KompoZer and Nvu User Guide

A Guide to using KompoZer and Nvu on systems running Windows

KompoZer is an open source, WYSIWYG (What You See Is What You Get), Web authoring system. The Guide is written in the hope that it will both prove useful for new users and act as a reference guide for those more experienced.

Contributed by Charles Cooke

KOMPOZER

New! The Guide now reflects the behaviour of KompoZer 0.7.10 which is significantly improved from earlier versions of both Kompozer and Nvu. Coverage of historic Nvu behaviour is retained but the main text follows Kompozer.

Table of Contents

1 Introduction

1.1 What is KompoZer? – What does it do?

KompoZer is a complete Web Authoring System which integrates web page development and web file management.

KompoZer is basically the same product as Nvu but has been developed further removing some of the faults found in Nvu and improving the user interface.

It provides a web page editor which has a simple graphical (wysiwyg – what you see is what you get) interface. With KompoZer, newcomers will quickly and easily be able to produce new web pages.

Alternative editing 'views' of the pages are provided right down to detailed code level.

The output code is compliant to a high extent with the latest issues of the appropriate web language specifications and pages may be checked for validity directly from Nvu using the official W3C validator.

KompoZer incorporates a Site Manager this gives rapid access to the files on both local machines and remote servers. It can cater for several sites and switch rapidly between them.

From within KompoZer pages and associated files may be uploaded to a remote server.

KompoZer supports the use of "Styles" through Cascading Style sheets (CSS) both embedded and external. It has an editor which generates CSS code conforming with CSS 2.1 specifications.

1.2 Who is KompoZer for?

KompoZer is suitable for anyone wishing to have a modern, free of charge, program for developing small web sites and who would like to learn modern web design techniques such as the use of CSS.

1.3 Scope of the guide

The guide has been written from the point of view of the Windows Operating system so details may differ for other systems. It is intended as a reference document although in a few places it uses a tutorial approach.

Although web pages can appear on screen, in print or in audio the guide is confined to on-screen uses.

1.4   Methodology

The approach taken by the guide attempts to satisfy the needs both of the comparative novice who has little knowledge of web design tools, and those making the transition from other tools and who may have considerable knowledge of the field.

Although the Guide is not intended for the complete beginner, concepts are introduced progressively and gradually added to. To avoid being too repetitive I have, in a number of places, referred to subjects yet to come. This applies, in particular to the use of styles in which Nvu is strong. I suggest that these references are disregarded at first reading and returned to later. This will not prevent understanding later parts.

1.5 Navigation

Navigate around the ten sections of the Guide and the appendices using the menu pane on the left hand side. In each section the menu expands to show more detail within that section. Cross references throughout are hyperlinked.

The buttons at the top of each page give access to some supplementary aids and information. The 'Site map' contains a detailed table of contents of the whole guide while the 'Quick index' gives access by subject.

1.6 KompoZer and Nvu

The development of Nvu was completed with the release of version 1 in June 2005. No further development is expected though the developer Daniel Glazman of Disruptive Innovations http://disruptive-innovations.com/ is working on a newer product which will probably supersede it. After version 1 release a number of shortcomings were revealed and an independent developer using the name of Kazé worked on several of these and released an enhanced product as KompoZer.

Early versions of KompoZer operated almost identically to Nvu but with version 0.7.10 a number of significant enhancements were introduced, most of them related to the use of the CSS editor CaScadeS, and the majority of Nvu users have probably transferred to using this version.

The program is completely free and can be downloaded from KompoZer's web site http://kompozer.sourceforge.net/.

You can install both Nvu and KompoZer on the same machine. From version 0.7.10 it is possible and safe to run both programs simultaneously.

This User Guide is written around the latest version of KompoZer at time of writing (see footer) and the methods described apply to that version but generally also to Nvu. Where the method for Nvu differs click the icon in the margin to reveal this.

In this User Guide where the term KompoZer is met it should be interpreted as including Nvu unless a distinction is made.

2 Quick start

2.1 Preliminary

Open KompoZer

The main window opens. At the top are a number of toolbars. The topmost is the Menu Bar. This carries a number of items (File, Edit etc) used to make selections. The next is the 'Composition Toolbar' which carries a number of 'Buttons' labelled 'New', 'Open' etc.

2.2 To create a new page:

On the Composition toolbar Click the 'New' button.

2.3 To open an existing page

Assuming that the page is stored on your local disk in HTML format:

On the menu Bar click 'File' then 'Open File'. Browse to the file and click 'Open'.

2.4 Editing a web page

Your web page – blank or otherwise – is in the large pane in the centre right of the KompoZer application window. Many editing functions are very similar to those in a word processor. The top four toolbars on the KompoZer application window provide a number of editing functions – to see what any do hover the cursor over an item and a hint will appear.

2.5 Saving a Page

To save a page:

On the Composition toolbar click 'Save'.

If it was a new document a dialog window will ask you to enter a title for the page. This will appear in the tab at the top of the page display area. NB this is NOT the file name. Click 'OK'; you will then be offered a normal save window which allows you to browse to a suitable location and name the file. The file extension offered will be HTML.

2.6   Browsing a page

To see how your page will look in your default browser on the Composition toolbar click 'Browse'.

2.7   Help

The help system should be a first resort in case of need. KDS Sahambihttp://nvu.kellertechnologies.com/. has written a useful tutorial for beginners which you may like to look at. It is shorter than this User Guide.

The forum at http://wysifauthoring.informe.com/ is a place for sharing experiences and obtaining and giving help. Many of the contributors are users of the programs and range from beginners to those with lots of experience.

3   Using KompoZer
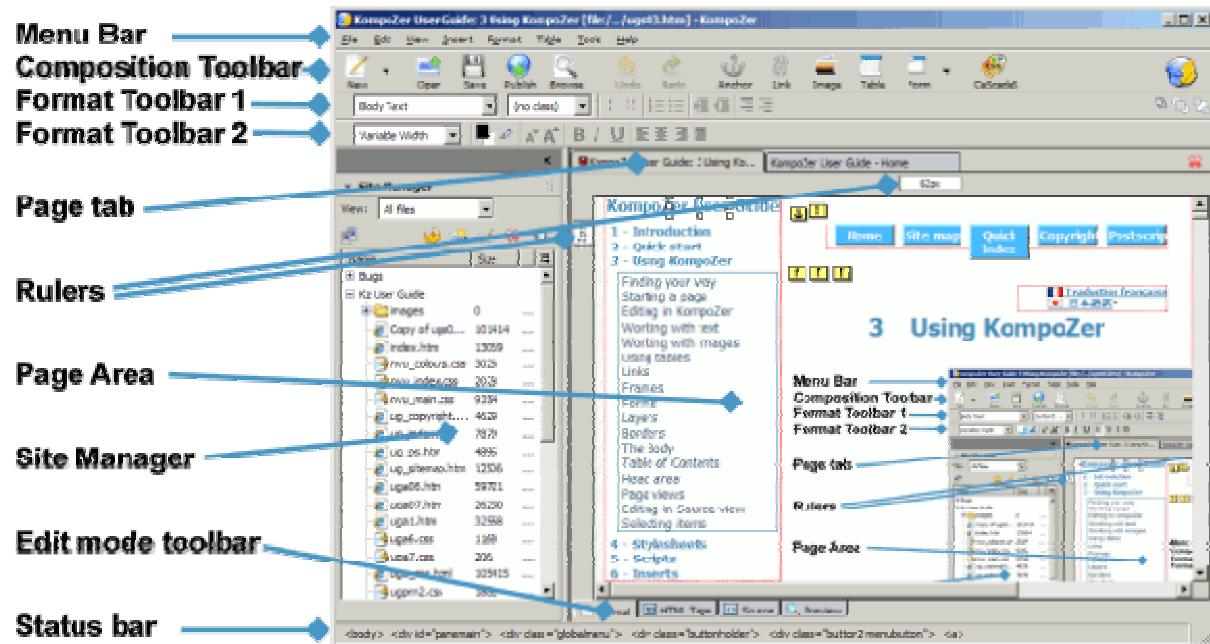
3.1 Finding your way around

When KompoZer starts the window carries a menu bar across the top (File, Edit, View etc).
Below this are three 'Toolbars'. To ensure that everything is visible on the Menu bar select
View > Show/Hide and see that each of the following is checked: Composition Toolbar,
Format Toolbar 1, Format Toolbar 2, Edit Mode Toolbar, Status bar, Rulers, Site Manager.

The three toolbars across the top of the window carry buttons (represented by icons). Hover
on any to find its function. If any are greyed out they are not functional in the current mode
as they are context sensitive.

Note Not all buttons are installed initially. The full list is given in section 9.3 Customising
toolbars

Across the centre of the window are two panes: the 'Site Manager' on the left and a blank
web page on the right. The Site Manager is a powerful tool which is covered in section 8.
Since it is not needed yet it may be closed by clicking on its close button or pressing F9.

At the top of the Page area there is a Page tab which carries the name of the page
('Untitled'). (KompoZer Help refers to this as the 'Tab Browser' toolbar.) If you had several
pages open, as shown in the figure, this tab would allow you to select one them rapidly.

At the bottom of the page area is the 'Edit Mode Toolbar' which carries four tabs which
select one of four 'Viewing modes' for a page ('Normal', 'HTML Tags', 'Source', 'Preview'.)

For details of viewing modes see section 3.15.

At the bottom of the window is the 'Status Bar'. This is a very powerful tool. Once a page is populated, by clicking any item in the page area its structure appears on the status bar. Any class or id applied to an element is shown and any bearing an inline style is indicated in italic type. Hovering reveals the style declaration. Additionally clicking an element marker highlights the element in both normal view and, on changing view, in source view thus simplifying navigation in source view.

Note The figure shows the buttons as they appear when KompoZer is first installed. They may be customised to different arrangements (see section 9.3). If this has been done some of the following may be difficult to follow. To restore defaults click View > Show/Hide > Customize Toolbar > Main Toolbar > Restore Default Set and repeat similarly for the Format Toolbar.
Using KompoZer right-click any toolbar to customise it.

3.2 Starting a page

A new page can be created either from a blank page or by opening any pre-existing page.

Note There are often several ways of getting KompoZer to do something. This guide will use the button on the Composition toolbar if that is available. For menu equivalents see section 9.3 Menu alternatives

3.2.1 Document types (Doctype)

Web pages use the HyperText Markup Language (HTML). There are number of variants of the basic language some details of which are outlined in Appendix 3. KompoZer is capable of creating and editing four variants but for simplicity of explanation **I recommend that users create the Strict HTML variant. I also recommend that you go the menu item Tools > Options > General and check the box 'Use CSS Styles instead of HTML Elements ...'** this will give you the maximum functionality. You can edit documents which have been originated in other variants and again 'Use CSS Styles ...' gives maximum functionality.

Note If you find a button or menu item disabled it is likely to be a because 'Use CSS Styles ...' is not set but could be because of the context of the action.

3.2.2 Options for starting a page

There are several ways to start new pages or open existing ones.

To start a new blank page.

On the menu bar click File > New.

A window headed 'Create a new document or template' appears. Check the boxes 'A blank document' and 'Strict DTD' and clear 'create a XHTML document'. (Without Strict DTD checked the document would have 'Transitional' DTD.)

Note If you wish to make these your defaults which are offered every time you create a new page click Tools > Options > Advanced and check the boxes Markup Language: 'HTML 4' and DTD: 'Strict'.

The use of templates is covered in <u>section 7</u>.

To open an existing page.

Either:

1. ▣Click the OPEN button to access a normal browse dialogue.
2. Click File > Recent Pages to get rapid access to those recently worked on.
3. Or use the Site Manager (<u>section 8</u>) which provides a powerful mini-browser and is very easily set up.

The doctype of an existing page will remain the same as before it was opened. It cannot be changed in KompoZer.

Each page opened starts in a new tab which can be clicked to select a document to work on.

3.3 Editing in KompoZer

Tip: The plus sign (+) means that you must press two keys surrounding the plus sign at the same time.

KompoZer supports all the standard Windows editing commands and shortcut keys. e.g. Copy Ctrl+C, Paste Ctrl+V etc. There are other KompoZer specifics. These are great time savers.

In 'HTML Tags' view KompoZer supports drag and drop editing for block items. (Select an item by pressing the Control key while clicking on the Tag.)

An extreme time saver is KompoZer's double click response. In several cases, such as links, images and tables, a very useful editing window is opened.

KompoZer supports many levels of Undo and Redo, however changes made in 'Source' view cannot be undone after you have changed the view.

3.3.1 Tutorial – Brief introduction to HTML

HTML stands for HyperText Markup Language. It is the publishing language of the World Wide Web on which KompoZer is based. Ideally it should be possible to use KompoZer without any knowledge of this language and certainly without a knowledge of the details. As

a precaution however a limited understanding is desirable. This tutorial covers a minimum of detail while introducing some useful features of KompoZer.

Start a new blank page as described above.

On the status bar click on the '<> Source' tab. You will see a number of terms enclosed within angle <> brackets. These are HTML **'Tags'**. Everything starting from the <**body**> and finishing with </body> will appear on your web page.

Note In most cases tags occur in pairs – opening tags and closing tags – Closing tags always start with a forward slash.

All the page consists of now is a <br> tag which is a line break and which is, of course, invisible.

Later as a page builds up many more tags will be added. For the most part the pairs of tags enclose **'contents'** from which the page is built up – items like Paragraphs, Headings, Tables, Images. The tags plus the contents are called **'elements'**. The tag names are usually either the same as element names or abbreviations of them. Some examples are 'table', 'img' for image, 'p' for paragraph, 'br' for line break (which is unusual in having no closing tag).

Nothing between the <**head**> and </head> tags will appear on the screen but may have quite useful functions nonetheless. For instance, between the <**title**> and </title> tags there isn't anything. Go for it, between them type "My first page" or something like that. Click the tab for 'Normal' view, now the tab at the top carries your page title and you've edited raw HTML. Well done!

If you were too timid to do this all is not lost. There is no need to use source view at all. Go to Format > Page Title and Properties. Here you can enter various parameters including the title and the language you are writing in. These will be entered in the HEAD area for you. In some circumstances they can be important; the Title and Description, for instance, can be used by search engines to index the page.

Back to '<> Source'. Notice the first tag is <**html**> and the last </html>. All web pages must carry these in these positions. Actually the case doesn't matter for any tag though it's good practice to use lower case. (For XHTML lower case is mandatory.)

The first line, in a different colour, specifies the **'Doctype'**. This is a reference to the type of HTML document. KompoZer determines the doctype according to the options selected on starting a page.

Tip: Filenames should generally be composed of unaccented letters, numbers and the underscore character. Some others are permissible but the rules for incorporating them are more complex.

Tip: Web pages are stored (hosted) on servers which usually run either a version of Windows or UNIX operating systems. UNIX filenames are case-sensitive while Windows

are not. For this reason if you use mixed cases in filenames you must be very rigorous. A site which runs perfectly in your local Windows environment or on a host running Windows if transferred to a UNIX system might require extensive and time-consuming changes.

Finally, give the document a name and save it.

3.3.2 Saving files

Go to File > Save as.

You are offered a 'Save Page as' window which allows you to browse to the folder you want to use. You will find the file name already completed with your page title. You will probably want to change this to a shorter, all lower case, name (but you don't have to). You will find the file extension completed as 'html' you may prefer to, and you may alter it to "htm".

Note If you have hidden extensions for file types this becomes unavailable. Hide or view extensions is controlled in Windows by Control Panel > Folder options > View then scroll down and clear 'Hide extensions for known file types'.

After you edit files you need to re-save periodically. Click the 'Save' button.

Note KompoZer includes a small red warning icon in the page tab to indicate if a page has not been saved.

There is nothing mysterious about the files. The curious may open the file using any text editor e.g. Windows notepad. What you will see will look just like the HTML Source code you've been looking at in 'Source' view. This shows that the source is just a simple text document.

3.3.3 Printing pages

The 'Print' button allows you to print the current page to a printer. This prints the page view and not the source code.

3.4 Working with text

3.4.1 Text on a page

3.4.1.1 Introduction

Text typed directly onto the KompoZer page defaults to appearing in the format for the 'body' element. HTML defines a small number of elements specifically for text and it is usually preferable to use these.

To format text in a standard element format select the text (it is actually necessary simply to click anywhere within the block of text) and click the first drop-down box on the format toolbar. This offers a selection of standard text formats. Paragraph is the most appropriate for general text.

Once formatted as a paragraph, when typing in a text area, use of the 'enter' key starts a new block of text i.e. a paragraph. To start a new line within the current paragraph press Shift+Enter; this generates a line break.

Note This is the default action and can be altered. (See Toolbars section 9.3)

Other standard text formats are Heading formats from Heading 1, the largest (for the main heading), to Heading 6, the smallest (for the least significant). Browsers generally render headings in bold type.

Text can be edited in any of the viewing modes and KompoZer responds to all the normal windows shortcut commands.

3.4.1.2 Block or Inline

Tip: The outline of each block of text can be revealed by selecting View >Block outline. The dimensions of each block in pixels are shown on the rulers at the top and side of the page.

The elements just described mark out 'blocks' of text.

- Blocks usually begin on a new line.
- Blocks have gaps above and below controlled by margins (see section 4.2.1.4).
- Vertical margins 'collapse' i.e. The larger margin is applied the smaller is ignored.

Within blocks it is possible to mark out sections of 'inline' text.

- Inline text does not usually start a new line.

3.4.2 Formatting text

Tip: Text to be formatted may be selected in one of several ways.

- Simply place the cursor anywhere in a block. Where the formatting desired can only apply to a complete block this is sufficient.
- Double-click within a block to select a word.
- Triple click within a block to select a line of text.
- Drag across a section of text to highlight and select that section only.
- In HTML Tags view press the Control key while clicking on the tag for the block to select the complete block.
- Using the status bar. See section 3.17

Text can be formatted in a number of ways using a format toolbar. The changes listed in the table can be applied (hover over a tool to discover what it does).

Choose a font *
Choose text colour *   Choose background colour *
Choose highlight colour *
Make text smaller *   Make text larger *
Embolden * Italicise *   Underline *
Format as a numbered list Format as a bulleted list
Align left   Align Centre Align right   Justify
Indent text   Outdent text
Emphasise *   Strongly emphasise *

Note 1 For the options marked with an asterisk select an inline section of text. This allows you to have a few words in a paragraph of a different size etc. (When you do this, if you use the 'HTML Tags' view, you will see the start of the change marked with a 'Span' marker.)

Note 2 'Choose background color' sets the colour for the background of the current block. If 'Use CSS Styles ...' is not checked, with KompoZer it changes the background colour for the complete page, with KompoZer it has no effect.

Note 3 The emphasis button will result in italicised text with most browsers and the strong emphasis button in bold text.

Warning. When using Strict Doctype check 'Use CSS styles instead of HTML elements and attributes' in Tools > Options > General to ensure availability of all options.

3.4.3 Positioning text

Generally the position of text is determined by the fact that one block flows on after the previous one etc.

The space between blocks is determined by the browser. Through the use of styles (see section 4.2.1.4) these settings can be altered.

Paragraph and heading blocks stretch from one side of the screen to the other. This a very elegant way of operating. Try resizing the KompoZer window. Until it becomes so narrow that nothing can fit, the text reformats so that it still fits – though narrower it becomes longer. It is fundamental to the way the web operates that it is designed for information transmission and not presentation. Web design tools allow us to refine presentation but, if we over constrain it, we may run into problems of getting it to fit on the screen. We really should allow the browser as much freedom as possible.

3.4.3.1 Fixed width text

KompoZer allows you to constrain the width of paragraphs. (The same applies to headings.) Try this. Set the window size so that it is somewhat narrower than the full screen width i.e. so that you have space either to narrow it more or to widen it.

Take two longish but similar paragraphs which fill about half of the window. Click in one, you will see the rulers both across the top and down the side. Grab the left edge of the ruler and draw it half way across the window. Do the same for the other paragraph but take the right edge of the ruler and drag it half way to the centre but in the opposite direction.

You probably think that you've achieved a rather elegant layout and discovered a flexible tool. Drag the window to be wider then narrower. Do you still think so? What you have done is to set the text to be in a fixed position and of fixed width. It no longer re-flows. Consider that visitors who view your pages may have smallish or very large screens. Will they all see what you want them to?

Tip: Instead of using HTML Tags view it is quicker just to click in the paragraph and click the corresponding <p> marker on the status bar. This will highlight the paragraph. After that just right click as described.

Supposing that you don't like what you have just done – you can just Undo it – press the Undo button or click Ctrl+Z. However if you leave it to later you would first have to undo all later work. You can re- or de-format a paragraph directly and individually. This allows you precise control rather than relying on dragging the ruler. Go to 'HTML Tags' view, click on the yellow 'p' marker for a paragraph, on the status bar right-click on the highlighted 'p' area and select 'Inline styles'. In the general tab that appears just delete the text and click 'OK'. Alternatively select 'Box Properties'. This allows precise editing of the paragraph within the 'box' in which it is contained.

3.4.4 Lists

3.4.4.1 Numbered and Bulleted lists

KompoZer can format a list of items giving each item a sequential number in any of several formats (HTML calls these ordered lists <ol>) or presenting them bulleted (unordered lists <ul>).

To start a list from scratch

1. Click one of the list buttons (Numbered List or Bulleted list) on the Format toolbar.
2. Type the first item.
3. Press Enter and type the next item.
4. To finish, on the last (blank) item press Enter

To change existing text into a list

1. Select the text required.
2. Click one of the list buttons on the Format Toolbar.

The text will be changed into a list a new item starting for each paragraph or other block item encountered.

To add items to a list

1. Click at the end of the last item in the list.
2. Press Enter and type the new item.

Numbering and format will continue from the previous item.

3.4.4.2 Formatting lists

To change the formatting for a list click in the list and, on the menu bar, select Format > List > List properties and choose the preferred option. Alternatively List properties is accessible by right-clicking in the list.

For numbered lists the numbering style may be selected, for bulleted lists the type of bullet may be changed.

3.4.4.3 Nesting lists

Lists can be nested.

To start a new level of list within an existing list

1. Click at the end of the line preceding the new list to be generated.
2. Press Shift+Enter
3. Click a list button on the format toolbar.
4. Type the content of each list item followed by Enter.

Each level of list in a nested set may have either the same or different format which may be set as above. KompoZer sets no format by default. Browsers normally by default render all numbered lists with decimal number format but each level of bulleted lists using a different bullet symbol.

3.4.5 More formatting

The menu selection FORMAT offers many ways of formatting text. In some cases you can simply click within the text region in others you must select the section of text involved. Some of the effects may be repeated several times for more effect, some will reverse on a second application.

The options offered include indenting text. Those unfamiliar with web design may look, in vain, for a tab option. HTML does not support tabs although the List feature does produce a tabbed effect. KompoZer substitutes for this in that pressing the Tab key introduces several spaces – of course these do not produce the alignment effects that real tabbing would. The spaces introduced by the Tab key are technically known as non-breaking spaces. Pressing the space bar more than once also introduces non-breaking spaces.

3.4.6 Importing text

Strictly KompoZer does not support importing text from other applications but it is possible to copy and paste text.
In normal view content from other web pages may be copied reliably. The result will be rendered according to any styling applied in your document; any reliance on external styles in the original document will be lost. See section 4 for more on styles.
Text from word processors such as Microsoft Word or OpenOffice.org in rtf or doc format or from text editors such as Windows notepad may also be copied and pasted.

Warning Word processor data should be copied in .doc or .rtf form. Word processors can also save pages in html form but this is unsatisfactory because the code is of poor quality. It will be almost impossible to alter the formats in KompoZer. To use such code first copy to a text editor, which removes all formatting, and then copy from the text editor to KompoZer. Apply any required formats from scratch.

When such text is pasted into KompoZer most formatting is lost.

Numbered lists will be retained.

Tabs will be rendered as three non-breaking spaces.

The contents of tables may be pasted, individual cells will be separated as if by tabs.

3.4.7 Special characters

Character		Entity	Decimal
nrule	–	–	–
mrule	—	—	—
ellipsis	…	…	…
left single quote	'	‘	‘
right single quote	'	’	’
left double quote	"	“	“
right double quote	"	”	”
infinity	∞	∞	∞
tilde	˜	˜	˜

capital Alpha   A   Α Α

lower case alpha   α   α α

and so on for other Greek letters

By special characters we refer to characters which do not have an equivalent keyboard key. HTML uses a system of characters known as 'Unicode'. This includes a large range of characters including all the international currency symbols and many thousands of others, though the fonts supplied on computers will support only a subset. A number of the more common, including accented ones, may be inserted using Insert > Characters and Symbols.

KompoZer does not directly support 'smart quotes' but text which has been copied from applications like Microsoft Word or OpenOffice.org which include smart quotes will retain these details and they will be correctly rendered in browsers.

If you need a character which is unobtainable by these methods there are several alternatives available.

1. Using the Windows Character Map – Find, copy and paste a character.
2. Directly from the keyboard – while pressing the ALT key input a zero followed by the three digit decimal code for the character. This code is shown for a number of characters at the bottom right of the Windows Character Map.
3. By editing the file in 'Source view'. See section 3.16. Characters may be inserted either using the decimal code for the character or using a mnemonic 'entity'. Precede the decimal code with &# and the entity with & and follow both with a semicolon. Use no spaces. The HTML specification (see appendix 2) in section 24 lists the codes.

Some useful examples are shown in the box.

There is more on special characters in appendix 6 sections A6.3.7 and A6.3.8

3.4.8 Checking spelling

Simplicity itself. In any view, other than 'Source', click on the 'Spell' button. The spell checker will work sequentially through the page. You will be offered one dictionary but you can add your own words to it. If the Spell button is not installed see Section 9.3

By first highlighting a section of a page only that section will be checked.

It is also possible to run the spell checker all the time. Any mis-spelled words underlined in red. To set this option
on the Menu Bar select Tools > Options. In the window which opens select the 'Advanced' button and set 'Underline misspelled words' – OK.

Warning This setting must be made after the page is opened. Otherwise cancel and re-set.

Note Spell checking uses enormous processor resources. Open only one document at a time when spell checking. When working on long documents, if your system grinds to a halt, disable this when not required.

3.4.8.1 Custom dictionaries

If you add words to the spell checker you create a custom dictionary. The spell checker will mark words as errors if they do not appear in either the standard or custom dictionaries. The custom dictionary is saved as a text file as persdict.dat in the chrome folder in your profiles folder. Refer to Appendix 4.2 for the location of profiles.

3.4.8.2 Installing other dictionaries

If you use a language other than that of the default dictionary or prefer a dictionary customised to regional spellings it is possible to install alternatives to cater for this.

With alternative dictionaries installed you can select which to use.

To select a dictionary.

1. Click the 'Spell' button.
2. In the 'Check spelling' window that opens click the drop down arrow in the 'Language' box.
3. Select the language required.

The language selected will become the default until intentionally changed.

Spelling will use this selected dictionary plus any custom dictionary.

If you have more than one profile in use the dictionary settings must be made, as required, for each profile. You can find details of profiles in Appendix 4.2.

Acquiring dictionaries

Dictionaries may be available from a variety of sources one of which is at http://dictionaries.mozdev.org/installation.html

Dictionaries are formatted with xpi extensions.

To acquire a dictionary first identify a source. Then

1. Open the source page in a browser.
2. Right click on the file.
3. Click 'Save link as'.
4. Navigate to a suitable folder (any will do)
5. Click 'Save'.

To install a new dictionary

1. Click Tools > Extensions > Install.
2. The 'Select an extension to install' window opens.
3. Navigate to the location where the xpi file has been saved, select it and click 'Open'.
4. The 'Software installation' window opens.
5. Click 'Install now'.

When the Spell Checker is next called the new dictionary appears in the drop down list.

[I am grateful to Greg and Elenora on the Forum for much of the above detail.]

3.4.9 Viewing text

The main page editing pane acts rather like a browser in that the text size may be viewed at different magnifications (this has nothing to do with the code for the final page). Set this using View > Text Zoom. This has no effect in Source view.

3.5 Working with images

With KompoZer it is easy to insert images on to your page. Browsers accept images in three formats 'gif', 'jpg' or 'png'. You may have to use a graphics package to get your image into a suitable format. You can insert an image located anywhere on your computer but, for the sake of order, it is a good idea to locate any images which you intend to use either in the same folder (directory) as your page document or in a sub-folder of this.

3.5.1 Inserting images

Warning Before inserting an image save your page. Otherwise KompoZer will have difficulty in referencing where your image is located.

As an exercise set your cursor in the middle of the paragraph, perhaps at the start of a sentence.

To insert an image

1. Click the 'Image' button on the Composition toolbar
2. The Image properties window opens. Click 'Choose File' and browse and select a file
3. Click 'Open'. Leave checked the box 'URL is relative to page location' this will allow you to move the page and image to another location, as you will have to when you upload them to a server. (If you de-select this and move the page, it will try to find the image at the original location.)
   Note If the box is 'greyed out' this is probably because the page has not been saved.
4. In the box labelled 'Alternate text' add a description of the image. (This forms the 'alt' attribute for the image and provides text which will appear in place of the image

with user agents (browsers) that cannot display images (screen readers and voice synthesisers). It will also be used by those with visual impairment. The content of this box must be carefully considered so as to be of maximum assistance in such cases.)
Note Where the image is purely decorative, and not necessary to understanding the page, alternative text is not required and should be omitted

5. In the box labelled 'Tooltip' you may optionally insert a 'Title' attribute for the image. Some browsers will show the text provided when the cursor hovers over the image
6. Click OK

3.5.2 Positioning and resizing images

You should see the image inserted into the middle of the paragraph. Maybe it has come out some ridiculous size. Click the image. A sizing box appears around the image. The box has 8 graphical handles (small white squares) one at each corner of the image and one in the middle of each side).

Drag on a handle to alter the size of the image. Make it some sensible size. Dragging a handle in the centre of a side will alter the shape of the image which is usually undesirable.

More precise control may be achieved via the Image properties window described after the note. Select the 'dimensions' tab.

Note Ideally images should be produced at the size required rather than resizing them. Importing large images and reducing the size wastes bandwidth and slows presentation. Resizing by small amounts can introduce undesirable artefacts.

Tip: The method described places the image within a paragraph. You may prefer to place the image directly in the body area particularly if you are familiar with using styles and want to control the image that way. You can get directly to the body using the method described in the tip in section 3.16.

Then just insert the image.

Images in the middle of paragraphs are not usually what is needed. You can place them at either the right or left side and have the text wrap round.

To align an image

1. Click on the image.
2. Click the Image button on the Composition toolbar: this will open the Image properties box.
3. Select the 'Appearance' tab
4. In the drop down box 'Align text to image' select the required alignment.
5. Click OK

Try the various options. The text starting from where the image is inserted will wrap around the image.

You may prefer the complete paragraph to wrap. Click on the image and drag the cursor to the start of the paragraph, or anywhere else. Moving images is easy!

You may find the first three options confusing. They refer to the position of the text relative to the image NOT the image relative to the text. Specifically you will find no way of putting an image in the middle of the page with text wrapping around it.

Note This type of presentation, common enough in print, is almost impossible to achieve using HTML even with full use of styles.

You may decide that the image is too close to the text, that you would like more space around it. Again 'Image Properties' allows this. On the 'Appearance' tab set 'Spacing' Top and Bottom and/or Left and Right as required. You can also put a border of selected width around a image. The border will be the same colour as the adjacent text.

An alternative way of aligning an image is to apply a style via a class as explained in sections 4.2.1.4 and 4.2.3.

3.6 Using tables

Tables allow data – images, links, forms, form fields, other tables, text, etc. – to be arranged into rows and columns of cells.

A table is basically a rectangular grid forming an array of boxes into which the data are placed. The **rules**, between cells, and the **border**, around the outside of the table, may be either visible or invisible. The boxes are called **'cells'** and may be coloured. The grid layout does not have to be regular – the first row may have one cell the second four, the third three; however, when first inserted, it will always be regular.

W3C recommends:

Tables should not be used purely as a means to layout document content as this may present problems when rendering to non-visual media. Additionally, when used with graphics, these tables may force users to scroll horizontally to view a table designed on a system with a larger display. To minimise these problems, authors should use style sheets to control layout rather than tables. (HTML 4.01 Specification Section 11.1)

3.6.1 Inserting tables

To insert a table

1. On the Composition Toolbar click the Table button. The 'Insert table' window appears
2. Leave the 'Quickly' tab selected and drag out a matrix then click the bottom right cell to define the table arrangement.
3. The cells appear on the screen with narrow outlines

Note If later the table border is set to zero these outlines disappear but KompoZer in normal view replaces them with a red outline. This does not appear in Preview or in a browser.

Tables have resizing boxes similar to those used with images

Alternatively for more control

1. On the Composition Toolbar click the Table button. The 'Insert table' window appears
2. Click the 'Precisely' tab
3. Specify the number of rows and columns required
4. Specify the width of the table either in pixels or as a percentage of window size. Alternatively clear the 'width' box. This will give 'fluid' width which will automatically adjust to the contents
5. If a border is required around the table enter the width in pixels otherwise enter '0'. If a border is specified each cell will also have a border but narrow

Note CSS styles give fuller control of borders. See section 4.2.1.3.

For more options

Right-click the table and select 'Table Cell properties'. The Table properties window opens. This has two tabs 'Table' and 'Cells' which allow overall control of several aspects of either the table or individual cells. This includes

a. Alignment of text within cells
b. Wrapping of text
c. Cell spacing - the gap between cells
d. Cell padding - the gap between the edge of the cell and the text within it
e. Size of table and cells
f. Background colour
g. Selection of cells as 'Normal' or 'Header' (Cells which are headings to rows or columns should be selected as 'Header'. Normally this results in them being rendered in bold type.)

3.6.2 Formatting tables

3.6.2.1 Table size

Tables created using the 'Quickly' tab have width set to 100% of the window size.

Note Sometimes KompoZer creates tables at a different width in which case this may be altered as given below.

Tables at full window width give the maximum available space for data. The width accommodates to screen or window size.

It is possible to set the width to a smaller percentage of window size or to a fixed size in pixels. Fixed sized tables are inflexible in use and could result in the user having to scroll horizontally to read the content so must be used with caution. It is also possible to leave the width unspecified. This is likely to result in the best presentation as the browser will set the width to the optimal value.

To adjust table size, with the cursor in the table, click the table button (or double-click the table). In the Table Properties window select the 'Table' tab. The width can be set either as a percentage of window width or in pixels. To leave it unspecified clear any number in the width box.

It is possible to adjust the size of a table using the sizing boxes. When you do this the size is specified in pixels.

3.6.2.2 Colouring, adding deleting and merging cells, rows and columns

Right-Click in a table or cell and several menu options appear which allow the table or cell to be formatted as desired. Some of the options are context sensitive.

'Table or cell background color' allows the colour of the complete table or any cell to be set.

'Table Insert' provides means of adding rows, columns, individual cells or even tables into a cell.

'Table delete' provides the opposite.

A cell can be joined with the cell on its right, thus merging two cells. Virgin cells cannot be split but joined cells can.

If adjacent cells (whether within a row or column) are selected together they can be joined.

Each cell, when clicked, reveals a set of six manipulation symbols (triangles and circles with x inside), these provide a rapid method of inserting or deleting rows and columns of cells as shown in the figure.

Setting the cell background colour for adjacent cells to a dark colour reveals gaps between the cells.

This may be convenient if the cells are to be filled with text as it avoids two items of text abutting each other.

At other times it is a problem. The gap is controlled by the 'Cellspacing' attribute.

3.6.2.3 Background images

It is possible to set a background image for a table or cell.

The CSS Specification in section 14.2.1 recommends

When setting a background image, authors should also specify a background color that will be used when the image is unavailable. When the image is available, it is rendered on top of the background color. (Thus, the color is visible in the transparent parts of the image).

The image appears behind the contents of the table. If the image is larger than the table the table is unaffected i.e. does not grow to accommodate the image, as much of the image as will fit is displayed. If the image is smaller than the table several options are offered e.g. the image may be centred or tiled horizontally, or vertically or in both directions until it fills the table.

To apply a background image

1. In 'HTML Tags' view click the tag for the table (or cell) while pressing the control key.
2. On the Status Bar right click on the highlighted tag.
3. Select 'Inline Styles', 'Background Properties'.

You can then select the tiling and/or position properties as required.

Note Background images are visible in Normal or Preview mode but not in HTML tags view.

3.6.2.4 Setting cell border and spacing

Right-click in a table, select 'Table Cell Properties' and the 'Table Properties' window appears. By selecting the 'Table' Tab the spacing is revealed as defaulting to 2 pixels between cells. If this is set to zero the gaps referred to in the previous paragraph disappear.

Note The 'Table Properties' window can also be accessed from within a table by clicking the 'Table' button.

The 'border' defaults to 1 pixel. This border appears around the edge of the table and rules appear around each cell rendering the outlines easily visible. If the border is set to zero both border and rules disappear.

If the border is set to a value greater than 1 pixel the rules re-appear and the border appears in the wider value specified. The exact appearance of the table border depends on the browser used.

Other aspects of these lines e.g. colour or suppressing rules when table border is selected cannot be controlled using these methods.

Note For full control of borders use a Stylesheet (see section 4.2.1 especially 4.2.1.3).

The 'Table Properties' window also allows the 'padding' to be set. Padding is the distance between the cell content, e.g. text, and the edge of the cell. This provides an alternative way of establishing space between the content of adjacent cells. Padding applies to all four edges of a cell and to all the cells in a table. Once again use a style for fuller control.

3.6.2.5 Colouring tables

This has already been referred to in section 3.6.2.2. KompoZer however provides several routes to colouring tables or cells. These include:

- Right-click on table and select Table Cell Properties.
- Use Table button to open Table Properties window.
- Right-click in table and select 'Table or Cell. Background Color'.

3.6.2.6 Cell sizes

As you work on a table the sizes of the cells may keep on changing. It is useful to understand how browsers treat tables. Unless the size of a table or cell is constrained in some way a browser will expand the cell in an attempt to accommodate any text inserted. Other cells in the same column will have the same width. This will tend to reduce the space available for any adjacent column of cells. Eventually the browser will attempt to balance the allocation to each column by wrapping the text. The overall effect will probably minimise the total height of the table.

Tables can have their width completely unspecified, specified to be a particular width in pixels or some percentage of window width.

When a table is started the Insert Table automatically selects the 'Quickly' tab. This allows KompoZer to make the decisions. If the 'Precisely' tab is selected the user can make selections specifying the width in pixels or as a percentage and specifying the border width. Tables specified as percentage or unspecified are more accommodating to differing window or screen sizes and are recommended.

Browsers do not have to slavishly follow instructions. If a table specification is too narrow to accommodate the cell contents the table will grow beyond its specified limit. This can occur if a large image is inserted or a long word like 'Antidisestablishmentarianism'.

3.6.2.7 Table properties

The table properties window opens when you

- Click a table and click the table button
- Right click anywhere in a table and select 'Table cell properties'.

Tables and cells can be re-formatted in many ways. This means that you don't have to plan everything in advance before you start your table as settings can be adjusted later.

The table properties window provides an option to insert a caption which can be positioned on any edge.

For those familiar with them, the 'Advanced Edit' button gives direct access to inline styles.

3.6.3 Inserting Text

Tip. Use HTML Tags view to see where to input the text.

Inserting text into a table cell is one of the simplest of tasks. Inserted text will be formatted as a TD element but may be re-formatted as a paragraph or using styles.

Even the simplest of can tasks become tedious if carried out repetitively possibly copying and pasting from pre-prepared text.

After selecting a piece of text KompoZer offers a useful facility via Tables > Create table from selection. Lines of text (whether terminated by line breaks or paragraph ends) will be converted to table rows, a new cell being generated each time the specified character is met. This allows tabbed lists to be created and overcomes the fact that HTML does not support tabs – Simply type a hyphen (or some other character) instead of the tab and convert the text to a table.

Using this method tables may be imported from applications like MS Word. In Word convert the table to text separating using a character that does not appear in the text.

Note Do not use tab as the separator as KompoZer will interpret this as several spaces. (See section 3.4.5)

3.6.3.1 Aligning text

Browsers normally display text aligned left and, in the vertical dimension, in then middle of the cell. To alter this place the cursor in a cell and click the Table button. The Table

properties window allows the alignment to be altered. To set the alignment for all cells first select Table > Select> All Cells.

3.6.4 Centring tables

To align a table to the centre of a page click the table, right-click 'Table Cell Properties', select 'Table' and set 'Table Alignment' to 'Center'.

Note If you are using Nvu (not KompoZer) to create a page using transitional doctype Internet Explorer aligns tables left. This may be corrected by editing the code in Source view to precede the Table with the code `<div align="center">` and follow the table with `</div>`. **N.B.** This code would not be valid to use for pages which have strict doctype.

3.6.5 Improving tables

There are a number of ways in which tables may be improved in presentation and accessibility. These include:

Caption - This is text describing the nature of the table. Insert this using the Table Properties window. The caption may be positioned on any edge of the table.

Summary - Captions will often be inadequate as a summary of the purpose and structure of the table from the perspective of people relying on non-visual user agents. W3C recommends "Authors should therefore take care to provide additional information summarizing the purpose and structure of the table using the summary attribute of the TABLE element. This is especially important for tables without captions." To add a summary, in the Table Properties window, select the Table tab and click the Advanced edit button. On the HTML attributes tab select the Summary attribute and insert the text in the 'Value' box.

Headings - Headings to rows and/or columns improve the visual presentation and make the meaning of the content clearer. By default headings are displayed bolder and centred. Headings are implemented in HTML by replacing td (table data) elements by th (table header) elements. To change a cell from td (normal cell) to heading select the cell and, using the Table Properties window, with the 'cells' tab selected tick the 'Cell Style' checkbox and set the style to 'Header'.

Accessibility - Adding summaries and headings represents the first step in improving accessibility of tables to non-visual users. HTML offers more powerful enhancements which improve the performance of text to speech devices. For more details about this see section 11.4 of the HTML 4.01 specification.

3.7 Links

Links provide the main means of navigating web sites. They allow you to move quickly from one place to either another on the same page, a different page on the same site or a page on an external site. Links may be attached to any element on a page, in fact almost any part of a

page content can form the basis of a link an element being generated specifically for the purpose. Usually however we use a few words of text or an image which we click on to activate the link.

3.7.1 Linking text

3.7.1.1 Linking to another file

To create a link

1. Select (highlight) a few words of text
2. On the Composition toolbar click on the 'Link' button, alternatively Right-click and select 'Create Link'. The 'Link Properties' window opens
3. Click on 'Choose File' and browse to the file that you want to link to
4. Click OPEN
5. Click OK

In Normal or Preview mode you will now find your text underlined and in a different colour (probably blue) You can now try double-clicking on the text, you will see the 'Link Properties' pop up again. This can be very useful if you want to change the file to which the link refers. (In a browser you would expect the file viewed to be replaced by the one to which the link refers!)

If you now switch to 'HTML Tags' view you will see your text preceded by a yellow marker with an 'A' in it. Your text has been defined as an 'A' (Anchor) element.

3.7.1.2 Inserting an email address

Instead of linking to a file it is possible to insert an email address. The result will be that, in use, when the link is clicked the email client on the visitor's machine will be opened with the correct address selected.

To do so proceed as under the previous heading. When the Link Properties window opens (or if Image Properties click the Link tab) enter the email address and check the box 'The above is an email address'

3.7.1.3 Inserting named Anchors

There is a second type of Anchor element the 'Named anchor'. Such an anchor is extremely useful as it can act as a type of bookmark defining a particular place on a page. Links can jump to such bookmarks.

To insert a named anchor

1. Place the cursor at the point you want to mark.

2. Click the 'Anchor' button on the Composition toolbar or, on the Menu Bar, select Insert > Named Anchor. The named anchor properties window appears.
3. Enter a unique name for the anchor.
4. Click OK.

In 'Normal' view anchors are marked by a picture of an anchor .

3.7.1.4 Linking to named anchors

Tip. If you have organised your page well using headings when appropriate KompoZer makes linking to places on a page extremely easy. Even if you have not inserted an anchor choose the drop down list as described. All headings will be included in the list and clicking on one will simultaneously create the anchor and link to it.

Start as above for linking to another file. When the 'Link Properties' window opens, instead of choosing file use the drop down list. Your anchor name should appear there preceded by a "#". Click it and OK. That's it! If you test your page on a browser when you click the link the view should move to show the position of the anchor.

Note If your page is short you may not notice this. You need to test with a long page with the link towards the bottom and the anchor towards the top. With most browsers the anchor should be sit on the top visible line.

3.7.1.5 Bigger jumps

You can jump directly to the position of named anchors in other pages. KompoZer does not make this quite so easy to achieve. Proceed as in 'Linking to another file' above. After you have chosen the file and clicked 'Open' you will see the name of the file in the box, immediately after this type "#" immediately followed by the name of the anchor. Thus, if your file was called 'pagetwo.html' and the anchor was called 'p2anchor' the complete entry should read 'pagetwo.html#p2anchor' (without the quotes). Be very careful to get the case of the text right. With some systems it doesn't matter but with others your link will fail if the case is wrong. (I recommend always using lower case for filenames and anchors.)

Note The anchor name is 'p2anchor' NOT '#p2anchor'. The number symbol merely tells a browser that the following text is an anchor.

You can also, and extremely usefully, jump to pages or places on other web sites. All you have to do is the enter the full URL (Web address) into the box. Again be very careful about getting the case correct. Make sure that the check box ('URL is relative to page location' is cleared (you may have no option).

While you can link to named anchors on other sites I suggest that you don't do this. Confine yourself to linking to the home page (usually ending /index.htm). To link to other pages or anchors presupposes that the webmaster will not re-arrange the structure of the site – in the end your visitors may land up with links that don't work.

3.7.1.6 Link colours

Links are normally underlined and in blue text. If you look carefully you may see the colour change and wonder why. Links can exist in several states the principal are:

unvisited links the initial state
visited links   have been visited in the past
hovered   cursor now hovering over

While these probably appear in different colours it is the browser which determines this.

Note For Transitional doctypes KompoZer offers a means of altering the colours via Tools > Options > New Page Settings. All new pages should use the new selections (they don't affect the currently open page). A better way to gain full, reliable control is to clear 'Use custom, colors' and set link colours using styles. See section 4.2.5.2.

3.7.2 Linking images

The techniques and possibilities are very similar to those used with text.

To create a link

1. Click on the image
2. On the Composition toolbar click on the 'Link' button, alternatively Right-click and select 'Create Link'. The 'Image Properties' window opens
3. Click on 'Choose File' and browse to the file that you want to link to. (The box 'URL is relative to page location' is checked. This means that if you move your page to a new folder you should move the image to a corresponding new folder. If you clear this box the absolute address of the image on the hard drive is given in full. If you move your page now it will look there for the image. As you start to organise a web site you will find that this is not a good arrangement and potentially disastrous when you upload the page to a server.)
4. Click OPEN
5. Click OK

3.7.3 Editing Links

To change the file to which a link refers, in Normal, Tags or Preview mode double-click on the link. The 'Link properties' window opens (for an image the 'Image Properties' window opens - click the Link tab). Edit the link.

To remove the link delete the link reference in the box.

3.8 Frames

If you don't know what frames are read no further – you'll be no worse off.

KompoZer does not support frames – at least not at present. If you open a frame document, you get a message 'This page can't be edited for an unknown reason' but it displays the frame content rather beautifully. Then you can do nothing with it except click on the 'Source' tab. You then see the source code and the system will seem to lock up. Actually it doesn't lock and you can load another page and revert to normal operation.

This is not a great limitation. Though the code for frames takes a little getting used to it is usually quite short and can easily be produced using a text editor. Once established it probably rarely needs to be altered. You can use KompoZer to develop the pages that go into the frames.

Frame-like presentations can be simulated using CSS.
`http://www.nvu.com/demos/frames/frameSimulate.html` provides useful guidance which works well with Mozilla type browsers but less well with Internet Explorer.

3.9 Forms

3.9.1 Introduction

Forms provide a mechanism by which a visitor to a site may send data to as sever for processing. This may be as simple as a box to write a message for e-mailing or as complex as looking up items from a catalogue and sending an order to a supplier. Forms collect data typed in or collected from check boxes or lists and pass them to the server. It follows that forms can be used only in association with compatible software running on the server.

Forms may be placed on standard web pages and act as block level elements. In normal view KompoZer shows forms surrounded by a dotted cyan box. Forms may contain other standard block level elements (paragraphs, headings …) as well as several specific elements known as 'form controls' which are designed for data collection. Since these form controls are essentially inline elements they must be laid out inside block level elements (typically paragraphs though divs would be equally suitable).

Each item of data sent to the server is tagged with information about which control has sent it. This is done by naming each control. Thus designers must give a unique name to each control to be used.

The data collected by a form will be sent to a URL specified in an element 'action' which KompoZer will add to the form. This URL is often on the server that hosts the web page but does not have to be. The data will be processed using one of two methods known as 'GET or 'POST' which also need to be specified. These details will be available from the provider of the associated software.

3.9.2 To set up a form

1.   Click the form button.

2. In the Form properties window give the form a name of your choosing
3. Complete the Action box with the correct URL and select the appropriate method
4. 'Encoding' and 'Target Frame' will frequently not be required but, if they are, select 'More Properties' and complete the boxes
5. Click OK
6. On the form place any headings, paragraphs and images ensuring that there is a placeholder for any controls needed. (If blank placeholders are needed it is probably sensible to put some dummy text in now and delete it later.)
7. Where controls are needed click the corresponding placeholder and using the drop down box beside the Form button select the required control
8. Give each control a unique name
9. Each control has specific information which needs to be entered. Enter it into the box in the window which appears

Note Since this is a guide to using KompoZer and not to designing forms this is as far as we go.

3.9.3 Form control styles

If the standard form styles do not appeal it is possible by the use of styles to customise them to personal preferences. For those familiar with using styles some specific details are given in section 4.2.5.3.

3.10 Layers

3.10.1 What layers are

 One of KompoZer's 'Tip of the day' says "Did you know that … you can move elements? Just place the caret in the block you'd like to move and click once on the Layer button to make the block movable."

Items forming layers are taken out of the normal document flow and placed in positions defined on the page absolutely e.g. in positions fixed relative to the page boundary.

While layers introduce a level of flexibility to page design they also involve complexities that newcomers may wish to avoid. This section is therefore for those who wish to delve further.

3.10.2 Creating layers

To create a layer.

1. Click in the block that you want to format as a layer.
2. Click the layer button on the Format (2) toolbar.
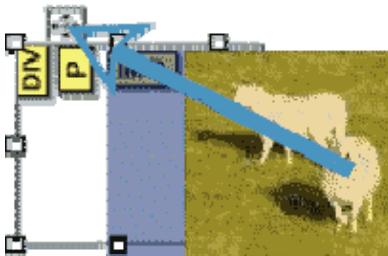
The block becomes a 'Layer'. Layers have sizing boxes which allow the size to be altered. At the top of the layer is a positioning handle (denoted by a four-way arrow)⊕. To move the layer click and drag the handle.

Layers do not have to consist of a single block – any sequence of elements that can be selected together can be converted into a layer. This may be several paragraphs with or without headings, images and tables.

Layers allow items to be placed on a page so that the absolute position, relative to the page, is defined. This provides a facility common in drawing and desk top publishing programs and allows items to be overlaid on top of each other. As such it is a very powerful tool.

Text items normally have transparent background so when overlaid become confused. By formatting them with a background colour they become solid. Similarly tables with solid background and non-transparent images may be overlaid on each other to produce clear effects.

3.10.3 Characteristics of layers

Items converted to layers have several characteristics which differ from 'normal' items.

- Layers in KompoZer are contained in 'div' elements. (For more on 'div' see section 4.5)
- Layers break out of the normal flow in which each block follows the previous and precedes the next.
- Layers, as created by KompoZer, have defined width so do not accommodate to narrowing or widening the viewing window.
- Layers do not however have defined height so will accommodate to changes in text size selected in the browser.
- Layers, as created, are characterised by three parameters 'top', 'left', and 'width'. These define, in pixels, how far the top left hand corner of the layer is from the top left hand origin of the page and how wide the layer is.
- Layers may be manipulated using the buttons on the Format 2 toolbar.

3.10.4 Manipulating layers

To move a layer click the positioning handle and drag it to the required position on the page.

To resize a layer drag a sizing handle as for other elements.

When positioning a layer, KompoZer allows pixel-accuracy movement. This can be difficult to achieve and, if it is attempted to line up items, small errors may become obvious. Better results may be achieved by snapping positions to a coarser grid. To do so select Format > Positioning grid and set the granularity required. Movement will now be constrained to the selection made.

**Note** This setting must be set each time a page is opened. It is not remembered.

Note Snapping applies to dragging the position not the size.

Warning If several overlapping layers occur on a page it may become impossible to click on a handle. In this event try resizing it so that the handles become accessible, alternatively temporally move another layer out of the way.

Note If the height of a layer is dragged, though previously undefined, it becomes defined but the contents will overflow so there will be no visible change.

 When positioning layers so that they overlap a mechanism is needed to define which is in front (so entirely visible) which behind (so partially obscured). Two buttons ('Bring to front' and 'Send to back') on the Format 2 toolbar take care of this. Select the layer and click the appropriate button.
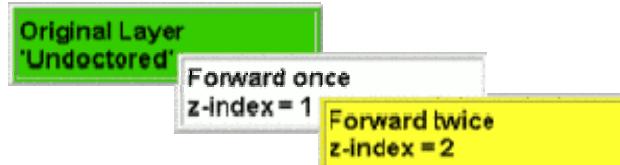
 Layers, as created, do not have a property 'z-index' allocated. When brought to the front a z-index with value =1 is attached. The z-index defines how far forward the layer is. If another layer is to be brought to the front of the first it needs z-index to be set to 2. Selecting it and bringing it to the front however merely sets the index to 1. By repeating the process the index is set to 2 and the layer brought further forward.

Note The buttons would be more appropriately labelled 'Bring forward', and 'Send back'.

3.10.5 Gluing

 Six other buttons on the Format 2 toolbar can be used to modify the behaviour of layers. They are

- Glue to the left border
- Glue to the right border
- Stay in the center

- Glue to the top
- Glue to the bottom
- Stay in the middle

Clicking the first (Glue to the left) initially has no effect since this is the default condition. If clicked after some conflicting condition was set it fixes the distance of the left hand side of the selected layer from the left edge.

Glue to the right fixes the distance of the right hand side (rather than the left hand side) of the layer from the right hand edge.

Glue to the center defines the position, not in absolute terms but as a percentage of the available width. Thus, if created at the centre of the space, it will remain in the centre, similarly for any other ratio.

Glue to the top also represents the default condition otherwise it fixes the distance from the top of the layer to the top edge.

Glue to the bottom and Stay in the center are best avoided by other than advanced users who will probably be using style sheets to achieve the effect required.

3.10.6 Doing it in style

Dragging an item is a rather imprecise action. KompoZer implements layers by using CSS absolute positioning styles. More precise control can be obtained by editing the style parameters.

To do so, in HTML tags view, right-click the 'DIV' tag for the layer and select 'Advanced properties', 'Inline style'.

It is instructive to note which properties are applied to achieve each effect.

			Gluing option				
	Initial	Left	Center	Right	Top	Middle	Bottom
Property		Values given to property					
position	absolute	absolute	absolute	absolute	absolute	absolute	absolute
width	px	px	px	px	px	px	px
top	px	px	px	px	px	%	auto
left	px	px	%	auto	px	px	px
right		auto	auto	px			
bottom					auto	auto	px

3.10.7 Using classes

Instead of setting up layers manually as described so far, a class may be allocated to the 'div' that contains the layer. The CSS editor allows classes with the required properties to be established.

In the CSS editor all the required properties are located under the 'Box' tab. The required properties and values are

Property	Value	Unit
Position	Position	absolute
Width	width	px
Top	Offsets - Top	px or %
Left	Offsets - Left	px or %
Right	Offsets - Right	px or %
Bottom	Offsets – Bottom	px or %
Z-index	Z-index	Integer

By using a class further options become available which cannot be achieved by using the toolbar buttons. The position selector provides options of Absolute, Relative, Static or Fixed. The 'fixed' option is used in the HTML version of this User Guide to position the menu. The layer remains in a fixed position on the screen, not scrolling. The effect is similar to that attainable using frames for a web site.

Warning Internet Explorer version 6 does not support fixed position.

3.11 Borders

Elsewhere a number of methods for applying borders to images and tables have been mentioned. The level of control possible has varied from case to case and the use of style sheets to gain full control has been referred to.

There is one further method which is very easy to apply and which is applicable not only to single elements (e.g. Table (and/or cells), Image, Heading, Paragraph, Form) but also to a group opf elements.

The method uses the full border capabilities of CSS which provides complete control of

- Colour of the border
- Thickness (width) of the border
- Style – Solid, dotted, dashed, double etc.
- Different choices for each side

The method does not cover other options which are sometimes available elsewhere e.g. Padding.

To apply a border select a block or highlight a group of blocks, then and click the 'Borders' button.

Note The Borders button is not installed by default. To access it customise the Format Toolbar. See section 9.3

Leave the check box ticked and set the style, width and colour for the top border. If all borders are to be the same click 'OK' otherwise clear the box and edit the settings for the other sides.

3.12 The Body

This is the background on which the whole page sits. The Body may be thought of as the Page itself. Normally it is of little concern to the designer but there are a few situations when it is.

The body can carry a background colour or image which lies behind everything else on the page in HTML documents (in XHTML documents, the background content for the body element defines only the background for the body element.).Thus the overall appearance of a page can be significantly determined by this content.

To set a background

1. On the Status bar click the <body> marker. All the contents of the page will be highlighted.
2. Right-click and select 'Inline Styles' and 'Background properties'.
3. Make the desired selections and click OK.

The 'Inline Styles' allows you to set several other properties for the whole page. This includes Text, Border and Box properties. This can be a quick way of styling a whole page since Text properties, for instance, will trickle down (technically they are 'inherited') to everything on the page for which they are not otherwise specifically stated. Border and box properties are not inherited.

Stylesheets (see section 4.4) often set several general properties for the body to avoid having to repeat them several times.

Inline styles may set the body width to be less than the full window width. In this case it may be desired to centre the page in the window.

To centre a page in the window

1. On the Status bar click the <body> marker. All the contents of the page will be highlighted
2. Right-click and select 'Inline Styles' and 'Box properties'
3. In the 'Width' box select the width desired for the page

4. In the 'Margins' area in both 'Left' and 'Right' boxes type 'auto'
5. Click OK

Note If you are using Nvu (not KompoZer) to create a page using transitional doctype in Internet Explorer 6 it will not centre. See also <u>Doctypes supported</u>.

3.13 Table of contents

3.13.1 Introduction

If you have a long document with sections headed using heading formats Heading 1, Heading 2 etc KompoZer can generate a Table of Contents (ToC) automatically. The table reflects the structure of the page, the content of the headings forming the text of the table.

By default the entries in the ToC will be linked to the heading to which they refer thus facilitating navigation around the page.

If required, the ToC can be numbered so that the Heading 1 levels are numbered from 1 upwards, Heading 2 levels are numbered from 1 upwards but this numbering restarts each time the level 1 number is incremented. Similarly for lower levels.

It is not necessary to list all heading levels in the ToC; for example it is possible to select only first and second levels. More importantly it is not necessary for the page to include any particular heading level so, if the page formatting uses Heading 5 and Heading 6 only, these may form the basis of the ToC; thus the use of Table of Contents imposes little restraint on the document structure.

If the headings in the page do not provide a satisfactory structure for the ToC it is possible to include short paragraphs specifically for this purpose. These may be included at any level in the ToC. The selected paragraphs are identified by applying a named class style (<u>section 4.2.3</u>) generated for the purpose (unless fortuitously pre-existing). Alternatively a <div> (<u>section 4.5</u>) to which a named class style has been applied may be used.

Generating and using a ToC in this way has the considerable advantage that it can be automatically maintained ensuring that changes to the page are accurately reflected.

A ToC works only within the page in which it resides. Links to other pages must be entered and maintained manually.

3.13.2 Inserting a Table of Contents

To insert a ToC

1. Place the cursor where the table is required
2. Click Insert > Table of Contents > Insert
3. The 'Table of Contents' window appears

4. In the column headed 'Tag' select the tag for each level e.g. against level 1 select h3 and against level 2 select h4 and for all the others select '--'
5. If, instead of using headings you wish to use classed paragraphs or a div, instead of selecting a heading tag select 'p' or 'div' and in the box in the column headed 'Class' enter the class required. (It is, of course, also possible to select headings by allocating a class.)
6. If you wish the contents to be numbered check the box 'Number all entries ...'
7. Click OK

The Table of Contents will be created.

Note The box 'Make the table of contents read-only' does not work.

To update a ToC after changes have been made to a page.

There is no need to place the cursor.

1. Click Insert > Table of Contents > Update
2. The 'Table of Contents' window appears showing the selections previously made
3. If desired, changes may be made to the selections
4. To update the ToC click OK

To delete a ToC.

There is no need to place the cursor.

1. Click Insert > Table of Contents > Remove

3.13.3 Methods

KompoZer builds the ToC using Ordered or Unordered lists. (section 3.4.4). These lists have to be nested to produce the structure required. The result is that each level of the contents is indented relative to the previous. The indentation may be altered using styles and specifying the OL or UL and LI elements. The changes required will be to the margin and padding (section 4.2.1.4). Any changes made will apply to all lists on the page (not only ToC).

In order to make links work, each tag listed in the table will have a named anchor added automatically when the table is created or updated. Deleting a ToC also deletes all these anchors.

3.14 The Head area

Every HTML document starts with a "Head" area. This provides information to browsers, search engines etc but is invisible on the page.

Items included on the head are gathered from a number of sources. These may include:

- Items dependent on the Doctype as set when creating the page ([section 3.2.1](#)).
- Items collected from New Page Settings ([section 9.1](#)). Includes Author, Language.
- Items set by the Format > Page Title and properties menu. Includes Title, Author, Description, Language, Writing direction Character set.
- Links to stylesheets ([section 4.1.3.3](#))
- Definitions for internal styles ([section 4.1.3.2](#))

Source view allows advanced users to add further information as desired.

3.15 Page views
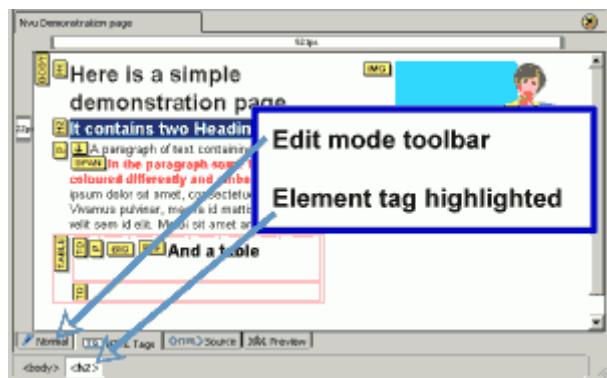
KompoZer provides four ways of viewing a page – 'Normal', 'HTML Tags', 'Source', 'Preview'. These are selected using the tabs on the Edit mode Toolbar. All provide full editing facilities.

- Preview mode offers almost the same view as in a browser with the addition of rulers and sizing boxes. The main differences are that scripts do not run (so their effects will not be seen), links do not operate and the style for links does not respond to any class set. Tooltips do work however.
- Normal view is very similar but here table outlines show and markers for named anchors appear.
- HTML Tags view assists those familiar with HTML. There is a yellow marker for the start tag for all elements (end tags are omitted). Clicking on a marker selects and highlights the whole of the element.

Note If clicking an element does not select it press the control key while clicking.

You will also see the element highlighted on the Status Bar (see figure). If you switch to Source view the element remains highlighted,

Note It will be on the screen but possibly right at the bottom.

This is of great assistance in editing. Marker names correspond with the HTML tag name with a few exceptions.

Note Named anchors are shown with a symbol whereas linked anchors are denoted by 'A', The marker 'Span' is used when certain changes are made inline with text – for instance if

the colour is changed for a single word. Span in HTML has a more restricted use being confined to changes implemented using styles. This may be confusing.

In HTML Tags view background images do not appear.

- Source view shows all detail of the HTML code. All except the DOCTYPE may be edited.

An additional view is provided via the 'Browse' button. This opens your default browser and gives full browser functionality and display to a page.

Warning If your default browser is Internet Explorer the page may open in source view instead of page view.

3.16 Editing in Source view

This mode is intended for those confident in coding in HTML. In this mode KompoZer is not simply a text editor. On switching to a different mode KompoZer carries out checks to ensure that the code is valid. If not it 'fixes' it. It will have made its best guess at what was intended – if it gets it wrong there is no going back – no 'Undo' button. While this may sometimes save you a lot of typing, if you know what you are doing, it is also possible to end up with a heavily 'messed up' page – it will display but half the content could have disappeared.

Needless to say good backup practice can save many disasters. The Site Manager (section 8) is actually a convenient tool for doing this. Create a backup folder and 'Publish' to this !

If you select an element via the status bar (see next section) to highlight it, on switching to source view the corresponding code will also be highlighted. This makes it possible to locate code very easily. (For an image the cursor will be adjacent to the code.)

The line numbers which appear are an editing convenience, they form no part of the code. They can help to find your place when switching modes. When switching between applications the numbers are only an approximate guide.

KompoZer provides some options in how lines are displayed as mentioned in section 9.2 on 'Preferences' or see KompoZer Help.

3.17 Selecting items - Reprise

By now KompoZer users should be familiar with selecting and manipulating items. A recapitulation may be useful however.

These comments apply in Normal, HTML Tags or preview mode.

KompoZer often offers a number of ways of doing something. Sometimes one method will apply sometimes another. Which works will depend on a) what is to be selected and b) what is to be done with the selection.

Tip: Occasionally you may wish to escape from some element to the body of the page. For instance if you have two tables immediately following each other pressing the down arrow at the end of one will take you into the next, you can't insert anything between them. If you have a paragraph with a special style the enter key will start a new paragraph with the same style, if you don't want that you have to clear the style. If you have a division which is the last item on a page you can't add anything after it.

In any of these situations click the last , or only, item in the element and, on the status bar, click the marker for the element, next press the down arrow key. This will take the pointer to below the div though the cursor will not be visible.

Then just type.

Selecting text.

- Simply place the cursor anywhere in a block. This is often sufficient where a format is to be applied to a complete block.
- Double-click within a block to select a word.
- Triple click within a block to select a line of text.
- Drag across a section of text to highlight and select that section only.

Many functions may be carried out on the selected text this includes

- Dragging to a different position,
- Applying a class
- Changing the element tag e.g. changing from a paragraph to a list.

Selecting in HTML tags view

Sometimes selections in normal or preview mode become difficult or impossible.

- In HTML Tags view press the Control key while clicking on the tag for the block to select a complete block.

Actions appropriate to complete blocks such as dragging or applying a class are possible.

Selecting via the Status bar

This is a quite excellent tool. Click in any text object or image, then, on the Status bar, click on the element marker farthest to the right. The object is selected. To the left of this are shown any elements higher in the document tree. Click any of these to select the corresponding elements. KompoZer also displays the class or id of each element allowing for easier selection.

Double clicking

This provides a powerful context-sensitive function which often provides the fastest route to appropriate actions.

- Double click an image to reach the 'Image properties' window which allows changes to image location, name, size, link etc.
- Double click a link or anchor to edit the name, properties etc.
- Double click a table or cell to access properties such as size, colour, alignment.
- Double click most things to obtain access to 'Advanced properties' via the 'Advanced edit' button which appears on the first window to open. While these require more detailed knowledge of HTML or CSS they can be used to apply extensive formatting options to items.

4 Style and stylesheets

4.1 Introduction to style

Note Where the [icon] icon occurs click it to reveal differences in the behaviour of Nvu if this interests you.

The use of styles is already well established in word processing and desk top publishing. In web site design it assumes an even more important role and can control almost every aspect of presentation.

Styles in HTML documents should comply with the requirements of the W3C style specifications. These are detailed in 'Cascading Style Sheets, level 2 CSS2 Specification' which is available at http://www.w3.org/TR/CSS21/.

A full introduction and many useful references may be found at http://en.wikipedia.org/wiki/Cascading_Style_Sheets

4.1.1 Power of styles

Styles specify how particular **elements** on a page appear on the screen, in print or whatever. This guide limits itself to on-screen considerations. By 'elements' we mean parts of the page structure, typically headings and paragraphs, but also stretching to many others including tables, bulleted and numbered lists etc. In fact most HTML 'Tags' may be specified though the same style may be applied to several.

Style may typically define such aspects of presentation as the font face, size and variant, the font colour, the background colour, whether an element is to be aligned right, centre or left, whether spaced away from other elements, surrounded by a border and, if so, what type or colour. Elements may be given an absolute position relative to the page (which allows elements to overlap). Elements such as paragraphs, tables and images are considered to exist within boxes or blocks and the sizes of these boxes may be specified.

Note You can see the outlines of the blocks by selecting 'View - Block outline'.

4.1.2 Classes

As well as allowing you to specify the style of elements it is possible to define styles and apply them selectively to only **some elements**. This is done through **'Classes'** – a 'Class' is just a style that can be applied as and when you choose.
'Classes' are applied to **'Tags'** (a marker attached to an element). The element to which the class is applied appears in the format defined by the class. Other similar elements without the class applied appear in the default format i.e. either the default specified by the browser or the style that the user has defined for the corresponding element.

On the status bar KompoZer shows classes along with the tag to which they are applied.

4.1.3 Using style

4.1.3.1 Inline style

Within a page styles may be used in three ways. These ways can be mixed and matched as you wish. The first, easiest and crudest is to define a style for the nonce, there and then. Such a style is listed in the code (in Source view) with the tag to which it applies (using the structure "`style = ….`" if you look at the code).

Don't worry KompoZer hides all this from you.

If you have another item with the same style this code must be repeated. This bloats the page. This is known as an 'inline style'. For the most part KompoZer users do not have to be concerned with this method although KompoZer will sometimes use it without you knowing.

4.1.3.2 Internal style

The second way to use a style is to embed a list of style definitions within the HEAD section of a page. (These definitions are referred to as 'style rules' or just 'rules')

These rules may be of one of two sorts. The first sort applies to all elements of a particular type (e.g. p, h1, table) the second is the class as discussed above.

Note If you look at the HTML source code you will see a structure like `<p class="mynicepara">` when the class `"mynicepara"` has been applied. The corresponding code where no class is applied is `<p>`. If you look at the structure of the style sheet for a class you will see the class name preceded by a dot (full stop) e.g. "`.mynicepara`" while a generally applied style will look like '`p`'. Generally applied styles must, of course, carry the name of the element to which they apply whereas a class can have any name you like to give it. To help maintain the site it is good to give it a name that describes its function NOT the resulting appearance.

4.1.3.3 External style – Linked stylesheets

While the first two methods are valid and have their uses, the third is the preferred method specifically because it is economical, reuses the same styles for many pages and helps in achieving consistency of appearance right across a complete web site. The method uses an external style sheet which is 'Linked' to a page, or to several pages (though each page must include the linking information for itself).

An external style sheet contains the same list of rules which would otherwise have been included in the internal list referred to above. (It is actually a simple text document such as you could construct using a text editor like Windows® Notepad.) The file is usually located

in the same folder (directory) as the page to which it is linked (though it can be elsewhere) and has the extension 'css'.

How does a page know to use this style sheet? A line of code which, KompoZer will insert for you in the head section of your page, will see to this.

Warning This is one of the few places where Nvu makes a mess of things. It inserts an absolute reference instead of a relative reference. This will work fine as long as your pages remain on your own computer but, if you move them to a new folder or upload them you will need to edit this line manually. See section 4.2.4.1

The code looks like `<link rel="stylesheet" type="text/css" href="mystylesheet.css>`

4.1.3.4   Cascading style sheets

You may wonder – if you have an external style sheet, perhaps internal rules and inline styles will there be confusion? This is where the cascading comes in. Your browser should first use whatever its default style would be. If it finds a matching style in an external sheet it should use that instead, it should then look at internal styles and, if any coincide with anything it is planning to use that should override them. Lastly inline styles prevail.

How could that be used? Well perhaps your site establishes a style using an external style sheet. Perhaps you have one page about emergency procedures and you decide that all its paragraphs should be in red. You simply add an internal rule on that page redefining the 'p' tag to be red but also link the page to your normal style sheet. All your normal styles will be applied except that all paragraph text will be in red, nothing else will be non-standard.

KompoZer will actually allow you to link several external sheets to a page. The browser reads these in the order encountered and acts on the most recent information read. Internal sheets still obtain priority irrespective of the order they appear in the code.

The full rules for prioritisation are rather complex. Perhaps in an attempt to assist here KompoZer allows you to switch off style sheets while testing pages. With the CSS editor open (see later) select a stylesheet and check the 'Disabled' box.

Style Nomenclature

Styles employ a bespoke nomenclature. uses this little since, except in source view, the code is hidden from the author. The terms involved do appear in this User Guide. Here they are along with some examples. The examples show the source code that would appear either in the head area of your page or in an external style sheet.

Simple rule with one declaration	p { font-family : Arial, sans-serif ;}
Rule with 2 declarations	p { font-family : Arial, sans-serif ; font-size : medium ;}
Rule with 2 selectors	h1 , h2 { color : #990000 ;}
Rule with class selector	.redtext { color : red ;}
Rule with complex selector *	#menu li a:link { text-decoration : none ;}

Key:

Rules appear within a blue bordered box

Declarations have a dashed border

Selectors have a red border and white background

Properties have a green border and white background

Values have a magenta border and white background

* This selector selects list items in the div with id 'menu' and which are linked.

4.2   Creating styles

4.2.1   Internal styles

CREATING INTERNAL STYLES USING NVU

Styles are created and edited using the built in Style sheet editor called CaScadeS. CaScadeS has two modes of operation 'Beginner Mode' and 'Expert Mode'. When it opens it opens in expert mode, should you wish to switch to Beginner Mode clear the check box. The difference is that in Beginner mode External style sheets cannot be created.

When internal styles are created Nvu will include the the rules for these in the 'head' area of the document.

To start creating styles:

1. Open Tools > CSS Editor.
2. Click 'Style elt', 'Create Stylesheet'.
3. Click RULE.
4. The General tab opens and offers the options 'Named Style' or 'Style applied to all elements of type' or 'Style applied to all elements matching the following selector'.

To style an element e.g. a paragraph:

5. Select 'Style applied to all elements of type'
6. In the box type "p", or the tag for the element (without the quotes)
7. Click 'Create style rule'.

To define how you want elements to look:

1. Click the tag for the element in the left hand pane. The General tab should show, listing the 'Selector' (at present the element tag) and any style 'Declarations' (initially blank).
2. Now select in turn as required the tabs for 'Text', 'Background', 'Border' etc and specify exactly how you wish that element to appear. The next section amplifies some details of how to do this.
3. Return to the general tab to see the full declarations that you have set for the Selector.

As we will see later style rules are stored in style sheets. KompoZer includes a CSS editor called CaScadeS which allows you to create stylesheets and define rules. Before you define a rule you must have a stylesheet but if you are working on a page which has none CaScadeS will create one for you.
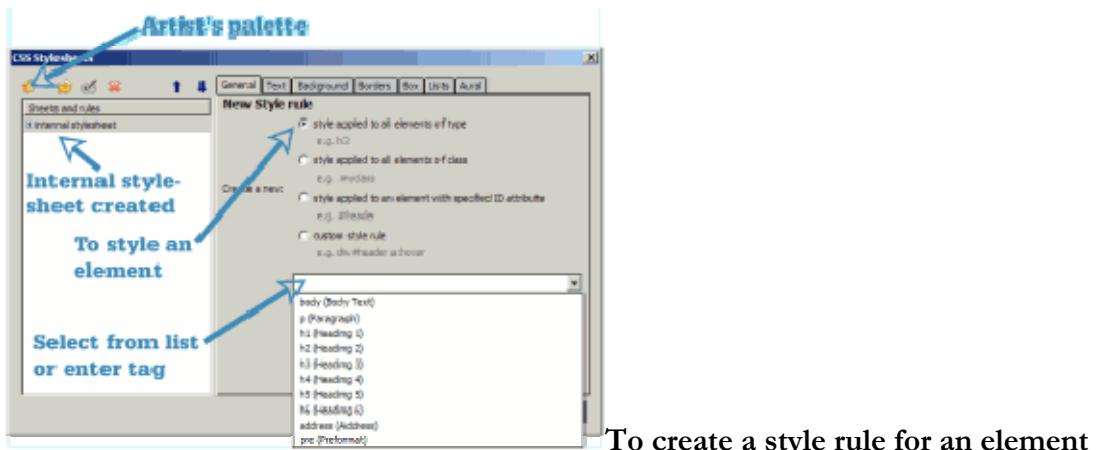
To create a style rule for an element

1. Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens.
2. Click on the artist's palette button 🎨 ⏷. In the sheets and rules box you will see an internal stylesheet has been created for you.
3. To create a rule click 'Style applied to all elements of type'
4. Beside the blank box click the drop down arrow. You will see listed a number of common elements. To create a style for one of these click it alternatively enter the tag for any other element.
5. Click 'Create Style rule'
6. You are now presented with a window headed 'Selector' followed by the tag for the element. The window actually lists the style declarations for that element, but of course that is now blank.

To define how you want elements to look:

1. Select in turn as required the tabs for 'Text', 'Background', 'Border' etc and specify exactly how you wish that element to appear. The next section amplifies some details of how to do this.
2. Return to the general tab to see the full declarations that you have set for the Selector
3. If you click the 'General' tabyou will see all the declarationsfor the rule. You can edit these here but it is better toleave the job to CaScadeS because if you make any errors the declaration will be deleted
4. When you are satisfied click 'OK'

Once created, rules may be edited by reopening CaScadeS, Click and expand the stylesheet and click the rule involved.

In the 'Sheets and rules' pane on the left you should see a structure showing each tag as it is defined.

I won't go through all the options, just mention a few.

First, do note you are not obliged to specify any particular properties. You specify as much or as little as you wish. It's just that the browser will revert to defaults if you don't. There are fairly complicated rules for what it uses, and browsers don't always follow the rules, however chances are that you'll be happy with whatever it does – if you're not – change it!

You may like to try setting declarations for the 'body' and apply it to all elements. In most cases declarations specified here with trickle down to everything else (they are 'inherited') unless you specify it explicitly elsewhere. Generally this works but can generate surprises.

Note The tabs available on CaScadeS allow a large number and all the most important style properties to be set. The CSS specification however includes many others. If you are familiar with these you can type declarations manually into the box (incorrectly) headed 'Style rule'. On closingCaScadeS the validity of the declaration will be checked and those which are valid will be added to the stylesheet.

4.2.1.1 Text Tab

Font family. You are offered three ways of specifying

- Unspecified
- Pre-defined
- Use custom font family

In common parlance 'font families' are referred to as 'fonts'. CSS always refers to them as 'font families'. The correct typographic term is 'Typeface'. (The English word for 'font' is 'fount' but sadly the American word has come into general use in the UK.)

'Unspecified' leaves everything to the browser

'Predefined' has a drop down selector offering one of three choices. Each choice has a set of 3 fonts which have fundamentally similar characteristics. The choices are:

- Sans serif proportional fonts.
- Serif proportional fonts.
- Fixed spacing fonts.

Each set has three entries. The first two are alternative similar fonts; the third entry is a statement of the generic font type – the browser should look for the first font listed, if it

can't find it, look for the second and, if that can't be found, use a font of the generic type of the final entry. Thus something suitable will always be found.

'Custom font family' is an option to be avoided or used with extreme caution. The list offered is taken from the fonts installed on the author's computer. If you choose one which is not available on your visitor's computer there is no control over what will be used. On the other hand, if carefully used, this option is very flexible and you can build your own set using the model from the 'Predefined' selector.

To build a set of fonts.

1. Note the structure of the 'Predefined' sets. The lists are built with each font name terminated by a comma and each set finished with a generic type.
2. Select a generic type from the following: serif, sans-serif, cursive, fantasy, monospace.
3. Select preferred fonts corresponding to the generic type required.
4. With the Text tab selected check 'Use custom font family' and in the box type the name of your preferred fonts each followed by a comma, finish with a generic type.

Note 1 You will probably wish to include one font available on Windows PC systems and an equivalent available on Macintosh computers.

Note 2 You may enter as many fonts as you wish.

Note 3 Using 'cursive' or 'fantasy' options is not recommended since the level of standardisation between visitors will render the results very uncertain.

Note 4 If you use a font which has a name which includes spaces (such as Times New Roman) the name must be enclosed within quotes e.g. "Times New Roman". If you fail to do this the result will probably still work but validation will fail.

Warning Unfortunately when Nvu builds sets of fonts itself it fails to do this. The fixed spacing option presents no problem but the two proportional options use fonts with spaces in the font names. Authors must either suffer the indignity of failing validation or edit the style values in source view to overcome the problem.

Font size. There are three sections offering different types of option. The first block is a miscellany offering various units. Because display screens vary in size and resolution and because people's eyes and preferences differ, browsers offer view options allowing the user to alter the text size. If this facility is disabled the user may not be able to see the page as he or she would wish. Some of the options in this first group (e.g. px, pt, cm, in, mm, pc) either apply to print media or are for fixed size units. These may disable browser zooming or result in unpredictable sizes. You will probably find the last set of named styles (starting xx-small)

is most satisfactory, alternatively use the percentage or em scales. As you select and alter the size you can see the result directly on your page.

Absolute and relative sizes.

Text in Division.
Size not specified - Inherited.

Text in table. Size not specified. Inherited from div.

Size for this cell set to 120%

Text in a second table

Text size for table set to 120%

Size for cell not specified inherited from table.

When setting font sizes for text lying directly in the body element 1em, 100% and 'medium' generally yield identical results. Font sizes are inherited so text in elements which themselves sit in the body will, unless specified separately, use the same sizes. Elements may however sit, not directly in the body but, within other elements such as a table or a division. Where such a table or division has a font size set on itself this may be inherited by text within that table or division. For instance, if a div has font size set to 150% and a paragraph within the div has font size set to 1em then the text will actually be rendered at 1.5em due to it inheriting the scale from the div.

Note A division is formed using the <div> element. See section 4.5

An example of how this can work is given in the adjacent figure. Text sits in table cells, which sit within a table which sits within a division. One of the tables and one of the cells has font size specified. Others do not.

Not all sizes scale in this way. Sizes set as a percentage or in em or ex do scale and are referred to as **relative sizes**. Sizes specified using the xx-small system do not scale and are referred to as **absolute sizes**

Note All sizes will zoom using the browser text size options with the exception of sizes specified in pixels. In this case browsers vary in response.

Font Colour. Either enter a colour into the box or click on the button to the right. If you click on the button the 'Text Color' window opens and you have three more options. (It's like buying an ice cream in the USA or making a phone call in the UK.) You can specify in terms of Hue, Saturation and Brightness (HSB), Red, Green Blue values (RGB) (range 0 to 255). Alternatively click on one of the 70 boxes of colours offered.

These are so-called 'web safe' colours because they work on lower performance displays. Actually the full range of 216 web-safe colours is not provided but, in any case, modern displays normally work in at least 24-bit colour so there is little justification for this limitation.

The next option is to click on the rainbow type area. Whichever you choose, the selected colour appears immediately on the square. Fine adjustments may be made using the RGB or HSB values. A box labelled Hex value shows the colour as a hexadecimal number. Actually you could have done this directly without opening the 'Text Color' window – when you close the window you'll see the hex value (preceded by a '#') in the box along with a sample of the colour on the button.

Named colours. You might have noticed, as you clicked on one of the 70 colours, that occasionally in the box labelled 'Name' the name of a colour appeared. (Try black or red to check.) The CSS recommendation recognises 17 colours and the names of these can be entered directly here, or, in fact, into the selector box without entering the 'Text Color' window.

The 17 CSScolours and names:

Black

Gray

Maroon

Red

Orange

Yellow

Lime

Green

Teal

Olive

Aqua

Blue

Navy

Fuchsia

Purple

Silver

White

.

Other text options

These appear below the colour option and are fairly self-explanatory.

4.2.1.2   Background Tab

Colour

This is selected in the same way as for <u>Font colour</u>.

Image

You can specify an image file as a background. This will appear behind anything else. Under 'Tile', if you select 'Don't tile' you get a single copy of the file which you can position with the buttons below. If you select 'Tile' the image repeats in the directions selected.

Warning. The opacity option uses a property not included in the CSS2 Specification, which is either not implemented or implemented incorrectly by most browsers.

4.2.1.3   Border

Items can be bordered and the options available provide great flexibility. Styles, thickness and colours may be set independently for each side.

Note For a border to show both width and style must be specified.

4.2.1.4   Box

The block structure in which most elements are located has already been mentioned (<u>section 3.4.1</u>). The Box options control the display within these blocks.
For those boxes which require a dimension to be entered: first type the number, then select the unit from the drop down box.

Margin Controls the distance between the element's box, i.e. where any border would be, and either the surrounding box or adjacent boxes.

Note Vertical margins 'collapse'. If a heading has a bottom margin set to 20px and is followed by a paragraph with a top margin set to 10px the separation between them is controlled by the greater (20px) not the sum (30 px).
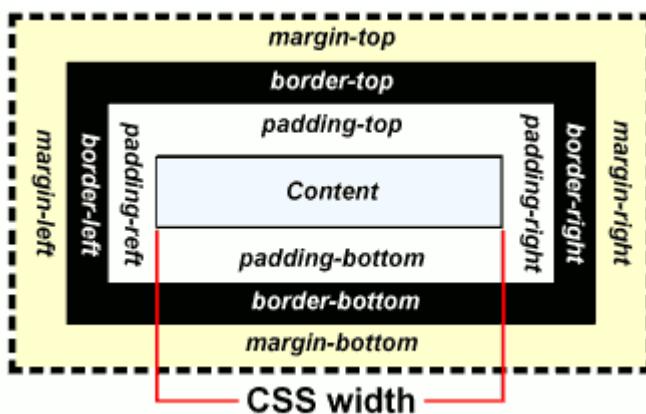
Padding Controls the distance between the box content and its border.

Width and **Height** Allow a specific size to be allocated to the content area of a box. This can be an absolute width or a percentage of the containing box.

Note The spin buttons beside each box allow fine adjustments to be made easily. In most cases the effect can be sen in real time on the screen.

Float allows blocks to be treated rather like images and to float them to the right or left hand side of a page (or to the edge of the enclosing block). Other elements may therefore appear to wrap around them.

Position, **Offset** and **z-index** are parameters related to the use of layers. These have been covered in <u>section 3.10.6</u>

4.2.2   Styling multiple elements

Instead of setting styles for one element at a time it may be easier to specify several at the same time. Perhaps all your headings are in one font (e.g. serif) and all others in an other (sans-serif). You can specify h1, h2, h3, h4, h5 and h6 as serif font in a single action. When you create a style rule at step, 4 instead of entering the tag for one element, just list all requiredand separate the tags by commas. Later you can specify, for example the size of, each heading individually in a separate rule.

EDITING SELECTORS

If you have already set up the style declarations required, perhaps for h1, but now want it to apply to additional elements just click the `Edit selector' button 🖉above the`Sheets and rules' window and type in the additional tags separated by commas.

4.2.3   Named Classes

Tip:Class names can contain letters, numbers, hyphens and underscores but not spaces and must not start with a number. They are case insensitive. **N.B. Class names do not include the initial dot.**

I've suggested above that you start by applying styles to all elements of a specific type. Such style definitions will apply automatically wherever that element is found on a page. This provides for great consistency within a document but maybe a touch of dullness! You can also define your own styles and apply them selectively. These styles are known as 'Classes'.

Using Nvu – To create a class

1. Open the CSS Editor, and if necessary create a style sheet, as described in section 4.2.1
2. Click on the stylesheet to select it. (If there is more than one stylesheet click on the one in which you wish to create the class.)
3. Decide on a name for the class.
4. Click on RULE, leave 'Named style' selected and enter your chosen class name in the box.
5. Click on 'Create style rule'. You should see the name of your style appear in the pane on the left but note that the name is preceded by a dot (full stop)
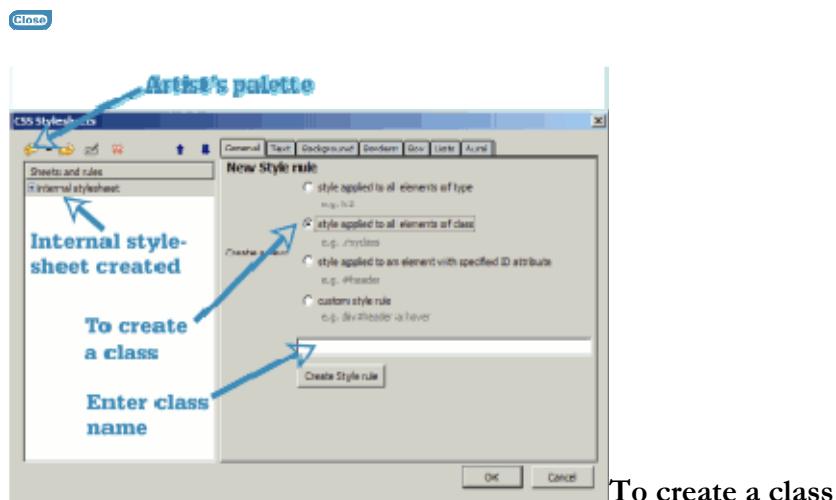
To create a class

1. Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens
2. If the stylesheet in which you wish to create the class exists click it followed by the artist's palette button . Else:
3. Click on the artist's palette button. In the sheets and rules box you will see an internal stylesheet has been created for you
4. To create a class click 'Style applied to all elements of class'
5. In the box you will see a full stop entered for you. Now either
   *   Follow that by a name of your choice which must start with a letter or

- Click the drop down arrow beside the box to get a list of already defined classes and click one of these

6. Click 'Create Style rule'
7. You are now presented with a window headed 'Selector' followed by your class name. The window actually lists the style declarations for that rule, but of course that is now blank

Now proceed as in section 4.2.1 '**To define how you want elements to look**'. In that case you could watch the effect of your styles appear immediately on your page (assuming that you had a page which used the corresponding elements). Now, of course, you have none, so first we must learn how to apply a class to an element.

First enter some details for your new class that will allow you to recognise it on the screen if it is applied. For instance use the border tab to produce a border styled as solid, 2px wide, coloured red.

It is possible to set styles for several classes at the same time just as it was for elements. Proceed as in 4.2.2 but precede all class names by a dot e.g. `.style1, .style2` (KompoZer automatically adds the dot for the first).

4.2.3.1 Applying Classes.
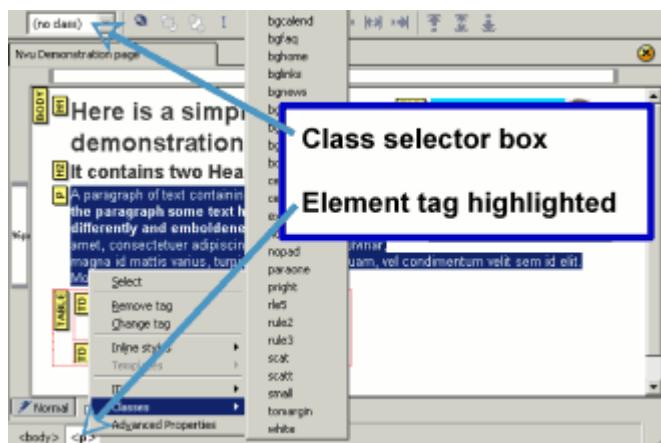
KompoZer provides two methods for applying classes. Method 1 is the simpler; sometimes it is difficult to select the item to use this method in which case method 2 is preferable.

Apply a class - Method 1

1. If necessary close the CSS editor.
2. Select the item to which the class is to be applied. For a heading or paragraph simply click in the text, to apply a class to an inline section of text (span) highlight the text.
3. Use the class selector box on the Format toolbar (see figure), click the drop down arrow and select the class required.
4. The result of applying the class should be seen in any view mode.

Note Although KompoZer is not a browser it replicates the browser display very well. However styles applied to links are not rendered correctly.

Apply a class - Method 2

Note This mode cannot be used to *create* the span element needed to style an inline section of text but can style such a section once the span exists.

1. If necessary close the CSS editor.
2. **EITHER** in 'HTML Tags' view Click on the yellow marker for the element for which you want to set a class. (If the element does not become selected, as indicated by outlining or highlighting on the screen, press the control key while selecting.)
   OR in Normal or Preview mode click in a block you wish to style then click on the Status bar on the right-most element listed.
3. Now both the block involved and the corresponding element and on the status bar should be highlighted. (See figure above.)
4. Right-click the highlighted element on the status bar and select 'Classes'. You should see a list of all classes pop up (This includes classes on internal and linked stylesheets.) Any class already applied to the element appears, marked with a tick ( ), at the top of the list above the divider line; other available classes are below the line.
5. To add a new class, click the class; To remove a class, click the class above the divider.

Once a class has been attributed to an element it is shown on the the tag on the status bar.

Note Elements can actually have several classes applied. If you do this you should ensure that the class properties don't conflict with each other. Most people will probably prefer to apply only one class at a time. You may need to check that you don't apply more by mistake.

4.2.3.2   Removing classes

A similar procedure may be used to disconnect an element from a class. Using the highlighted tag method simply select the ticked class, this deselects it. Using the class selector box select the style again, this should cancel it. Alternatively, after selecting the element, right-click on the highlighted tag on the status bar and select 'Advanced properties. The selected class will be listed select it and click 'Remove'.

4.2.4   External styles

There are three ways to start working with external style sheets

4.2.4.1   Create a new external style sheet

External style sheets are created in almost exactly the same way as internal sheets.

To create an external style sheet

1. Open the CSS Editor. (Tools > CSS Editor).
2. Make sure that you are in 'Expert Mode' and Click 'Style elt.'
3. Type a title in the box and click 'Create Stylesheet'.
4. Click 'Export Stylesheet …'
5. Give it a name with extension css and save it, preferably in the same folder as your page.

Now click on 'RULE' and you should be on familiar territory.

When we created internal styles they were entered on our page in the HEAD section. Now the styles are being created in an external file. If you looked, in 'Source' View, in the head area you would see a line referring to the external file and linking it to your page.

Warning – Unfortunately Nvu makes a meal of inserting the filename. It puts the absolute address of the file as it appears on your hard drive. When you upload or move your page the stylesheet won't be found. You will need to edit the HTML Source code in the line starting `<link rel="stylesheet"` (note the 'line', which may spread over several lines, finishes `type="text/css">`) Delete all the code like `file:///C:/Documents%20…` right up to and including the / before the filename. The result should contain text like `href="filename.css"`. If your stylesheet is in a sub-folder the text should be like `href="subfolder/filename.css"`.

4.2.4.2   Change an internal style listing into an external stylesheet

If you already have set up your page using internal styles you can convert them into an external stylesheet and link to that.
Open the CSS Editor. In the 'Sheets and rules' pane click on the stylesheet and under the general tab click on 'Export stylesheet and switch to exported version' and proceed.

4.2.4.3   Link to an existing stylesheet

If you have a stylesheet that you created for another page or intend to use right across your site you can link your page to that.
To link to an existing external style sheet

1. Open the CaScadeS. (Tools > CSS Editor).
2. Make sure that you are in 'Expert Mode' and Click 'Link elt.'
3. Click 'Choose file' and browse to the location of the stylesheet you want to use.
4. Click 'Open' then 'Create stylesheet'. You should then see the stylesheet name appear in the 'Sheets and Rules' pane to the left.

You can now close the CSS editor, but, of course you can work on the stylesheet in the usual way.

NB the same warning applies as in section 4.2.4.1

External style sheets are created in almost exactly the same way as internal sheets.

4.2.4.1 Create a new external style sheet

To create an external style sheet if the file has no internal or external stylesheet.

1. Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens showing that an internal stylesheet has been created for you
2. Click on the name of the stylesheet 'Internal stylesheet'
3. Click the dropdown arrow beside the artist's palette button and select 'Linked stylesheet'
4. Click 'Create stylesheet'
5. Click the name of the stylesheet again
6. Click 'Export stylesheet and switch to exported version'
7. Give the stylesheet a name with the extension .css
8. Give it a name with extension css and save it, preferably in the same folder as your page
9. Click 'Save'

Once the sheet has been created you may use it to store any style rules that you wish.

When we created internal styles they were entered on our page in the HEAD section. Now the styles are being created in an external file. If you looked, in 'Source' View, in the head area you would see a line referring to the external file and linking it to your page.

4.2.4.2 Change an internal style listing into an external stylesheet

If you already have set up your page using internal styles you can convert them into an external stylesheet and link to that.
Do this

1. Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens showing the existing internal stylesheet
2. Click on the name of the stylesheet 'Internal stylesheet'
3. Click 'Export stylesheet and switch to exported version'
4. Give the stylesheet a name with the extension .css
5. Give it a name with extension css and save it, preferably in the same folder as your page
6. Click 'Save'

4.2.4.3   Link to an existing stylesheet

If you have a stylesheet that you created for another page or intend to use right across your site you can link your page to that.
To link to an existing external style sheet

1. Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens
2. Click the dropdown arrow beside the artist's palette button ✏ and select 'Linked stylesheet'
3. Click 'Choose file' and browse to the file that you need.
4. Click 'Open'
5. Click 'Create stylesheet'
6. Click 'OK

You can now close the CSS editor, but, of course you can work on the stylesheet in the usual way.

4.2.4.4 Saving stylesheets

Once you have linked and saved a stylesheet, if changes are made later, the changes will appear in your document immediately and the sheet will be re-saved when the page is saved.

Once you edit a stylesheet in KompoZer its name is marked by a red symbol indicating that it has not been saved. When you close CaScadeS changed sheets are saved immediately.

Warning If the sheet is linked to several pages, any others which are open at the time will not recognise the change until they have been closed and re-opened. (File > Revert)

4.2.5 Other style selectors

4.2.5.1 Introduction

Element styles ([section 4.2.1](#)) and named classes ([section 4.2.3](#)) are the two main 'selectors' used for styles. A selector may be thought of as the link which attaches a particular style declaration to a particular item on a page.

Note Strictly the class selector consists of the class name and a prefixed dot.

In addition to these the CSS specification provides a large range of more complex selectors most of which lie outside the scope of this Guide. For details see the CSS Specification

section 5. (See <u>appendix 2</u>.) There are however a few cases which are needed more frequently and for less complex sites and these are covered in the next two sections.

4.2.5.2 Link styles

Browsers render links in default colours and provides some options for controlling these (<u>see 3.7.1.6</u>). Through the use of styles full control of each state of a link is possible. This is achieved through the use of selectors called link 'pseudo-classes'.

The more important pseudo-classes are

Selector	Descripton
a:link	for all links not yet visited
a:visited	for links that have been visited
a:hover	for a link over which the cursor is hovering

Note The following procedures are based on the use of internal style sheets. Very similar procedures may be used with external stylesheets.

To create link pseudo-classes

1. Proceed as if creating a class (see <u>section 4.2.3</u>)
2. At step 4, if using Nvu select 'Style applied to all elements matching the following selector' and, in the box enter the selector as listed above.
3. At step 5 the selector name should appear as entered.

It is now possible to style these options fully, i.e. not only with text colour and decoration but with background, boxes, the lot.

Note that the a:hover must be placed after the a:link and a:visited rules, since otherwise the cascading rules will hide the a:hover rule.

To create link pseudo-classes

1. Proceed as if creating a class (see <u>section 4.2.3</u>)
2. At step 4 select 'Custom style rule' and, in the box enter the selector as listed above.
3. At step 5 the selector name should appear as entered.

It is now possible to style these options fully, i.e. not only with text colour and decoration but with background, boxes, the lot.

Note that the `a:hover` must be placed after the `a:link` and `a:visited` rules, since otherwise the cascading rules will hide the `a:hover` rule.

Warning in Nvu 1.0 the 'Down' button is disabled. Using KompoZer both buttons work correctly.

The CSS editor enters styles in the order encountered. The order may be altered by selecting an item and clicking the 'Up' or 'Down' button.

Note These styles will now be applied to links automatically. They do not have to be applied manually as do classes.

4.2.5.3 Form control styles

Setting up forms to appear as required may require styles to be set for specific controls. These are set exactly as just described for links it is merely necessary to set styles for the required selectors. Some of the main ones are listed following.

Control	Element	Selector
Submit button	Input	`input[type=submit]`
Reset button	Input	`input[type=reset]`
Text input	Input	`input[type=text]`
Text area	textarea	`textarea`
Label	label	`label`
Field set	fieldset	`fieldset`
Legend	legend	`legend`

4.2.5.4 ID styles

Ids (or Unique identifiers) haven't been mentioned so far and I won't explain fully. They can be used for several purposes though support in Nvu at present is limited.

Since an id is a unique identifier it can be used once only (on any page). Most commonly ids are used in conjunction with the div element to mark a particular area on a page such as a table of contents, a menu or some such. It is helpful in such cases to give the id a name which indicates its use.

To create a style based on an ID

1. Select a name for the id. This must follow the same rules as for a class. See section 4.2.3.
2. Proceed as if creating a stylesheet and styling an element (see section 4.2.1).
3. At step 6 enter the name of the ID preceded by the number sign e.g. #menu.

To apply an ID

While, in principle, ids are similar to classes it is not possible in Nvu to apply an id via the class selector box but the method is still easy.

1. If necessary close the CSS editor.
2. Click in the div to which you wish to apply the id.
3. On the status bar click the <div> tag. If there are more than one, select that furthest to the right. Both the tag and the content of the div should be highlighted.
4. Without moving the cursor right-click and select 'id'.
5. From the pop-up list select and click the id required.
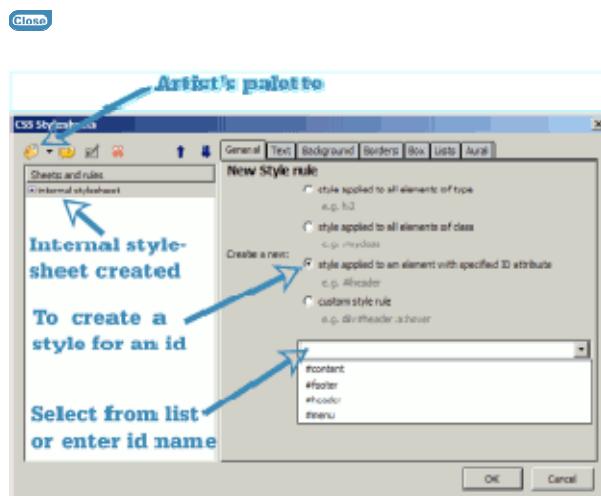
Tip: id names must start with a letter otherwise they may use the same characters as class names.

To create a style based on an id

1. Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens
2. If the stylesheet in which you wish to create the class exists click it followed by the artist's palette button 📷. Else:
3. Click on the artist's palette button. In the sheets and rules box you will see an internal stylesheet has been created for you
4. Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens.
5. To create a rule for an id click 'style applied to an element with specified id attribute'

6. Beside the blank box click the drop down arrow. You will see listed all the ids in use on the page. If you want to use one of these click it. Else:
7. If you want to create a style for an id not yet in use enter the name for the id in the box.
8. Click 'Create Style rule'
9. You are now presented with a window headed 'Selector' followed by the tag for the element. The window actually lists the style declarations for that id, but of course that is now blank.

To apply an ID

While, in principle, ids are similar to classes it is not possibleto apply an id via the class selector box but the method is still easy.

1. If necessary close the CSS editor
2. Click in the div to which you wish to apply the id
3. On the status bar click the <div> tag. If there are more than one, select that furthest to the right. Both the tag and the content of the div should be highlighted
4. Without moving the cursor right-click and select 'id'
5. From the pop-up list select and click the id required

Warning You can apply each id only once on a page. KompoZer at version 0.7.10 will ensure that this occurs however if you apply an id that is already in use no warning is given and the id will be applied as requested with the first instance being removed.

4.3 Removing styles

The CaScadeS allows you to remove styles in a similar way to adding them.
In the 'Sheets and rules' pane select the rule you want to remove and click 'Remove' .

Similarly you can remove a stylesheet. Select the sheet and click 'Remove'. If you select an internal stylesheet it is deleted from the file completely.

Warning this deletion is not reversible using 'Undo'.

If you remove an external stylesheet the link on the page is removed irreversibly but the external file is left intact.

4.4 Inheritance

A web page has a hierarchical structure e.g. text in a table cell sits under the 'Table'. You see some of this structure in 'HTML Tags view'.

Inheritance means that style properties set at a high level in the structure e.g. at 'Body' level will be applied at lower levels unless specifically re-specified. So a font family set the 'Table' element will be applied to everything in the table but if you include a paragraph in the table

and set a different font for the 'p' element then that will be applied. <u>Section 4.2.1.1</u> has already shown a small demonstration of the effect.

Inheritance is very useful as it allows properties to be set once in a style sheet rather than having to be applied to every element that you want to use that property.

You will have to consult a detailed specification (such as appendix F 'Full property table' of the CSS2.1 specification) to see what is inherited and what is not but, in general, you can assume that all properties are inherited except things like Background, Border, Margin, Padding, Width and positioning.

4.5 The division <div> element

The <div> element has been mentioned only in passing so far. A div establishes a 'division', i.e. a section of a page, to contain a number of elements. (Within the div styles are often applied in particular ways as will be touched on later.) The remainder of this section is not intended as a full explanation of what a div is or how to use it rather it is an explanation of how to create and populate a div using. Other aspects of divs are likely to be beyond the interest of beginners.

4.5.1 Creating divs

To create a div

1. Place the cursor where the div is to be created.
   Note Do not try to create a div as the last item on a page unless you do not need to add items after it.
   Note It may be easier to see what is happening in HTML tags view with View > Block outline selected.
2. Click the first drop-down box on format 1 toolbar.
3. Click the last option at the bottom of the list - Generic container (div).
4. The div is created formatted as body text.

It is now possible to insert any desired item into the div e.g. Text or images. Text may then be selected and formatted as paragraphs etc. If required, divs may be inserted so that a set of nested divs is produced.

Sometimes difficulty may be experienced such that elements in a div disappear when an attempt is made to populate them. To avoid this, it is undesirable to insert empty elements, so either insert, for example, one paragraph at a time or enter some dummy text as a placeholder.

To apply a class to a div

Proceed in the normal way (see <u>section 4.2.3.1</u>). The class selector box may be used to apply a class to a div but it will not display the class which has been set. The preferred method to use is via the highlighted tag on the status bar which can be used to add or remove classes.

To inspect or change a class

Use the status bar to access the class.

Elements within a div may have styles applied in the all the normal ways.

Tip: If you do get 'trapped' in the last division on page need to work between two touching divs in the flow there is a way to escape. Click on the last item in the division and then on the marker for the div on the status bar. Next press the down arrow key. This will take the pointer to below the div though the cursor will not be visible.

You can use the same technique to escape from other block elements.

To leave a div

Click in any area outside the div. This is why a div should never be the last item on a page it will be impossible to leave it except to visit an item higher on the page.

Unique divisions

A common use of divs is for unique regions of a page. Examples might be a menu, a table of contents or the main text area. In such cases rather than applying a class to a div it is more appropriate to use an 'id'.

4.5.2 Styles within a div

A div is a useful way of applying a different set of styles within a limited region on a page. Of course a set of classes could be designed and applied to the elements within the region but this is laborious and bloats the code. A simpler way is to establish a set of styles for the elements in the div and to tag them in such a way that they apply only within the div. This is done using an appropriate selector. Suppose the div has a class 'divclas'. Examples of appropriate selectors are:

```
div.divclas p
div.divclas td
div.divclas a:visited
```

Or when using ids

```
#menu p
#contents li
```

These may be set up in the same way as for pseudo-classes see section 4.2.5.2.

4.5.3 Creating a layer using a div

In section 3.10.3 we noted that 'Layers' are contained in 'divs'. In fact a layer is simply a div which has a position, and optionally a z-index, style attached. When KompoZer creates a layer it does so by attaching an inline style to the div.

It is equally possible to create a layer by setting up an appropriate class and attaching this to a div.

When a layer is created in this way KompoZer depicts it as a normal layer and renders it with a surrounding box with sizing and position handles. Visually it is indistinguishable from any other layer.

Such layers may, of course be precisely located and are not subject to the vagaries of manual dragging.

Warning If a layer, created using a class, is subsequently dragged or resized, KompoZer attaches an inline style. Any styling so added will override the class styling.

5   Scripts

Scripts are short programs that are included in HTML documents (and consist of alphanumeric and other characters) which can change the presentation of pages in some way. Scripts must be written in an appropriate script language but this does not form part of the HTML specification. Support for scripting language depends on the browser and/or server employed.

The most common scripting language is probably JavaScript. It runs on the browser and most modern browsers provide support but this can sometimes be disabled if desired by the visitor.

Another common script is PHP. This runs on and depends on support provided by the server. When including PHP code authors should use the fuller tags e.g. `<? php … ?>` and not the short tags `<? … ?>` which KompoZer does not support.

KompoZer allows the use of scripts on pages but does not respond to them. The effect of the script will not be seen on the Normal, HTML Tags or Preview view (but the code will of course appear in Source view).

To see the code at work click on 'Browse'.

Note Some scripts include 'special characters' which, when encoded by KompoZer, result in errors when the script runs. To avoid this, in Tools > Preferences > Advanced, check the box 'Don't encode special characters in attribute values'. You can find more in formation about this in [Appendix 6](#)

In the menu there is a command Tools > JavaScript Console – This Guide does not cover this.

6   Inserts

The menu command Insert provides several options.

Image. See <u>section 3.5.1</u>.

Table. See <u>section 3.6.1.</u>

Form. See <u>section 3.9</u>.

Link. See <u>section 3.7.</u>

Named anchor. See <u>section 3.7.1.3</u>.

Horizontal line. Inserts a horizontal line across the width of a block at the point selected. This doesn't have to be in the simple default style you can define a class and apply it to customise the line. Alternatively the Horizontal Line Properties (double click on the line) allows the width and height, alignment to be set. Colour can be set via the Advanced Property editor using the 'Inline Styles' tab and setting a value for 'background-color'.

Characters and symbols. Provides a wide range of characters not readily available from the keyboard including accented and copyright. There is more about special characters in <u>Appendix 6</u>.

Table of contents. Inserts a table of contents, based on heading styles which can be selected. The table entries are linked to the corresponding headings. See <u>section 3.13</u>.

Smart Widgets. If you're proud of your design and that fact that KompoZer helped you, insert an image and a link to publicise it.

Templates. See <u>section 7</u>.

HTML. Allows you to insert some hand-coded HTML code at any point on a page. (If you want to add code in the HEAD area you'll have to use "Source" view instead.)

Note If you wish to insert HTML into your page this is probably the safest way to do so. While you can still make mistakes in the code KompoZer will carry out some checks and ensure that the result is valid HTML (whether or not what you intended). Editing in Source view (see <u>section 3.16</u>) can cause massive corruption of your page.

PHP Code
Clicking opens a window into which PHP code may be typed or copied. On closing, the code along with the required php tags is added to the source code.

Note KompoZer does not support short tags like <? ?> .

Comments are commonly used by coders in order to improve maintainability of code. Comments do not appear on a page. KompoZer helpfully displays an exclamation mark in viewing modes other than preview which when hovered over reveals the content of the comment.

Break below images. If you have an image aligned right or left (with text wrapping around it) this command breaks the wrap at the point selected, the text continuing below the image.

7 Templates

7.1 What are templates?

Templates are basically pages having some content (eg a letter head) which can be *re-used* to create other pages which will have the same underlying page structure and, often, the same graphical layout.

Templates are not altered in use and can be used over and over again. The simplest template is probably a blank sheet which links to a stylesheet for use throughout a site. More common is a page which has a banner and perhaps a menu to appear on every page. Last might be a complete page layout for use on all, or many, pages of a site but which includes areas for customising individually.
Templates may be considered as having two parts – the fixed part or 'boilerplate' which remains the same for every page and the editable part which changes.

KompoZer recognises two types of editable part. Block and flow. Block items are like any other html block and may consist of paragraph-like items or divs (which may contain several blocks). Blocks may be made repeatable - so that several similar items may be added if required.

Flow items will be contained within other block items in line with the text, for instance to change one or two words in a paragraph.

7.2 Create a new template
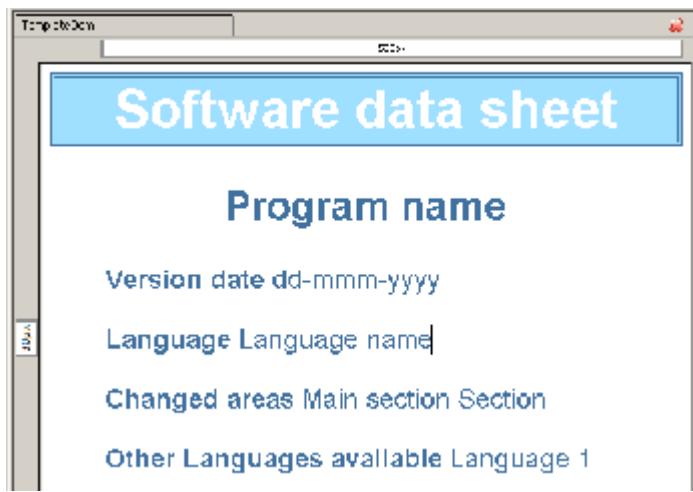
Templates are prepared and edited using KompoZer just like any other page.

To create a template

1. Click File > New then select 'A blank template' and 'Create'. **Note** All templates will be created with the Transitional HTML Doctype.
2. Add any content which is to appear on all pages based on the template.

3. Format this in the normal way.
4. If the formatting is to be based on a style sheet prepare this in the normal way as detailed in <u>section 4.2.4</u>.
Note For this to be successful the template must first be saved.
5. Add the areas which are to be made editable.

Fill these in using dummy text so that a complete page layout is achieved even though some of it may be meaningless. Format this as required.

Note Useful dummy text can be found at `http://www.lipsum.com/`.

Note There are other approaches to page layout but the method described is simple and reliable.

The first figure shows a page prepared as described.

7.3   Saving templates

To save a template

1. Click File > Save or File > Save As.

The extension 'mzt' will be selected automatically.

7.4   Create a template from a page

A pre-existing document may be transformed into a template

1. Click Format > Page Title And Properties
2. Check the box 'This page is a template'.
3. Click OK.
4. Click File > Save as The file type 'HTML Template' will be completed.
5. Name and save the file as normal.

7.5   Setting up template details

The content of the page - boilerplate and examples of editable content - should already be set up and formatted.

To make blocks editable
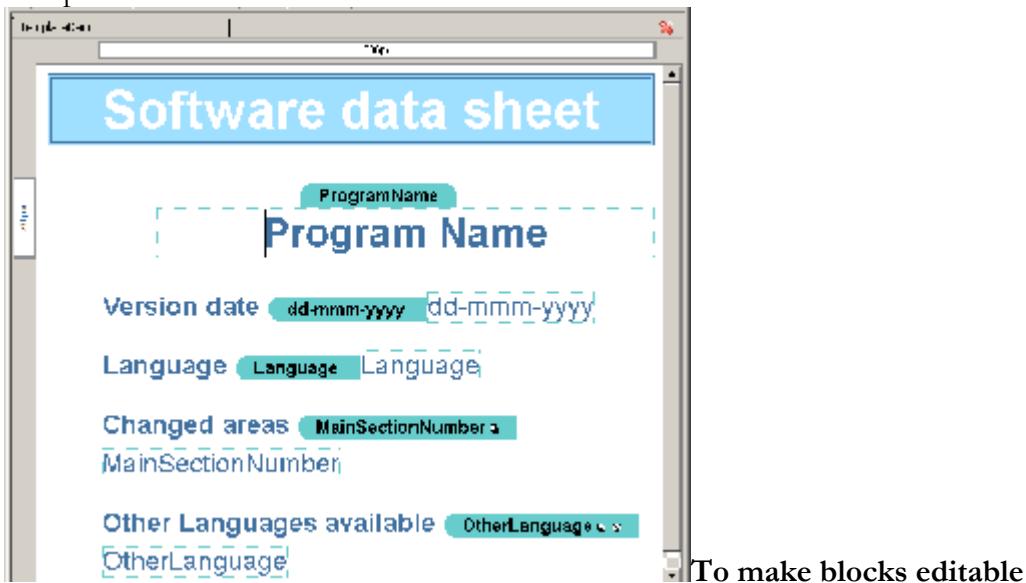
1. In turn, select each block that you wish to make editable.
2. In HTML tags view select the block by clicking its tag.
3. On the status bar right-click the corresponding highlighted tag.
4. Click Templates > Make editable.
5. In the 'Insert an editable area' window give the block a recognisable name. Now check the options boxes if required.

Note The option 'Area is optional' allows it to be deleted easily when the page is in use. 'Area is repeatable' allows copies to be made rapidly. 'Area is moveable' turns the area into a movable layer (see section 3.10).

Warning In KompoZer 0.7.10 this option is disabled

6   Click OK.

To make a flow selection editable

1. In turn select (highlight) each section of text that you wish to make editable.
2. Click Insert > Templates >Insert editable area.
3. In the 'Insert an editable area' window give the block a recognisable name.
4. Leave checked the option 'Flow of text'.
5. Check the options boxes if required as described above.

Note The option 'Area is moveable' is inappropriate for flow areas).

6   Click OK

When you have finished save the template as described in <u>section 7.3</u>

The second figure (above) shows a template at this stage. It has one non-repeatable block item - The program name, Two non-repeatable flow items - The Date and Language, Two repeatable flow items - The Changed areas and Other languages; the second of these is optional.
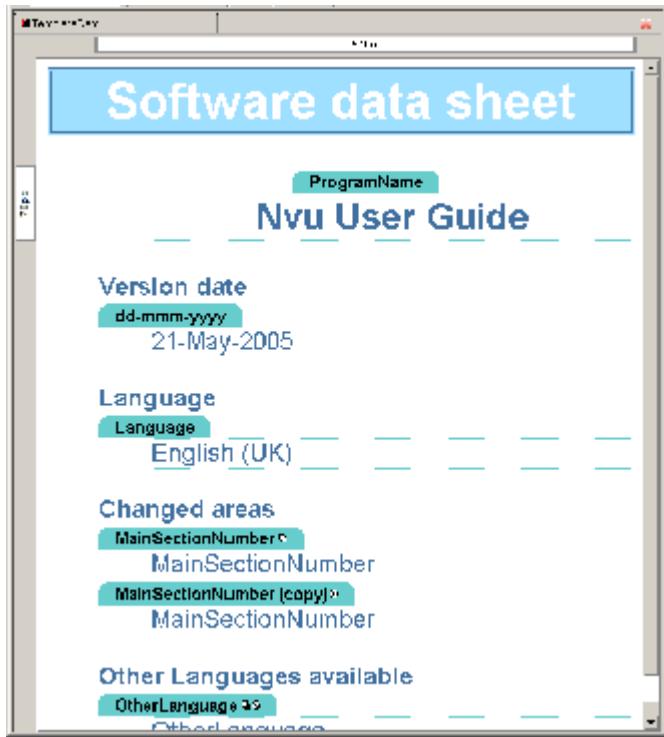
7.6   Using templates

To base a page on a template first ensure that the template itself has been saved and closed – a page cannot be based on an open template.

Note All pages which KompoZer creates which are based on templates are created as HTML documents using the Transitional doctype. It is not possible to create XHTML documents or those with Strict doctypes from templates.

To create a page

1. Click File > New > A new document based on a template > Choose File.
2. Select the Template (Note templates have the file extension 'mzt')
3. Click 'Create'.

The page that appears carries

- Boilerplate items
- Labels for the editable areas within coloured rectangles (with rounded upper corners)
- Sample text (same as the labels) within surrounding editable boxes (with dashed borders).

To use the page

Warning - At KompoZer 0.7.10 the text for repeatable items cannot be edited at this stage, however copies may be made for editing later.

1. Click in turn in each editable areas.

   Software data sheet

   Nvu User Guide

   Version date 21-May-2005

   Language English-UK

   Changed areas 3, 4, 5

   Other Languages available French (in preparation), Japanese (in preparation)

2. Select and delete the sample text and replace it with new text.
   2a If the editable area was repeatable a small square appears within the label, hovering turns it red and clicking makes a copy. Copies have small circles which act as delete buttons. The third figure (above) illustrates the template with one flow item (the Changed areas) repeated once. The first three items have the text replaced.
   2b If any area was optional a small circle with an x appears within the label. Hovering turns it red and clicking deletes it. The same figure shows this for the 'Other languages'. Because of the limitation described it has not been possible to fill in all the editable areas.
3. When all editable boxes have been completed detach the page from the template by clicking Edit > Detach from template. The page now assumes its final appearance.
4. Save the page in the normal way. The last figure shows the final result. The areas which could not be edited earlier have been completed.

Now it is possible to edit any item and as a workaround the frozen repeatable items may be added.

7.7   Editing templates

Templates which have already been saved may be altered after opening using menu commands File > Open File and selecting 'Files of Type' then 'HTML Templates'.

Note The default opening setting is 'HTML Files' which will not open templates.

8   Site Manager

8.1   Overview

The site manager allows you to navigate your site or between sites easily.

To toggle the Site Manager on or off either press F9 or use View > Show/Hide > Site Manager.

Site Manager can deal with sites irrespective of whether they reside on a local machine or on a remote server. In the latter case, if you are on a dial-up network, Site Manager will dial and make the connection for you. Since generally you will set up a site on a local machine and later 'publish' to a remote server we will deal first with setting up on a local machine.

Site Manager provides a directory tree view of a site similar to the view with Windows Explorer. It however lists only directories which you have specifically set up as 'Sites'. You can set up many sites, they appear in Site Manager irrespective of where they appear in a normal directory tree.
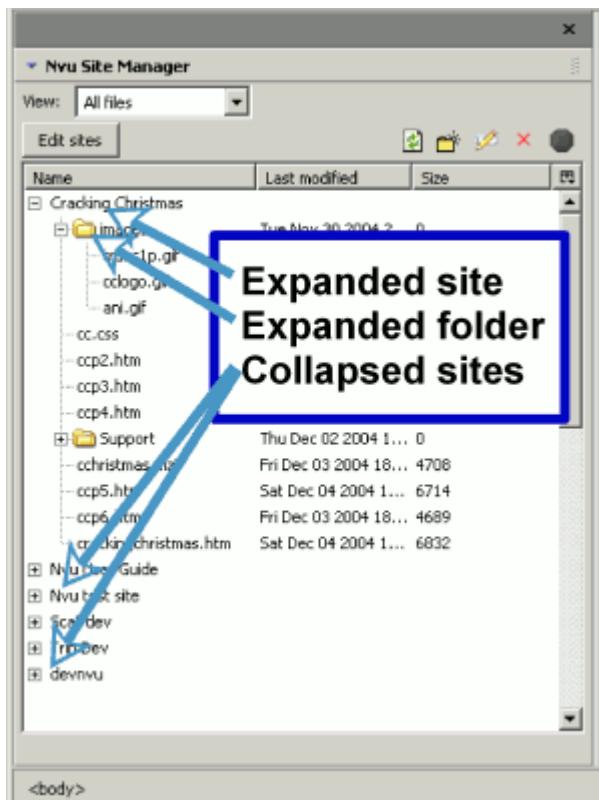
8.2   Setting up sites

To set up a new site (assuming that pages for the site have already been created) open Site Manager, click 'Edit Sites'. This opens the 'Publish Settings' window.

Choose a convenient name for the site (You may like to call it something like "My Site Local" to differentiate it from a later "My Site Remote") and enter it in the box 'Site Name'.

In the 'Publishing server' area click 'Select directory' and browse to the folder where the site is located, select the folder and OK. The box should read something like "`file:///C:/Documents ....`" (Ignore the hint line which applies to remote sites only.)

Leave the other boxes (Web site information, User name etc.) blank.

Back in Site Manager in the column headed 'Name' the new site should be listed and can be expanded to show its contents.

8.3   Site Manager Functions

The main window lists all sites which have been set up and, for any sites which have been expanded, the files contained in it. (See figure.)
Double-click any site to expand it.

At the top of the Site Manager window a drop down box allows you to view all files or to select to view only html files (which includes htm files) or only image files (these include gif, jpg, jpeg and png files).

For files it is possible to display the file size and modified date. To select or de-select these options, in the column heading click the right-most division and select the options required (see figure). (You may have to widen the site manager window to make this possible.)
It is not possible to change the order of the listing.

Double click on any html file to load it directly to the page area for editing.
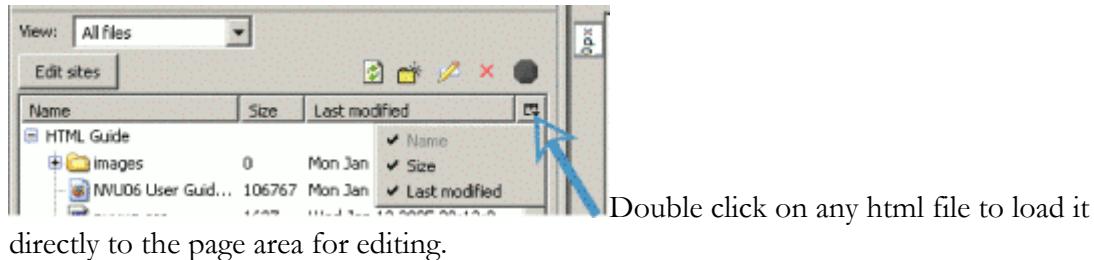
By using the buttons at the top of the Site Manager window, and selecting a file if required, it is possible to rename and delete files and to create folders.

Note all these actions alter the actual files concerned. By using the 'Edit Sites' function and the 'Publish Settings' window, sites may be removed from the Site Manager but this has no effect on the actual folders or files involved only on the view in Site Manager.

8.4   Remote sites

Remote sites – on the server hosting a site – may be set up in almost exactly the same way as local sites (see Section 8.2). This permits the same browsing, display and editing functions as for a local site.

This time all the boxes on the 'Publish Settings' window must be completed. Hints are provided for the content of each. The HTTP address will be used by the Site Manager to find the site.

Warning Remember that any firewall in place must allow KompoZer to access to the site. Unfortunately KompoZer provides little help in resolving any problems which arise while setting up sites.

9 Setting preferences

9.1 Preferences

You can set up a number of features in KompoZer according to personal preferences. Several of the options are grouped under the Tools > Optons menu. In addition you can customise toolbars via the View > Show/hide menu.

Defaults

The defaults set by KompoZer will generally be found satisfactory.

	Menu selection	Nvu default	Kz Default
	Maximum number of pages	10	10
	Retain original source formatting	Cleared	**Checked**
	Reformat HTML Source	Checked	Cleared
Tools > Options > General	Save images when saving pages	Checked	**Cleared**
	Always show publish dialog	Cleared	Cleared
	Maintain table layout	Checked	Checked
	Use CSS styles	Checked	Checked
	Always open a document in a new tab	Checked	Checked
Tools > Options > Fonts	Allow documents to use other fonts	Checked	Checked
	for others see text		
	Author	Blank	Blank (see below)
	Reader's default colors	Checked	Checked
Tools > Options > New Page settings	Background image	Blank	Blank (see below)
	Language	Blank	Blank (see below)
	Writing direction	No direction …	No direction …
	Character set	ISO-8859-1	ISO-8859-1
Tools > Options > Advanced	Set up Proxies	Direct connection	Direct connection
	Markup - Language	HTML 4	HTML 4
	Markup - DTD	Transitional	**Strict**
	Return in paragraph always creates new	Checked	Checked
	Underline misspelled words	Checked	Checked
	Output the following characters	HTML4 Special characters	**Only & < > and non …**

Don't encode '>' outside attribute	Cleared	Cleared
Don't encode special characters	Cleared	Cleared

9.2 Editing preferences

KompoZer can be customised in several ways through the Menu selection Tools > Options mechanism.

All the options may be set at any time. All take effect immediately except for 'New page settings' which do not apply to any existing page.

General group

Retain original source formatting/Reformat HTML Source

If you reformat the source some tags will be indented for easier reading but extra blank lines will be inserted. Note that the source view and the actual file content do not agree exactly.

Save images and other associated files when saving pages.

When saving a file to a new location using 'Save as ..' if this box is checked the images and style sheets will be saved to the same folder as the page. This may be useful occasionally but, where a site is structured with different kinds of files in different folders, that structure will not be preserved.

Warning. Not only will the files not be copied to the file structure specified in the source file but the new saved file will be altered to reflect the new (flat) structure. If you wish to use more complex structure on your site, checking this box will inflict considerable damage to your file necessitating a lot of rework. The original file in the original location will however remain as designed.

Always show publish dialog when publishing pages

If checked, KompoZer always displays the Publish Page dialog box when you Publish a page. If not checked, KompoZer only displays the dialog box if it needs more information in order to publish the page.

Maintain table layout when inserting or deleting cells

Tables are normally created as a regular matrix i.e. every row has the same number of cells. With this box checked inserting a cell in one row should also insert cells in other rows to maintain regularity. With KompoZer 0.7.10 this does not work.

Use CSS styles instead of HTML elements and attributes attributes

Originally HTML formatted pages using elements and attributes. Many of these are not allowed when using strict doctypes but CSS provides alternatives and greater flexibility.

Always open a document in a new tab

When opening pages KompoZer normally creates a new tab to hold it. If this box is checked when a page is opened using Site Manager a new instance of KompoZer is created to open the page. The previous instance remains available. This does not apply when pages are opened by other means.

Fonts

In the absence of a definition of the font to be used a browser will make a choice on some basis built in to the browser. KompoZer acts in the same way and this area allows customisation of the choice made. If the box 'Allow documents to use other fonts' is checked KompoZer will use the font specified in the style or other means. If the box is unchecked KompoZer will not respect styles and will use a font from the list specified on the upper part of this window.

The boxes specifying sizes and screen resolution appear not to work.

New page settings

These define several of the settings used when creating new pages. (Others will be found on the 'Create a new document or template' window which appears via the File > New menu.)

Author's name - Set this if you want the author's name to appear in the head section of all pages. (For a specific page this may be set using Format > Page Title and Properties.)

Readers default colors - Leave this checked. The alternative of 'Use custom colors' allows the colours of links to be specified but uses deprecated attributes which are not permitted for pages using strict doctype. CSS styles provide alternative means of doing the same thing. (Section 4.2.5.2).

Background image - Set this if you want a particular background for all pages.

Language - The main language for the page should be set to facilitate use by automated tools, for instance speech synthesisers. Many languages offer regionalisation alternatives, either select one of these or use the generic.

Writing direction Usually this may be left at the default of 'No direction specified'. Setting the language defines the writing direction.

Character set - Normally for western languages leave this set to the default of ISO-8859-1. If in doubt set to UTF-8. You can find more about choice of encoding in Appendix 6.

Note This is badly entitled. It should read 'Character encoding'. Though it sets a 'charset' parameter it dos not define a character set.

Advanced

Set up Proxies - For most people click 'Connection settings' and check that 'Direct connection' is checked. if you need to set up proxies refer to KompoZer Help.

Markup - The Markup language and doctype may be set as desired. I have recommended the use of Strict HTML.

Note Properly this should be included under New page Settings. **NB** It does not alter an existing page.

Return in paragraph always creates new paragraph - The action of the Return (Enter) key in or at the end of text objects is described in the following table.

	Return in paragraph	Return at end of paragraph	Return in heading	Return at end of heading
Box checked	New paragraph	New paragraph	New heading	Body
Box cleared	New line	New line	New heading	Body

When the box is checked it is possible to create a new line in or at the end of a paragraph or heading by pressing Shift + Enter.

When the box is cleared it is possible to escape from a paragraph by pressing the Return key twice. This creates a new paragraph but also leaves a line break at the end of the previous paragraph. Most people will prefer to leave the box checked.

The action of the return key in other circumstances depends on the context, for instance in a table cell it generates a new line, in a list it creates a new list item.

Underline misspelled words - When set this is a useful reminder to poor spellers or typists who forget to run the spell checker separately, however it consumes resources which may slow the system, in particular the time taken to switch between views may become excessive. It is usually better to clear this setting.

Output the following characters as entities - In normal use this should be set as recommended. No setting will damage a file but other settings are intended for use only during development of pages.

Don't encode '>' outside attribute - Set as recommended.

Don't encode special characters as entities - Set as recommended.

9.3   Toolbars

Buttons on Composition toolbar

New file	Open file	Save file	Publish	Browse	Undo

Redo	Insert or edit named anchor	Insert or edit Link	Insert or edit Image	Insert or edit table	Insert or edit Form

Open CaScadeS
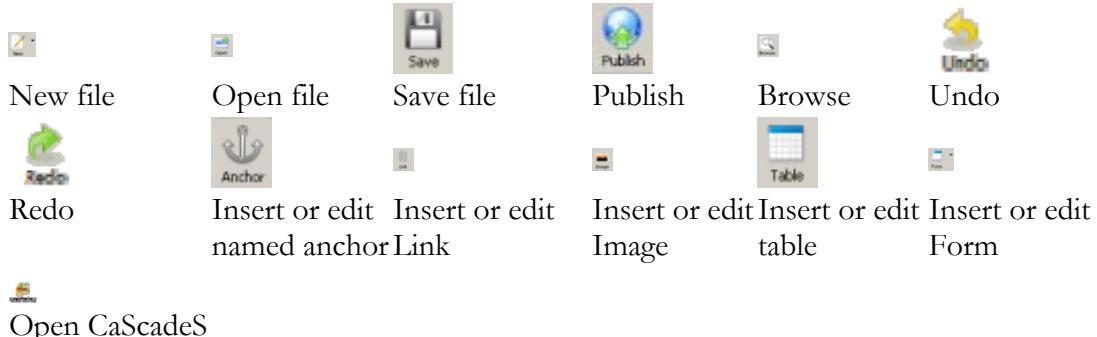
Buttons on Format 1 toolbar

Emphasise	Strong emphasis	Numbered list	Bulleted list	Indent text	Outdent text

Definition term	Definition description

Buttons at extreme right of toolbar

Layer	Bring forward	Send back

Buttons on Format 2 toolbar

Text and background colour	Highlight colour	Make text larger	Make text smaller	Bold text	Italic text

Underline text	Align text left	Align text centre	Align text right	Justify text
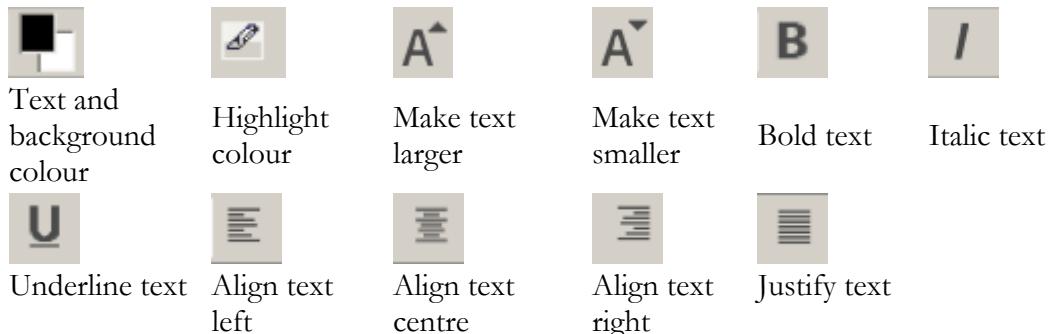
Buttons not normally installed

Cut	Copy	Paste	Print page	Find	Insert horizontal line

Check Spelling

Borders	Left to right	Right to left	Glue to left border	Stay in centre	Glue to right border
Glue to top border	Stay in middle	Glue to bottom			
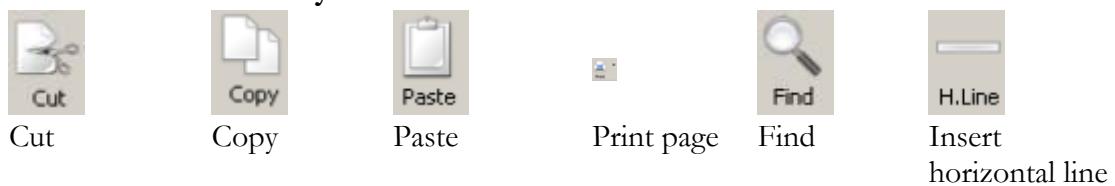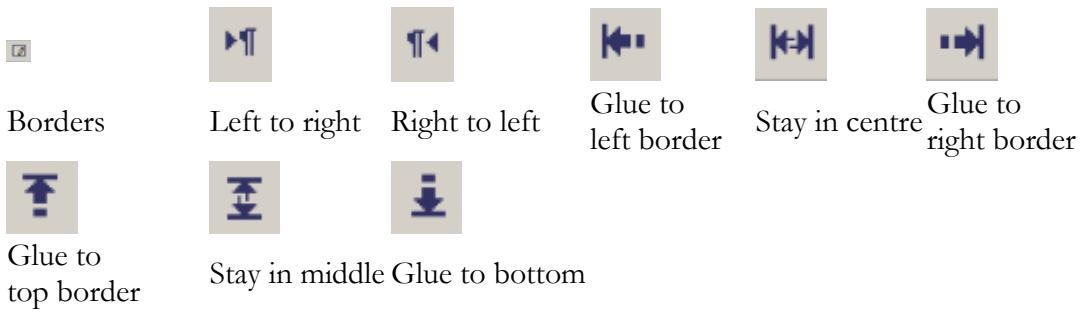
Customising Toolbars

The list above shows the buttons, and the corresponding icons, available on KompoZer toolbars. The buttons are normally located on the toolbar indicated. A button may not be installed by default in which case the bar may be customised to make it available.

You can view or hide the toolbars. On the Menu Bar select View > Show/hide then check which toolbar you wish to display. You cannot hide the menu bar.

You can customise the toolbars to change which items appear on them. To do so right-click on the bar and select 'Customize' then drag an item to or from the toolbar.

As you customise a toolbar you can select whether you wish to display small or large icons. This selection is specific to each bar.

When customising toolbars three additional symbols will be found. 'Separator' provides a vertical bar to separate sections on a toolbar, 'Space' provides an invisible space of fixed size. 'Flexible space' provides a space which will expand to fill any space available.

Menu alternatives

Since the Composition Toolbar is often the simplest way of carrying out a function this guide has generally assumed that buttons available for that bar are visible. If you have disabled the Composition or Format toolbar or a button, functions may be still accessed via the Menu bar. The corresponding selections are shown in the table opposite. If there is a shortcut key (with control key) it is also listed.

Button	Menu bar selection	Shortcut with Ctrl
Anchor	Insert > Named anchor	
Align center	Format > Align > Center	
Align justify	Format > Align > Justify	
Align left	Format > Align > Left	

Align right	Format > Align > Right	
Bold	Format > Text style > Bold	B
Browse	File > Browse page	
Bulleted list	Format > List > Bulleted	
Choose color for text	Format > Text Color	
Choose color for background	n/a	
Choose highlight color for text	n/a	
Copy	Edit > Copy	C
Cut	Edit > Cut	X
Find	Edit > Find	F
Form	Insert > Form	
Horizontal Line	Insert > Horizontal line	
Image	Insert > Image	
Indent text	Format > Increase indent	]
Italic	Format > Text style > Italic	I
Larger font size	Format > Font size > Larger	+
Link	Insert > Link	L
New	File > New	N
Numbered list	Format > List > Numbered	
Open	File > Open File	O
Outdent text	Format > Decrease indent	[
Paste	Edit > Paste	V
Print	File > Print	P
Publish	File > Publish	
Save	File > Save	S
Smaller font size	Format > Font size > smaller	-
Spell	Edit > Spell checking	K
Table	Insert > Table	
Underline	Format > Text style > Underline	U

10   Publishing

10.1 Introduction

Publishing a site means transferring the site, i.e. the pages, images and stylesheets involved, to a web server from which they may be accessed, usually but not necessarily, by the public. This process is called 'Uploading'. Prior to publishing there are a few checks which should be carried out.

10.2   Validating the site.

Validating simply means checking that the page or site operates as intended. Designers may by now have a plethora of files some of which are obsolete or are not required for the published site. A good practice is to create a new folder structure and in it place those files, and only those files, which are needed.

This can be done using your usual file manager e.g. Windows Explorer but the KompoZer File Manager can be set up to do this task by 'Publishing' these to a new folder.

10.2.1   Markup Cleaner

KompoZer provides a Markup Cleaner Tools > Markup Cleaner which can be used to remove redundant code which clutters the file but does not contribute to the page content. This does not necessarily remove all redundant content but makes a good contribution to it.

10.2.2   Browser testing

The next stage of validation has already been taken care of by KompoZer. Users have a high assurance that the code generated represents valid HTML or CSS. The site should next be run using a browser. Possible problems include you having omitted to transfer files to the correct folders, naming the folders incorrectly, including getting the case wrong.

Tip. Those having access to only one computer running Windows XP may like to know that they can run several versions of Internet Explorer on it. Joe Maddalone developed the method and Yousif **Al Saif** has produced an installer. Initially this did not run on Windows Vista but a method of doing so has been developed.

The site should be checked on a range of different browsers. Several modern Mozilla based browsers behave very similarly so do not provide additional confidence. Check with browsers from other sources and be very conscious that a few visitors may still be using very old browsers.

In mid 2007 a minimal set of browsers to test on might be Firefox (latest version), Opera (latest version), Internet Explorer 6 and Internet Explorer 7. Both these latter two behave

differently from each other and also from other browsers so use both. In spite of the availability of Internet Explorer 7 version 6 remains the most widely deployed so it is essential to check with this.

10.2.3   Formal validation

Browsers are often very tolerant of incorrect code. A much more rigorous test is to submit pages to the W3C validators. There are two: one checks that the page code conforms to the HTML specification, the second that the style sheet (internal and external linked) code conforms to the CSS specification. KompoZer provides direct access to the first of these.

To validate a page. Open it in KompoZer. On the Menu choose Tools > Validate HTML. A validator window opens and the page is automatically submitted for validation. The result will be displayed in the window.

To validate pages with internal stylesheets or separate external stylesheets the validators may be used independently. The URLs are given in the References (Appendix 2). Pages may be validated either from the local disk or after the files have been uploaded.

While the W3C validators check rigorously for conformity with specifications this does not guarantee that the code will run correctly. Unfortunately none of the current browsers fully support the specifications! This is why browser validation is an essential step.

10.3   Publishing

You can publish your web pages to a web server on which you have an account. Your ISP probably offers limited free space but you can buy space from professional hosting providers.

To publish a site you need to know the settings for the space so that you can set up your system.

Files may be uploaded using any available FTP program but KompoZer has built-in facilities which are integrated with the Site Manager which makes this task very simple.

While uploading using KompoZer is simple the facilities and the folder structures supported are limited. You may prefer to use one of the available open source or freeware FTP programs.

10.3.1   Setting up your site

While setting up Site Manager you may already have configured the 'remote' site (sections 8.4 and 8.2), if not, either proceed as detailed there, go directly to Publish Settings via Edit > Publishing Site settings.

 Enter the following details:

- In the 'Site names' box enter the name that you want to know the site by.
- HTTP address (URL) of your site. From your ISP (see hints).
- Publishing address – This is the ftp address to which you will publish.
- User name – From your ISP.
- Password – From your ISP.
- If you wish to, check 'Save Password'.

If you have several sites set up and you have one site that you always or usually publish to you may wish to click on the name of this site then 'Set as default'. This simplifies uploading.

Click OK.

10.3.2   Uploading

Open the page that you want to upload. An easy way to do this is from the Site Manager.

1. Click the 'Publish' button.
2. On the 'Publish Page' window on the 'Publish' tab, if it is not your default, in the 'Site name' box select the site to which you want to publish. The 'Page title' and 'File name' should already be completed.
3. If the page is to be uploaded to a sub-directory, rather than the root directory, enter the name of a sub-directory and any of the other data if required.
   Note This directory must exist. KompoZer cannot create it.
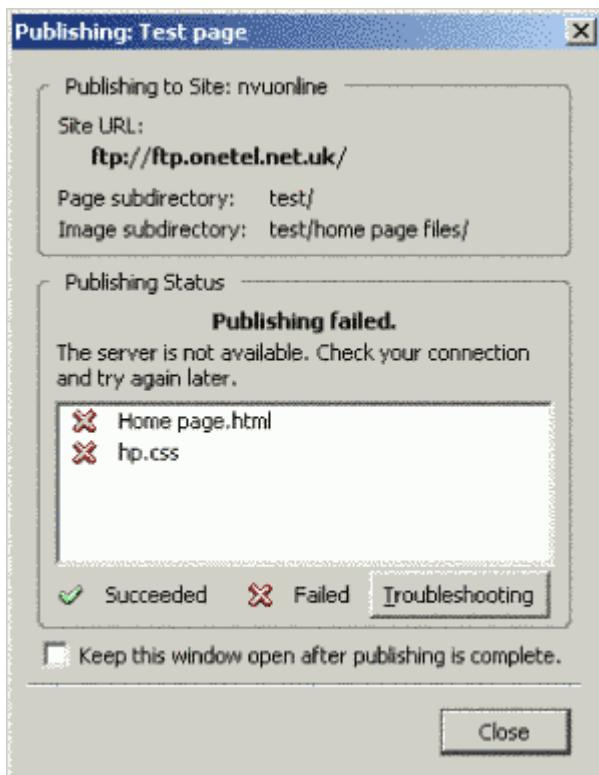
4. If it is the first time to upload the page, and if it includes images or uses external style sheets, check the box 'Include images and other files'. (If it is not the first time and these other files have not changed the box may be left unchecked.) The files will be placed in the same directory as the page. If you want them to go in a sub-directory check the box 'Use this site sub-directory' and name the directory. In this case the directory will be created if needed.
Note this is a sub-directory of the root directory not any directory detailed in the previous point. If you want to use such a directory this must be explicitly detailed e.g. pagesubdirectory/filessubdirectory/
Note all the Images and style sheets will be placed in the same directory. This is the only arrangement supported by the Site Manager Publisher.
5. You should not need to refer to the 'Settings' tab as the data should be collected via the Site name you have selected but you may view the data and change if you wish.
6. Click 'Publish'. A 'Publishing' window will appear and uploading will commence. (If you are on a dial-up connection this will be connected.)
7. Within a short time you should receive confirmation of correct publication similar to the first figure above.
8. One possible source of problems occurs if you are prevented from accessing the site by a firewall. In this case you may receive a 'Publishing failed' message similar to that shown in the second figure.

The 'Troubleshooting' button takes you to the KompoZer help system but this is short of aid in this area at present.

Other possible problems include

- Some required files are missing
- File or directory names incorrect e.g. Wrong case

Once you have published a page, if you need to publish it again, your settings (e.g. subdirectories) should be remembered by KompoZer. You will not see steps 2 to 4 again unless changes have been made to the page.

C'est fini !

Appendices

Appendix 1 – Glossary

This Glossary defines terms as used in this document. As far as possible, when terms used also occur in the HTML or CSS specifications, they have the same meanings as in those specifications. In such cases more precise definitions may be obtained there.

ASCII
> American Standard Code for information Interchange. Defines a set of 95 printable characters.

BMP
> Basic Multilingual plane.

Button
> A small picture, or icon, which when clicked results in an action. KompoZer buttons are of various shapes.

CaScadeS
> The CSS editor that comes with KompoZer.

CSS
> Cascading Style Sheets.

Declaration
> The second part of a style rule. Contains format details of the style.

Deprecated
> A feature marked as deprecated is one which is considered obsolete, and whose use is discouraged. Generally CSS provides better features. Transitional Document types allow the use of deprecated features. (The 'Strict' type does not.)

Directory
> Equivalent to what Windows calls a 'folder'.

Doctype
> Document type Declaration - see appendix 3.

DTD
> Document type Definition - see appendix 3.

Element
> A small part of a web page such as a paragraph or an image which can be manipulated by a browser in ways determined by the code associated with the element.

Extension
> The final section of a filename (after the dot) indicating the type of file.

FTP
> File Transfer Protocol. Used to transfer files to a server.

Handle
> Symbols, usually small squares, on the edges of sizing boxes which, when clicked, allow the size or position to be dragged.

HTML
> HyperText Markup Language. The publishing language of the World Wide Web.

ISP

Internet Service Provider.

KompoZer

The open source software package available at http://kompozer.sourceforge.net/, on which this guide is based.

Mozilla

I use the term to refer to a number of similar browsers including Netscape 6 and later, Mozilla and Firefox.

MSIE

Microsoft Internet Explorer. The browser from Microsoft Corporation.

Non-breaking space

A space character which, though invisible, acts as a normal character in that a 'word' which includes a non-breaking space will not be split to wrap a line.

Nvu

A predecessor software package on which KompoZer is based. The final version available at http://www.nvu.com/, contains bugs which are gradually being eliminated in each new version of KompoZer.

Quirks Mode

For documents with transitional doctype MSIE offers two modes of operation — 'Standards mode' and 'Quirks mode'. In quirks mode it emulates the behaviour of MSIE5. Quirks mode is triggered when the doctype does not comply with the specification and omits the DTD.

Selector

The first part of a style rule. (May be thought of as the name of the rule.)

Sizing box

A box, denoted by small squares, (Handles) which in KompoZer appears around some items when clicked. See also 'Handle'.

Strict

see 'Deprecated'.

Stylesheet

A text file, that may be a part of the page in use, that describes the position, appearance or behaviour of some or all of the html elements of which the page consists. Some terms used in developing stylesheets are explained in section 4.2.1

Tab (window)

Part of many window structures which emulate the tab on a paper filing system designed to give quick access to part of the file. If a tab is clicked a new display will appear within the same window.

Tab (key)

The keyboard key intended to emulate the action of the tab key on a typewriter.

Tag

A piece of code in a web page associated with an element. Typically each element has a start tag and an end tag. Tags include the element name and are surrounded by angle brackets. E.g. <p> and </p>.

UCS

Universal character set. A set of about 100,000 characters.

Upload (ing)

The process of transferring web files from a local computer to the computer which will host them on the web.

URL

Uniform Resource Locator. The web address of an item.

W3C

World Wide Web Consortium http://www.w3.org/ A consortium of the 500 biggest IT corporations who got together to define specifications and recommendations so that languages like HTML or CSS can inter-operate without problems on different platforms, devices, operating systems and media.

Appendix 2 – References

HTML 4.01 Specification	http://www.w3.org/TR/html4/cover.html
CSS2 Specification	http://www.w3.org/TR/CSS21/
W3C	http://www.w3.org/
HTML Validator	http://validator.w3.org/
CSS Validator	http://jigsaw.w3.org/css-validator/

Appendix 3 – Doctypes

A 3.1   Introduction

Doctype abbreviates 'Document type Declaration' which tells a browser which 'Document type definition' (DTD) to use to interpret a page. The Doctype also includes the URL of a file containing the DTD details.

The Doctype must appear on the first line of code for a page.

Note Actually on an XHTML page an XML declaration may precede this but this can result in problems with some browsers. KompoZer does not insert this declaration. Files without an XML declaration use UTF-8 character coding.

A 3.2   Doctypes supported

Note Where the [Nvu]icon occurs click it to reveal differences in the behaviour of Nvu if this interests you.

The Doctypes supported by KompoZer are
HTML Transitional
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
HTML Strict
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
XHTML Transitional
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
XHTML Strict
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

Note For HTML Doctypes when Nvu inserts a document type declaration, it omits the URL This has the undesirable effect of triggering some versions of MSIE into the so called 'quirks mode'.

To set the Doctype for a new file:

On the menu bar click File > New and check the appropriate boxes.

A 3.3 Differences between transitional and strict DTD

In HTML 4.01 strict, elements and attributes which are dedicated to presentation are deprecated in favour of CSS. Therefore, HTML 4.01 strict is actually a trimmed version of HTML 4.01 and is likely to be supported by a larger number of user agents and media. The features which have been trimmed out can be replaced using CSS so that no functionality is lost.

The benefits of clear separation of content from presentation thanks to CSS implementation are:

- smaller files and faster download (Note 1)
- better layout control
- better interoperability across platforms, devices and media (Note 2)
- easier to maintain

Note 1: See Max Design article Why use CSS to separate content from presentation?

Note 2: "Experience has shown that separating the structure of a document from its presentational aspects reduces the cost of serving a wide range of platforms, media, etc., and facilitates document revisions." Ref HTML 4.01 specification

A 3.4 Choice of language and DTD

Beginners to web authoring often ask 'Which doctype declaration should I use?' That is a good and reasonable question.

When working on pre-existing documents KompoZer is capable of working with any of the doctypes supported and you may continue to use these if you want to avoid upgrading to a preferred format.

When creating a new document my recommendation is to **use HTML 4.01 strict doctype.**

I do not recommend XHTML 1.0 doctype declarations (transitional or strict) unless you **know** that you need to serve your web page as content type `application/xhtml+xml`.

Note Internet Explorer 6 and Internet Explorer 7 will not serve a document as content type `application/xhtml+xml` when the proper content type for XHTML should be `application/xhtml+xml`.

Further explanation on why XHTML is not recommended – at least at present – would go beyond the scope of this User Guide.

People who want to know more on this issue are recommended to read the following resources:

- [Say No to XHTML](): an excellent article on misunderstood claims of XHTML – at least, on serious misunderstandings about XHTML – and problems related to XHTML.
- [Beware of XHTML]() Another excellent article, worth reading carefully. It has 16 examples and demonstrations showing the issues involved.
- [Sending XHTML as text/html Considered Harmful]() from Ian Hickson, author of Web Hypertext Application Technology, Applications 1.0.
- [XHTML is dead]() from Tommy Olsson.
- [XHTML's Dirty Little Secret]() from Mark Pilgrim
- [XHTML - What's the point?]() from Henri Sivonen

Even if you are working on an existing document you may consider that it is worth whatever additional work would be required to upgrade to using the HTML 4.01 strict doctype.The benefits of using web standards (W3C recommendations and language specifications regarding web page authoring) are numerous, important, often underestimated and often misunderstood.

- [Using Web Standards in Your Web Pages]() is an excellent introduction to this issue which deserves careful study.

A 3.5 KompoZer capability

KompoZer can create pages using any of these doctypes. Pages created from Templates are always generated using HTML 4.01 Transitional doctype.

Note It may be possible to edit a template external to KompoZer and change the doctype so that pages created from it use the corresponding Doctype. I have not tested this.

HTML pages created will offer, on saving, the "html" extension. XHTML pages will also offer "xhtml" extension. Regarding XHTML pages, the choice is important as an xhtml file extension will lead browsers to serve such document as content type `application/xhtml+xml`. If you save your XHTML document with the "html" file extension, then your XHTML document will be served as text/html.

KompoZer can open and edit pages with any of these doctypes. The doctype is not editable but KompoZer will not change any doctype encountered.

When creating and editing documents, KompoZer will generate code which is appropriate to the doctype declared using only appropriate elements and attributes. This means that some features may be disabled or be inaccessible depending on the type of document encountered. Typically this may necessitate the use of a CSS style to obtain some effect which is not accessible using an HTML attribute.

KompoZer cannot convert from one doctype to another so, if the doctype is not consistent with the code on the page the result will be unreliable. It is because KompoZer cannot convert doctypes that I suggest that you continue with the existing doctype when working on pre-existing documents . If you do wish to convert a page it is possible to take content

from one document to another with a different doctype by cutting and pasting in normal view. However, when you paste, you must use the 'Paste without formatting' option and reapply all the formats needed. Whether or not you take this option will depend on circumstances.

A 3.6 Browser responses

While browsers will render pages irrespective of the doctype, detailed differences in response occur.

Doctype switching by David Hammond has the results of some excellent research and documenting the responses of all the main current browsers.
Information for older browsers was provided by Henri Sivonen.
CSS Enhancements in Internet Explorer 6 gives fuller details for this browser.

Appendix 4 – Installing KompoZer and Extensions

A4.1 Installing KompoZer

Note Where the ![Nvu icon] icon occurs click it to reveal differences in the behaviour of Nvu if this interests you.

![Nvu icon]

Installing Nvu is straightforward.

1. Download the Windows full installer from the downloads page at http://www.nvu.com/ there is no charge for this.
2. Save the program wherever you wish. Possibly in My Documents\Downloads.
3. Double click the file to start the installation process.
4. Answer the prompts. You will probably want to install to C:\Program Files\Nvu

![Close]

The program may be downloaded from KompoZer's web sitehttp://kompozer.sourceforge.net/.

At the time of writing no installer is available for KompoZer but this is expected to be available soon. With an installer available:

1. Download the Windows full installer from the downloads page
2. Save the program wherever you wish. Possibly in My Documents\Downloads
3. Double click the file to start the installation process
4. Answer the prompts. You will probably want to install to C:\Program Files\KompoZer

If no installer is available:

1. Download the Win32 tarball
2. Save the program wherever you wish. Possibly in My Documents\Downloads. The file will have the extension 'zip'
3. Decide on the folder in which you wish to install the files – probably C:\Program Files\KompoZer
4. Double click the 'zip' file and extract to the required location
5. In the same location find the file `kompozer.exe'
6. Create a shortcut to this and place it either on the desktop or the taskbar

The program may now be run by clicking the shortcut. The first time it is run it will create a 'profile'. (See next section.)

Note With earlier versions of KompoZer you were advised not to run Nvu and KompoZer simultaneously. This no longer applies.

A4.2 Reinstalling/updating

It is not expected that Nvu will be updated after version 1.0 but you may need to upgrade from an earlier version or possibly to repair a broken installation.

Nvu stores a number of settings in files located in the 'Profiles' folder. KompoZer does the same thing and maintains a separate, independent profiles folder.

The profile includes:

- Some default settings
- Details of the sites which you have set up in 'Site Manager'
- Publishing Site Settings
- Details of any add-ons installed
- Your custom spelling dictionary.

The Nvu team recommend that this folder be deleted prior to installing a new version of Nvu. This means that these settings will be lost and have to be re-entered following the new installation. It is advisable to follow this recommendation.

The default profiles folder is located at `%AppData%\Nvu` (i.e. Something like `c:\Documents and Settings\<login settings>\Application Data\Nvu`). (Similarly for KompoZer.)

The procedure for updating is therefore (applicable to Nvu only)

1. Download the latest version of the program
2. Make sure that you have a record of your Publishing site settings
3. De-install the program in the usual way e.g. From Start > Settings > Control panel > Add or remove Programs.
4. Delete the profiles folder.
5. Install the new version of Nvu
6. Set up Site Manager and Publishing Settings

A4.3 Installing extensions
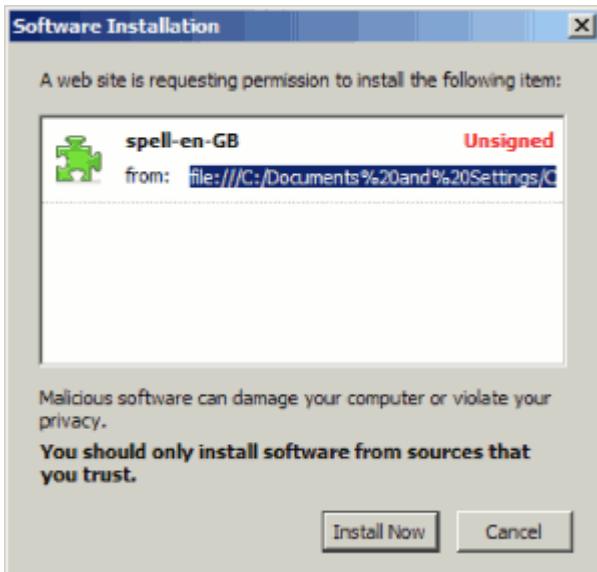
There are a small number of extensions available for KompoZer including alternative spelling dictionaries.

Extensions are installed differently from normal programs. the procedure is as follows:

- Locate the web page from which the extension may be downloaded
- Download and save save the extension on your hard drive using the link 'Save link target as…'.
- Launch KompoZer.
- Tools > Extensions > Install.
- Browse to the folder where the extension file is located and select the file.
- In the window that opens click 'Install now'.
- Close and relaunch KompoZer.

If you have extensions installed you may sometimes wish to run KompoZer without the extension. For this you need to have one profile which has the extension installed and one without. Managing profiles explains how to achieve this.

A4.4 Managing profiles

In order to manage profiles you must start KompoZer with profiles enabled.

To enable profiles:

1. Right click the shortcut icon which you use to start the program.
2. Click 'Properties'.
3. On the shortcut tab .`Target' will show something like "C:\Program Files\Kompozer\kompozer.exe".
4. Change this by adding a space followed by -p to read something like "C:\Program Files\Kompozer\kompozer.exe" -p

If you use more than one shortcut route you have can set others in the same way.

When you launch the program you will now get a window like that shown on the right.

You can launch the program from here but you may prefer first to add other profiles.

To add a profile:

1. Click the 'Create Profile' button
2. Answer the prompts

You will be offered the opportunity of giving the profile a more meaningful name. You will also have the opportunity of selecting where you wish to store the files for the profile. If you change this to somewhere in My Documents it may be easier to manage your routine backups.

Note You should create a dedicated folder for each profile.

You can also rename and delete profiles as shown in the second figure.

Before you add any additional profiles the 'Default profile' will be the profile that you were previously using. It will include all the customisations that you had prepared.

Any additional profiles added will be created in a 'virgin' state with none of your history included. You will therefore have to add any site manager and publishing settings that you wish. Because of this you may prefer to set up additional profiles for specific purposes and retain others for more basic functions.

[I am grateful to Neil Parks on the Wysi Forum for some of the above information.]

Appendix 5 – Revision History

29- Sept-2007

Updated to reflect KompoZer 0.7.10 which involves changes to the way CaScadeS works. Main changes to section 4 but references to Nvu now mainly hidden to make KompoZer methods clearer.

28-Jul-2007

Revised throughout.
Now incorporates KompoZer
Pages are printable
Appendices 6 (Character encoding) and 7 (Printing pages) added.
Profiles explained and more details of spelling.
Links to translations of each page
Many minor changes throughout.

10-Jun-2006

Section 3.7.2 - Amended

30-Mar-2006

A3.2 - Note amended

13-Mar-2006

A number of minor corrections throughout

07-Jan-2006

Minor changes 3.6, 3.6.2.1, 3.6.2.4, 6, 3.3.2, 3.4.1.2, 3.6.2.4, 4.1.3.2, 4.1.3.4, 4.2.1.1, 4.2.3, 4.4, 4.5, 4.5.1, 4.5.2, A3.4
Rewrite 3.6.5 (subject change), 9
Corrections 3.10.7, 4.2.5.3, 4.2.5.4
Revisions 3.10.5, 4.1.3.3, 4.2.1.1, 4.2.1, 4.2.1.2, ~~4.3.2.4~~, 4.2.3.1, 4.5

30-Nov-2005

Section 3.7.2 - Rewritten (was completely wrong), Minor addition 3.4.2

30-Jul-2005

Minor corrections - Clarification in 3.7.2, Formatting error section 4.5.2, Wrong figure and associated text Section 7

15-Jul-2005

Based on Nvu 1.0

Section 3.2.1 - Document types -- rewritten.
Section 3.4.2 - Notes reorganised, 2 & 3 added.
Section 3.4.4 - Lists – rewritten
Section 3.4.7 - Special characters – enhanced.
Section 3.4.8 - Spelling – rewritten
Section 3.5.2 - Positioning images -- revised
Minor changes. 3.6.1, 3.6.2.1, 3.16, 9.1.
Warnings no longer applicable deleted – Sections 3.4.4, 3.5.2

06-Jun-2005

Based on Nvu 1.0PR

Using different doctypes incorporated - mainly section 3.2
Guide uses Strict doctype.
Top menu and linked pages added.
Section 3.10 - Forms - rewritten.
Section 3.13 - Table of Contents - added.
Section 4.2.5 - Style selectors - added.
Section 7 - Templates - largely rewritten.
Appendix 3 - Doctypes - added.
Appendix 4 - Updating Nvu - added.
As a result of above some renumbering has occurred.
Throughout, smaller editorial and styling improvements have been incorporated.

01-Apr-2005

First version available directly via nvudev.org.
Translations into French and Japanese commenced.
Section 9.2 on Toolbar Buttons added.
Section 4.6 on Styles Special Cases (Links, Class selectors) added.

17-Mar-2005

Based on Nvu 0.9

Nvu 0.9 new or changed features: Border Button (section 3.11), email link (3.7.1.2), <div> element 4.5, set Icon size (9.1), Table properties (3.6.2.4), HTML Source view renamed. Previously omitted items added: Printing (3.3.3), Position grid (3.10.4), CaScadeS Beginners mode (4.2.3.5), Image sizing using Property window (3.5.2).

Clarifications or corrections to: Drag and drop (3.3), Comments (6), Sizing boxes refer to 'Handles', other minor items.

07-Feb-2005

Based on Nvu 0.81

HTML version unframed. Optimised for Mozilla-type browsers.
Section 3.6 on tables rewritten.
Section 3.10 Layers (previously Absolute positioning) expanded.
Section 7 Templates corrected and expanded.
Throughout - extensive re-work, minor corrections, clarifications and expansions.
Warnings differentiated from notes.
The writing style of the guide is being altered progressively but still incomplete.

25-Jan-2005

Based on Nvu 0.7

First version available in both HTML and pdf formats
HTML version based on frames
Footnotes incorporated into main text
Major rewrites Sections 8 (Site Manager), 10 (Publishing)
Additional material Sections 3.4.6-7 (Text), 3.6.2 (Tables), 3.10 (Absolute positioning), 3.14 (HTML Editing).
Minor editorial changes throughout.

Appendix 6 – Character Encoding

A6.1 Introduction

Encoding refers to the details of how the characters in the source file for a web page are coded for transmission over the web. For the most part the author can leave all the details to be handled by KompoZer or the browser. This leaves a few options available to the author these generally provide a means of optimising a file only in rare cases do they affect functionality.

A6.2 Quick start

KompoZer defaults to using ISO-8859-1 encoding with the following settings. You may check or reset this as follows.

- Use Format > Page Title and Properties and in the Internationalization area see that the 'Character set' box reads 'ISO-8859-1'. To change it using the drop down menu ' Choose a charset'.
- Use Tools > Preferences> Advanced and in the 'Special characters' area check that 'Output the following characters as entities' is set to 'Only & < > and non-breakable whitespace'.
- For the two check boxes below this:
  Clear 'Don't encode > outside of attribute values'
  Check 'Don't encode special characters in attribute values'

These settings are completely suitable for pages using English and adequate, though not necessarily optimised, for most other languages used in Western and Northern Europe.

If using an other European language and some other languages a different selection from ISO-8859 may be preferable. KompoZer offers the full available range. Wikipedia [Ref 14] has a useful article detailing the coverage.

A6.3 Encoding text

A6.3.1 ASCII and Latin 1 characters

Table A6.3-1

Printable ASCII and Latin 1 characters

Hex code for character - msd in row lsd in column

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF		
0x																	⇑	⇑
1x																	A	
2x	sp	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	S	L
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	C	A

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	nbsp	!	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	shy	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ê	Ë	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

The characters sp (space), nbsp, (No-break space),
shy (soft hyphen) are printable but (normally) invisible.

Early computers used the ASCII (American Standard Code for Information Interchange) which provides a set of 95 printable characters dating from the teleprinter era. An eight bit byte however allows a doubling of this number (while reserving a number of codes for control purposes) and gives rise to the Latin-1 set illustrated in table A6.3-1. The row and column headings indicate the more and less significant parts of the code (in hexadecimal) corresponding to each character. For the code for character 'A' is 41.

Latin-1 corresponds to the ISO-8859-1 set which is sufficient for web pages in English and many other western European languages. Include the appropriate code in a file and the corresponding character will appear.

A6.3.2 ISO-8859

The needs of many languages, European and other, can be satisfied by similar sets of characters, all share the ASCII characters and substitute some in other positions. This give rise to 15 standards in the ISO-8859 series. You can find which language, along with the list of characters, supported by each encoding in the Wikipedia article [Ref 9] referred to above.

To implement this it is clear that more than 256 characters are needed although only 256 locations (less control positions) are available to address them. The characters required to satisfy all in the series are drawn from a much larger set.

A6.3.3 Universal Character Set

The Unicode Consortium [Ref 17] have standardised a universal character set (UCS), i.e. a standard that defines, in one place, all the characters needed for writing the majority of living languages in use on computers. It aims to be, and to a large extent already is, a superset of all other character sets that have been encoded.

>Unicode (as the UCS is commonly referred to) can access over a million characters of which about 100,000 have already been defined. These include characters for all the world's main languages along with a selection of symbols for various purposes.

HTML specifies a Document Character Set which is a list of the character repertoire available along with the corresponding code points (sometimes referred to as code positions). For HTML (and XHTML) the Document Character Set is identical to the UCS which means that, in principle, any character in the UCS may be used in any HTML document. In practice support for the complete character range is uneconomic and systems provide support for subsets only.

Table A6.3-2
**Printable ASCII and Greek characters
Using ISO-8859-7 encoding**

Hex code for character - msd in row lsd in column

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	sp	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	nbsp	'	'	£	€	₯	¦	§	¨	©	ͺ	«	¬	shy		—
Bx	°	±	²	³	΄	΅	Ά	·	Έ	Ή	Ί	»	Ό	½	Ύ	Ώ
Cx	ΐ	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
Dx	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί
Ex	ΰ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
Fx	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	

(Right margin label spanning rows 1x–7x: ⇑ A S C I I ⇓)

A6.3.4 Character Encoding

Character Encoding, at its simplest, refers to the process whereby the codes for the characters are mapped to the code points for the Unicode characters appropriate to the language in use. In the case of ISO-8859-1 the character codes are mapped to identical Unicode code points. (The first 256 Unicode characters being the same as the Latin-1 set.) As another example, ISO-8859-7 encodes Greek characters displacing many from the Latin-1 set to make room. (Compare table A6.3-2 to table A6.3-1.) In this case the code EA instead of being mapped to Unicode code point EA (giving e circumflex ê) is mapped to

code point 03BA which returns a small kappa κ. In fact ISO-8859-7 does not include the ê character.

All ISO-8859 encodings retain the ASCII characters at the original positions.

This document uses `ISO-8859-1` encoding but, in spite of this, has no difficulty in representing the full repertoire of the Greek characters covered by `ISO-8859-7` as can be seen in the table. How this is achieved is explained in the next section.

Authors should note that every page uses one character encoding, and one only, irrespective of the number or range of languages encountered on a page.

In HTML pages character encoding is specified using the 'charset' parameter in the head area for each page. Several options are permissible but KompoZer always uses the form `<meta http-equiv= "content-type" content= "text/html; charset=ISO-8859-7">`.

Note 'charset', in spite of its name, does not specify a character set. The character set for HTML documents is always the UCS. 'charset' specifies the encoding.

A6.3.5 Character references

Table A6.3-3
Important entity references

Character	Entity	Numeric character reference	
€	€	€	€
<	<	<	<
>	>	>	>
×	×	×	×
÷	÷	÷	÷
&	&	&	&
"	"	"	"
no-break space			

ISO-8859 uses a single byte per character to represent all the characters commonly expected in a language but clearly there may be a need to represent uncommon characters. HTML provides two mechanisms – Character entity references (entities) and numeric character references. Using these methods any character in the UCS may be reached by using a sequence of ASCII characters to point to the required character. Entities take the form `€` and numeric references the form `€` or `€` all representing the euro symbol. The `8464` and `x20AC` represent the Unicode code point for the symbol in decimal and hexadecimal notation.

These methods free the author to employ Unicode characters, irrespective of the encoding in use, at the expense of increasing file size. Where such use is limited this is inconsequential.

The list of entities is included at section 24 of the [HTML specification [Ref 16]](). About 250 are defined, numeric character references must be used for characters outside this range. Characters do not have to be out of range of the encoding for entity references to be provided as is clear from Table A6.3-3 which lists some of the most frequently used including some in the ASCII set.

Note Entities are case sensitive thus `É` represents upper case E with an acute accent (É) while `é` represents the corresponding lower case letter (é). `&EacutE;` does not represent anything (&EacutE;). (The error just gets printed out.)

Note Irrespective of the ISO-8859 encoding employed the entity or numeric reference to be input remains the same. So, although in ISO-8859-7 the euro symbol is represented as byte A4, entering the code `¤` will generate a ¤ symbol not a euro symbol. The code to be input is the entity or numeric character reference for the character required.

A6.3.6 UTF encodings

ISO-8859 is fine when using one language at a time but becomes clumsy and slow when languages are mixed. UTF coding releases us from this restriction and provides a mechanism for addressing the full range of Unicode characters quickly. KompoZer allows coding in either UTF-8, UTF-16 or UTF-32 formats which are based on units of 8, 16 or 32 bits respectively. UTF-32 is not usually used for coding web pages.

UTF-8 uses 1 to 4 bytes to represent a character. It uses 1 byte to represent characters in the ASCII set, two bytes for the next 1920 characters (including the Latin alphabet characters with diacritics, Greek, Cyrillic, Coptic, Armenian, Hebrew, and Arabic characters) and three bytes for the rest of 65,000 characters in the Basic Multilingual plane (BMP). Supplementary characters use 4 bytes.

UTF-16 uses 2 bytes for any character in the BMP, and 4 bytes for supplementary characters.

UTF-32 uses 4 bytes for all characters

KompoZer offers additional practical advantages when UTF-8 encoding is used for pages with multiple languages. Irrespective of the encoding in use in Normal or Preview mode KompoZer depicts all characters correctly using the corresponding 'glyph'. In source view encodings which require characters to be represented as character references show the character reference. This reduces readability significantly. With UTF encoding all characters can be represented as glyphs so the problem is avoided.

A6.3.7 Symbols

Traditionally computers have relied on special fonts like 'Symbol' or 'Wingdings' to produce symbols. This is not necessary on web pages. Since such fonts do not support Unicode any attempt to use them will yield unreliable results which may vary from browser to browser.

Fortunately Unicode supports a large range of symbols which fulfills many needs.

Inputting special symbols

Table A6.3-4
Some useful symbols

Symbol		Key code	Unicode
En dash	–	alt+0150	U+2013
Em dash	—	alt+0151	U+2014
ellipsis	…	alt+0133	U+2026
Left single quote	'	alt+0145	U+2018
Right single quote	'	alt+0146	U+2019
Left double quote	"	alt+0147	U+201C
Right double quote	"	alt+0148	U+201D
Euro	€	alt+0128	U+20AC
Pound	£	alt+0163	U+00A3
Degree sign	°	alt+0176	U+00B0
Multiply sign	×	alt+0215	U+00D7;
Divide sign	÷	alt+0247	U+00F7;

There are several ways of inserting symbols into a page using KompoZer.

1. Via the Insert menu. Using Insert > Characters and Symbols. Covers accented characters and a number of other common symbols
2. Using Windows Character Map. Using the standard font in use find the symbol required, select and copy it, then in KompoZer paste it as required. Recent versions of Character map allow you to group characters by Unicode subrange which makes it easier to find a particular symbol
3. If you use a particular symbol frequently it may be easier to insert it by keystroke. Several characters permit this and, where it can be done, the keystroke is shown at the bottom right corner of Character Map. The euro symbol, €, for instance, may be inserted using `ALT+0128`
4. AllChars [Ref21]http://allchars.zwolnet.com/ is a useful utility that allows any program to insert any Windows-1282 (see below) character using a few easily discovered keystrokes
5. In KompoZer Source view any character may be inserted using the Numeric character reference. In the bottom left corner of Character Map this is given in hex format. Thus for euro you insert `€`
   Note The code is shown in Character map as U+20AC This is the conventional way of representing Unicode characters i.e. by 'U+' followed by the hexadecimal code point

6. In source view entity references may be inserted similarly. Many are easy to memorise e.g. `€ > <`

The first four methods can be used in normal or preview mode, the last two in source view.

The keen-eyed may observe that the key codes are neither the character codes nor the Unicode code points for the character required. In fact they are the (decimal) character codes derived from Windows-1252 encoding. KompoZer will convert these to whatever code is appropriate depending on the encoding selected and the character involved. (So the key code never appears in the source.) Incidentally Windows-1252 [Ref 13] encoding is a possible alternative to ISO-8859-1 suitable for western languages. It increases the number of available character codes to 218 characters by re-allocating some of the codes in the range 80 to 9F which are normally unused.

Alan Wood's website [Ref 2] is a useful resource listing entities (where defined) and Numeric character references for a large number of characters from the Monotype Typography Symbol font (as on Windows XP) including Greek, Mathematical and Punctuation [Ref 6] and also the Microsoft Wingdings font [Ref 7]. (For Windings in several cases there is no Unicode equivalent.)

A6.3.8 Unicode support

A6.3.8.1 Fonts

Although Unicode offers tremendous potential the usual caveats apply when choosing fonts. See for instance section 4.2.1.1 but, in this case, it is important to check that all fonts in the list include the characters required. No font covers the full range of Unicode, or even a small single digit percentage of it. To check the supported Unicode ranges of a font Microsoft supply an extension [Ref 12] for Windows Explorer. With it installed, right-click any TrueType (TTF) font file in Windows Explorer and select the Properties tab. Particular characters can be searched for using Character map.

Checking for support is more than usually difficult if unusual characters are required. Compatible fonts must be installed on any visitor's computer and, where in a style sheet the font-family is specified as a prioritised list of font family names (as it should be), ideally all fonts in the list should be checked.

Note A font list lists fonts the first of which will be used if available. It does not check that the character required is supported by the font. So even if support is provided by a font lower in the list that may not be accessed.

Check the rendering of a page on as many different browsers as possible. Mozilla browsers do authors only a partial service since, if a character is encountered which is not included in the font(s) listed they will make an attempt to find the character on other fonts installed on the machine. Authors should however check all pages using MSIE 6 which does not offer this capability. (MSIE7 will substitute for a few characters.) Visitors may finish up looking at square boxes instead of the character required.

Alan Wood offers several pages which are extremely useful in this respect. Using special characters from Windows Glyph List 4 (WGL4) in HTML [Ref 3] lists characters in the WGL4 set and which are likely to be widely available. Unicode fonts for Windows computers [Ref 4] lists which fonts carry specific ranges of Unicode characters and more interestingly shows distribution of the fonts so that authors may check likely availability to visitors.Those wishing to use a rarer character may check which fonts include them at Unicode character ranges and the Unicode fonts that support them [Ref 5].

A6.3.8.2 Examples

Example

Character U+21D1 not included any font used from the list specified (Tahoma, Arial, Helvetica, sans-serif).

⇑

The arrow appears as a square when using MSIE ≤ 7

Same demonstration but set up spanning the arrow with `font-family: 'Lucida Sans Unicode'` .

⇑

While preparing this page, for instance, Table A6.3-1 displayed correctly in Firefox and KompoZer but in MSIE the arrows originally appeared as squares. The issue is reproduced in the box on the right. The arrows use comparatively rare characters that do not appear in the Tahoma font used but, on the writer's machine at least, the Gecko engine was able to retrieve them, possibly from Lucida Sans Unicode.

The result is that visitors using MSIE see boxes instead of arrows but those using Firefox or Opera may see the arrows if Lucida Sans Unicode or some other font with the characters is installed on their machine.

A work-around this issue is possible, as also shown in the box. The font for the arrow is specified as 'Lucida Sans Unicode'. If this is available it will be used, otherwise the inherited font will be used. Alan Wood shows that this font is supplied with Windows XP and Windows 2000 which cover 90% of installations (mid 2007).

This is a moderately, but not very, robust solution. Had the availability of the arrows been critical to understanding the table it would have been necessary to change the design.

While the arrows may be considered rare and unusual characters even characters covered by some ISO-8859 options may not be reliable. In viewing Table A6.3-2, depending on the browser in use and fonts installed there are two characters, Drachma sign (Code A5) and

Greek ypogegrammeni (Code AA), which may not display correctly. In cases like these checking the WGL4 list may provide a warning because neither of the characters is listed.

A6.4 Special characters

A6.4.1 Entities or numeric character references

A6.4.1.1 Output the following characters

The way in which characters are coded in the source for the page may be altered in KompoZer using Tools > Preferences > Advanced. In the Special Characters area there are four options under 'Output the following characters as entities'

- Only & < > and non-breakable whitespace
- The above and Latin-1 letters
- HTML 4 special characters
- Use &#…; notation for all non-ASCII characters

The options refer to characters typed onto the page which will be readable by a visitor to a site. Irrespective of the option set the visual appearance on screen will remain the same.

- **First option**

Note In normal practice the character referred to as 'non-breakable whitespace' is called 'no-break space' (entity).

Only & < > and non-breakable whitespace

The section on 'Preferences' recommends this as the preferred option.

This is the minimal setting. The characters listed must always be encoded whatever option is selected. With this selection the encoding will be as entities. Since the character < occurs in HTML code to mark the start of an element, if it is included in the page text the browser would expect to to start a new element and the page would become corrupted. It must always be encoded. The > character marks the end of an element and should be safe to use but W3C recommend that it also be encoded since it may confuse older browsers See section 5.3.2 of the HTML Specification [Ref 16]. If you wish to override this check the box 'Don't encode > outside of attribute values'.

Since entities and numeric character references start with an ampersand (&) a similar problem occurs with this character.

Outputing the no-break space as an entity is convenient since it would otherwise look like a normal space in a listing.

With this option the output will be an entity where specified, else for encodable characters, it will be the code for the character else the character will be output as a decimal numeric character code.

Before publishing a page always select this option since it will result in the smallest file size.

- **Second option**

The option **'The above and Latin-1 letters'** Strictly should read 'The above and Latin-1 characters except ASCII characters'. That refers to characters in the Latin-1 set with codes in the range A0 to FF.

The output code for ASCII characters is the character code, for the remaining Latin-1 characters is the entity, else is a decimal numeric character reference.

- **Third option**

The option **'HTML 4 special characters'** refers to all characters for which The HTML 4 specification [Ref 16] (Section 24) provides an entity reference.

The output code for ASCII characters is the character code, for all characters for which an entity exists is the entity else it is a decimal numeric character reference.

- **Final option**

The **final option** should output decimal numeric character reference for all non-ASCII characters but has a bug so that it uses a mixture of character codes (for ASCII characters) and entities or numeric references for others.

Note The output will always produce valid and operating code for the corresponding character so files run correctly. With Nvu 1.0 and KompoZer 0.7.10, for all options, there are a few detailed non-compliances with the above description.

The options provided may be useful during development if you work in source code and may make it easier for you to read. They may also be required if transferring source code to some other applications. If, for instance, your file was encoded as UTF-8 and you wanted to transfer the content to an application which could accept only text the fourth option would be useful.

This ensures that only ASCII characters appear in the file. Unfortunately this is one option that does not work in Nvu 1.0 and KompoZer 0.7.10

Note. In source view you may enter characters using any of these systems irrespective of the option selected. After leaving and re-entering source view, the display will conform to the option selected and not to the form in which it was entered.

In Normal and Preview mode KompoZer will attempt to render all characters correctly, irrespective of the option selected, subject to the limitations described under 'Unicode support' above.

If you need an ascii only file an alternative way to obtain one is via the File > Save and change Character Encoding menu item. Check the 'Export to text' box and save the file as text. The original file remains intact. Again with UTF encoding this appears to be faulty but if the encoding is temporally switched to ISO-8859-1 it works.

A6.4.1.2 Don't encode > outside of attribute values

This is explained under the first option above.

The section on 'Preferences' suggests that this be left in the default, cleared, state but this is optional.

A6.4.2 Encoding in attribute values

One final option in Tools > Preferences > Advanced provides the option 'Don't encode special characters in Attribute values'. That is not a very precise definition of what happens because the whole page must be encoded in the same way. What it changes is the way in which character references are used in attribute values.

When you check the box special characters appearing in the value of any attribute will be left untouched and not encoded as entities or numerical character references.

A6.4.2.1 Don't encode special characters in attribute values

An example will serve to illustrate.

The **table element** has an attribute 'summary' and might appear as:
```
<table summary="Resumé of results">
```

The attribute is 'summary' and the value is 'Resumé of results'. If the characters were encoded as entities the source would read
```
<table summary="Resum&eacute; of results">
```

Even worse if the characters were encoded as numeric character references the source would read
```
<table summary="Resum&#233; of results">
```

Any readout on the screen, whether through a properties inspector or at the bottom of the browser window, will be normal but those doing lot of work in source view might prefer not to encode in this situation.

You are recommended to **check the box 'Don't encode special characters in attribute values'** unless you have good reason to do otherwise.

Where the attribute value is a URL a different encoding method is covered in the following section.

A6.5 URL encoding

A6.5.1 Percentage encoding

Table A6.5-1

Characters permissible in a URL

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																⇑
2x	sp	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{			}	~	
8x																
9x																
Ax	nbsp	!	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	shy	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ê	Ë	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

⇑ A S C I I

L A T I N 1

⇓

⇓

Special considerations apply to characters in a URL. URLs can occur as the value of an attribute to many elements the most common being the 'href'.

Any Latin 1 character may occur in a URL but only those shown against a green background in Table A6.5-1 may be used freely. This set includes alphabetics, hyphen and underscore. A number of other characters which may have specific meanings are reserved. This includes the majority of the remaining ASCII characters. Such characters may be used to separate one part of the structure from an other e.g the colon separates the protocol from the domain. These characters, from current specifications, are shown against an orange background. Whenever such a character is used other than for the specific reserved purpose it must be encoded to avoid confusion. Use of the remaining characters depends on specifics of the URL or part of the URL involved.

When encoding is required in a URL a new method referred to as 'percentage encoding' is used. Put simply, percent encoded characters consist of a percentage sign followed by two

characters representing the hexadecimal position of the character in the Latin 1 set. Thus %20 represents a space.

Full details of URL encoding are covered in [RFC 3986 [Ref 14]](#), [Wikipedia [Ref 11]](#) provides a simpler explanation.

Authors often note that the names of saved files appear with spaces replaced by %20. As explained this is quite safe and indeed some operating systems prohibit unencoded spaces in file names. It is always preferable to avoid spaces when naming files. Use the underscore as an alternative.

It is actually possible to use percentage encoding for any character in the Latin 1 set.

When KompoZer percent encodes it errs on the cautious side, that is, it may encode when this is not strictly necessary. This occurs for two reasons.

- KompoZer often cannot know the ultimate purpose of the URL so cannot necessarily know whether or not it is safe not to encode.
- The specifications of what *may*, *may not* or *optionally may be* encoded have varied from time to time and many old systems which may not comply to the latest specifications are still deployed.

Since it is permissible to percent encode any character this should not matter. Unfortunately it sometimes does. Again systems may not comply with the current specifications.

Eric Meyer has provided a URL Decoder Encoder in his [toolbox [ref 20]](#) which allows you to see the results of encoding.

A6.5.2 Don't encode special characters in attribute values

The check box 'Don't encode special characters in attribute values' controls whether or not KompoZer percent encodes special characters in URLs.

As an example KompoZer encodes the pipe character '|' because an earlier specification required this. But some systems will not decode this correctly and malfunction. On the Nvu forum it was reported that

```
<img src="/cgi-sys/Count.cgi?df=bassman7.dat|display=Counter|
ft=1|md=6|frgb=100;139;216|dd=E&reload=N">
```

was being encoded as

```
<img src="/cgi-sys/Count.cgi?df=bassman7.dat%7Cdisplay=Counter%7C
ft=1%7Cmd=6%7Cfrgb=100;139;216%7Cdd=E&amp;reload=N">
```

and that this was not being corectly recognised.

To solve this problem **check the box 'Don't encode special characters in attribute values'.**

KompoZer is not capable of selectively encoding characters in one URL and not in another. Nevertheless this limitation does not appear to be a problem to authors.

A6.6 References

Jukka Korpela

[1] Characters and Encodings http://www.cs.tut.fi/%7Ejkorpela/chars/covers almost every aspect of the subject.

Alan Wood

[2] Excellent general resource http://www.alanwood.net/unicode/index.html offers several pages including Unicode test pages, setting up of browsers, Fonts with Unicode support, Operating systems. Also specific pages on

[3] Using special characters from Windows Glyph List 4 (WGL4) in HTML http://www.alanwood.net/demos/wgl4.html

[4] Unicode fonts for Windows computers http://www.alanwood.net/unicode/fonts.html

[5] Unicode character ranges and the Unicode fonts that support them. http://www.alanwood.net/unicode/fontsbyrange.html

[6] Greek, Mathematical and Punctuation http://www.alanwood.net/demos/symbol.html

[7] Microsoft Wingdings font http://www.alanwood.net/demos/wingdings.html

Wikipedia has good articles on

[8] ASCII http://en.wikipedia.org/wiki/ASCII

[9] ISO 8859-1 http://en.wikipedia.org/wiki/ISO/IEC_8859-1 also other parts

[10] Mapping of Unicode characters to Unicode planes http://en.wikipedia.org/wiki/Mapping_of_Unicode_characters

[11] URL encoding http://en.wikipedia.org/wiki/Percent-encoding

Microsoft has

[12] Extension http://www.microsoft.com/typography/property/property.htm for the properties tab of Windows Explorer to check Unicode support.

[13] Windows 1252 http://www.microsoft.com/globaldev/reference/sbcs/1252.mspx character listing

Standards and Specifications

[14] RFC 3986 Uniform Resource Identifier Generic Syntax http://www.ietf.org/rfc/rfc3986.tx

[15] CSS 2.1 specification http://www.w3.org/TR/CSS21/

[16] HTML 4.01 specification href="http://www.w3.org/TR/html4/cover.html

[17] Unicode site http://www.unicode.org

Other resources

[18] Penn State University http://tlt.its.psu.edu/suggestions/international/web/unicode.html also has an excellent page on 'Getting Started: Unicode' which covers OS, Browsers, fonts etc

[19] W3C http://www.w3.org/International/tutorials/tutorial-char-enc/en/all.html#Slide0010 offer a somewhat more technical tutorial dealing mainly with encoding and covering material that KompoZer hides from the author

[20] Eric Meyerhttp://meyerweb.com/eric/tools/ has a toolbox which includes a URL Decoder Encoder.

[21] AllCharshttp://allchars.zwolnet.com/ has a freeware utility that allows you to enter any Windows 1282 character using a few keystrokes

Appendix 7 – Making pages Printable

A7.1 Preliminary

A7.1.1 Introduction

When web pages are printed the results are sometimes unsatisfactory. Using CSS authors can often improve this significantly though browser support for some print rules is still somewhat limited.

Although web pages are most commonly viewed on screen site visitors sometimes wish to print them to hard copy. This is particularly so with pages carrying information about an order or order confirmation, though it could occur for almost any page.

One approach is to offer a special 'printer friendly' version of a page. This is accessed via a link on the page and can be formatted to produce a good printed result. One penalty is that two versions of a page must be maintained with concomitant risk that the content of the two may differ; another is that the visitor may miss the link and use the normal method of printing so the effort is wasted.

Fortunately CSS styles provide an alternative approach. A special printer style sheet can be developed all pages on a site can be linked to it and no special editing of individual pages is required. A page may then be printed using menu instructions such as File > Print in a browser.

Those unfamiliar with CSS should return to [Section 4](#) of the User Guide and learn how to use Styles in KompoZer before proceeding further.

A7.1.2 Printer style sheets

Printer style sheets can be either internal style sheets or external (linked) sheets. What is needed is some means of identifying such sheets with the purpose for which they are intended. This is done using one of two methods.

- For complete style sheets specify the 'media' attribute for the link or style element.
- For individual style rules or groups of rules use the @media rule.

A7.1.2.1 Style sheets – media="print"

The normal form of a link to an external style sheet, in the header of a file, is `<link rel="stylesheet" href="stylefile.css" type="text/css">`.

The normal form for the opening tag for an internal style sheet is `<style type="text/css">`

With KompoZer, when you open the CSS editor and click a sheet, in the general tab the 'Media list' shows 'all'. This indicates that the styles in the file apply to all media.

Normal style sheets can be changed to 'print' style sheets very easily. Proceed as follows

1. Open the CSS editor
2. In the Sheets and rules window click on the stylesheet. The general tab will be opened.
3. In the Media list box the word 'all' will appear. Overtype it with the word 'print'.
4. Close the CSS editor. (If the OK button is inaccessible click any stylesheet then any rule and then click OK.)

If you now inspect the code it will have been changed to

For an external stylesheet `<link media="print" rel="stylesheet" href="stylefile.css" type="text/css">`

For an internal stylesheet `<style media="print" type="text/css">`

Note. When using KompoZer, if you re-open the CSS editor the entry in the 'Media list' box will not have changed. This is a bug. To correct this save the file and revert.

This shows that the sheet applies only to print media. In the absence of a media attribute a sheet applies to all media.

Note Although the CSS editor may show the media list as 'all' and this is in fact a recognised media type the stylesheet itself does not contain this information. However it will default to applying to all media types.

A7.1.2.2 Individual style rules – @media

Sheets that apply to all media may have individual rules tagged to apply only to print media. To do this rules must be enclosed in the @media rule.

For example the body could be styled to remove the margin and background and specify the font size in printer units

```
@media print {
body { margin: 0; background-image: none; font-size: 12pt; }
}
```

This shows a single rule being changed but rules may be grouped.

```
@media print {
p { font-size: 10pt; }
h1 { font-size: 14pt; }
h2 { font-size: 12pt; }
}
```

KompoZer does not offer any way of implementing this method automatically so if you want to use the @media rule you must edit the style sheet to add the @media by hand. Once created the CSS editor I can be used to edit, add or delete rules.

A7.1.3 Media types

This appendix is about 'Print' type media. The CSS specification includes the following types: Braille, Embossed, Handheld, Print, Projection, Screen, Speech, tty and TV. At present only Screen and, to a lesser extent, print are widely supported.

An earlier version of the specification had an 'Aural' type but this has been replaced by 'Speech'.

A7.2 Styles for printing

So what changes might you want to make to make a page print better?

A7.2.1 Dimensions

The first thing to consider is probably whether the dimensions and units of measurement in the style sheet are appropriate for printing.

The CSS Specification provides specific methods for specifying the characteristics of the printed page which includes size and margin and allows different margins for first, left- and right-hand side pages and even allows classes to be applied to different parts of a document to allow these to be changed. The method is based on the @page rule. Unfortunately support for this rule is almost non-existent with current browsers so this guide recommends alternative, albeit not as flexible, methods.

A7.2.1.1 Margins

Printed pages are usually presented with much wider margins than pages viewed on screen. The body element of a web page frequently has margins of only about 10px. How wide this appears on a printed page depends on the resolution of a printer and, with a high resolution printer, may be little more than a gnats whisker. For a printed page this should be quite satisfactory as will be explained.

Many printed pages use generous margins of up to 3cm. The minimum margin should probably not be less than 1.2 cm since some printers may require this. Visitors needing to file printed pages and punch them to do so may prefer a left margin of about 2cm.

All browsers apply some default margin to a printed page and the main browsers allow this to be altered. Any margin specified within a web page will be applied in addition to the margin set in the browser. Web authors should therefore not apply large margins to pages for printing but leave it to the visitor to set the browser. Where the standard style sheet

applies a margin of 10 or 15px this can probably be ignored but where it is greater the print style sheet should set this to zero.

(The competition between the two sets of margins may well account for the fact that browsers avoid supporting the @page rule.)

If the style of the page is to have a wide space on each side with centralised content, it is quite possible to specify the width, which should be in inches or centimetres, and set left and right margins to auto.

A7.2.1.2 Fonts

After margins the most important thing to consider is fonts. Although the CSS rules allow font size for printed media to be specified in any of the normal ways the unit almost universally used for print media is the point and there appears to be no good reason to use any other.

A possible print style sheet might look like

```
@media print {
p, h4, ul, ol, td, a { font-size: 12pt; }
h1 { font-size: 24pt; }
h2 { font-size: 18pt; }
h3 { font-size: 14pt; }
h5 { font-size: 10pt; }
}
```

Font colour also needs to be considered. Many printers print in monochrome. Rather than hope that any colours used will be interpreted suitably it is better to take the bull by the horns and change `color` to black.

A7.2.1.3 Other dimensions

Where items are positioned, indented or given sizes on a web page this is frequently expressed in pixels. The specification does not include any rules for mapping these to paper dimensions, so those wanting to be really fussy may wish to substitute inches or centimetres. In practice the effort is rarely justifiable as browsers make reasonable assumptions. Dimensions expressed as percentages work as normal.

A7.2.2 Selections

The second issue is exactly what should be printed — is there material on the page that does not need printing?

The issues are twofold.

- Links on the page will be inactive when printed. Internal links may usefully be omitted but external links which are not explicit may need to become so.

- Printer ink and print time can be a precious resource. Minimising use may be helpful to the user.

A7.2.2.1 Menus

On a page like this, to print the navigation menu is a waste of time. Items like this can be prevented from printing by using the style rule `display: none;` available on CSS editor Box tab. This rule can then be attached to any item which is not to be printed by using a class like 'noprint'. A rule to do this could be:

```
@media print {
.noprint { display: none; }
}
```

Simply tag any item not to be printed with the class 'noprint'. This will have no effect on screen since it is defined only for print media. Being able to tag items with several classes is convenient since adding the noprint class has no effect on any other class attached to an item.

The menu of this page is enclosed in its own division so the treatment described is easy. Depending on the structure of the document it may be helpful to enclose several items in a div rather than tag each individually to suppress printing.

A7.2.2.2 External links

Many authors don't show the URL of external links on screen. This makes the page tidier and links still work when clicked. When printed such links become unrecognisable and useless. An alternative is to repeat the URL after each link but to style it as `display: none;` in the general style sheet and `display: inline;` in the print style sheet. Just take care that it is surrounded by spaces when printed, otherwise it will look ugly.

As an example here is how I coded the first reference in section A7.4 below.

```
<p><a href="http://www.w3.org/TR/CSS21/">CSS2 Specification </a>
<span class="extlink">http://www.w3.org/TR/CSS21/</span> introduces
media in section 7 and covers paged media in section 13</p>
```

The words 'CSS2 Specification' form a clickable link. The URL of the link is then repeated but enclosed in a <span> which has the class 'extlink'.

The page is linked to two stylesheets. The sheet applicable to all media is loaded first and includes the rule

```
.extlink { display: none; }
```

The print stylesheet is loaded later and includes the rule

```
.extlink { display: inline; text-decoration: underline; }
```

Underlining the links makes them stand out and you may like to consider printing them in blue.

The order of loading the sheets is important because, if the print stylesheet were loaded first and the unspecific ('all') sheet loaded subsequently this would override the print sheet. See section 7.3.1

A7.2.2.3 Images

Images are the raison d'être of some pages so must be printed, in others they vary from major irritants to incidental material which may be omitted to save time and ink. The same approach can be used as for menus or, if all images are to be suppressed, the img element may be given the style `display: none;`

A7.2.2.4 Backgrounds

Some browsers will not print backgrounds at all. Others offer options, but this cannot be relied on. This applies both to background images and solid colour. On a page like this, if the menu is not suppressed it is likely to print unreliably. The background colour, if printed, may bleed into the white text, particularly with low quality paper, and become unreadable. In such cases it is much better to set up a new set of styles for the buttons using colour schemes which are workable.

A7.2.3 Page media

Having decided what to print and where to print it, the next thing to consider is page breaks. This is a concept completely alien to screen devices. Of course the page author cannot know the size of the sheet that the page will be printed on, so manual page breaks are out of the question and we must resort to using automatic page breaks. Anyone who has used a word processor or desk top publisher will know that this is a 'dodgy' process. No less with web pages, but the CSS Specification does provide some assistance.

The CSS editor offers a tab for developing style sheets for 'aural' media (this type is now deprecated) but not print media. As a result what follows can be implemented only by editing the code by hand.

The CSS2 Specification covers paged media in section 13. As already stated, the @page method is largely unsupported as yet, but two properties which may improve printed layout are page-break-before and page-break after. These may be set to a value of 'always, 'avoid' or 'auto'.

Typically authors may force a page break before a major heading, for instance:

```
h1 {page-break-before: always}
```

Similarly other headings can be kept with their associated content by:

```
        h3 {page-break-after: avoid}
```

The 'page-break-after' property is less well supported, at least with the value 'avoid'.

The specification also includes the property page-break-inside. This is intended to keep together associated content such as a table or div. Potentially this is an extremely useful property and overcomes one of the most annoying and spoiling features of printed pages but as of 2007 appears to be unsupported.

To keep a block item together use a class like:

```
        .keeptogether {page-break-inside: avoid}
```

A7.3 Getting results

A7.3.1 The right order

Having set up printer style sheets as described, all should be well but there are a few precautions to take.

Normally there will be two sets of styles, one for printed media, one for all media. If the two sets were for printed media and screen media no ambiguity could result but there are good reasons to avoid this approach:

- Visitors using other media would be completely disenfranchised.
- Both style sheets would have to be comprehensive, this would lead to more development and more maintenance.

Instead of this, it is normal to use a style sheet covering all media and a separate one for print media which includes only a few styles – those few that need to be different for print.

The specifications that control the cascading order for styles dictate that the later encountered style rules override those encountered earlier. Thus if the rules for print media are read first, followed by those for all media, the latter will prevail and the print media rules will be ignored. **It is therefore important that the link to the print style sheet should appear after that to the normal style sheet.**

The CSS editor enters style sheets in the head section in the same order as shown in the 'Sheets and rules' window.

To alter the order of linked, or internal, style sheets:

- Open the CSS editor
- In the sheets and rules window click a style sheet
- Click the 'Up' or 'Down' button to change its place in the listing.

A7.3.2 Inline styles

As a general rule, styles for print media cannot be applied using inline styles. Possible exceptions are the page-break properties. These apply only to paged media and are therefore meaningless and will be ignored by screen and most other devices, however both print and projection are paged media so may require different breaks.

The cascading rules dictate that inline styles have precedence over all others. Authors will probably use inline styles sparingly and, other than the above, should certainly avoid specifying any styles which do not apply to all media. **Inline style declarations will override any print styles so any which could give problems must be avoided**.

A7.3.3 Browsers

Pages and style sheets designed as suggested will be rendered satisfactorily, though not perfectly, on most browsers. At the time of writing (mid 2007) all of the major browsers have difficulty with some aspect of print styles. Authors should not hope to be really prescriptive about how the final output appears. Web Devout has a useful page detailing Web browser CSS support. Coverage will undoubtedly improve with time.

A7.3.4 Development aid

Readers will know that KompoZer masquerades as a passable browser. This can be pressed into use when developing print style sheets. For testing purposes, once the style sheets have been linked in the correct order it is simple to relink the print style sheet as an 'all' style sheet (reversing the process described in A7.1.2.1). After closing and reloading the file the Normal view or Preview mode then become print view modes. The style sheet can then be refined while immediately seeing the result. **Don't forget to change back to 'print' before finalising!**

A7.4 References

[1] CSS2 Specification http://www.w3.org/TR/CSS21/ introduces media in section 7 and covers paged media in section 13

[2] Web Devout http://www.webdevout.net/browser-support-css has a useful page detailing CSS support for several modern browsers.

[3] WestCiv http://www.westciv.com/style_master/academy/browser_support/index.html has summary information about a wide range including older browsers .

[4] A number of sites cover various aspects of printing though none are at all comprehensive.

[5] A List Apart http://www.alistapart.com/articles/alaprintstyles has an excellent article by Eric Meyer on choosing styles for printing. Probably the best available.

[6] About http://webdesign.about.com/cs/css/a/aa042103a.htm gives a general introduction by Jennifer Kyrnyn to considerations for printed pages.

[7] Opera offers some useful general information.
http://my.opera.com/community/dev/device/css-media/

[8] WestCiv offers more specific information about @page and page properties.
http://www.westciv.com/style_master/academy/css_tutorial/advanced/printing.html

[9] Rich in style has a fairly informative page about paged media.
href="http://www.richinstyle.com/guides/fontface2.html This will become more useful
when @page is better supported.

Copyright

Postscript

This guide started as notes for my personal use. I revealed its existence to the Nvu forums and after receiving favourable and encouraging remarks decided to press on.

I express my thanks to those, including several from the Nvu and WysiFree forums, who have made a number of useful suggestions which I have incorporated wherever possible. Particular thanks however go to Gérard Talbot http://www.gtalbot.org/ who's knowledge of the web, html and CSS greatly exceeds mine and who has offered a number of corrections and made many suggestions which I believe have significantly improved the result. My thanks also go to Jukka "Yucca" Korpela http://www.cs.tut.fi/%7Ejkorpela/ who checked and provided several corrections to the appendix on Encoding. Mistakes and omissions, of course, remain mine.

Gérard must also be thanked for undertaking the translation into French as must Tsutomu Ohmori for translating into Japanese.

Thanks are also due to Daniel Glazman at http://www.disruptive-innovations.com/ Disruptive Innovations who developed Nvu from the earlier Netscape Composer and to http://fabiwan.kenobi.free.fr/ Fabien Cazenave (Kazé) who developed it into KompoZer. Thanks also to http://www.linspire.com Linspire who have provided hosting.

User Guide - Quick Index

This short subject index links either to the main treatment of the subject or to an important sub-topic which then leads to the main topic.

See also the site map for a full contents list.

AllChars

Anchors

Background image (body)

Background Image (table)

Block outline

Body

Border

Box styles

Bring to front

Character set (UCS)

Classes

Colour

CSS (Cascading Style Sheets)

Dictionaries

div element

Doctype

Email address

Encoding characters

Entities

Export stylesheet

Extensions

Firewall

Font see also Unicode

Forms

Frames

Glue to edge

Head

id

Images

Inheritance

Installing KompoZer

ISO 8859

KompoZer

Latin 1

Layer

Links

Lists

Made with KompoZer

Margin

Markup Cleaner

Nvu

@ Media

Padding

Percent encoding

Positioning grid

Profiles

Pseudo-classes

Publishing

Quotes

Selectors (style)

Selecting Items

Special characters

Spelling checking

Style

Site manager

Tables

Table of Contents

Templates

Unicode

Updating KompoZer

URL Encoding

Views

Validate

XHTML

z-index.