 {\Alphabet \CONextension s, Spacing Modifier Letters}
\Def 0300name1 {\Combining Diacritical Marks,}
\Def 0300name2 {\Greek}
\Def 0400name1 {\Cyrillic}
\Def 0500name1 {\Cyrillic \CONsupplement ary,}
\Def 0500name2 {\Armenian, Hebrew}
\Def 0600name1 {\Arabic}
\Def 0700name1 {\Syriac, Thaana}
\Def 0900name1 {\Devanagari, Bengali}

```

```

\Def 0A00name1 {Gurmukhi, Gujarati}
\Def 0B00name1 {Oriya, Tamil}
\Def 0C00name1 {Telugu, Kannada}
\Def 0D00name1 {Malayalam, Sinhala}
\Def 0E00name1 {Thai, Lao}
\Def 0F00name1 {Tibetan}
\Def 1000name1 {Myanmar, Georgian}
\Def 1100name1 {Hangul Jamo}
\Def 1200name1 {Ethiopic/1}
\Def 1300name1 {Ethiopic/2, Cherokee}
\Def 1400name1 {\CONcanadian \CONspace /1}
\Def 1500name1 {\CONcanadian \CONspace /2}
\Def 1600name1 {\CONcanadian \CONspace /3, Ogham, Runic}
\Def 1700name1 {Tagalog, Hanunoo, Buhid,}
\Def 1700name2 {Tagbanwa, Khmer}
\Def 1800name1 {Mongolian}
\Def 1E00name1 {\CONlatin \CONspace \CONextended \CONspace Additional}
\Def 1F00name1 {Greek \CONspace \CONextended}
\Def 2000name1 {General Punctuation, Super- and Subscripts, Currency}
\Def 2000name2 {\CONSsymbols, Combining Diacritical Marks for \CONSsymbols}
\Def 2100name1 {Letterlike \CONSsymbols,}
\Def 2100name2 {Number Forms, Arrows}
\Def 2200name1 {\CONmath al Operators}
\Def 2300name1 {\CONmisc \CONspace Technical}
\Def 2400name1 {Control Pictures, Optical Character Recognition,}
\Def 2400name2 {Enclosed Alphanumerics}
\Def 2500name1 {Box Drawing, Block Elements,}
\Def 2500name2 {Geometric Shapes}
\Def 2600name1 {\CONmisc \CONspace \CONSsymbols}
\Def 2700name1 {Dingbats, \CONmisc \CONspace \CONmath al \CONSsymbols-A,}
\Def 2700name2 {\CONSsupplement al Arrows-A}
\Def 2800name1 {Braille Patterns}
\Def 2900name1 {\CONSsupplement al Arrows-B, \CONmisc}
\Def 2900name2 {\CONmath \CONspace \CONSsymbols-B}
\Def 2A00name1 {\CONSsupplement al \CONmath al}
\Def 2A00name2 {Operators}
\Def 2B00name1 {}
\Def 2C00name1 {}
\Def 2D00name1 {}
\Def 2E00name1 {Chinese-Japanese-Korean Radicals \CONSsupplement}
\Def 2F00name1 {KangXi Radicals,}
\Def 2F00name2 {\CONideographic Description Characters}
\Def 3000name1 {CJK \CONSsymbols \CONspace and Punctuation,}
\Def 3000name2 {Hiragana, Katakana}
\Def 3100name1 {Bopomofo, Hangul \CONcompatibility, Jamo, Kanbun,}
\Def 3100name2 {Bopomofo \CONextended, Katakana Phonetic \CONextension s}
\Def 3200name1 {Enclosed CJK Letters and Months}
\Def 3300name1 {CJK Compatibility}
\Def 3400name1 {CJK Unified \CONideograph s}
\Def 3400name2 {\CONextension \CONspace A}
\Def 4E00name1 {CJK Unified \CONideograph s}
\Def A000name1 {Yi Syllables}
\Def A400name1 {Yi Radicals}
\Def AC00name1 {Hangul Syllables}
\Def F600name1 {\CONcorporateuse \CONspace \CONadobe}
\Def F700name1 {\CONcorporateuse \CONspace \CONadobe}
\Def F800name1 {\CONcorporateuse \CONspace (Apple Dingbats)}
\Def FC00name1 {\CONcorporateuse \CONspace \CONadobe}
\Def FD00name1 {\CONcorporateuse \CONspace \CONadobe}
\Def F900name1 {CJK \CONcompatibility \CONspace \CONideograph s}
\Def FB00name1 {Alphabetic \CONpresentation,}
\Def FB00name2 {Arabic \CONpresentation-A}
\Def FE00name1 {Variation Selectors, Comb. Half Marks, CJK Comp.}
\Def FE00name2 {Forms, Small Form Variants, Arabic Pres. Forms-B}
\Def FF00name1 {Halfwidth and Fullwidth Forms,}
\Def FF00name2 {Specials}
\Def 10300name1 {Old Italic, Gothic}
\Def 10400name1 {Deseret}

\Def 12300name1 {Klingon}
\Def 1D000name1 {Byzantine Musical \CONSsymbols}
\Def 1D100name1 {Musical \CONSsymbols}
\Def 1D400name1 {\CONmath al Alphanumeric \CONSsymbols}
\Def 20000name1 {CJK Unified \CONideograph s}
\Def 20000name2 {\CONextension \CONspace B}
\Def 2F800name1 {CJK \CONcompatibility \CONspace \CONideograph s}
\Def 2F800name2 {\CONSsupplement}
\Def E0000name1 {Tags}

% URL: =====
\let \URLgedcomimage \empty

% Distinguish internal and external destinations
% (handled by pdf viewer and web client):

\bgroup \CATotherhashmark \CATparameterdegree
\gdef \URLget #1{%
  \expandafter \urlA #1#\relax
  \expandafter \urlB #1://\relax
  \expandafter \urlC #1mailto:\relax
  \ifGEDCOM \expandafter \urlD #1#@.\relax \fi
  \expandafter \urlE #1news:\relax
  \expandafter \urlF #1.htm#GMS-STOP\relax
  \expandafter \urlG #1://#\relax } \egroup
\bgroup \CATotherhashmark \CATparameterdegree
\gdef \urlA #1#2{\edef \urlAY {#1}\edef \urlAZ {#2}% local
  \ifx \ANCRext \empty \urlAA \fi
  \ifx \urlAZ \CONrelax \urlAB \else \urlAC \fi \URLlocalnext } \egroup
\def \urlAA {\let \ANCRext \urlAY}%
\def \urlAB {\let \URLlocalnext \relax \edef \ANCRint {\ANCRint \urlAY}}
\def \urlAC {\ifx \urlAZ \CONhashmark \else \urlACHash \fi}
\def \urlACHash {\let \URLlocalnext \urlA \let \ANCRint \urlAZ}
\def \urlB #1://#2\relax {% internet % Fixme: used globally
  \edef \urlBZ {#1}\ifx \urlBZ \CONftp \URLinternettrue
    \else \ifx \urlBZ \CONhttp \URLinternettrue
      \else \ifx \urlBZ \CONhttps \URLinternettrue \fi \fi \fi}
\def \urlC #1mailto:#2\relax {% mail
  \def \urlCZ {#2}\ifx \urlCZ \empty \else \URLinternettrue \fi}
\def \urlD #1@#2.#3@#4\relax {% gedcom mail
  \def \urlDZ {#4}\ifx \urlDZ \COMdotat \URLinternettrue
    \xdef \ANCRext {mailto:#1@#2.#3}\glet \ANCRint \ANCRext \fi}
\def \urlE #1news:#2\relax {% news
  \def \urlEZ {#2}\ifx \urlEZ \empty \else \URLinternettrue \fi}
\bgroup \CATotherhashmark \CATparameterdegree
\gdef \urlF #1.#2htm*3#4\relax {% htm
  \def \urlFX {#2}\def \urlFY {#3}\def \urlFZ {#4}%
  \ifx \urlFX \empty \urlFA \else \urlFB \fi } \egroup
\def \urlFA {\ifx \urlFY \COML \urlFAA
  \else \ifx \urlFY \empty \urlFAA \fi \fi}
\def \urlFAA {\ifx \urlFZ \CONstopmark \else \URLinternettrue \fi}
\def \urlFB {\ifx \urlFY \CONdothtm
  \else \ifx \urlFY \CONldothtm \else \ifx \urlFY \empty
    \else \ifx \urlFZ \CONstopmark \URLinternettrue \fi \fi \fi \fi}
\def \urlG #1://#2://#3\relax {\edef \urlGX {#1}% relative - Fixme: used gl.
  \ifx \urlGX \CONfilelc \URLinternetfalse \edef \urlGX {#2}%
  \let \ANCRint \urlGX \let \URLhref \ANCRint \fi}

\let \URLhref \empty

\new \ifURLinternet

% VALUE: =====
% Handle numbers, dimensions and per cent values:

\net \VAL

```



```

\def \VALgetint {\expandthree \valiA % integer value
\expandafter {\the \VAL}\let \VALinteger \VALi}
\def \VALi {0}\new \ifVALi % integer found
\def \valiA #1{\let \VALi \empty \VALitruer \valiB #1\end
\ifVALi \else \def \VALi {0}\fi \ifx \VALi \empty \def \VALi {0}\fi}
\def \valiB #1{\edef \valiBX {#1}%
\ifx \end #1\let \VALinext \relax \else
\count 8 = 0 \let \VALiold \VALi % Fixme: reserve counter
\loop \ifnum \count 8 < 10
\valiC {\the \count 8} \advance \count 8 by 1 \repeat
\ifx \VALiold \VALi \VALifalse \fi \let \VALinext \valiB \fi \VALinext}
\def \valiC #1{\edef \valiCX {#1}%
\ifx \valiBX \valiCX \edef \VALi {\VALi \valiBX }\fi}
\def \VALgetlc {\expandafter \VALgetlcA \the \VAL \relax}
\def \VALgetlcA #1\relax {\lowercase {\edef \VALlowercase {#1}}}
\def \VALgetuc {\expandafter \VALgetucA \the \VAL \relax}
\def \VALgetucA #1\relax {\uppercase {\edef \VALuppercase {#1}}}
\def \VALgetwidows {\edef \VALgetwidowstest {\the \VAL}%
\ifx \VALgetwidowstest \CONnumone \global \widowpenalty = 9999 \else
\ifx \VALgetwidowstest \CONnumtwo \global \widowpenalty = 10000 \fi \fi}

\def \VALset #1{\edef #1{\the \VAL}} % set control sequence #1 to value
% Set dimension register #1 to value #2 given absolutely or relatively:
\bgrouper \CATotherpercent \CATcommentbar
\gdef \VALsetdim #1#2{\VALgetlc
\ifx \VALlowercase \CONauto \CHKerrrunsupattprop {"\CONauto"} \VAL = {50%}\fi
\let \valDIMunt \empty \valDIMuntrue \valDIMpxfalse
\edef \VALsetdimtest {#1#2\the \VAL \relax} | "%" is no comment here!
\expandafter \valdA \VALsetdimtest \}egrouper
\new \ifvalDIMnum % number
\new \valDIMprc % per cent
\new \ifvalDIMpx % pixel
\new \ifvalDIMunt \let \valDIMunt \empty % unit
% #1 has to be a register that will be set to #3 (per cent of #2 if #4=""):
\bgrouper \CATotherpercent \CATcommentbar
\gdef \valdA #1#2#3#4\relax {| dimension value
\def \valdAW {#1}\def \valdAX {#2}|
\def \valdAY {#3}\def \valdAZ {#4}|
\ifx \valdAZ \CONpercent \valdAA \else \valdB \fi \}egrouper
\def \valdAA % percent, e.g \VAL = {80%}
\valdAW = \valdAX \valDIMprc = \valdAY pt
\divide \valDIMprc by 65536 \divide \valdAW by 10
\multiply \valdAW by \valDIMprc \divide \valdAW by 10 \}
\def \valdB {\ifvalDIMnumtrue \expandafter \valdBA \valdAYZ \CONstopmark
\valdD \ifvalDIMnum \valdE \else \valdF \fi} % no percent
\bgrouper \CATotherpercent \CATcommentbar
\gdef \valdBA #1{\edef \valdAZ {#1}\let \valdBAA \valdBA | extract unit
\ifx \valdAZ \CONstopmark \valdBAB \else \valdC \fi \valdBAA \}egrouper
\def \valdBAB {\let \valdBAA \relax}
\def \valdC {\ifx \valdAZ \empty \else \valdCA \fi}
\def \valdCA {\ifx \valdAZ \CONdot \else % . decimal, - negative
\ifx \valdAZ \CONhyphen \else \ifx \valdAZ \CONnumzero \else
\ifx \valdAZ \CONnumone \else \ifx \valdAZ \CONnumtwo \else
\ifx \valdAZ \CONnumthree \else \valdCAA \fi \fi \fi \fi \fi \fi
\ifvalDIMnum \else \edef \valDIMunt {\valDIMunt \valdAZ }\fi}
\def \valdCAA {\ifx \valdAZ \CONnumfour \else
\ifx \valdAZ \CONnumfive \else \ifx \valdAZ \CONnumsix \else
\ifx \valdAZ \CONnumseven \else \ifx \valdAZ \CONnumeight \else
\ifx \valdAZ \CONnumnine \else
\valDIMnumfalse \fi \fi \fi \fi \fi \fi}
\def \valdD % check unit
\ifx \valDIMunt \CONpx \valDIMuntrue \valDIMpxtrue \else
\ifx \valDIMunt \CONpt \valDIMuntrue \else
\ifx \valDIMunt \CONmm \valDIMuntrue \else
\ifx \valDIMunt \CONcm \valDIMuntrue \else
\ifx \valDIMunt \CONex \valDIMuntrue \else
\ifx \valDIMunt \CONem \valDIMuntrue \else
\ifx \valDIMunt \CONin \valDIMuntrue \else
\valdDA \fi \fi \fi \fi \fi \fi \fi}

\def \valdDA % no HTML unit % Fixme: do error message
\ifx \valDIMunt \CONpc \valDIMuntrue \else
\ifx \valDIMunt \CONdd \valDIMuntrue \else
\ifx \valDIMunt \CONcc \valDIMuntrue \else
\ifx \valDIMunt \CONbp \valDIMuntrue \else
\ifx \valDIMunt \CONsp \valDIMuntrue \fi \fi \fi \fi \fi}
\def \valdE % number, e.g \VAL = {80}
\valDIMprc = \valdAY sp
\ifnum \valDIMprc < 2000 \valdEA \valdAW = \valDIMprc
\else \valdAW = 1cm \fi}
% Seems to be a number, not a dimension, so assume pixels:
\def \valdEA {\multiply \valDIMprc by 65536 }% 1pt = 65536sp
\def \valdF % dimension, e.g \VAL = {80mm}
\ifvalDIMpx \valdFA % multiply number with 1px
\else \ifvalDIMunt \valdAW = \valdAY \relax % just accept
\else \CHKerrinvalidunit \valDIMunt \fi \fi}
\def \valdFA {\edef \valdAZ {\valdAY}%
\expandafter \valdFAA \valdAZ}
\def \valdFAA #1px{\edef \valdFAAtest {#1}%
\ifx \valdFAAtest \empty \else \valdAW = #1pt \fi}
\def \VALsetint #1{\VALgetint \edef #1{\VALinteger}}
\def \VALsetglobal #1{\xdef #1{\the \VAL}}
\def \VALsetlc #1{\VALgetlc \edef #1{\VALlowercase}}
\def \VALsetrelax #1{\edef #1{\the \VAL \relax}}

\message {win-entities.} %=====
% HTML entities that are contained in Windows codepages:
\def \WINamp {26}% 38
\def \WINbdquo {84}% 8222
\def \WINbull {95}% 8226
\def \WINcirc {88}% 710
\def \WINdagger {86}% 8224
\def \WINdagger {87}% 8225
\def \WINeuro {80}% 8364
\def \WINfnof {83}% 402
\def \WINgt {3E}% 62
\def \WINhellip {85}% 8230
\def \WINlt {3C}% 60
\def \WINldquo {93}% 8220
\def \WINlsquo {8B}% 8249
\def \WINlsquo {91}% 8216
\def \WINmdash {97}% 8212
\def \WINndash {96}% 8211
\def \WINOelig {8C}% 338
\def \WINoelig {9C}% 339
\def \WINrdquo {94}% 8221
\def \WINrsquo {9B}% 8250
\def \WINrsquo {92}% 8217
\def \WINsbquo {82}% 8218
\def \WINScaron {8A}% 352
\def \WINscaron {9A}% 353
\def \WINtilde {98}% 732
\def \WINtrade {99}% 8482
\def \WINYuml {9F}% 376
\def \WINZcaron {8E}% 381
\def \WINzcaron {9E}% 382

% Unicode references that are contained in Windows codepages:
\def \WINdef #1 #2 {\Def WIN#1 {\GAPsgmapp {\char "#2\RFRrelax }}}
\WINdef 338 8C % OElig
\WINdef 339 9C % oelig
\WINdef 352 8A % Scaron
\WINdef 353 9A % scaron
\WINdef 376 9F % Yuml
\WINdef 381 8E % Zcaron
\WINdef 382 9E % zcaron

```

```

\WInDef 402 83 % fnof
\WInDef 710 88 % circ
\WInDef 8211 96 % ndash
\WInDef 8212 97 % mdash
\WInDef 8216 91 % lsquo
\WInDef 8217 92 % rsquo
\WInDef 8218 82 % sbquo
\WInDef 8220 93 % ldquo
\WInDef 8221 94 % rdquo
\WInDef 8222 84 % bdquo
\WInDef 8224 86 % dagger
\WInDef 8225 87 % Dagger
\WInDef 8226 95 % bull
\WInDef 8230 85 % hellip
\WInDef 8240 89 % permil
\WInDef 8249 8B % lsaquo
\WInDef 8250 9B % rsaquo
\WInDef 8364 80 % euro
\WInDef 8482 99 % trade

% 3. Markup #####

\echo {- Recognized attributes:} %=====

% A lot of attributes are only made known to Markup Shredder to enable the
% doctype check, without giving functional support:

\NEWattquiet abbr
\NEWatttagunsupported abbr td
\NEWatttagunsupported abbr th

\NEWattquiet accept
\NEWatttagunsupported accept form
\NEWatttagunsupported accept input

\NEWattquiet accept-charset
\NEWatttagunsupported accept-charset form

\NEWattquiet accesskey
\NEWatttagunsupported accesskey a
\NEWatttagunsupported accesskey area
\NEWatttagunsupported accesskey button
\NEWatttagunsupported accesskey input
\NEWatttagunsupported accesskey label
\NEWatttagunsupported accesskey legend
\NEWatttagunsupported accesskey textarea

\NEWattquiet action
\NEWatttagunsupported action form

\NEWatt align
\NEWatttagtemplate align applet img
\NEWatttagtemplate align caption table
\NEWatttag align col \ALGNtextbyvalue
\NEWatttagtemplate align colgroup col
\NEWatttagtemplate align div col
\NEWatttagtemplate align h1 col
\NEWatttagtemplate align h2 col
\NEWatttagtemplate align h3 col
\NEWatttagtemplate align h4 col
\NEWatttagtemplate align h5 col
\NEWatttagtemplate align h6 col
\NEWatttagtemplate align hr col
\NEWatttagtemplate align iframe img
\NEWatttagdeprecated align img {\VALsetlc \IMGalign}
\NEWatttagtemplate align input img
\NEWatttagtemplate align legend img
\NEWatttagtemplate align object img
\NEWatttagdeprecated align p \ALGNtextbyvalue

\NEWatttagdeprecatedunsupported align pre
\NEWatttagdeprecated align table {\VALset \TABLEalign}
\NEWatttagtemplate align tbody col
\NEWatttagtemplate align td col
\NEWatttagtemplate align tfoot col
\NEWatttagtemplate align th col
\NEWatttagtemplate align thead col
\NEWatttagtemplate align tr col

\NEWattquiet alink % active link color unused
\NEWatttagdeprecated alink body \empty

\NEWatt alt
\NEWatttagdeprecated alt applet \TAGimgATTalt
\NEWatttagtemplate alt area img
\NEWatttag alt img {\VALsetrelax \IMGalt}
\NEWatttagtemplate alt input img

\NEWattquiet aoff

\NEWattquiet archive
\NEWatttagdeprecatedunsupported archive applet
\NEWatttagunsupported archive object

\NEWattquiet axis
\NEWatttagunsupported axis td
\NEWatttagunsupported axis th

\NEWatt background
\NEWatttagdeprecated background body {%
\BKGDsourceset \BKGDload \CLRbgbodytrue}

\NEWatt bgcolor
\NEWatttagdeprecated bgcolor body \CLRsetbgbody
\NEWatttagdeprecated bgcolor table {\CLRsetbgtable \CLRbordertrue}
\NEWatttagtemplate bgcolor td table
\NEWatttagtemplate bgcolor th table
\NEWatttagtemplate bgcolor tr table

\NEWatt border
\NEWatttagdeprecated border img \BRDset
\NEWatttagtemplate border object img
\NEWatttag border table \BRDset

\NEWattproprietary bordercolor
\NEWattproprietary bordercolordark
\NEWattproprietary bordercolorlight

\NEWattquiet bottommargin

\NEWatt cellpadding
\NEWatttag cellpadding table \PROPpadding

\NEWatt cellspacing
\NEWatttag cellspacing table \PROPMargin

\NEWattquiet char
\NEWatttagunsupported char col
\NEWatttagunsupported char colgroup
\NEWatttagunsupported char tbody
\NEWatttagunsupported char td
\NEWatttagunsupported char tfoot
\NEWatttagunsupported char th
\NEWatttagunsupported char thead
\NEWatttagunsupported char tr

\NEWattquiet charoff
\NEWatttagunsupported charoff col
\NEWatttagunsupported charoff colgroup

```

```

\NEWattagunsupported charoff tbody
\NEWattagunsupported charoff td
\NEWattagunsupported charoff tfoot
\NEWattagunsupported charoff th
\NEWattagunsupported charoff thead
\NEWattagunsupported charoff tr

\NEWattquiet charset
\NEWattagunsupported charset a
\NEWattagunsupported charset link
\NEWattagunsupported charset script

\NEWattquiet checked
\NEWattagunsupported checked input

\NEWattquiet cite
\NEWattagunsupported cite blockquote
\NEWattagunsupported cite del
\NEWattagunsupported cite ins
\NEWattagunsupported cite q

\message {class,}
% Fixme: proprietary for base, basefont, head, html, meta,
% param, script, style, title

\NEWattquiet classid
\NEWattagunsupported classid object

\NEWattquiet clear
\NEWattagdeprecatedunsupported clear br {\VALgetlc
\ifLlNKunrendered \else
\ifx \VALlowercase \CONall \par \vfil \FLTget \noindent \fi \fi}

\NEWattdef code \empty % Gedcom
\NEWattagdeprecatedunsupported code applet

\NEWattquiet codebase
\NEWattagdeprecatedunsupported codebase applet
\NEWattagunsupported codebase object

\NEWattquiet codetype
\NEWattagunsupported codetype object

\NEWatt color
\NEWattagdeprecated color basefont \CLRsetforeground
\NEWattagdeprecated color font \CLRsetforeground

\NEWattquiet cols
\NEWattagnotstrictunsupported frameset cols
\NEWattagunsupported cols textarea

\NEWatt colspan
\NEWattag colspan td {\SPANcolpost \BOXwidthdefault}
\NEWattag colspan th {\SPANcolpost \BOXwidthdefault}

\NEWatt compact
\NEWattagdeprecatedunsupported compact dir
\NEWattagdeprecated compact dl \OPENDlcompacttrue
\NEWattagdeprecatedunsupported compact menu
\NEWattagdeprecatedunsupported compact ol
\NEWattagdeprecatedunsupported compact ul

\NEWatt content
\NEWattag content meta {\VALset \METAcont}

\NEWattquiet coords
\NEWattagunsupported coords a
\NEWattagunsupported coords area

\NEWatt data
\NEWattag data object {\VALset \URLhref}

\NEWattquiet datetime
\NEWattagunsupported datetime del
\NEWattagunsupported datetime ins

\NEWattquiet declare
\NEWattagunsupported declare object

\NEWattquiet defer
\NEWattagunsupported defer script

\NEWatt dir
\NEWattdef dir \BIDIsitone % sic!
\NEWattagproprietaryunsupported dir basefont
\NEWattagproprietaryunsupported dir br
\NEWattagproprietaryunsupported dir frame
\NEWattagproprietaryunsupported dir frameset
\NEWattagproprietaryunsupported dir iframe
\NEWattagproprietaryunsupported dir param
\NEWattagproprietaryunsupported dir script

\NEWattquiet disabled
\NEWattagunsupported disabled button
\NEWattagunsupported disabled input
\NEWattagunsupported disabled optgroup
\NEWattagunsupported disabled option
\NEWattagunsupported disabled select
\NEWattagunsupported disabled textarea

\NEWattquiet enctype
\NEWattagunsupported enctype form

\NEWatt face
\NEWattagdeprecated face basefont {\FONTfam = \VAL \FONTchoose}
\NEWattagtemplate face font basefont

\NEWattquiet font-weight

\NEWattquiet for
\NEWattagunsupported for label

\NEWattquiet frame
\NEWattagunsupported frame table

\NEWattquiet frameborder
\NEWattagnotstrictunsupported frameborder frame
\NEWattagnotstrictunsupported frameborder iframe

\NEWattproprietary framespacing

\NEWattquiet guid

\NEWattquiet headers
\NEWattagunsupported headers td
\NEWattagunsupported headers th

\NEWatt height
\NEWattagtemplate height applet td
\NEWattagnotstrict height iframe \TAGimgATtheight
\NEWattag height img {\VALsetdim \IMGheight \vsize}
\NEWattagtemplate height object img
\NEWattagdeprecated height td {%
\VALsetdim \BOXheight \vsize \BOXheightgiventru}
\NEWattagtemplate height th td
\NEWattagproprietary height tr \TAGtdATtheight

\NEWattquiet hover

```

```

\NEWatt href
\NEWatttag href a {\VALset \URLhref}
\NEWatttagtemplate href area a
\NEWatttagtemplate href base a
\NEWatttagtemplate href link a
\NEWatttagtemplate href xml-stylesheet a %%%

\NEWattquiet hreflang
\NEWatttagunsupported hreflang a
\NEWatttagunsupported hreflang link

\NEWattquiet hspace
\NEWatttagdeprecatedunsupported hspace applet
\NEWatttagdeprecatedunsupported hspace img
\NEWatttagdeprecatedunsupported hspace object

\NEWatt http-equiv
\NEWatttag http-equiv meta {\VALset \metahtpequiv} % Fixme: \META...?

\NEWatt id
\NEWattdef id {\VALset \LINKid \let \LINKname \LINKid
\ifx \LINKname \empty \else \ANCRbyname \LINKname \fi}
\NEWatttagproprietaryunsupported id base
\NEWatttagproprietaryunsupported id head
\NEWatttagproprietaryunsupported id html
\NEWatttagproprietaryunsupported id meta
\NEWatttagproprietaryunsupported id script
\NEWatttagproprietaryunsupported id style
\NEWatttagproprietaryunsupported id title

\NEWattproprietary ids

\NEWattquiet ismap
\NEWatttagunsupported ismap img
\NEWatttagunsupported ismap input

\NEWattquiet label
\NEWatttagunsupported label option
\NEWatttagunsupported label optgroup

\NEWatt lang
\NEWattdef lang {\VALset \LANGabbr \LANGset}
\NEWatttagproprietaryunsupported lang applet
\NEWatttagproprietaryunsupported lang base
\NEWatttagproprietaryunsupported lang basefont
\NEWatttagproprietaryunsupported lang br
\NEWatttagproprietaryunsupported lang frame
\NEWatttagproprietaryunsupported lang frameset
\NEWatttagproprietaryunsupported lang iframe
\NEWatttagproprietaryunsupported lang param
\NEWatttagproprietaryunsupported lang script

\NEWatt language
\NEWatttagdeprecated language script \SCRIPTset

\NEWattquiet leftmargin

\NEWattquiet link % unvisited link color used for external links
\NEWatttagdeprecated link body \CLRsetlink

\NEWattquiet longdesc
\NEWatttagnotstrictunsupported longdesc frame
\NEWatttagnotstrictunsupported longdesc iframe
\NEWatttagunsupported longdesc img

\NEWattquiet lowsrc

\NEWattproprietary marginbottom

\NEWattquiet marginheight
\NEWatttagnotstrictunsupported marginheight frame
\NEWatttagnotstrictunsupported marginheight iframe

\NEWattproprietary marginleft
\NEWattproprietary marginright

\NEWattproprietary margintop

\NEWattquiet marginwidth
\NEWatttagnotstrictunsupported marginwidth frame
\NEWatttagnotstrictunsupported marginwidth iframe

\NEWattquiet maxlength
\NEWatttagunsupported maxlength input

\NEWatt media
\NEWatttag media link {\VALsetlc \LINKmedia}
\NEWatttagtemplate media style link

\NEWattquiet menu

\NEWattdef method \empty % Gedcom
\NEWatttagunsupported method form

\NEWattquiet multiple
\NEWatttagunsupported multiple select

\NEWatt name
\NEWatttag name a {\VALset \LINKname}
\NEWatttagdeprecated name applet \empty
\NEWatttagnotstrict name frame \empty
\NEWatttagnotstrict name iframe \empty
\NEWatttag name img \empty
\NEWatttag name input \empty
\NEWatttag name map \empty
\NEWatttag name object \empty
\NEWatttag name param \empty
\NEWatttag name map \empty
\NEWatttag name meta {\VALset \METAname}
\NEWatttag name select \empty
\NEWatttag name textarea \empty

\NEWattproprietary newvalues

\NEWattquiet nohref
\NEWatttagunsupported nohref area

\NEWattquiet noresize
\NEWatttagnotstrictunsupported noresize frame

\NEWattquiet noshade
\NEWatttagdeprecatedunsupported noshade hr

\NEWattquiet nowrap
\NEWatttagdeprecatedunsupported nowrap td
\NEWatttagdeprecatedunsupported nowrap th

\NEWattquiet object
\NEWatttagdeprecatedunsupported object applet

% Fixme: proprietary for several elements:
\NEWattunsupported onblur
\NEWattunsupported onchange
\NEWattunsupported onclick
\NEWattproprietary oncopy
\NEWattunsupported ondblclick

```

```

\NEWattunsupported onfocus
\NEWattunsupported onkeydown
\NEWattunsupported onkeypress
\NEWattunsupported onkeyup
\NEWattnotstrictunsupported onload
\NEWattunsupported onmousedown
\NEWattunsupported onmousemove
\NEWattunsupported onmouseout
\NEWattunsupported onmouseover
\NEWattunsupported onmouseup
\NEWattunsupported onreset
\NEWattunsupported onselect
\NEWattunsupported onsubmit
\NEWattnotstrictunsupported onunload

\NEWattquiet point-size

\NEWattquiet profile
\NEWatttagunsupported profile head

\NEWattquiet prompt
\NEWatttagdeprecatedunsupported prompt isindex

\NEWattquiet readonly
\NEWatttagunsupported readonly input

\NEWattquiet ref
\NEWatttag ref link {%
  \ifGEDCOM \VALset \URLhref \else \CHKerrproprietaryatt {ref}\fi}

\NEWatt rel
\NEWatttag rel a \empty
\NEWatttag rel link {\VALsetlc \LINKrel}

\NEWatt rev
\NEWatttagunsupported rev a
\NEWatttagunsupported rev link

\NEWattquiet rightmargin

\NEWattquiet rows
\NEWatttagnotstrictunsupported rows frameset
\NEWatttagunsupported rows textarea

\NEWatt rowspan % Fixme: rowspan needed before class attribute in 3c2stagl.htm
\NEWatttag rowspan td {\SPANrowpost \BOXspancolexec \BOXfillerbottom}
\NEWatttag rowspan th {\SPANrowpost \BOXspancolexec \BOXfillerbottom}

\NEWattquiet rules
\NEWatttagunsupported rules table

\NEWattquiet scheme
\NEWatttagunsupported meta scheme

\NEWattquiet scope
\NEWatttagunsupported scope td
\NEWatttagunsupported scope th

\NEWattquiet scrolling
\NEWatttagnotstrictunsupported scrolling frame
\NEWatttagnotstrictunsupported scrolling iframe

\NEWattquiet selected
\NEWatttagunsupported selected option

\NEWattquiet shape
\NEWatttagunsupported shape a
\NEWatttagunsupported shape area

\NEWatt size
\NEWatttagdeprecated size basefont {\SIZESwitch {\the \VAL}}
\NEWatttagtemplate size font basefont
\NEWatttagunsupported size hr
\NEWatttag size input {\VALsetint \INDENTinputsize}
\NEWatttagtemplate size select input

\NEWatt span
\NEWatttagtemplate span col colgroup
\NEWatttag span colgroup \SPANcolgroupset

\NEWatt src
\NEWatttagnotstrict src frame {\VALset \URLhref}
\NEWatttag src img {\VALset \URLhref}
\NEWatttagtemplate src iframe frame
\NEWatttagtemplate src input img
\NEWatttagtemplate src script img

\NEWattunsupported standby

\NEWatt start
%\NEWatttagdeprecated start ol {\VALsetint \Noli
% \CHKerrrunsupattprop \CONstart} %% Fixme: used in tds.htm

\message {style,}
% Fixme: style is proprietary for base, basefont, head, html, meta,
% param, script, style, title.

\NEWatt summary
\NEWatttag summary table {\edef \TABLEsummary {\the \VAL \relax}}

\NEWattquiet tabindex
\NEWatttagunsupported tabindex a
\NEWatttagunsupported tabindex a
\NEWatttagunsupported tabindex area
\NEWatttagunsupported tabindex button
\NEWatttagunsupported tabindex input
\NEWatttagunsupported tabindex object
\NEWatttagunsupported tabindex select
\NEWatttagunsupported tabindex textarea

\NEWattquiet target
\NEWatttagnotstrict target a \empty
\NEWatttagnotstrict target area \empty
\NEWatttagnotstrict target base \empty
\NEWatttagnotstrict target form \empty
\NEWatttag target link {\ifGEDCOM \else \CHKerrnotstrictatt \fi}

\NEWattproprietary test

\NEWatt text
\NEWatttagdeprecated text body \CLRsetforeground

\NEWatt title
\NEWattdef title \empty
\NEWatttag title a \empty
\NEWatttag title abbr \empty
\NEWatttag title acronym \empty
\NEWatttag title b \empty
\NEWatttagproprietaryunsupported title base
\NEWatttagproprietaryunsupported title basefont
\NEWatttagproprietaryunsupported title head
\NEWatttagproprietaryunsupported title html
\NEWatttag title i \empty
\NEWatttagproprietaryunsupported title meta
\NEWatttagproprietaryunsupported title param
\NEWatttagproprietaryunsupported title script
\NEWatttag title span \empty
\NEWatttagproprietaryunsupported title title

```

```

\NEWattquiet topmargin

\NEWattdef type \empty % Gedcom
\NEWatttagtemplate type a link
\NEWatttagunsupported type button
\NEWatttag type input {\VALsetlc \INDENTinputtype}
\NEWatttagdeprecatedunsupported type li
\NEWatttag type link {\VALset \LINKtype}
\NEWatttag type object {\VALset \OBJECTtype}
\NEWatttagdeprecatedunsupported type ol
\NEWatttagtemplate type param link
\NEWatttag type script {\VALset \SCRIPTtype}
\NEWatttagtemplate type style link
\NEWatttagdeprecatedunsupported type ul
\NEWatttagtemplate type xml-style sheet link %%%

\NEWattquiet url

\NEWattquiet usemap
\NEWatttagunsupported usemap img
\NEWatttagunsupported usemap input
\NEWatttagunsupported usemap object

\NEWatt valign
\NEWatttag valign col \BOXvalignset
\NEWatttagtemplate valign colgroup col
\NEWatttagtemplate valign tbody col
\NEWatttagtemplate valign td col
\NEWatttagtemplate valign tfoot col
\NEWatttagtemplate valign th col
\NEWatttagtemplate valign thead col
\NEWatttagtemplate valign tr col

\NEWatt value
\NEWatttagtemplate value button input
\NEWatttag value input {\VALset \INDENTinputvalue}
\NEWatttagtemplate value li input
\NEWatttagtemplate value option input
\NEWatttagtemplate value param input

\NEWattquiet valuetype
\NEWatttagunsupported valuetype param

\NEWattquiet version
\NEWatttagdeprecatedunsupported version html

\NEWattquiet visibility

\NEWatt vlink % visited link color used for internal links
\NEWatttagdeprecated vlink body \CLRsetvlink

\NEWattquiet vspace
\NEWatttagdeprecatedunsupported vspace applet
\NEWatttagdeprecatedunsupported vspace img
\NEWatttagdeprecatedunsupported vspace object

\NEWatt width
\NEWatttagtemplate width applet td
\NEWatttag width col {\TAGtableATTwidth \BOXspancolexec}
\NEWatttagtemplate width colgroup col
\NEWatttagdeprecated width hr {\VALsetdim \BOXwidth \hsize}
\NEWatttagnotstrict width iframe {\VALsetdim \IMGwidth \hsize}
\NEWatttag width img {\VALsetdim \IMGwidth \hsize}
\NEWatttagtemplate width object img
\NEWatttagtemplate width pre td
\NEWatttag width table {
  \VALsetdim \BOXwidth \hsize \BOXwidthgiventrue}
\NEWatttagdeprecated width td {\TAGtableATTwidth \BOXspancolexec}

\NEWatttagtemplate width th td

\NEWattproprietary wrap

\NEWattdef xmlns {\VALsetglobal \NAMESPACE}
\NEWattdef xmlns:html {\VALsetglobal \NAMESPACE}

\NEWattproprietary xmlns:utility

\NEWattdef xml:lang \ATTlang
\NEWattunsupported xml:space

\message {...}

\echo {- Pre-defined classes:} %=====

% Fixme: outsource

\message {break,}
\def \CLASSbreak {\VAL = {always}%
  \csname PROPage-break-before\endcsname}

\message {breakifeven,}
\def \CLASSbreakifeven {%
  \ifodd \OUTnpage \else \VAL = {always}%
  \csname PROPage-break-before\endcsname \fi}

\message {breakifodd,}
\def \CLASSbreakifodd {%
  \ifodd \OUTnpage \VAL = {always}%
  \csname PROPage-break-before\endcsname \fi}

\message {kerning,}%
\def \CLASSkerning {\KRNtrue \KRNcheck}

\message {nofloat,}%
\let \CLASSnofloat \OPENfloatsuppresstrue

\message {noindent,}%
\def \CLASSnoindent {\TEXTindent = 0pt}

\message {nokerning,}%
\def \CLASSnokerning {\KRNfalse \KRNcheckback}

\message {nolink,}%
\let \CLASSnolink \LINKunlinkedtrue

\message {nopagenumbers,}%
\def \CLASSnopagenumbers {%
  \global \OUTnpagefalse \global \OUTnpageonttrue}

\message {noprint,}%
\let \CLASSnoprint \LINKunrenderedtrue

\message {noscreen,}%
\let \CLASSnoscreen \empty

\message {pagenumbers,}%
\def \CLASSpagenumbers {%
  \global \OUTnpageodottrue \global \OUTnpageontfalse}

\message {twocolumns,}%
\let \CLASStwocolumns \OUTdoubletrue

\echo {- Block-level elements:} %=====

\ELEMENTnew address
  \A \GAPwrite \B \ELEMENTvalueget \FONTdo \BIDIbegin \C
  \D \GAPwrite \ALGNtextend \BIDIend \par \E \F

```

```

\ELEMENTnewblock blockquote % Fixme: alignments not supported
  \A \B \initstart
    \advance \leftskip by 2\LINEheight \advance \leftskip by -lex
    \advance \rightskip by 2\LINEheight \advance \rightskip by -lex
  \NOParreset \C \D
  \E \NOParresetglobal \F

\ELEMENTnew body
  \A \B \OUTbody \CONbody \C
  \D \GAPwrite \par \LINEupskip
    \ifOPENobject \else \global \CLRfalse
    \global \CLRbgfalse \vfill \break \fi \E \F

\Def TAGbr/ {\TAGbrexec \CHKerrmissingspace}
\def \TAGbr {\TAGbrexec}
  \def \TAGbrexec {\ELEMENTcount \ELEMENTvalueget \GAPwrite
    \ifLINKunrendered \else \ifGAPspacenow \hskip -\spaceskip \fi
    \ALGNtextend \break \global \GAPspacenowtrue \fi}
\Def TAG/br {\relax}

\ELEMENTnew caption
  \A \ifGEDCOM \GAPwrite \fi \B
    \ifGEDCOM \ELEMENTblockfalse \ELEMENTvalueget
    \ifELEMENTblock \par \FLTget \noindent \fi
    \else \BRDreset \ELEMENTvalueget \fi \C
  \D \GAPwrite \ifGEDCOM \else \par \fi \E \F

\ELEMENTnewblock center
  \A \B \ALGNtext \CONcenter \initstart \OUTdoublebegin
    \CHKerrdeprecated \CONcenter \C
  \D \OUTdoubleend \E \NOParreset \F

\Message {col /}
\def \TAGcol {\ELEMENTvalueget \SPANcolnoadvance \BOXspancolexec}

\ELEMENTnew colgroup
  \A \SPANcolgroupinit \ELEMENTvalueget \B \SPANcolgroupexec \C \D \E \F

\ELEMENTnewblock dd
  \A \nobreak \B \NOParreset \LISTnoddlstsetglobal \LISTleftmarginset
    \FONTitalictrue \initstart \FONTmodechange \C \D \E \F

\ELEMENTnewblock div
  \A \B \global \hangindent = 0pt \global \hangafter = 0
    \initstart \OUTdoublebegin \C \D \OUTdoubleend \E \F

\ELEMENTnewblock dl
  \A \hangindent = 0pt \hangafter = 0
  \B \OPENddfalses \OPENdtfalses \OPENlifalses \LISTnostart \initstart \C
  \D \E \NOParresetglobal \F

\ELEMENTnewblock dt
  \A \B \FONTbdtrue \FONTmodechange \LISTleftmarginsetdt \initstart \C
  \D \nobreak \ifOPENdlcompact \LINEupskip \CONspace \fi \nobreak \E \F

\ELEMENTnewblockh 1 5 3 3 1
\ELEMENTnewblockh 2 4 3 2.25 0.75
\ELEMENTnewblockh 3 3 2 2.25 0.75
\ELEMENTnewblockh 4 2 2 1.5 0.5
\ELEMENTnewblockh 5 1 2 0.75 0.25
\ELEMENTnewblockh 6 0 1 0.75 0.25
\let \ELEMENTnewblockh \undefined

\Def TAGhr/ {%
  \GAPwrite \TAGhrexec \ELEMENTvalueget \TAGhrexecindent \CHKerrmissingspace}
\def \TAGhr {\GAPwrite \TAGhrexec \ELEMENTvalueget \TAGhrexecindent}
  \def \TAGhrexec {%
    \strut \ELEMENTcount \ifOPENp \par \fi
    \ifOPENtd \vskip -0,67\LINEheight % <hr /> is allowed inside <li>.
    \else \ifOPENth \vskip -0,67\LINEheight \fi \fi
    \ifLINKunrendered \let \backupunrend \empty
    \else \let \backupunrend \undefined \fi
    \bgroup \BOXwidth = \hsize}
  \def \TAGhrexecindent {%
    \ifLINKunrendered \else \ifhmode \par \fi
    \centerline {\vbox to \LINEheight {\vss \CLRizeforeground
      \hrule width \BOXwidth height 0.033\LINEheight \vss}}%
    \NOParresetglobal \fi \egroup
    \ifOPENtd \vskip -0.33\LINEheight \fi
    \ifx \backupunrend \undefined \LINKunrenderedfalse \fi}
\Def TAG/hr {\relax}

\ELEMENTnew iframe
  \A \B \CHKerrnotstrict \OPENobjecttrue
    \ELEMENTvalueget \CHNobjectloadurl \CHNiftable \C
  \D \E \global \ENDfalse \beginmarkupCAT \F % Fixme: \global args

\Message {img /}
\def \TAGimg {%
  \ELEMENTcount
  % \ifOPENpre \ifGAPfilled \else
  % \LINEupskip \hskip -2\LINEheight \hskip 1ex \fi \fi
  \GAPwrite \IMGinit \ALGNfalse
  \ELEMENTvalueget \IMGget}

\Message {input /}
\Def TAGinput/ {%
  \GAPwrite \ELEMENTvalueget
  \TAGinputindent \CHKerrmissingspace}
\def \TAGinput {%
  \ELEMENTcount \GAPwrite \ELEMENTvalueget \TAGinputindent}
\def \TAGinputindent {%
  \ifLINKunrendered \else
  \ifx \INDENTinputtype \empty \TAGinputindenttext \else
  \ifx \INDENTinputtype \CONTEXT \TAGinputindenttext \fi \fi \fi}
\def \TAGinputindenttext {% Fixme: handle \VALradio and \VALcheckbox too
  \ifvmode \FLTget \noindent \fi
  \INDENTinputvalue \leaders \hrule \hfill \CONspace
  \ifOPENTable \else \par \fi}
\Def TAG/input {\relax}

\ELEMENTnewblock li
  \A \NOParreset \B \LISTnolastsetglobal
    \initstart \LISTleftmarginset \LISTmark \C
  \D \E \LISTnoadvance \LISTpenalty \F

\ELEMENTnewblock map
  \A \B \OPENpfalses \initstart \CHKerrunsupported \CONmap \C \D \E \F

\ELEMENTnew object
  \A \B \OBJECTinit \ELEMENTvalueget \OBJECThandle \C
  \D \OPENobjectfalses \E \global \ENDfalse \beginmarkupCAT \F

\ELEMENTnewblock ol
  \A \hangindent = 0pt \hangafter = 0
  \B \OPENlifalses \LISTnostart \initstart \C \D \E \NOParresetglobal \F

\ELEMENTnew option
  \A \B \ifLINKunrendered \else \par \FLTget \noindent \hskip -\LINEheight
    \CLRize {\the \CLRuldot \CONspace rg \the \CLRuldot \CONspace RG}%
    \hbox {\vbox {\hrule height 0.1ex \hbox to 0.8ex {\vrule width 0.1ex
      \hfil \vbox to 0.8ex {\vfil }}\hfil
      \vrule width 0.1ex }\hrule height 0.1ex }}%
    \hskip \LINEheight \hskip -1ex \fi \C \D \par \E \F

\ELEMENTnewblock p
  \A \NOParadvance \B \NOParlastsetglobal \initstart \ifLINKunrendered \else

```

```

\ifTEXTindentgiven \FLTget \indent \else \ELEMENTblockindentbyno \fi \fi \C
\D \E \F

\ELEMENTnewblock pre % Fixme: alignments not supported
\A \Noparadvance \B \GAPobeytrue
\advance \leftskip by 2\LINEheight
\advance \leftskip by -1ex \parindent = 0mm
\FONTdomono \OBEYlines \OBEYspaces \initstart \C \D \E \OBEYend \F

\ELEMENTnewblock select
\A \B \advance \leftskip by 2\LINEheight
\advance \leftskip by -1ex \initstart \C \D \E \F

\ELEMENTnew table
\A \B \OPENTdfalse \OPENThfalse \OPENTrfalse
\OPENTbodyfalse \OPENTfootfalse \OPENTheadfalse \OPENfontinnerfalse
\TABLEinit \BOXinit \ELEMENTvalueget
\ifLINKunrendered \else \GAPsetstrut \MSGkerning
\advance \TABLEnesting by 1 \FLTtablealign
\ifx \TABLEsummary \empty \CHKerrempty \CONsummary \fi
\BOXlineheightset \SPANlayout \TABLEbegin \BOXindent \fi \C
\D \ifLINKunrendered \else \TABLEend \par \FLTtable \SPANrowclear \fi
\E \ifLINKunrendered \else
\ifOPENTd \LINEupskip \else \ifOPENTh \LINEupskip \fi \fi \fi
\SPANrowrest \Noparreset \NESTtableendouter \F

\ELEMENTnewblock tbody
\A \B \initstart \C \D \E \F

\ELEMENTnew td
\A \SPANcolnoadvance \B
\ifLINKunrendered \else \BOXbegin \BIDbegin \fi \C
\D \ifLINKunrendered \else %%%
\NESTtdthend \GAPwriteorstrut \BIDIend \BOXend \fi \E \F

\ELEMENTnewblock tfoot % Fixme: not rendered as table footer, if not last row
\A \B \initstart \C \D \E \F

\ELEMENTnew th
\A \SPANcolnoadvance \B \ifLINKunrendered \else
\FONTbdttrue \BOXbegin \BIDbegin \FONTmodechange \fi \C
\D \ifLINKunrendered \else
\NESTtdthend \GAPwriteorstrut \BIDIend \BOXend \fi \E \F

\ELEMENTnewblock thead
\A \B \initstart \C \D \E \F

\ELEMENTnew title
\A \ifGEDCOM \GAPwrite \fi
\B \ifGEDCOM \ELEMENTblockfalse \ELEMENTvalueget
\ifELEMENTblock \par \FLTget \noindent \fi \fi \C
\D \ifGEDCOM \GAPwrite \else \METAtitle \fi \E \F

\ELEMENTnew tr
\A \BOXrownoadvance
\B \BOXgridtrbegin \ELEMENTvalueget % Fixme: class cannot be 'body'
\BOXheightswappedinit \C
\D \SPANrowfill \BOXgridtrend
\TABLErowend \ifBOXheightgiven \BOXheightswapped % sic!
\E \TABLErowbegin \BOXindent \F

\ELEMENTnewblock ul
\A \hangindent = 0pt \hangafter = 0
\B \OPENlifalse \LISTnostart \initstart \C
\D \E \Noparresetglobal \F

\echo {- Inline-level elements} %-----

\ELEMENTnew a
\A \GAPwrite \nobreak \ANCRfalse
\B \ANCRinit \ELEMENTvalueget \ANCRbegin \C
\D \GAPwrite \ANCRend \E \ANCRfalse \F

\ELEMENTnew abbr
\A \GAPwrite \B \FONTitalictrue \ELEMENTvalueget \C
\D \FONTmodechange \GAPwrite \E \F

\ELEMENTnewdefa {acronym} {\FONTitalictrue \FONTmodechange}

\ELEMENTnew b
\A \GAPwrite \B \FONTbdttrue \ELEMENTvalueget \FONTmodechange \C
\D \GAPwrite \E \F

\def \TACbasefont {\ELEMENTcount \ELEMENTvalueget \FONTswitchspan
\CHKerrdeprecated \CONbasefont}

\ifx \eTeXversion \undefined
\message {bdo ! Unsupported (eTeX required).}%
\else
\ELEMENTnew bdo
\A \GAPwrite \B \ELEMENTvalueget \BIDIbeginone \C
\D \GAPwrite \BIDIendone \E \F \fi

\ELEMENTnewdefa {big} {\SIZEbaselarger \CLRizeforeground \FONTdo}

\ELEMENTnewdefa {blink} {% Fixme: should not appear as a message
\CHKerrproprietary \CONblink \FONTbdttrue \FONTitalictrue
\FONTmodechange \FONTunderline \GAPwrite}

\ELEMENTnewdefa {bold} {% Fixme: should not appear as a message
\CHKerrproprietary \CONbold \FONTbdttrue \FONTmodechange}

\ELEMENTnew cite
\A \GAPwrite \B \FONTitalictrue \ELEMENTvalueget \FONTmodechange \C
\D \GAPwrite \E \F

\ELEMENTnew code
\A \GAPwrite \B \FONTdomono \ELEMENTvalueget \FONTchangenamename \C
\D \GAPwrite \E \F

\ELEMENTnew del
\A \GAPwrite \B \C \D \GAPstriketrue \GAPwrite \E \F

\ELEMENTnew dfn
\A \GAPwrite \B \FONTitalictrue \ELEMENTvalueget \FONTmodechange \C
\D \GAPwrite \E \F

\ELEMENTnew em
\A \GAPwrite \B \FONTitalictrue \ELEMENTvalueget \FONTmodechange \C
\D \GAPwrite \E \F

\ELEMENTnew font
\A \GAPwrite \B \FONTdoinner \ELEMENTvalueget
\CHKerrdeprecated \CONfont \FONTswitchspan \C
\D \GAPwrite \E \F

\ELEMENTnew i
\A \GAPwrite \B \FONTitalictrue \ELEMENTvalueget \FONTmodechange \C
\D \GAPwrite \E \F

\ELEMENTnew ins
\A \GAPwrite \B \ELEMENTvalueget \C \D \FONTunderline \GAPwrite \E \F

\ELEMENTnew kbd
\A \GAPwrite \B \FONTdomono \ELEMENTvalueget \FONTchangenamename \C
\D \GAPwrite \E \F

\ELEMENTnew s

```



```

\A \GAPwrite \B \CHKerrdeprecated \CONS \ELEMENTvalueget \C
\D \GAPstrikertrue \GAPwrite \E \F

\ELEMENTnew samp
\A \GAPwrite \B \FONTdomono \ELEMENTvalueget \FONTchangenamename \C
\D \GAPwrite \E \F

\ELEMENTnewdefa {small} {\SIZEsmaller}

\ELEMENTnew span
\A \GAPwrite \B \ELEMENTvalueget \FONTswitchspan \C
\D \LNGde \GAPwrite \E \F

\ELEMENTnew strike
\A \GAPwrite \B \C
\D \CHKerrdeprecated \CONstrike \GAPstrikertrue \GAPwrite \E \F

\ELEMENTnew strong
\A \GAPwrite \B \FONTbdtrue \ELEMENTvalueget \FONTmodechange \C
\D \GAPwrite \E \F

\ELEMENTnew sub
\A \GAPwrite \B \FONTswitchsubbegin \GAPnostrutstrue \C
\D \GAPwrite \FONTswitchsubend \E \F

\ELEMENTnew sup
\A \GAPwrite \B \FONTswitchsupbegin \GAPnostrutstrue \C
\D \GAPwrite \FONTswitchsuspend \E \F

\ELEMENTnewdefa {tt} {%
\FONTdomono \CLRizeforeground}

\ELEMENTnew u
\A \GAPwrite \B \CHKerrdeprecated \CONu \ELEMENTvalueget \C
\D \FONTunderline \GAPwrite \E \F

\ELEMENTnew var
\A \GAPwrite \B \FONTitalictrue \ELEMENTvalueget \FONTmodechange \C
\D \GAPwrite \E \F

\echo {- Unrendered elements:} %-----

\Message {form}
% In GMS, forms do not open another nesting level.
% In practice, forms are too often misnested within tables.
\def \TAGform {\ELEMENTcount \OPENformtrue}
\Def TAG/form {\OPENformfalse}

\ELEMENTnew head
\A \B \GAPwriteomit % Fixme: \global args
\global \CSSaddedfalse \C \D \E \F

\ELEMENTnew html
\A \MSGhtml \B \METApdfinfocreator \METApdfcatalogmenu
\ELEMENTvalueget \C
\D \E \ENDtrue \CHNcheck
\ifOPENobject \strut \hfill \par \fi % Fixme: same in <GEDCOM>
\F % Fixme: do not render object gap

\ELEMENTnewdefa {label} {}

\Message {link /}
\def \TAGlink {%
\ELEMENTcount \ifGEDCOM \GAPwrite \ANCRgedcom
\else \LINKinit \ELEMENTvalueget \LINKexec \fi}

\Message {meta /}
\def \TAGmeta {\ELEMENTcount \METAinit \ELEMENTvalueget \METAget}

\ELEMENTnew script
\A \GAPwrite \B \SCRIPTinit \ELEMENTvalueget \SCRIPThandle \C
\D \E \beginmarkupCAT \F

\bgroup
\ELEMENTnew style
\A \B \GAPwriteomit \SHTstylebegin \ELEMENTvalueget \C
\D \SHTstyleend \E \F \egroup

\def \TAGtrace {% Trace TeX macros for GMS debugging:
\GAPwrite \Message {trace}\MSGproprietary \CHKerrtracebegin}
\Def TAG/trace {\GAPwrite \CHKerrtraceend \Message {/trace}}

\Def TAGxml {\MSGhtml \Message {?\CONxml \CONspace ?}\ELEMENTcount}

\Def TAGxml-stylesheet {%
\message {<?\CONxml-\CONstylesheet}\ELEMENTcount \LINKinit \ELEMENTvalueget
\let \LINKrel \CONstylesheet \LINKexec \message {>}}

%echo {- Unsupported elements:} %-----

\ELEMENTnewdefa {applet} {% Fixme: should not appear as a message
\CHKerrrunsupported \CONapplet \CHKerrdeprecated \CONapplet}

\def \TAGarachne {\ELEMENTcount \CHKerrproprietary {arachne}}

\def \TAGarea {\ELEMENTcount \CHKerrrunsupported {area}}

\def \TAGbase {\CHKerrrunsupported \CONbase}
\Def TAG/base {}

\ELEMENTnewdefa {button} {% Fixme: should not appear as a message
\CHKerrrunsupported \CONbutton}

\def \TAGcomment {% used internally
\ifOPENtr \ifOPENTd \GAPwrite \else \ifOPENTh \GAPwrite \fi \fi
\else \GAPwrite \fi}

\ELEMENTnewdefa {dir} {% Fixme: should not appear as a message
\CHKerrrunsupported \CONdir \CHKerrdeprecated \CONdir}

\def \TAGend {\CHKerrdebugend} % used internally

\ELEMENTnewdefa {fieldset} {% Fixme: should not appear as a message
\CHKerrrunsupported \CONfieldset}

\def \TAGframe {\ELEMENTcount \CHKerrrunsupported \CONframe \CHKerrnotstrict}

\ELEMENTnewdefa {frameset} {\CHKerrrunsupported \CONframeset \CHKerrnotstrict}

\def \TAGisindex {%
\CHKerrrunsupported \CONisindex \CHKerrdeprecated \CONisindex}

\ELEMENTnewdefa {layer} {% Fixme: should not appear as a message
\CHKerrproprietary \CONlayer}

\ELEMENTnewdefa {legend} {% Fixme: should not appear as a message
\CHKerrrunsupported \CONlegend}

\def \TAGmenu {\CHKerrdeprecated \CONmenu}
\Def TAG/menu {\csname TAG/ul\endcsname}

\ELEMENTnew nobr % Fixme: should not appear as a message
\A \GAPwrite \B \CHKerrproprietary \CONnobr \CHKerrrunsupported \CONnobr \C
\D \E \F

\ELEMENTnewdefa {noframes} {% Fixme: should not appear as a message
\CHKerrrunsupported \CONnoframes \CHKerrnotstrict}

```

```

\ELEMENTnewdefa {noscript} {% Fixme: should not appear as a message
  \CHKerrrunsupported \CONscript}

\def \TAGopgroup {\TAGoptgroup \CHKerrmisspelled}
\Def TAG/opgroup {%
  \csname \CONtag /\CONoptgroup \endcsname \CHKerrmisspelled}

\ELEMENTnew optgroup % Fixme: should not appear as a message
  \A \B \CHKerrrunsupported \CONoptgroup \C \D \E \F

\def \TAGparam {\ELEMENTcount \CHKerrrunsupported \CONparam}

\ELEMENTnewdefa {q} {% Fixme: should not appear as a message
  \CHKerrrunsupported \CONq}

\def \TAGsdb {\CHKerrproprietary {sdb}}

\def \TAGspacer {\CHKerrproprietary {spacer}}

\def \TAGtextarea {\CHKerrrunsupported \CONtextarea}
\Def TAG/textarea {}

\def \TAGwbr {\CHKerrproprietary \CONwbr}

\echo {- Recognized style properties:} %=====

\NEWproproprietary align \ALGNtextbyvalue

\NEWpropunsupported azimuth

\NEWpropunsupported background
%%%\NEWpropnew background \CLRsetbgspan %%% supporting only color
\NEWpropnewtag body background \BKGDclrsetbody
\NEWpropnewtag table background \CLRsetbgtable
\NEWpropnewtag tbody background \CLRsetbgtable
\NEWpropnewtag td background \CLRsetbgtable
\NEWpropnewtag tfoot background \CLRsetbgtable
\NEWpropnewtag th background \CLRsetbgtable
\NEWpropnewtag thead background \CLRsetbgtable
\NEWpropnewtag tr background \CLRsetbgtable

\NEWpropunsupported background-attachment

\NEWpropmessage background-color
\NEWpropnew background-color \CLRsetbgspan %%%
\NEWproptagunsupported h1 background-color %%% Fixme
\NEWproptagunsupported h2 background-color %%%
\NEWproptagunsupported h3 background-color %%%
\NEWproptagunsupported h4 background-color %%%
\NEWproptagunsupported h5 background-color %%%
\NEWproptagunsupported h6 background-color %%%
\NEWpropnewtag body background-color \CLRsetbackgroundbody
\NEWpropnewtag table background-color \CLRsetbgtable
\NEWpropnewtag tbody background-color \CLRsetbgtable
\NEWpropnewtag td background-color \CLRsetbgtable
\NEWpropnewtag tfoot background-color \CLRsetbgtable
\NEWpropnewtag th background-color \CLRsetbgtable
\NEWpropnewtag thead background-color \CLRsetbgtable
\NEWpropnewtag tr background-color \CLRsetbgtable

\NEWpropmessage background-image
\NEWpropnewtag body background-image {\BKGDgeturl {\the \VAL}}
\NEWpropnewtag div background-image {\BKGDgeturl {\the \VAL}}

\NEWpropunsupported background-position

\NEWpropnew background-repeat \BKGDsetrepeat

\NEWproproprietary behavior \empty

\NEWpropunsupported border
\NEWpropunsupported border-bottom
\NEWpropunsupported border-left
\NEWpropunsupported border-right
\NEWpropunsupported border-top

\NEWpropunsupported border-collapse

\NEWpropnew border-color \CLRsetborder

\NEWpropunsupported border-bottom-color
\NEWpropunsupported border-left-color
\NEWpropunsupported border-right-color
\NEWpropunsupported border-top-color

\NEWpropunsupported border-spacing

\NEWpropnew border-style {\VALset \BRDprofile}

\NEWpropunsupported border-bottom-style
\NEWpropunsupported border-left-style
\NEWpropunsupported border-right-style
\NEWpropunsupported border-top-style

% Fixme: if a profile like "groove" is given, use "thick"er width:
\NEWpropnew border-width \BRDset
\NEWpropquiet border-bottom-width {\BRDsetone \BRDwb}
\NEWpropquiet border-left-width {\BRDsetone \BRDwl}
\NEWpropquiet border-right-width {\BRDsetone \BRDwr}
\NEWpropquiet border-top-width {\BRDsetone \BRDwt}

\NEWpropunsupported bottom

\NEWpropunsupported caption-side

\NEWpropunsupported clear

\NEWpropunsupported clip

\NEWpropnew color \CLRsetforeground
\NEWpropnewtag ul color \CLRsetlistdot

\NEWpropnew content {\VALset \ELEMENTcontent}

\NEWpropunsupported counter-increment

\NEWpropunsupported cue
\NEWpropunsupported cue-after
\NEWpropunsupported cue-before

\NEWpropunsupported cursor

\ifx \eTeXversion \undefined
  \message {! Unsupported: direction (eTeX required),}
  \NEWpropunsupported direction
\else \message {\CONDdirection,}%
  \NEWpropnew direction \BIDIset \fi

\NEWpropnew display \GAPdisplay
\NEWpropnewtag p display \GAPdisplay

\let \PROPdummy \empty % used internally for CSS selector parsing
\Def PROP;dummy \empty

\NEWpropunsupported elevation

\NEWpropunsupported empty-cells

```

```

\NEWpropproprietary filter \empty
\NEWpropmessage float
\NEWpropnewtag img float \TAGimgATTalign

\NEWpropunsupported font
\NEWpropnew font-family \FONTfamset
\NEWpropnew font-size \SIZEset
\NEWpropnew font-size-adjust \FONTsizeadjustset
\NEWpropunsupported font-stretch
\NEWpropnew font-style \FONTmodestyle
\NEWpropunsupported font-variant
\NEWpropnew font-weight \FONTmodeweight
\NEWpropnew height {\VALsetdim \BOXheight \vsize \BOXheightgiventrue}
\NEWpropnewtag a height {
  \VALsetdim \ANCRheight \LINEheight \ANCRheightgiventrue}
\NEWpropnewtag img height \TAGimgATTtheight

\NEWpropproprietary hspace \empty
\NEWpropunsupported left
\NEWpropunsupported letter-spacing
\NEWpropnew line-height {\VALsetdim \LINEheight \LINEheight
  \LINEheightgiventrue \GAPsetstrut}
\NEWpropnewtag body line-height {%%% Fixme: shorten
  \VALsetdim \LINEheight \LINEheight %%%
  \LINEheightgiventrue \GAPsetstrut %%%
  \topskip = 0.7083\LINEheight} %%%

\NEWpropunsupported list-style
\NEWpropunsupported list-style-image
\NEWpropunsupported list-style-position
\NEWpropunsupported list-style-type

\NEWpropnew margin \MARGINset
\NEWpropnew margin-bottom { \VALsetdim \MARGINbottom \vsize
  \MARGINgiventrue \MARGINgivenbottomtrue}
\NEWpropnewtag img margin-bottom {\VALsetdim \IMGmarginb \vsize}

\NEWpropnew margin-left {\VALsetdim \MARGINleft \hsize \MARGINgiventrue}
\NEWpropnewtag p margin-left {\VALsetdim \MARGINleft \hsize \MARGINgiventrue
  \advance \leftskip by \MARGINleft}

\NEWpropnew margin-right {\VALsetdim \MARGINright \hsize \MARGINgiventrue}
\NEWpropnewtag p margin-right {\VALsetdim \MARGINright \hsize \MARGINgiventrue
  \advance \rightskip by \MARGINright}

\NEWpropnew margin-top {\VALsetdim \MARGINtop \vsize
  \MARGINgiventrue \MARGINgiventoptrue}
\NEWpropnewtag img margin-top {\VALsetdim \IMGmargin \vsize}

\NEWpropunsupported marker-offset
\NEWpropunsupported marks
\NEWpropunsupported max-height
\NEWpropunsupported max-width

\NEWpropunsupported min-height
\NEWpropunsupported min-width
\NEWpropproprietary -moz-opacity \empty
\NEWpropnew orphans {\VALset \test
  \ifx \test \CONnumone \global \clubpenalty = 9999
  \else \ifx \test \CONnumtwo \global \clubpenalty = 10000 \fi \fi}
\NEWpropunsupported outline
\NEWpropunsupported outline-color
\NEWpropunsupported outline-style
\NEWpropunsupported outline-width
\NEWpropunsupported overflow
\NEWpropproprietary overflow-x \empty
\NEWpropproprietary overflow-y \empty
\NEWpropquiet padding \PDGset
\NEWpropnew padding-bottom {\VALsetdim \PDGbottom \vsize}
\NEWpropnew padding-left {\VALsetdim \PDGleft \hsize}
\NEWpropnew padding-top {\VALsetdim \PDGtop \vsize}
\NEWpropnew padding-right {\VALsetdim \PDGright \hsize}

\NEWpropunsupported page % Fixme: check vs. @page
\NEWpropunsupported page-break-after
\NEWpropnew page-break-before \OBEYbreak
\NEWpropnewtag h1 page-break-before \OBEYbreakh
\NEWpropnew page-break-inside {\VALset \TABLEnobreaktest
  \ifx \TABLEnobreaktest \CONavoid \TABLEnobreaktrue \fi}
  \new \ifTABLEnobreak
\NEWpropunsupported pause
\NEWpropunsupported pause-after
\NEWpropunsupported pause-before
\NEWpropunsupported pitch
\NEWpropunsupported pitch-range
\NEWpropunsupported play-during
\NEWpropunsupported position
\NEWpropunsupported quotes
\NEWpropunsupported richness
\NEWpropunsupported right
\NEWpropunsupported rotation-code
\NEWpropmessage size
\NEWpropnewtag @page size \OUTsize
\NEWpropnewtag select size \empty
\NEWpropunsupported speak
\NEWpropunsupported speak-header
\NEWpropunsupported speak-numeral
\NEWpropunsupported speak-punctuation
\NEWpropunsupported speech-rate
\NEWpropunsupported stress
\NEWpropunsupported table-layout

```

```

\NEWpropnew text-align \ALGNtextbyvalue
\NEWpropnewtag pre text-align {%
  \CHKerrunsuppattprop \CONtextalign \ALGNtext \CONleft}

\NEWpropnew text-decoration {\VALset \test
  \ifx \test \CONnone \FONTdecorationfalse \fi}

\NEWpropnew text-indent {\TEXTindentgiventrue \VALsetdim \TEXTindent \hsize}

\NEWpropproprietary text-justify \empty

\NEWpropunsupported text-shadow
\NEWpropunsupported text-transform

\NEWpropunsupported top

\NEWpropunsupported unicode-bidi

\NEWpropnew vertical-align \BOXvalignset

\NEWpropunsupported visibility

\NEWpropunsupported voice-family

\NEWpropunsupported volume

\NEWpropunsupported white-space

\NEWpropnew widows \VALgetwidows

\NEWpropproprietary vspace \empty

\NEWpropnew width {\TAGtableATTwidth \BOXspancolexec}
\NEWpropnewtag img width \TAGimgATTwidth

\NEWpropunsupported word-spacing

\NEWpropunsupported z-index

\message {...}

\echo {- Gedcom XML elements:} %=====

% New 2007/03/08:

% Elements <caption>, <link>, <title> used in HTML too

\ELEMENTnew email % Fixme: partly doubles <uri>, so outsource
  \A \GAPwrite \nobreak \ANCRfalse
  \B \ELEMENTblockfalse \ELEMENTvalueget \ELEMENTblockindent
    \ELEMENTnewgenericpseudo \CONbefore \C
  \D \ANCRinit \edef \TAGuritest {\CONspace \the \GAP \CONspace}%
    \expandafter \TAGuriurlhrefget \TAGuritest
    \glet \ANCRext \URLhref \glet \ANCRint \URLhref
    \ANCRbegext \GAPwrite \ANCRend
    \ifx \URLhref \CONDot \else \TAGuriimage \fi \E \ANCRfalse \F

% Fixme: A few element definitions let :after and :before are unsupported:

\def \TAGexternalid {% Fixme: unsupported
  \ELEMENTcount \Message {\CONexternalid \CONspace /}}

\ELEMENTnew gedcom % like <html> + <body>
  \A \GEDCOMtrue \B \METApdfinfocreator \METApdfcatalogmenu
    \OUTbody \CONgedcom \C
  \D \ELEMENTnewgenericpseudo \CONafter \GAPwrite
    \par \LINEupskip %% Fixme: same as <body>
    \ifOPENobject \else \global \CLRfalse
      \global \CLRbgfalse \vfill \break \fi
  \E \ENDtrue \CHNcheck \ifOPENobject \strut \hfill \par \fi \F

\ELEMENTnew uri % Fixme: partly doubles <email>, so outsource
  \A \GAPwrite \nobreak \ANCRfalse
  \B \ELEMENTblockfalse \ELEMENTvalueget
    \ELEMENTblockindent \ELEMENTnewgenericpseudo \CONbefore \C
  \D \ANCRinit \edef \TAGuritest {\CONspace \the \GAP \CONspace}%
    \expandafter \TAGuriurlhrefget \TAGuritest
    \glet \ANCRext \URLhref \glet \ANCRint \URLhref
    \ANCRbegext \GAPwrite \ANCRend
    \ifx \URLhref \CONDot \else \TAGuriimage \fi \E \ANCRfalse \F

\def \TAGuriurlhrefget #1#2 #3{\xdef \URLhref {#1#2}\global \GAP = {#1#2}#3}

\new \ifTAGuriimage

\def \TAGuriimage {\TAGuriimagefalse % Fixme: rename
  \expandafter \TAGuriimageA \URLhref .jpeg\relax
  \expandafter \TAGuriimageB \URLhref .JPEG\relax
  \expandafter \TAGuriimageC \URLhref .jpg\relax
  \expandafter \TAGuriimageD \URLhref .JPG\relax
  \expandafter \TAGuriimageE \URLhref .pdf\relax
  \expandafter \TAGuriimageF \URLhref .PDF\relax
  \expandafter \TAGuriimageG \URLhref .png\relax
  \expandafter \TAGuriimageH \URLhref .PNG\relax
  \ifTAGuriimage \ELEMENTblockindent %%
    \IMGinit \let \IMGalt \empty \ALGNfalse \IMGget
    \ifIMGabsent \strut \LINEupskip \fi \fi}
  \def \TAGuriimageA #1.jpeg#2\relax {\TAGuriimageHA {#2}\CONDotjpeg}
  \def \TAGuriimageB #1.JPEG#2\relax {\TAGuriimageHA {#2}\CONDotjpeg}
  \def \TAGuriimageC #1.jpg#2\relax {\TAGuriimageHA {#2}\CONDotjpg}
  \def \TAGuriimageD #1.JPG#2\relax {\TAGuriimageHA {#2}\CONDotjpg}
  \def \TAGuriimageE #1.pdf#2\relax {\TAGuriimageHA {#2}\CONDotpdf}
  \def \TAGuriimageF #1.PDF#2\relax {\TAGuriimageHA {#2}\CONDotpdf}
  \def \TAGuriimageG #1.png#2\relax {\TAGuriimageHA {#2}\CONDotpng}
  \def \TAGuriimageH #1.PNG#2\relax {\TAGuriimageHA {#2}\CONDotpng}
  \def \TAGuriimageHA #1#2{\lowercase {\def \TAGuriimageHAY {#1}}%
    \ifx \TAGuriimageHAY #2\TAGuriimagetrue \fi}

% Fixme: elements should be generated by reading the document type declaration

% Top-level:
\ELEMENTnewgeneric ContactRec
\ELEMENTnewgeneric EventRec
\ELEMENTnewgeneric FamilyRec
\ELEMENTnewgeneric GroupRec
\ELEMENTnewgeneric HeaderRec
\ELEMENTnewgeneric IndividualRec
\ELEMENTnewgeneric LDSOrdRec
\ELEMENTnewgeneric MultiMediaRec
\ELEMENTnewgeneric RepositoryRec
\ELEMENTnewgeneric SourceRec

% Sub-level:
\ELEMENTnewgeneric Addressee
\ELEMENTnewgeneric Article
\ELEMENTnewgeneric ArticleDoc
\ELEMENTnewgeneric Association
\ELEMENTnewgeneric AssocIndiv
\ELEMENTnewgeneric Author
\ELEMENTnewgeneric AuthorDoc
\ELEMENTnewgeneric BasedOn
\ELEMENTnewgeneric BirthDateDoc
\ELEMENTnewgeneric BirthPlaceDoc
\ELEMENTnewgeneric CallNbr
\ELEMENTnewgeneric Changed
\ELEMENTnewgeneric Child
\ELEMENTnewgeneric ChildNbr
\ELEMENTnewgeneric Citation
\ELEMENTnewgeneric CitationText
\ELEMENTnewgeneric Contact

```

```

\ELEMENTnewgeneric Coordinates
\ELEMENTnewgeneric Copyright
\ELEMENTnewgeneric Date
\ELEMENTnewgeneric Day
\ELEMENTnewgeneric DeathDateDoc
\ELEMENTnewgeneric DeathPlaceDoc
\ELEMENTnewgeneric DeathStatus
\ELEMENTnewgeneric Description
\ELEMENTnewgeneric DupIndiv
\ELEMENTnewgeneric Event
\ELEMENTnewgeneric Extract
\ELEMENTnewgeneric FamilyNbr
\ELEMENTnewgeneric FamParticipant
\ELEMENTnewgeneric FileCreation
\ELEMENTnewgeneric Format
\ELEMENTnewgeneric Gender
\ELEMENTnewgeneric GivenName
\ELEMENTnewgeneric GivenNameDoc
\ELEMENTnewgeneric GivenNameDoc
\ELEMENTnewgeneric HusbFath
\ELEMENTnewgeneric HusbFathLiving
\ELEMENTnewgeneric IndivDoc
\ELEMENTnewgeneric IndivName
\ELEMENTnewgeneric IndivParticipant
\ELEMENTnewgeneric IndNameVariation
\ELEMENTnewgeneric Level
\ELEMENTnewgeneric Living
\ELEMENTnewgeneric MaidenName
\ELEMENTnewgeneric MailAddress
\ELEMENTnewgeneric MarrDateDoc
\ELEMENTnewgeneric MarrPlaceDoc
\ELEMENTnewgeneric Member
\ELEMENTnewgeneric Month
\ELEMENTnewgeneric MultiMedia
\ELEMENTnewgeneric Name
\ELEMENTnewgeneric NameDoc
\ELEMENTnewgeneric NameSuffix
\ELEMENTnewgeneric Note
\ELEMENTnewgeneric OrdStat
\ELEMENTnewgeneric OrganizationName
\ELEMENTnewgeneric ParentGroup
\ELEMENTnewgeneric Participant
\ELEMENTnewgeneric PersonalTitle
\ELEMENTnewgeneric Phone
\ELEMENTnewgeneric Place
\ELEMENTnewgeneric PlacePart
\ELEMENTnewgeneric PlaceVariation
\ELEMENTnewgeneric Product
\ELEMENTnewgeneric ProductId
\ELEMENTnewgeneric Publishing
\ELEMENTnewgeneric PublishingDoc
\ELEMENTnewgeneric References
\ELEMENTnewgeneric ReferencesDoc
\ELEMENTnewgeneric Relationship
\ELEMENTnewgeneric RelativeDoc
\ELEMENTnewgeneric Religion
\ELEMENTnewgeneric RelToFath
\ELEMENTnewgeneric RelToMoth
\ELEMENTnewgeneric Repository
\ELEMENTnewgeneric ResidenceDoc
\ELEMENTnewgeneric Role
\ELEMENTnewgeneric SameIndiv
\ELEMENTnewgeneric Source
\ELEMENTnewgeneric Start
\ELEMENTnewgeneric Stop
\ELEMENTnewgeneric Submitter
\ELEMENTnewgeneric Supplier
\ELEMENTnewgeneric SurName
\ELEMENTnewgeneric SurNameDoc
\ELEMENTnewgeneric TempleCode
\ELEMENTnewgeneric TitleDoc
\ELEMENTnewgeneric Units
\ELEMENTnewgeneric Version
\ELEMENTnewgeneric WhenRecDoc
\ELEMENTnewgeneric WhenRecorded
\ELEMENTnewgeneric WhereInDoc
\ELEMENTnewgeneric WhereInMM
\ELEMENTnewgeneric WhereInSource
\ELEMENTnewgeneric WifeMoth
\ELEMENTnewgeneric WifeMothLiving
\ELEMENTnewgeneric Year

% Gedcom attributes:
\NEWattdef calendar \empty
\NEWattdef date \empty
\NEWattdef level \empty
\NEWattdef ref \empty
\NEWattdef time \empty
\NEWattdef translationtype \empty
\NEWattdef vitaltype \empty

#####

% Compatibility with external files: -----

% In prologue.tex, epilogue.tex, font.cfg, plugin.cfg, typeset.cfg,
% charsets.tex, these names are used:

\let \CELLdefaultno \SPANcolpredefault
\let \CODEPAGE \CPG
\let \CODEPAGEadd \CPGadd
\let \CODEPAGEaddalias \CPGaddalias
\let \CODEPAGEenable \CPGnbl
\let \CODEPAGEencwrite \CPGencwrite
\let \CODEPAGEloadthem \CPGload
\let \COLORanchorex \CLRanchorex
\let \COLORanchrint \CLRanchorint
\let \COLORuldot \CLRuldot
\let \ERRORabsent \CHKerrabsent % script.htm
\let \FONTfamily \FONTfam
\let \FSIZE \SIZE
\let \FSIZEbase \SIZEbase
\let \GLYPHsadd \GLYPHadd
\let \GLYPHsnamesload \GLYPHload
\let \IMAGEResolution \IMGresolution
\let \KERNING \KRN
\let \KERNINGloadtables \KRNget
\let \KERNINGparseline \KRNparseline
\let \LANGUAGEadd \LNGadd
\let \LANGUAGEloadpatterns \LNGloadpatterns
\let \MAPfontadd \MAPadd
\let \MAPfontaddalias \MAPaddalias
\let \MAPfontaddencoding \MAPaddencoding
\let \MAPfontaddfamily \MAPaddfamily
\let \MAPfontaddfamilystd \MAPaddfamilystd
\let \MAPfontaddfamilyuni \MAPaddfamilyuni
\let \MAPfontaddmates \MAPaddmates
\let \MAPfontaddpair \MAPaddpair
\let \MAPfontaddspecific \MAPaddspecific
\let \MAPload \MAPget
\let \PAGEheight \OUTheight
\let \PAGEwidth \OUTwidth
\let \PAGEmarginbottom \OUTmarginb
\let \PAGEmarginleft \OUTmarginl
\let \PAGEmarginright \OUTmarginr
\let \PAGEmargintop \OUTmarginl

```

```

\let \UNICODEadd \UCDadd
\let \UNICODEencswrite \UCDwrite
\let \UNICODEslotsload \UCDwriteslot
\nec \UNICODEslottempo
\let \VALcolor \CONcolor
\let \VALfontfamily \CONfontfamily
\let \VALfontsize \CONfontsize
\let \VALfontstyle \CONfontstyle
\let \VALheight \CONheight
\let \VALlineheight \CONlineheight
\let \VALmargin \CONmargin
\let \VALpadding \CONpadding
\let \VALsize \CONsize
\let \VALtextalign \CONtextalign
\let \VALtextindent \CONtextindent
\let \VALwidth \CONwidth

\CPGgetdefaultname
\FONTfbkset

\fillmessage 6a{typeset}3 \input typeset.cfg \fillmessage 6-{/typeset}2
\fillmessage 6b{alias}5 \input alias.cfg \fillmessage 6-{/alias}4
\message {fallback}\message {fonts;}
\message {cmr}\message {(default),}
\message {\FONTfbkmonospace}\message {(monospace),}
\message {\FONTfbksansserif}\message {(sans-serif),}
\message {\FONTfbkserif}\message {(serif);}
\FILEsearch {files.cfg}\ifFILEexist
\fillmessage 6c{files}5 \input files.cfg \fillmessage 6-{/files}4 \fi
\fillmessage 4-{/interface}2 \input plugin.cfg
\fillmessage 2-{/initialize}3
\GMSendabout \GMSgoodbye

% Load third-party macros and start shredder: .....
\input epilogue.cfg

```

```

% Setup: .....

```

[GMS_ROOT]/tex/hyphen

Hyphenation Patterns

ca.tex	hu.tex
cs.tex	ia.tex
da.tex	id.tex
de.tex	is.tex
de-rf.tex	it.tex
el.tex	la.tex
en-UK.tex	nl.tex
en-US.tex	no.tex
es.tex	pl.tex
et.tex	pt.tex
eu.tex	ru.tex
fi.tex	sr.tex
fr.tex	sv.tex
ga.tex	tr.tex
hr.tex	uk.tex
	wen.tex

License

*Gerolf Markup Shredder
Copyright (c) 1999–2008 by
Gerolf Diethelm Brettschneider,
Luchtbergstr. 27, D–28237 Bremen.
All rights reserved. This GMS soft-
ware comes without ANY warranty.
You may freely distribute and use it.*

<The>

Gerolf

<Markup>

Shredder

<Handbook>

Version 0.08a

</Handbook>

2008-01-07

</Markup>

Enjoy!

</The>