

HTML TIDY - Release Notes

[Dave Raggett dsr@w3.org](mailto:dsr@w3.org)

Public Email List for Tidy: <html-tidy@w3.org>

I have set up an archived mailing list devoted to Tidy. To subscribe send an email to html-tidy-request@w3.org with the word subscribe in the subject line (include the word unsubscribe if you want to unsubscribe). The [archive](#) for this list is accessible online. Please use this list to report errors or enhancement requests.

Things awaiting further attention

These have been moved to the [pending page](#), which includes all the suggestions for improvements and bug fixes. I am looking for volunteers to help with these as my current workload means that I don't get much time left to work on HTML Tidy.

August 2000

Ann Navarro comments that the "appears to" message is confusing when it differs from the doctype declaration. Perhaps it would make sense to also report the doctype? Tidy will now report the FPI when present, and then the apparent version as deduced from the elements and attributes present in the rest of the document.

John Russell sent in an example which featured a script element in a frameset document where the script element appears after the head and before the frameset. This is I believe illegal, but Tidy proceeds to do the dumb thing discarding the frameset element! I think it should move the script element into the head and continue. This is now implemented.

Jacques Steyn says that Tidy doesn't know about the HTML4 char attribute for col elements. Now fixed.

Carlos Piqueres Ayela would like Tidy to detect all cases of repeated attributes, e.g. repeated valign in table cells. This was introduced a few releases back, but I forgot to apply this check for the elements with special purpose attribute checking methods. Now fixed. Tidy will issue a warning for each repeated attribute. In principle Tidy could merge repeated class attributes, but this will require more work. My apologies to Carole Mah for not having the time to do this now.

Henry Zrepa would like an option to suppress whitespace munging on selected attributes used for legacy scripts passed as parameters to plugins. I have added a new boolean option "literal-attributes" which can be set to yes to preserve whitespace within attribute values. A better solution would be to make this selectable on a per element basis, but I don't have time to explore this now.

Edward Zalta spotted that Tidy always removed newlines immediately after start tags even for empty elements such as img. An exception to this rule is the br element. Now fixed.

July 2000

Edward Zalta sent me an example, where Tidy was inadvertently wrapping lines after an image element. The problem was a conditional in pprint.c, now fixed.

Andy Quick offered a bug fix for the AddClass() function in clean.c. My thanks to Terry Teague for bringing this to my attention. Davor Golek reported a problem with the -f option. I discovered a bug in line 898 in tidy.c, now fixed.

June 2000

Fixed bug in NormalizeSpaces (== in place of =) on line 1699.

I have added a new config option "gnu-emacs" following a suggestion by David Biesack. The option changes the way errors and warnings are reported to make them easier for Emacs to parse.

Tony Leneis noticed that Tidy didn't know that width and height attributes on the img element aren't allowed in HTML 2.0. He also noted that Tidy didn't know that HTML 2.0 allows img as a direct child of body. Both of these bugs are now fixed.

I have refined CanPrune() to block pruning empty elements with if they have id or name attributes. Previously any attribute would prevent an empty element from being pruned. The rationale is that such empty elements are placed there to be filled dynamically by a script. This is unlikely to occur unless the element can be referenced via id or name.

Denis Barbier sent in details patches that suppresses numerous warnings when compiling tidy, especially:

- 'static' declaration of subroutines when possible
- initialization of variables when it might be used before assignment
- change name of local variables when it overrides global ones (count, index, fp)
- suppression of long jump, buffers are closed in FatalError

Fixed memory leak in CoerceNode. My thanks to Daniel Persson for spotting this. Tapio Markula asked if Tidy could give improved detection of spurious </ in script elements. Now done.

My thanks to John Russell who pointed out that Tidy wasn't complaining about src attributes on hr elements. My thanks to Johann-Christian Hanke who spotted that Tidy didn't know about the Netscape wrap attribute for the text area element.

Sebastian Lange has contributed a perl wrapper for calling Tidy from your perl scripts, see sl-tidy.pl.

Stephen Reynolds would like comments that end with a line break to retain this property when tidied. I have added a new boolean property to the node structure which is set by the end comment parser in lexer.c and acted on by the comment formatting code in pprint.c

Henry Zrepa (sp?) reported that XHTML <param> elements were being discarded. This was due to an error in ParseBlock, now fixed.

Carole E. Mah noted that Tidy doesn't complain if there are two or more title elements. Tidy will now complain if there are more than one title element or more than one base element.

May 2000

Following a suggestion by Julian Reschke, I have added an option to add xml:space="preserve" to elements such as pre, style and script when generating XML. This is needed if these elements are to be correctly parsed without access to the DTD.

April 2000

Randy Wacki notes that IsValidAttribute() wasn't checking that the first character in an attribute name is a letter. Now fixed.

Jelks Cabaniss wants the naked li style hack made into an option or at least tweaked to work in IE and Opera as well as Navigator. Sadly, even Navigator 6 preview 1 replicates the buggy CSS support for lists found in Navigator 4. Neither Navigator 6 nor IE5 (win32) supports the CSS marker-offset property, and so far I have been unable to find a safe way to replicate the visual rendering of naked li elements (ones without an enclosing ul or ol element). As a result I have opted for the safer approach of adding a class value to the generated ul element (class="noindent") to keep track of which li's weren't properly enclosed.

Rick Parsons would like to be able to use quote marks around file names which include spaces, when specifying files in the config file. Currently, this only effects the "error-file" option. I have changed that to use ParseString. You can specify error files with spaces in their names.

Karen Schlesinger would like tidy to avoid pruning empty span elements when these have id attributes, e.g. for use in setting the content later via the DOM. Done.

I have modified GetToken() to switch mode from IgnoreWhitespace to MixedContent when encountering non-white textual content. This solves a problem noticed by Murray Longmore, where Tidy was swallowing white space before an end tag, when the text is the first child of the body element.

Tidy needs to check for text as direct child of blockquote etc. which isn't allowed in HTML 4 strict. This could be implemented as a special check which or's in transitional into the version vector when appropriate.

ParseBlock now recognizes that text isn't allowed directly in the block content model for HTML strict. Furthermore, following a suggestion by Berend de Boer, a new option `enclose-block-text` has the same effect as `enclose-text` but also applies to any block element that allows mixed content for HTML transitional but not HTML strict.

Jany Quintard noted that Tidy didn't realise the width and height attribute aren't allowed on table cells in HTML strict (it's fine on HTML transitional). This is now fixed. Nigel Wadsworth wanted border on table without a value to be mapped into `border="1"`. Tidy already does this but only if the output is XHTML.

Jelks Cabaniss wanted Tidy to check that a link to an external style sheet includes a type attribute. This is now done. He also suggested extending the clean operation to migrate presentation attributes on body to style rules. Done.

March 2000

I have been working on improving the Word2000 cleanup, but have yet to figure out foolproof rules of thumb for recognizing when paragraphs should be included as part of ul or ol lists. Tidy recognizes the class "MsoListBullet" which Word seems to derive from the Word style named "List Bullet". I have yet to deal with nested lists in Word2000. This is something I was able to deal with for html exported from Word97, but it looks like being significantly harder to deal with for Word2000.

Tidy is now able to create a pre element for paragraphs with the style "Code". So try to use this style in your Word documents for preformatted text. Tidy strips out the p tags and coerces non-breaking spaces to regular spaces when assembling the pre element's content.

I would very much welcome any suggestions on how to make the Word2000 clean up work better!

Changed `Style2Rule()` in `clean.c` to check for an existing class attribute, and to append the new class after a space. Previously you got two class attributes which is an error

Changed default for `add-xml-pi` to no since this was causing serious problems for several browsers.

Joakim Holm notes that tidy crashes on ASP when used for attributes. The problem turned out to be caused by `CheckUniqueAttribute()` which was being inappropriately applied to ASP nodes.

John Bigby noted that Tidy didn't know about Microsoft's data binding feature. I have added the corresponding attributes to the table in `attr.c` and tweaked `CanPrune()` so that empty elements aren't deleted if they have attributes.

Tidy is now more sophisticated about how it treats nested `<b>`'s etc. It will prune redundant tags as needed. One difficulty is in knowing whether a start tag is a typo and should have been an end-tag or whether it starts a nested element. I can't think of a hard and fast rule for this. Tidy will coerce a `<b>` to `</b>` except when it is directly after a preceding `<b>`.

Bertilo Wennergren noted that Tidy lost `<frame/>` elements. This has now been fixed with a patch to `ParseFrameSet`.

February 2000

Dave Bryan spotted an error in `pprint.c` which allowed some attributes to be wrapped even when `wrap-attribute` was set to no. On a separate point, I have now added a check to issue a warning if `SYSTEM`, `PUBLIC`, `//W3C`, `//DTD` or `//EN` are not in upper case.

Tidy now realises that inline content and text is not allowed as a direct child of body in HTML strict.

Dave Bryan also noticed that Tidy was preferring HTML 4.0 to 4.01 when doctype is set to strict or transitional, since the entries for 4.0 appeared earlier than those for 4.01 in the table named `W3C_Version` in `lexer.c`. I have reversed the order of the entries to correct this. Dave also spotted that `ParseString()` in `config.c` is erroneously calling `NextProperty()` even though it has already reached the end of the line.

January 2000

I have added a new function `ApparentVersion()` which takes the doctype into account as well as other clues. This is now used to report the apparent version of the html in use.

Thanks to the encouragement of Denis Barbier, I finally got around to deal with the extra bracketing needed to quiet gcc -Wall. This involved the initialization of the tag, attribute and entity tables, and miscellaneous side-effecting while and for loops.

`PPrintXMLTree` has been updated so that it only inserts line breaks after start tags and before end tags for elements without mixed content. This brings Tidy into line with current wisdom for XML editors. My thanks to Eric Thorbjornsen for suggesting a fix to `FindTag` that ensures that Tidy doesn't mistreat elements looking like html.

`<table border>` is now converted to `<table border="1">` when converting to XHTML.

I have added support for CDATA marked sections which are passed through without change, e.g.

```
<![CDATA[ .. markup here has no effect .. ]]>
```

A number of people were interested in Tidied documents be marked as such using a meta element. Tidy will now add the following to the head if not already present:

```
<meta name="generator" content="HTML Tidy, see www.w3.org">
```

If you don't want this added, set the option `tidy-mark` to no.

In the January 12th release, `ParseXMLElement` screwed up on doctypes and toplevel comments, causing a memory exception. This has now been fixed. `PPrintXMLTree` now uses zero indent for comments to avoid progressive indentation as an XML document is repeatedly tidied. I have added a blank line after elements unless they are the last in the parent's content.

Johnny Lee reports that Tidy didn't realise that HTML4 allows the object element in the document head. Now fixed. Rainer Gutsche noticed that Tidy wasn't moving an initial space after an anchor start tag to just before the element. I have streamlined the trimming of spaces.

Johannes Zellner spotted that newly declared preformatted tags weren't being treated as such for XML documents. Now fixed.

December 1999

Tidy now generates the XHTML namespace and system identifier as specified by the current [XHTML Proposed Recommendation](#). In addition it now assumes the latest version of HTML4 - HTML 4.01. This fixes an omission in 4.0 by adding the name attribute to the `img` and `form` elements. This means that documents with rollovers and smart forms will now validate!

James Pickering noticed that Tidy was missing off the `xhtml-` prefix for the XHTML DTD file names in the system identifier on the doctype. This was a recent change to XHTML. I have fixed `lexer.c` to deal with this.

This release adds support for [JSTE](#) psuedo elements looking like: `<# #>`. Note that Tidy can't distinguish between ASP and JSTE for psuedo elements looking like: `<% %>`. Line wrapping of this syntax is inhibited by setting either the `wrap-asp` or `wrap-jste` options to no.

Thanks to Jacek Niedziela, The Win32 executable for tidy is now able to example wild cards in filenames. This utilizes the `setargv` library supplied with VC++.

Jonathan Adair asked for the hashtables to be cleared when emptied to avoid problems when running Tidy a second time, when Tidy is embedded in other code. I have applied this to `FreeEntities()`, `FreeAttrTable()`, `FreeConfig()`, and `FreeTags()`.

Ian Davey spotted that Tidy wasn't deleting inline emphasis elements when these only contained whitespace (other than non-breaking spaces). This was due to an oversight in the `CanPrune()` function, now fixed.

Michel Lemay spotted some bugs in if statements and provided some sample html files that caused Tidy to

crash. On further study, I found a bug in the code that moves font elements inside anchors. I have fixed this and added a new method to test the tree for internal consistency in its bidirectional links: `CheckNodeIntegrity()`.

I have also refined the code for handling `noframes` to make it more robust. It will now handle `noframes` within a body within a `noframes` etc. (something permitted by HTML4). It will also recover if the `noframes` end tag is missing or is in the wrong place.

I have fleshed out the table for mapping characters in the Windows Western character set into Unicode, see `Win2Unicode[]`. Yahoo was, for example, using the Windows Western character for bullet, which is in Unicode `U+2022`.

David Halliday noticed that applets without any content between the start and end tags were being pruned by Tidy. This is a bug and has now been fixed.

I have changed the way Tidy handles empty paragraphs when the `drop-empty-paras` is set to `no`. HTML4 doesn't allow empty paragraphs so I am now replacing them by a pair of `br` elements, so that the formatting is preserved. When `drop-empty-paras` is set to `yes`, empty paragraphs are simply removed.

Darren Forcier asked for a way to suppress fixing up of comments when these include adjacent hyphens since this was screwing up Cold Fusion's special comment syntax. The new option is called: *fix-bad-comments* and defaults to `yes`.

Using Michel's examples I have improved the way the table parser deals with unexpected content. This is now consistently moved before the table, or to the head element as appropriate. Microsoft and Netscape differ in how an unclosed blockquote renders when found at the table or `tr` level. Netscape indents the table but Microsoft does not. This is getting too tricky for me to deal with!

Using a sample page from Yahoo, I discovered that Netscape Navigator doesn't implement the `text-align` style property on `tr` or table elements. As a result I have added a special check for this in `BlockStyle()` to avoid translating the `align` attribute on `tr` or table into a style rule.

Richard Allsebrook would like to be able to map `b/i` to `strong/em` without the full clean process being invoked. I have therefore decoupled these two options. Note that setting logical-emphasis is also decoupled from `drop-font-tags`.

30th November 1999

This is an interim release to provide a bug fix for a bug introduced earlier in the month. I have fixed a bug in the emphasis code which looks for start tags which are most likely intended as end tags. This bug only appeared in the November release and could cause a crash or indefinite looping. My thanks to a respondent calling himself "Michael" who provided a collection of files that allowed me to track this down.

I have also added page transition effects for the slide maker feature. The effects are currently only visible on IE and above, and take advantage of the meta element. I will provide an option to select between a range of transition effects in the next release.

November 1999

David Duffy found a case causing Tidy to loop indefinitely. The problem occurred when a blocklevel element is found within a list item that isn't enclosed in a `ul` or `ol` element. I have added a check to `ParseList` to prevent this.

Takuya Asada tells me that in Raw mode Tidy is incorrectly mapping `0xA0` to the entity causing problems for `Shift_JIS` etc. Now fixed. Larry Virden reported a problem with `ParseConfig` when one of the arguments was null. I have added a check for this.

Thomas McGuigan notes that Tidy issues a warning for `noframes` elements without a body element. HTML4 is defined so that the content of the `noframes` element is restricted to a single body element. However, it also allows you to omit the start and end tags for body, something that isn't allowed for XHTML. I have changed the code to only issue the warning when generating XML.

Added new `--version` or `-v` option that reports the release date to the error stream. `ParseConfig()` now returns

false if it doesn't use the parameter. This avoids the next argument on the command line from being swallowed inadvertently, e.g. for unknown options. Tidy now warns about unrecognized options.

I have revised the way Tidy deals with comments to avoid problems with repeated hyphens. First "--" is illegal in XML, and second, the comment syntax for SGML is very error prone when it comes to when and where you can use hyphens. As a result, Tidy will now replace repeated hyphens with "=" characters. My thanks to Yudong Yang and Randy Waki for their input on this.

Emphasis start tags will now be coerced to end tags when the corresponding element is already open. For instance <u>...<u>. This behavior doesn't apply to font tags or start tags with attributes. My thanks to Luis M. Cruz for suggesting this idea.

Jonathan Adair would like Tidy to warn when the same attribute appears more than once in the same element. This is an error for both SGML and XML. The best way to make this check would be to sort the attributes and look for duplicate entries. Other people have asked for the attributes to be sorted, but I need further input on the appropriate sort order. As an interim solution, Tidy uses a simple test which generates n+1 warnings if an attribute is repeated n times.

October 1999

On Unix systems you can get Tidy to look for a config file in ~/.tidyrc or ~/your/.tidyrc etc. when the HTML_TIDY environment variable isn't set. To enable this feature don't forget to uncomment SUPPORT_GETPWNAM in the platform.h file. This feature won't work on Windows. My thanks to Todd Lewis who contributed the code.

Darren Forcier reports that Cold Fusion uses the following syntax:

```
<CFIF True IS True>
  This should always be output
<CFELSE>
  This will never output
</CFIF>
```

After declaring the CFIF tag in the config file, Tidy was screwing up the Cold Fusion expression syntax, mapping 'True' to 'True=""' etc. My fix was to leave such pseudo attributes untouched if they occur on user defined elements.

Jelks Cabaniss noticed that Tidy wasn't adding an id attribute to the map element when converting to XHTML. I have added routines to do this for both 'a' and 'map'. The value of the id attribute is taken from the name attribute.

Larry Cousin noted that Tidy is now screwing up on option elements. This proved to be a recently introduced error, which I have now fixed. Peter Ruevski forwarded an example that caused Tidy to loop endlessly. The problem was caused by an ol start tag followed by a b start tag and then an li element. I have solved the problem with a fix to ParseBlock.

I have revised the way Tidy deals with unexpected content in lists. Tidy now wraps such content in list items with the style attribute set to "list-style: none" to suppress list bullets. If an li element is found unexpectedly in the body or block-level content, it is wrapped into a ul element with the style attribute set to "margin-left: -2em". This provides a closer match to the observed rendering on current browsers. I use a couple of postprocessing steps (List2BQ and BQ2Div) to further clean this up to use div elements. My thanks to Thomas Ribbrock for sending me a challenging example that led me to this solution.

A number of people have asked for a config option to set the alt attribute for images when missing. The alt-text property can now be used for this purpose. Please note that YOU are responsible for making your documents accessible to people who can't view the images!

Terry Teague spotted a bug in ParseConfigFile() that prevented Tidy from parsing more than one file. This has been fixed by setting the char buffer to zero in the call to InitConfig() before parsing. Terry also noted a few places where I had slipped back into using malloc and free rather than MemAlloc and MemFree, now fixed.

Bjoern Hoehrmann notes that the September 27th release mapped empty paragraphs to br elements, which introduces extra whitespace in IE and Navigator. The former behavior to strip empty paragraphs is as per HTML4 and works fine on most browsers with the exception of Lynx. I have reverted to stripping empty P's, but have added an option to leave them alone.

Bjoern also drew my attention to a bug in the September release where table content is lacking a preceding td or th start tag. Tidy moves such content to before the table element to match the observed rendering. This is now working as planned. I have tweaked the printing behavior when the omit end tags option is set. It now omit the </html> as well as the optional start tags for html, head and body.

Pao-Hsi Huang had problems with the contents of the option element being discarded. I was unable to reproduce this problem, but did notice that I unintentionally preserving newlines within option text. This is now fixed. Shane Harrelson spotted that table cells containing a single font element, when cleaned dropped the for element without getting the corresponding style. Now fixed via a tweak to InlineStyle().

Andre Hinrichs wanted Tidy to do a better job on font elements with relative size changes. This is in fact rather tricky. Currently, Tidy uses percentage scaling values for fonts rather than the enumeration defined by CSS [xx-small | x-small | small | medium | large | x-large | xx-large]. The first problem is to match these 7 values onto the 6 define by the font element. The next problem is caused by the fact that CSS doesn't provide matching relative font size values that you could match to the ones defined for the font element. I have done my best using percentage values, base on tests with IE and Navigator. If anyone can come up with a better approach, please let me know.

Tom Berger reported a problem when quote-marks was set to yes. Using his test file everything is now working fine. Several people asked for a way to turn off line wrapping. Tidy will now interpret zero as meaning disable wrapping. Johannes Zellner wants to include some tcl code in his XML markup and asks for a way define new tags that behave in the same way as HTML's pre element. The new option is new-pre-tags.

September 1999

Tidy will now add a type attribute to the style and script attributes when this is missing. Tidy examines the language attribute to determine what media type to use. I have also added code to create an id attribute for anchors when a name attribute is present, and to report a warning if id and name don't match.

Added support for cleaning up HTML generated by Microsoft Word 2000 when you save as "Web Page". When you set "word-2000: yes" Tidy makes a Herculean effort to clean up the mess created when Word 2000 exports to HTML. Word bulks out HTML with presentation information that allows it to round-trip documents between HTML and Word without lost of information. This makes the HTML hard to edit and can cause some very popular browsers to crash! I haven't dealt with the VML markup Word uses for line drawings.

Applied fix to InsertNodeAfterElement() to set node->next->prev. My thanks to "Advocate" for this. This was only encountered when dealing with PRE tags containing content illegal for PRE. (Called twice by ParsePre to move illegal PRE content to be a later sibling of PRE, then open PRE again afterward)

Change to table row parser so that when Tidy comes across an empty row, it inserts an empty cell rather than deleting it. This is consistent with browser behavior and avoids problems with cells that span rows.

Baruch Even sent extensive patches for improved support for the PHP preprocessing psuedo tags. You can now use the 'wrap-php: no' to suppress line wrapping within PHP instructions. In the process of this work, I have created a new function InsertMisc() for dealing with comments, processing instructions, ASP and PHP.

I have update the table of tags to include additional proprietary tags such as server, ilayer, layer, nolayer and multicol. Using patches sent in by Edward Avis, Tidy now offers a quiet mode which suppresses the initial welcome message and the summary report on the number of errors or warnings. Jason Tribbeck sent in patches to allow config options normally set in the config file to be set on the command line, by preceding then with a "--" (no intervening space), for example:

```
tidy --break-before-br true --show-warnings false
```

Kenichi Numata discovered that Tidy looped indefinitely for examples similar to the following:

```
<font size=+2>Title
<ol>
</font>Text
</ol>
```

I have now cured this problem which used to occur when a tag was placed at the beginning of a list element. If the example included a list item before the Tidy will now create the following markup:

```
<font size=+2>Title</font>
<blockquote>Text </blockquote>
```

```
<ol>
<li>list item</li>
</ol>
```

This uses blockquote to indent the text without the bullet/number and switches back to the ol list for the first true list item.

I have worked hard to improve support for server side preprocessing instructions such as ASP, PHP and Tango. Tidy now allows you to replace attribute values by such instructions and is able to fix up the case where the instruction appears without delimiting quote marks. Tidy supports ASP and PHP in element content and also in place of attribute value pairs. Support for Tango is limited to attribute values only.

John Love-Jensen contribute a table for mapping the MacRoman character set into Unicode. I have added a new charset option "mac" to support this. Note the translation is one way and doesn't convert back to the Mac codes on output.

Some people place <p> at the end of their list items to introduce whitespace before the next item. I have modified TrimEmptyElement to coerce empty p elements to br elements to reproduce this rendering. If a p start tag is found in dt elements, I now coerce the p to a br. Satwinder Mangat has alerted me to several such problems. First, text as a direct child of dl should be wrapped in a dt and not a dd element. Second, unlike other inline tags, browser only close anchors on a anchor start or end tag. Actually Navigator and IE differ in how they handle this. Try the following example:

```
<p><b><a href=foo>some text</i> which should be in the label</a></p>
<p>next para and guess what the emphasis will be?</p>
```

Navigator 4 renders the second paragraph in normal text while IE renders it in bold. If you substitute <a> for the </i>, once again the browsers differ. IE stops underlining at the <a> text while Navigator continues until the , although it realizes that you can't click there.

Satwinder continues: browsers happily interpret center within a heading. Tidy now moves the center element to be the parent of the rest of the heading, splitting it as needed, rather than prematurely ending the heading. The same applies to a div element within a heading. Satwinder notes that Tidy inserts a ul when an li is encountered as a direct child of body.

This is a case where you can't produce a legal HTML file that renders the same way as browsers handle this. The same applies to a dt or dd element without an enclosing dl element. I can report that W3C's HTML working group was unwilling to bless naked li's etc. A similar problem arises for dt elements when they contain hr, center or div. The specs say this is illegal, but browsers render it fine!

I have done my best for hr, splitting the dt as needed and enclosing the hr within a dd. The hr doesn't look the same, sadly, as it now starts at the left margin for the dd'st rather than the left margin for dt's. I wasn't sure how to deal with center and div within dt, and chose to discard them.

</br> is now mapped to
 to match observed browser rendering. On the same basis, an unmatched </p> is mapped to

. This should improve fidelity of tidied files to the original rendering, subject to the limitations in the HTML standards described above.

Vlad Harchev spotted that Tidy was swallowing the first and last spaces within inline elements when in a pre element. Now fixed. Zac Thompson spotted that Tidy didn't know that the tags s, strike and u weren't allowed in HTML4 strict. I have now fixed this.

Tidy now preserves the last modified time for the files it writes back to. This was introduced on the suggestion of René Fritz, who uses the SiteCopy utility to upload recently modified files to his Web server. By preserving file timestamps Tidy can be used on all files in a directory without impacting which ones will be uploaded, the next time SiteCopy runs. This is implemented using the fstat and futime system calls. If your platform doesn't support these calls, set PRESERVEFILETIMES to 0 in platform.h

I have fixed a bug on lexer.c which screwed up the removal of doctype elements. This bug was associated with the symptom of printing an indefinite number of doctype elements.

August 1999

Added lowsrc and bgproperties attributes to attribute table. Rob Clark tells me that bgproperties="fixed" on the

body elements causes NS and IE to fix the background relative to the window rather than the document's content.

Terry Teague kindly drew my attention to several bugs discovered by other people: My thanks to Randy Waki for discovering a bug when an unexpected inline end-tag is found in a ul or ol element. I have added new code to ParseList in parser.c to pop the inline stack and discard the end tag. I am checking to see whether a similar problem occurs elsewhere. Randy also discovered a bug (now fixed) in TrimInitialSpace() in parser.c which caused it to fail when the element was the first in the content. John Cumming found that comments cause problems in table row group elements such as tbody. I have fixed this oversight in this release.

Bjoern Hoehrmann tells me that bgsound is only allowed in the head and not in the body, according to the Microsoft documentation. I have therefore updated the entry in tags.c. The slide generation feature caused an exception when the original document didn't include a document type declaration. The fix involves setting the link to the parent node when creating the doctype node.

26th July 1999

Jussi Vestman reported a bug in FixDocType in lexer.c which caused tidy to corrupt the parse tree, leading to an infinite loop. I independently spotted this and fixed it. Justin Farnsworth spotted that Tidy wasn't handling XML processing instructions which end in ?> rather than just > as specified by SGML. I have added a new option: assume-xml-procins: yes which when set to yes expects the XML style of processing instruction. It defaults to no, but is automatically set to yes for XML input. Justin notes that the XML PIs are used for a server preprocessor format called PHP, which will now be easy to handle with Tidy. Richard Allsebrook's mail prompts me to make sure that the contents of processing instructions are treated as CDATA so that < and > etc. are passed through unescaped.

Bill Sowers asks for Tidy to support another server preprocessor format called Tango which features syntax such as:

```
<b><@include <@cgi><appfilepath>includes/message.html></b>
```

I don't have time to add support for Tango in this release, but would be happy if someone else were to mail in appropriate changes. Darrell Birsak reports problems when using DOS on Win98. I am using Win95 and have been unable to reproduce the problem. Jelks Cabaniss notes that Tidy doesn't support XML document type subset declarations. This is a documented shortcoming and needs to be fixed in the not too distant future. Tidy focuses on HTML, so this hasn't been a priority to date.

Jussi Vestman asks for an optional feature for mapping IP addresses to DNS hostnames and back again in URLs. Sadly, I don't expect to be able to do this for quite a while. Adding network support to Tidy would also allow it to check for bad URLs.

Ryan Youck reports that Tidy's behavior when finding a ul element when it expects an li start tag doesn't match Netscape or IE. I have confirmed this and have changed the code for parsing lists to append misplaced lists to the end of the previous list item. If a new list is found in place of the first list item, I now place it into a blockquote and move it before the start of the current list, so as to preserve the intended rendering.

I have added a new option - enclose-text which encloses any text it finds at the body level within p elements. This is very useful for curing problems with the margins when applying style sheets.

9th July 1999

Added bgsound to tags.c. Added '_' to definition of namechars to match html4.decl. My thanks to Craig Horman for spotting this.

Jelks Cabaniss asked for the clean option to be automatically set when the drop-font-tags option is set. Jelks also notes that a lot of the authoring tools automatically generate, for example, <I> and in place of and (MS FrontPage 98 generated the latter, but FP2000 has reverted to the former - with no option to change or set it). Jelks suggested adding a general tag substitution mechanism. As a simpler measure for now I have added a new property called logical-emphasis to the config file for replacing i by em and b by strong.

7th July 1999

Fixed recent bug with escaping ampersands and plugged memory leaks following Terry Teagues suggestions. Changed IsValidAttrName() in lexer.c to test for namechars to allow - and : in names.

2nd July 1999

Chami noticed that the definition for the marquee tag was wrong. I have fixed the entry in tags.c and Tidy now works fine on the example he sent. To support mixing MathML with HTML I have added a new config option for declaring empty inline tags "new-empty-tags". Philip Riebold noted that single quote marks were being silently dropped unless quote marks was set to yes. This is an unfortunate bug recently introduced and now fixed.

Paul Smith sent in an example of badly formed tables, where paragraph elements occurred in table rows without enclosing table cells. Tidy was handling this by inserting a table cell. After comparison with Netscape and IE, I have revised the code for parsing table rows to move unexpected content to just before the table.

26th June 1999

Tony Leneis reports that Tidy incorrectly thinks the table frame attribute is a transitional feature. Now fixed. Chami reported a bug in ParseIndent in config.c and that onsubmit is missing from the table of attributes. Both now fixed. Carsten Allefeld reports that Tidy doesn't know that the valign attribute was introduced in HTML 3.2 and is ok in HTML 4.0 strict, necessitating a trivial change to attrs.c.

Axel Kielhorn notes that Tidy wasn't checking the preamble for the DOCTYPE tag matches either "html PUBLIC" or "html SYSTEM". Bill Homer spotted changes needed for Tidy to compile with SGI MIPSpro C++. All of Bill's changes have been incorporated, except for the include file "unistd.h" (for the unlink call) which isn't available on win32. To include this define NEEDS_UNISTD_H

Bjoern Hoehrmann asked for information on how to use the result returned by Tidy when it exits. I have include a example using Perl that Bjoern sent in. Bodo Eing reported that Tidy gave misleading warning when title text is emphasized. It now reports a missing </title> before any unexpected markup.

Bruce Aron says that many WYSIWYG HTML editors place a font element around an hypertext link enclosing the anchor element rather than its contents. Unfortunately, the anchor element then overrides the color change specified by the font element! I have added an extra rule to ParseInline to move the font element inside an anchor when the anchor is the only child of the font element. Note CSS is a better long term solution, and Tidy can be used to replace font elements by style rules using the clean option.

Carsten Allefeld reported that valign on table cells caused Tidy to mislabel content as HTML 4.0 transitional rather than strict. Now fixed. A number of people said they expected the quote-mark option to apply to all text and not just to attribute values. I have obliged and changed the option accordingly.

Some people have wondered why "</" causes an error when present within scripts. The reason is that this substring is not permitted by the SGML and XML standards. Tidy now fixes this by inserting a backslash, changing the substring to "<\>". Note this is only done for JavaScript and not for other scripting languages.

Chami reported that onsubmit wasn't recognized by Tidy - now fixed. Chris Nappin drew my attention to the fact that script string literals in attributes weren't being wrapped correctly when QuoteMarks was set to no. Now fixed. Christian Zuckschwerdt asked for support for the POSIX long options format e.g. --help. I have modified tidy.c to support this for all the long options. I have kept support for -help and -clean etc.

Craig Horman sent in a routine for checking attribute names don't contain invalid characters, such as commas. have used this to avoid spurious attribute/value pairs when a quotemark is misplaced. Darren Forcier is interested in wrapping Tidy up as a Win32 DLL. Darren asked for Tidy to release its memory resources for the various tables on exit. Now done, see DelnitTidy() in tidy.c

Darren also asks about the config file mechanism for declaring additional tags, e.g. **new-blocklevel-tags: cfoutput, cfquery** for use with Cold Fusion. You can add inline and blocklevel elements but as yet you can't add empty elements (similar to br or hr) or to change the content model for the table, ul, ol and dl elements. Note that the indent option applies to new elements in the same way as it does for built-in elements. Tidy will accept the following:

```
<cfquery name="MyQuery" datasource="Customer">
  select CustomerName from foo where x > 1
</cfquery>
```

```
<cfoutput query="MyQuery">
  <table>
    <tr>
      <td>#CustomerName#</TD>
    </tr>
  </table>
</cfoutput>
```

but the next example **won't** since you can't as yet modify the content model for the table element:

```
<cfquery name="MyQuery" datasource="Customer">
  select CustomerName from foo where x > 1
</cfquery>

<table>
  <cfoutput query="MyQuery">
    <tr>
      <td>#CustomerName#</TD>
    </tr>
  </cfoutput>
</table>
```

I have been studying richer ways to support modular extensions to html using assertions and a generalization of regular expressions to trees. This work has led a tool for generating DTDs named **dtdgen** and I am in the process of creating a further tool for verification. More information is available in my note on [Assertion Grammars](#). Please contact me if you are interested in helping with this work.

David Fallon is interested in using Tidy to dynamically repair markup in an HTML editor as people type. My recommendation is to take advantage of the tables in tags.c and attrs.c for this, and to defer to application of the full range of heuristics to such a time as saving to disk or when explicitly requested. The CM_OPT property in the tags table indicates that the end tag is optional, while CM_EMPTY indicates that an element is *empty*, i.e. has no content.

Betsy Miller reports: *I tried printing the HTML Tidy page for a class I am teaching tomorrow on HTML, and everything in the "green" style (all of the examples) print in the smallest font I have ever seen (in fact they look like tiny little horizontal lines). Any explanation?*

Yes. This is a problem with Internet Explorer and Style Sheets. The Tidy page includes a CSS style sheet that tries to make the size of the font used for the examples 80% smaller than for normal text. Internet Explorer gets this wrong, picking a very much smaller font. I am hoping this bug is fixed in the IE 5.0 release. I have changed the style sheet to work around this.

Francisco Guardiola writes that Tidy wasn't fixing frameset documents with body elements unenclosed in noframes elements. Now fixed. Frederik Fouvry found that comments after the html end tag generated a warning for content after body. I can't reproduce this symptom and assume it was fixed in an earlier release.

Indrek Toom wants to know how to format tables so that tr elements indent their content, but td tags do not. The solution is to use *indent: auto*. Jelks Cabaniss noted that the clean option created style rules with tag names in uppercase, which would cause problems for Extensible HTML (xhtml). This prompted me to overhaul Tidy to switch to lower case for that tag tables and literals. I have adopted Jelks' suggestion for adding support for a doctype property in config files. This supports **omit**, **auto**, **strict**, **loose** or a string specifying the fpi (formal public identifier).

Johannes Koch notes that Tidy doesn't fix up the doctype correctly when bursting to slides. He says that if a document contains the HTML 4.0 strict DT declaration, then the slides also include the same strict DT declaration, but also contain the center tag which does not appear in the strict DTD. I have applied a simple work around, which is to remove the original doctype when bursting to slides.

I have extended the support for the ASP preprocessing syntax to cope with the use of ASP within tags for attributes. I have also added a new option `wrap-asp` to the config file support to allow you to turn off wrapping within ASP code. Thanks to Ken Cox for this idea.

Larry Virden asked for a compile-time option for setting the config file, he says "The reason it would be useful is to be able to define a set of commonly used additional tags. For instance, our site is starting to use a lot of ColdFusion. I would love to be able to put the CF tags into a site wide file so that users of tidy automatically get them defined". You can now do this by defining `CONFIG_FILE` in platform.h

Loïc Trégan asks: Is there a way to generate a "light" xml, with no "<!DOCTYPE...>" and "xmlns=..."? I have tweaked the code to allow the doctype property to apply when outputting XML, and added a new property

"add-xml-pi" to control whether an `<?xml?>` processing instruction is added or not. To generate a minimal XML document, you can set the `xml-out` property to yes, the `doctype` and `add-xml-pi` property to no.

Marc Jauvin has been using Windows Application to generate Web pages and found that some of them generate very "non-portable" HTML. One of the problems that is often introduced is the use of "\" in URLs instead of "/" which confuses Unix Web servers. To deal with this I have introduced the "fix-backslash" property. This has been set by default to yes, but can be set to no if that causes problems.

The new property `indent-attributes` when set to yes places each attribute on a new line. Note that the attributes are only indented one space. Paul Ossenbruggen asked for something slightly different, where the second and subsequent attributes start on a new line and are indented to line up under the first attribute. That proved to involve rather more work to implement than I have time for right now. I plan to work some more on this for a future release.

Peter Jeremy reported that when an error file is specified to tidy (`-f file`), the error file is opened for every HTML file specified on the command line, but not closed until all HTML files have been processed. If a large number of files are specified on the command line (e.g. processing the FreeBSD handbook), this can overflow the process or system file descriptor table. I have now fixed this so that the error file is only opened once.

Rafi Stern notes: I have entered `output-xml: yes` in my config file, not `output-xhtml`. Tidy second guesses me and adds the `xmlns` attribute for XHTML at the head of my file, which I then have to remove as this interferes with my XSLT parser. Fixed along with the other bugs reported by Rafi.

Steffen Ullrich and Andy Quick both spotted a problem with attribute values consisting of an empty string, e.g. `alt=""`. This was caused by bugs in `tidy.c` and in `lexer.c`, both now fixed. Jussi Vestman noted Tidy had problems with `hr` elements within headings. This appears to be an old bug that came back to life! Now fixed. Jussi also asked for a config file option for fixing URLs where non-conforming tools have used backslash instead of forward slash.

An example from Thomas Wolff allowed me to the idea of inserting the appropriate container elements for naked list items when these appear in block level elements. At the same time I have fixed a bug in the table code to infer implicit table rows for text occurring within row group elements such as `thead` and `tbody`. An example sent in by Steve Lee allowed me to pin point an endless loop when a head or body element is unexpectedly found in a table cell.

15th April 1999

Another minor release. Jacob Sparre Andersen reports a bug with `"` in attribute values. Now fixed. Francisco Guardiola reports problems when a body element follows the `frameset` end tag. I have fixed this with a patch to `ParseHTML`, `ParseNoFrames` and `ParseFrameset` in `parser.c`. Chris Nappin wrote in with the suggestion for a config file option for enabling wrapping script attributes within embedded string literals. You can now do this using `"wrap-script-strings: yes"`.

14th April 1999

Added check for `Asp` tags on line 2674 in `parser.c` so that `Asp` tags are not forcibly moved inside an HTML element. My thanks to Stuart Updegrave for this. Fixed problem with `&` entities. Bede McCall spotted that `&am` was being written out as `&`. The fix alters `ParseEntity()` in `lexer.c`.

12th April 1999

Added a missing "else" on line 241 in `config.c` (thanks for Keith Blakemore-Noble for spotting this). Added `config.c` and `.o` to the Makefile (an oversight in the release on the 8th April).

8th April 1999

Localization:

All the message text is now defined in `localize.c` which should make it a tad easier to localize Tidy for different languages.

Config file support:

I have added support for configuring tidy via a configuration file. The new code is in config.h which provides a table driven parser for RFC822 style headers. The new command line option `-config <filename>` can be used to identify the config file. The environment variable "HTML_TIDY" may be used to name the config file. If defined, is parsed before scanning the command line. You are advised to use an absolute path for the variable to avoid problems when running tidy in different directories.

Allan Kuchinsky:

Reports that the XML DOM parser by Eduard Derksen screws up on `<img alt=" " data-bbox="142 197 911 227">`, naked `&` and `%` in URLs as well as having problems with newlines after the `'='` before attribute values.

I have tweaked PrintChar when generating XML to output ` ` in place of ` `; and `&` in place of `&`. In general XHTML when parsed as well-formed XML shouldn't use named entities other than those defined in XML 1.0. Note that this isn't a problem if the parser uses the XHTML DTDs which import the entity definitions.

Allan Odgaard:

When tidy encounter entities without a terminating semi-colon (e.g. `"©"`) then it correctly outputs `"©"`, but it doesn't report an error.

I have added a ReportEntityError procedure to localize.c and updated ParseEntity to call this for missing semicolons and unknown entities.

Andreas Buchholz:

Tidy warns if table element is missing. This is incorrect for HTML 3.2 which doesn't define this attribute.

The summary attribute was introduced in HTML 4.0 as an aid for accessibility. I have modified CheckTABLE to suppress the warning when the document type explicitly designates the document as being HTML 2.0 or HTML 3.2.

Andy Brown:

I have renamed the field from class to tag_class as "class" is a reserved word in C++ with the goal of allowing tidy to be compiled as C++ e.g. when part of a larger program.

I have switched to Bool and the values yes and no to avoid problems with detecting which compilers define bool and those that don't.

Andy would prefer a return code or C++ exception rather than an exit. I have removed the calls to exit from pprint.c and used a long jump from FatalError() back to main() followed by returning 2. It should be easy to adapt this to generate a C++ exception.

Sometimes the prev links are inconsistent with next links. I have fixed some tree operations which might have caused this. Let me know if any inconsistencies remain.

Ann Navarro:

Would like to be able to use:

```
tidy file.html | more
```

to pause the screen output, and/or full output passing to file as with

```
tidy file.html > output.txt
```

Tidy writes markup to stdout and errors to stderr. 'More' only works for stdout so that the errors fly by. My compromise is to write errors to stdout when the markup is suppressed using the command line option `-e` or "markup: no" in the config file.

html-kit@chamisplace.com

Writes asking for a single output routine for Tidy. Acting on his suggestion, I have added a new routine `tidy_out()` which should make it easier to embed HTML Tidy in a GUI application such as HTML-Kit. The new routine is in `localize.c`. All input takes place via `ReadCharFromStream()` in `tidy.c`, excepting command line arguments and the new config file mechanism.

Chami also asks for single routines for initializing and de-initializing Tidy, something that happens often from the GUI environment of HTML-Kit. I have added `InitTidy()` and `DeInitTidy()` in `tidy.c` to try to satisfy this need. Chami now supports an online interface for Tidy at the URL:

<http://www.chamisplace.com/asp/hk.asp>

He further asks for Tidy to optionally output a length parameter whenever possible. This could represent the length of the element, attribute or code block related to the error. An online validator could then highlight the starting and ending columns which may be easier for beginners to understand, rather than pointing to a single character column. I will investigate this for a future release.

Chang Hyun Baek:

Reports a problem when generating XML using `-iso2022`. Tidy inserts `?` before `<p>` rather than `</p>`. I tried Chang's test file but it worked fine with in all the right places. Please let me know if this problem persists.

Christian Ruetgers:

When using `-indent` option Tidy emits a newline before which alters the layout of some tables.

I note that browsers aren't conforming to the SGML spec on generally ignoring a newline immediately after start tags and immediately before end tags. Netscape does this for pre elements but not for other tags! My work around is to avoid additional newlines for the content of `th` and `td` elements, except where their content starts with a block level element. This kind of thing is getting really hairy!

Christian Pantel:

Would like the `servlet` tag added to tidy. This looks very similar to `applet` and used for preprocessing document content before delivery. `Servlet` acts as a container for `param` elements and fallback content to be shown if the server doesn't support `servlet`. I have added it as a proprietary tag and parse it in the same way as `applet`.

Christian also reports that `<td><hr/></td>` caused Tidy to discard the `<hr/>` element. I have fixed the associated bug in `ParseBlock`.

Chuck Baslock:

Points out that an isolated `&` is converted to `&` in element content and in attribute values. This is in fact correct and in agreement with the recommendations for HTML 2.0 onwards.

Craig Horman:

Reports that Tidy loops indefinitely if a naked `LI` is found in a table cell. I have patched `ParseBlock` to fix this, and now successfully deal with naked list items appearing in table cells, clothing them in a `ul`.

Craig Johnson:

Reports that Tidy gets confused by `</comment>` before the doctype. This is apparently inserted by some authoring tool or other. I have patched Tidy to safely recover from the unrecognized and unexpected end tag without moving the parse state into the head or body.

Daniel Vogelheim:

Asks for Tidy to recognize obsolete elements such as `LISTING` and to replace them by more modern equivalents, in this case `pre`. I have added code to issue a warning and replace such elements as `xmp`, `listing`,

plaintext by pre, and dir and menu by ul. Daniel also asks for a means to suppressing warnings, i.e. to only report errors. I have added the boolean "show-warnings" to the config file support to deal with this and split off warnings to ReportWarnings().

Dan Rudman:

Would love a version of Tidy written in Java. This is a big job. I am working on a completely new implementation of Tidy, this time using an object-oriented approach but I don't expect to have this done until later this year.

DEFERRED

David Brooke:

Reports that when tidying an XMLfile with characters above 127 Tidy is outputting the numeric entity followed by the character. I have fixed this by a patch to PPrintChar() for XmlTags.

David Getchell:

Reports that Tidy thinks an ol list is HTML 4.0 when you use the type attribute. I have fixed an error in attrs.c to correct this feature to first appearing in HTML 3.2.

Drew Adams:

Reported problems when using comments to hide the contents of script elements from ancient browsers. I wasn't able to reproduce the problem, and guess I fixed it earlier.

Drew also reported a problem which on further investigation is caused by the very weird syntax for comments in SGML and XML. The syntax for comments is really error prone:

```
<!--[text excluding --][[whitespace]*--[text excluding --]]--*>
```

This means that <!--> is a complete comment but <!--> is not since the parser is expecting a matching terminating -- and as it doesn't find the -- it ploughs on and on treating the rest of the markup as a comment unless it finds another end comment. I have added a rule of thumb (a heuristic) for detecting this situation. Basically I count the number of comment groups without other characters and if the count is > 2 and a '>' is seen, a warning is generated.

Drew goes on to comment on the -clean option. This made me take another look at the relative font sizes I am using for the absolute font sizes for 0 through 6. I have tweaked them to get a reasonable match before/after applying -clean as viewed on NS4 and IE4. Font size=3 is taken as the normal body font size and as such the font element is silently dropped unless it also defines a color.

I have also added InlineStyle to deal with the cases where an inline element has as its only child a font element. A further possibility would be to promote style properties common to all children of an element to the element. will have to leave this for future work.

Drew asks why </ is not allowed in script content. The answer is that SGML treats </ as delimiting the end of CDATA element content, so that it ends prematurely before the </script> end tag. Browsers tend not to follow the SGML standard in this respect, but Tidy is designed to help you do so.

Guus Goos:

Notes that tidy *.html doesn't work under DOS. This is because DOS unlike Unix doesn't expand names with wildcards to the list of matching file names. This is a right nuisance and one more reason why Linux is gaining popularity. I plan to provide a work around in a future release of Tidy. Are there any free drop-in replacements for the DOS shell that fix this problem?

Jack Horsfield:

Like a number of others would like list items and table cells to be output compactly where possible. I have added a flag to avoid indentation of content to tags.c that avoids further indentation when the content is inline, e.g.

```
<ul>
  <li>some text</li>
  <li>
    <p>
      a new paragraph
    </p>
  </li>
</ul>
```

This behavior is enabled via "smart-indent: yes" and overrides "indent: no". Use "indent-spaces: 5" to set the number of spaces used for each level of indentation.

Jeff Young:

Has a few suggestions that will make Tidy work with XSL. Thanks, I have incorporated all of them into the new release.

Jelks Cabaniss:

Reports that the Tidy thinks the end tag is missing if the script element has no content. I have patched ParseScript to fix this. Jelks also asks for a way to ask Tidy to hide the contents of script and style elements; a way to avoid promoting inline styles with -clean to style rules as a work around for a bug in IE for URLs with relative URLs; finally, a way to avoid empty elements being discarded, especially if they define an ID for scripting. Very reasonable, but I would prefer leave these to a future release. (This release is big enough right now!).

One thing I can satisfy right away is a mailing list for Tidy. html-tidy@w3.org has been created for discussing Tidy and I have placed the details for subscribing and accessing the Web archive on the Tidy overview page.

Johannes Koch:

Reports that Tidy isn't quite right about when it reports the doctype as inconsistent or not. I have tweaked HTMLVersion() to fix this. Let me know if any further problems arise.

John Tobler:

Wants to know how to get Tidy to preserve his explicit entities e.g. " and . Currently Tidy interprets all entities as character values and as such has no way to distinguish whether these were derived from entities or not. To help John with this release you can use "quote-marks: yes" in the config file if you want all " marks to appear as " and "quote-nbsp: yes" if you want non-breaking spaces to be shown as entities. Note that for XML in general is not-predeclared, so you should also use "numeric-entities: yes". This doesn't apply to XHTML though.

John also reports that the weirdly complex URLs using the javascript: scheme as used by www.bookmarklets.com can cause Tidy indigestion. I have made Tidy aware of which attributes are using Javascript and disabled the missing quote mark heuristic for these. I have also tweaked the way unknown entities are reported to say that the markup have contain unescaped ampersands.

Mathew Cepi:

Notes that dir and menu are deprecated and not allowed in HTML4 strict. I have updated the entry in the tags table for these two. I also now coerce them automatically to ul when -clean is set.

Maurice Buxton:

Reports that some implementations of gcc don't work with the current compiler directive Tidy uses to avoid duplicate typedefs for uint and ulong. I don't have a truly platform independent solution for this, so you may need to edit platform.h if the code doesn't compile out of the box on your platform.

Osma Ahvenlampi:

Found that Tidy is confused by map elements in the head. Tidy knows that map is only allowed in the body and thinks the author has left out the

start tag. Thereafter elements which it knows only belong in the head are moved to the head, so things should work out ok. Osma also reports having difficulties with non-breaking spaces, but I was unable to reproduce these with the new release of Tidy, so perhaps the problems have been fixed.

Paul Ward:

Reports that Tidy caused JavaScript errors when it introduced linebreaks in JavaScript attributes. Tidy goes to some efforts to avoid this and I am interested in any reports of further problems with the new release.

Rafi Stern:

Would like Tidy to warn when a tag has an extra quote mark, as in `<a href="xxxxx">`. I have patched ParseAttribute to do this.

Rene Fritz:

Reported a space being inserted at the end of lines when a the text is wrapped at the start of hypertext links. This isn't occurring with this release, so I guess the problem was solved a while back. Rene also suggests that Tidy could be used to add and remove metadata and attributes etc. for a group of files, e.g. to add a link to a style sheet or to assert attribution. This sounds like a good idea for work in the future.

Shane McCarron:

Reports that Tidy sometimes wraps text within markup that occurs in the context of a pre element. I am only able to repeat this when the markup wraps within start tags, e.g. between attribute values. This is perfectly legitimate and doesn't effect rendering.

Steven Lobo:

Notes that Tidy doesn't remove entities such as ` ` or `©` which aren't defined by XML 1.0. That is true these entities **are** fine if you are using XHTML. If you want to generate generic XML then you need to use the `-r` option or to set "numeric-entities: yes" in the config file. This will then output all such entities in their numeric form or as direct character values according to the character encoding flags.

Steven Pemberton:

Comments that he would like Tidy to replace naked `&` in URLs by `&`. You can now use "quote-ampersands: yes" in the config file to ensure this. Note that this is always done when outputting to XML where naked `'&'` characters are illegal.

Steven also asks for a way to allow Tidy to proceed after finding unknown elements. The issue is how to parse them, e.g. to treat them as inline or block level elements? The latter would terminate the current paragraph whereas the former would not.

If treated as inline, presumably, unknown tags should be treated specially, for instance, normal inline end tags close the currently open inline element, but this doesn't feel right for unknown tags. What should the content model for unknown tags be - flow? Again its far from obvious. One way to avoid these difficulties would be to provide a means for authors to declare unknown tags in the config file.

You can now declare new inline and block-level tags in the config file, e.g.:

```
define-inline-tags: foo, bar
define-blocklevel-tags: blob
```

The content model for new tags allows for block or inline content. Steven further comments that some authors use `ul` without an `li` to indent content. Tidy currently coerces these to wrap the content within an `li` which alters the rendering. He suggests using blockquote instead. I have done this, and if you use the `-clean` option at the same time, it gets replaced by a `div` element with a class and style rule for indenting the content.

Stuart Updegrave:

Would like to be able to coerce attributes to uppercase. I have added support for "uppercase-attributes: yes" for this. Stuart also asks for Tidy to support Microsoft's ASP tags. These are part of Microsoft's server-side scripting model (similar to CGI). I have treated ASP tags in the same way as processing instructions, and they don't effect the version of HTML as they are assumed to have been interpreted before delivery to the client.

Stuart is also interested in having Tidy reading from and writing back to the Windows clipboard. This sounds interesting but I have to leave this to a future release.

Terry Cassidy:

Points out that Tidy doesn't like "top" or "bottom" for the align attribute on the caption element. I have added a new routine to check the align attribute for the caption element and cleaned up the code for checking the document type.

Xavier Plantefeve:

Suggests that I should ensure that the options are self consistent, e.g. if -asxml is set, then this should imply lower case and override any instruction to omit optional end tags. Accordingly, I have introduced a new routine AdjustConfig() that is applied after reading the command line and config files and before tidying any files.

Xavier wonders whether name attributes should be replaced or supplemented by id attributes when translating HTML anchors to XHTML. This is something I am thinking about for a future release along with supplementing lang attributes by xml:lang attributes.

Zdenek Kabelac:

Asks for headings and paragraphs to be treated specially when other tags are indented. I have dealt with this via the new smart-indent mechanism.

22nd February 1999

Tidy can now fix up XML empty tags for which the attribute values are unquoted, e.g. <br clear=all/>. Care is taken to avoid this being applied to tags with URLs, e.g. where the / is part of the attribute value and doesn't signify an empty tag. Authors are advised to always quote attribute values to avoid such problems!

22nd January 1999

Tidy no longer complains about a missing </tr> before a <tbody>. Added link to a free [win32 GUI for tidy](#).

11th January 1999

Added a link to the OS/2 distribution of Tidy made available by Kaz SHIMZ. No changes to Tidy's source code.

7th January 1999

Fixed bug in ParseBlock that resulted in nested table cells.

Fixed clean.c to add the style property "text-align:" rather than "align:".

Disabled line wrapping within HTML alt, content and value attribute values. Wrapping will still occur when output as XML.

16th December 1998

This release fixes a problem with missing quotemarks in attribute values introduced in the December 14th release. It also fixes problems with parsing tables when the table cells include naked list items and when unexpected end tags are encountered for td and tr cells. Warnings are now generated for unknown entities

(those not defined by HTML 4.0). It may be worth thinking about a new option to determine how to handle these, especially for XML.

14th December 1998

Rewrote parser for elements with CDATA content to fix problems with tags in script content.

New pretty printer for XML mode. I have also modified the XML parser to recognize xml:space attributes appropriately. I have yet to add support for CDATA marked sections though.

script and noscript are now allowed in inline content.

To make it easier to drive tidy from scripts, it now returns 2 if any errors are found, 1 if any warnings are found, otherwise it returns 0. Note tidy doesn't generate the cleaned up markup if it finds errors other than warnings.

Fixed bug causing the column to be reported incorrectly when there are inline tags early on the same line.

Added -numeric option to force character entities to be written as numeric rather than as named character entities. Hexadecimal character entities are never generated since Netscape 4 doesn't support them.

Entities which aren't part of HTML 4.0 are now passed through unchanged, e.g. &precompiler-entity; This means that an isolated & will be pass through unchanged since there is no way to distinguish this from an unknown entity.

Tidy now detects malformed comments, where something other than whitespace or '--' is found when '>' is expected at the end of a comment.

The
 tags are now positioned at the start of a blank line to make their presence easier to spot.

The -asxml mode now inserts the appropriate Voyager html namespace on the html element and strips the doctype. The html namespace will be usable for rigorous validation as soon as W3C finishes work on formalizing the definition of document profiles, see: [WD-html-in-xml](#).

13th November 1998 and earlier releases

Fixed bug wherein <style type=text/css> was written out as <style type="text/ss">.

Tidy now handles wrapping of attributes containing JavaScript text strings, inserting the line continuation mark as needed, for instance:

```
onmouseover="window.status='Mission Statement, \
Our goals and why they matter.'; return true"
```

You can now set the wrap margin with the -wrap option.

When the output is XML, tidy now ensures the content starts with <?xml version="1.0"?>.

The Document type for HTML 2.0 is now "-//IETF//DTD HTML 2.0//". In previous versions of tidy, it was incorrectly set to "-//W3C//DTD HTML 2.0//".

When using the -clean option isolated FONT elements are now mapped to SPAN elements. Previously these FONT elements were simply dropped.

NOFRAMES now works fine with BODY element in frameset documents.