

TagWrite 3.0

for Windows

Operator's Guide

ZANDAR Corporation

TagWrite 3.0 For Windows
Operator's Guide

Copyright protection claimed includes all forms and matters of copyrightable material and information now allowed by statutory or judicial law or hereinafter granted, including without limitation, material generated from the software programs which are displayed on the screen such as icons, screen display looks, etc. Reproduction of this manual by any means, including xerography, scanning, or any electronic means, is prohibited.

Copyright ©1994 ZANDAR Corporation. All rights reserved.

TagWrite ® is a registered trademark of the ZANDAR Corporation.

"Ami Pro" is a trademark of Lotus Development Corporation.

"Apple" and Apple "Macintosh" are trademarks of Apple Computer Corporation.

"IBM" and "Bookmaster" are registered trademarks of the International Business Machines Corporation

"Interleaf" is a trademark of Interleaf Corporation.

"Microsoft", "Microsoft Word", "Microsoft Word For Windows", "Windows" and "Rich Text Format (RTF)" are registered trademarks of Microsoft Corporation.

"Norton Utilities" is a trademark of Symantec Corporation.

"PageMaker" is a trademark of the ZANDAR Corporation.

"PC Tools" is a trademark of Central Point Software.

"Quark" is a trademark of Quark, Incorporated.

"WordPerfect" is a trademark of the WordPerfect Corporation.

"Ventura" & "Ventura Publisher" are trademarks of Corel Corporation.

All other product names are registered trademarks of their manufacturers or owners.

Information in this manual is subject to change without notice and does not represent a commitment on the part of the vendor. The TagWrite software is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement.

ZANDAR Corporation, creator of TagWrite, is a pioneer in the development of SGML tools and other tagging applications. Our products specialize not only in tagging but also in untagging SGML and reformatting it for display and other uses.

ZANDAR offers additional software tools and services to expand your SGML application. An expanded tool set is available to automatically and bi-directionally transform Rich Text Format (RTF) tables and fully formatted SGML CALS tables including complete table declaration, support for spanning cells, grid and box, and all character formatting. Extensions can be developed to fully transform tables between SGML and virtually any other markup.

For SGML and other tagging/untagging development services and software tools, contact ZANDAR Corporation through our telephone listing in Burlington, Vermont, USA. For information on development services and software call our sales-only line at 800-639-5818. (Sorry, no support calls can be accepted. For support, contact Corel Technical Support at 1-800-818-1848)

ZANDAR Corporation
P.O. Box 467
Jericho, Vermont 05465

Table of Contents

The Function of TagWrite	1
What is “Tagging” or “markup” or “coding” ?	1
Code	1
Tag	2
What does a tag look like?	2
Why do you tag a document?	2
Why tag every structural element of a document?	2
Standard Generalized Markup Language—SGML	3
Why tag a document automatically?	3
What tagging systems can TagWrite support?	3
What word processing formats can TagWrite read and write?	3
TagWrite Reads and Writes	3
How fast is TagWrite?	4
What kinds of documents are most likely to be used with TagWrite?	4
Structure Is Different From Style	4
Untagging and reformatting your document.	4
What Happens to the Original Text File?	5
Save as ASCII	5
More about the operation of TagWrite	5
Sample Applications Included	6
Contents of the TagWrite Directories	7
Getting Started Tutorial	9
Overview	9
Starting and Exiting TagWrite	10
Start	10
Exit	10
Choose Template	10
Template Name	10
Function of Choose Template	11
Select Files To Tag	11
Optional features of the “Select Files To Tag” Dialog Box	12
Location of the Template on your computer	14
Find Templates	15
Setting A New Default Directory	16
Template Editor	16
Information about Template and Index Files	16
The Template Manager	17
Template Manager	17
Copy A Template	17
Renaming Templates	19
Delete Template	19

Working with word-processed Text files _____ 21

Overview	21
Use The Correct Word Processor With a Template	22
Templates Support A Specific Document Type	22
The Key Entry (typing) Convention For a Document Type	22
Match of the Template and the Key Entry Convention	23
Styles	23
Conventional Typing	23
Controlling The Match—Template and Word-processor	24
The Filename Extension	24
<i>Save Your Text Files With The Correct Extension</i>	24
How To Use the Extensions Button	24
Pictures	25
Save File as ASCII — Picture Stripped Out	25
Keeping Track Of Pictures Under Windows	25

Working With ASCII Files _____ 27

Definition of ASCII	27
Overview of ASCII and TagWrite	27
Save Word Processed Input File As An ASCII Output File	28
To Save In ASCII	28
The Result Of Save As ASCII In DOS Or UNIX Format	28
Accept The Input File in ASCII Format	29
The ASCII Input File Options Dialog Box	29
Select ASCII Input File Format	30
Save ASCII File In	30
Save With WordPerfect or RTF Style	33
Overview	33
How To Use “With Styles”	33
Raw ASCII Text From Scanners	36

Problems and Error Messages _____ 37

Problems	37
TagWrite runs very slowly (slower than three seconds per page)	37
TagWrite functions endlessly with no apparent progress.	38
Strange codes appear in the document.	38
No Templates appear in “Choose Template” Dialog Box	38
Error Message: STOP : Document is wrong type for {templatename}	38
Error Message: Microsoft Word Binary is no longer supported OR Document is not an RTF document	39
Template names do not display	39
List of Error Messages	40
Can’t read the template index; must be corrupted.	40
Unable to process document; style definition is too long	40
Document is wrong type for template {name}	40
You may not edit this template	40

Out of Memory — close another application!	40
Read file Errors:	40
Write file Errors:	41
Error deleting temporary file!	41
Errors related to the document's file format:	41
Bad RTF control word {name} in CODES	41
Unknown code {name} run error check on template	41
Microsoft Word Binary is no longer supported	42

The Function of TagWrite

TagWrite is known as an “autotagger”. It is also an “untagger”. TagWrite “marks up” your file with “tags” or “codes.” It automatically marks the structure, the format, and the content of your file.

What is “Tagging” or “markup” or “coding” ?

“Markup” is an old fashioned word taken from the typesetting industry in the era when a skilled craftsman set metal, moveable type. Prior to setting the type, an editor would mark-up a manuscript with special signs to tell the typesetter how the document should appear.

“Tagging”, and “coding” mean about the same thing in today’s computer world; however, we distinguish between a “tag” and a “code”.

Code

A “code” is a word processing format command that only has meaning within the word processing software. In this tutorial, a code specifically is a WordPerfect or Microsoft Word (RTF) command. Such codes only make sense to WordPerfect or Microsoft Word or compatible products.

TagWrite can read and write the internal codes of WordPerfect and Microsoft Word (RTF) file formats.

WordPerfect and Microsoft Word (RTF) **Styles** are internal codes that tell the word processing software how to format a paragraph or a piece

of text. If your version of TagWrite supports Styles, TagWrite can read and write Style codes in addition to most other WordPerfect and Microsoft RTF codes.

Tag

A tag is a descriptive piece of information—a special set of letters, numbers, and characters—visibly inserted into the word processed document file. Sometimes it is referred to as an “embedded tag” because it is embedded in the flow of the text file. You can see it and read it when present.

The tag has meaning. It identifies the structure, the format, or the content of the text. In typesetting systems and in document storage and retrieval systems like the Standard Generalized Markup Language (SGML), one or more tags must be inserted into each paragraph of the document.

What does a tag look like?

All tags have a “delimiter” (a begin and an end marker) and a tagname. There are many kinds of delimiters. TagWrite supports virtually any tagging system. For example, if you are using typesetting systems like PageMaker, Interleaf, SGML, XICS and others, the delimiter is the angle bracket `< >`. The tags look like this `<TAGNAME>`. Ventura Publisher typesetting software uses an “at” sign `@` and an equal sign `=` as delimiters. Ventura Publisher tags look like this: `@TAGNAME =`. The U. S. Government Printing Office uses the “bell code” (ASCII hex 07) as a delimiter. Troff and other systems use dot commands that look like this: `.TAG`.

Why do you tag a document?

Every document has a structure made up of many parts (or components) like a title, a chapter heading or a paragraph heading. Every piece of structure may require a tag. A typical document may have twenty or more distinct structural components. An SGML document can have many more distinct tags.

The correct tag embedded into the proper place of a text file distinguishes a title from a chapter heading from a paragraph heading or distinguishes a first level paragraph from a second level paragraph and so forth.

Why tag every structural element of a document?

If each structural piece of a document is correctly tagged, then the document can be said to contain intelligent information about itself. The document itself then tells you how it is organized and stored.

This information allows the document to be used constructively. For example: with the structure tagged, typesetting software can be display and print the document to meet style specifications. Hypertext applications can compile a tagged document. In complex storage and retrieval systems, the same tagged document can be stored, moved, and

used for different purposes on different hardware and software platforms.

Standard Generalized Markup Language—SGML

SGML is an international standard (ISO 8879) for tagging (marking-up) documents. The SGML Document Type Definition (DTD) specifies every piece of structure of the document that must be tagged if the document is to be certified as being valid.

SGML, and its hybrid variants, have the purpose of declaring meaningful structure in a document.

Why tag a document automatically?

Each document may contain thousands of tags. If you enter these tags by hand from the keyboard or with a mouse, you will spend many boring hours and make many errors. TagWrite enters these tags automatically in seconds. Experience shows that used correctly, TagWrite reduces the likelihood of error.

What tagging systems can TagWrite support?

TagWrite can be configured to support almost any coding system including Standard Generalized Markup Language (SGML), Aldus PageMaker, Ventura Publisher, Interleaf, Troff, QuarkXPress, Framemaker, the Xerox Integrated Composition System (XICS), Government Printing Office Bell Codes, and all others that use ASCII characters to create tag names.

What word processing formats can TagWrite read and write?

TagWrite reads native WordPerfect 5 series including WordPerfect For Windows, and all Microsoft Word products saved as RTF including Microsoft Word For Windows, and ASCII.

Note: With TagWrite, you must save Microsoft Word products in RTF format. Do **not** save your Microsoft Word files as “Normal”. TagWrite will not process Microsoft Word files saved as Normal.

TagWrite Reads and Writes

TagWrite can perform the following:

- Reads WordPerfect 5 series and writes WordPerfect or Straight ASCII
- Reads Microsoft RTF and writes RTF or Straight ASCII
- Reads Straight ASCII and writes WordPerfect, or Microsoft RTF or ASCII

TagWrite 3.0 can read from ASCII (for example, SGML tagged files), strip the SGML tags, and write fully formatted WordPerfect or Microsoft Word.

TagWrite can read WordPerfect in one formatted version and write WordPerfect in a different formatted version, or read Microsoft Word RTF in one formatted version and write RTF in a different formatted version.

Many word processing products save their files in WordPerfect 5 series or Microsoft RTF format. It is likely that TagWrite applications can be written to accept input files from these products.

How fast is TagWrite?

TagWrite's speed is dependent on your hardware, the application, and the word processor. Generally, on a 33 MHz 80386 machine with a complex document and a correct TagWrite application, tagging occurs at about 100-120 pages per minute for a WordPerfect document, and about 40-60 pages per minute for an RTF document.

What kinds of documents are most likely to be used with TagWrite?

TagWrite is used most often with "structured documents". Structure denotes the components, or the identifiable parts, of a document. Usually, each structural component of document must be tagged.

Each document type has its own structure. Some examples of structural components are: paragraph heading levels, chapter titles, tables, footnotes, graphics, references to linked data like a Dynamic Data Exchange (DDE), and others.

TagWrite can be configured for the structure of virtually any document type.

Structure Is Different From Style

It is important to distinguish "structure" from "style". Style denotes the appearance.

A structural component, like a "chapter heading", can be printed in more than one style. For example, the structure "chapter heading" may be printed in 14 point, bold Helvetica, centered type or it may be printed in 18 point, Times Roman, flush right, italic. Regardless of the style, the structural component remains a "chapter heading".

TagWrite is very powerful when used for structured documents of virtually any style.

Untagging and reformatting your document.

TagWrite applications can be made to untag a document to restore a clean word processing file. You can also remove other information (like unwanted binary information) that does not belong in the cleaned-up file. After untagging and clean up, you can bring the file back into your word processor for a new round of editing.

TagWrite can untag and simultaneously reformat your document so that it can be imported into WordPerfect or Microsoft Word or most typesetting software.

You can untag the *same* text file and retag it for different coding systems.

What Happens to the Original Text File?

With TagWrite, your original text file is protected from corruption. A correct TagWrite application is designed so that the original document file is preserved, and a new file, containing the correct tags or codes, is written to the disk.

Save as ASCII

You can choose from a dialog box to have the new, tagged (or untagged) file stored in DOS or Unix ASCII format.

More about the operation of TagWrite

(Read this if you want to know more about the internal operation of TagWrite.)

Internally, TagWrite is rather complex. Unless you plan to create TagWrite Templates, it is not really necessary to explore the deep, inner workings of the software. You should, however, understand the basic ideas behind TagWrite.

When you type a document on a computer using WordPerfect or Microsoft Word, the computer creates an electronic file that represents alphabetic letters, numbers, punctuation characters and many word processing and other format codes including styles.

TagWrite is designed to understand that information.

In the most simple case, imagine that you keyenter a chapter title by centering the title like this:

Heading Title

You can write a TagWrite Template to contain a rule that will search for a word processing center code and then text. (The Template may contain many other rules.) The Template is stored in your computer.

The simplest explanation of TagWrite is that it is a "parser". When TagWrite is activated, it looks at the first character in your word processed file. Let us assume that a center code occurs first. TagWrite remembers the center code and then looks into the TagWrite Template and tries to match the word processing center code with a TagWrite Input rule.

If an Input rule in the TagWrite Template is looking for a center code, then that Template rule is partly satisfied, and TagWrite remembers this fact.

TagWrite then goes back to the word processed file to look at the next character. The next character in our example is an ordinary alphabetic character.

TagWrite now looks back into the Template to see if the rule that starts with center is looking next for an ordinary alphabetic character. If the rule is looking for the alphabetic character, then it remains active.

TagWrite goes on looking at the next character or format code in the word processed file comparing it with the Template rule. This process goes on until a rule in the Template is totally fulfilled.

If a Template Input rule is totally fulfilled, then TagWrite looks to the corresponding Output rule in the Template. The Output rule tells TagWrite what to do with a fulfilled Input rule.

In the most common application of TagWrite, if an Input rule is fulfilled, then the Output rule will write back to disk a typesetting code (like <TAGNAME>) plus the desired text. Since TagWrite allows the Output rule to write back information differently from the way it reads in the information, TagWrite is able to enter tags or styles, remove tags, and is able to enter and remove other information and reconfigure your word processed file.

TagWrite Templates can be set-up to look at very complex input data streams and make logical decisions as well as write back complex formatting different from the original file. It is fair to say that TagWrite has some intelligence.

Sample Applications Included

- | |
|--|
| <ol style="list-style-type: none">1. \TAGWRITE\EXAMPLES\SGMLDEMO.RTF
This illustrates transformation from RTF to a version of SGML.2. \TAGWRITE\EXAMPLES\SGMLDEMO.VEN
This illustrates transformation from Ventura to SGML.3. \TAGWRITE\EXAMPLES\TECHMAN.RTF
This is a simple tutorial template to help you get started. |
|--|

Contents of the TagWrite Directories

The install procedure creates the following directories on your hard drive under the COREL50 directory: \TAGWRITE, \TAGWRITE\STYLES, \TAGWRITE\TEMPLATES, \TAGWRITE\TOOLS, \TAGWRITE\EXAMPLES and \TAGWRITE\MCONVERT.

- \TAGWRITE is the directory which contains all of the TagWrite system files including the Templates that govern your application.
Caution: Do not erase or alter any files in the \TAGWRITE directory. To add or remove files from \TAGWRITE you must use the TagWrite Copy utility which is explained later in this manual. If you are reinstalling TagWrite, your Templates will not be erased.
- \TAGWRITE\STYLES subdirectory is reserved to contain TagWrite's special WordPerfect or RTF style header. TagWrite is delivered with a default, "Normal" style header for WordPerfect and RTF.
Custom Style Headers can be prepared if your version of TagWrite supports the Styles Module. For technical details, see *Preparing a Styles Application* supplied with the Styles Module.
- \TAGWRITE\TEMPLATES subdirectory is the default directory for your own TagWrite template files.
- \TAGWRITE\TOOLS subdirectory contains special files used by a Template Designer including the tool to make the Style

Header. Its contents are explained in Appendix C of the *Template Designer's Guide*.

- \TAGWRITE\EXAMPLES is a convenient place for you to keep your text files. However, you are *not* required to keep your files in \TAGWRITE\EXAMPLES. You may keep your text files in any directory on any drive. After you are fully familiar with the TagWrite product, you will decide how to manage your text files.

The directory contains sample files and you will need to use \TAGWRITE\EXAMPLES for the learning portion of this manual.

- \TAGWRITE\MCONVERT directory contains files for the Corel "Style Matching" filter.

Getting Started Tutorial

The Getting Started tutorial guides you through the TagWrite Windows Menus and Dialog Boxes. The purpose of this Chapter is to step through the process of tagging a document. Do not skip this exercise.

We use the “Techman” sample application as a reference point. The “Techman” application must be installed on your computer. See the previous chapter for instructions.

To get maximum benefit, you must do this tutorial on the computer. The whole process should take about an hour.

Overview

Here is a guide to the functions of the main TagWrite menus and dialog boxes. Some subjects are discussed in the cited chapters, the others are discussed in this chapter.

- Start and Exit TagWrite
- Choose Template
- Select Files To Tag

From “Select Files To Tag”, you can also make a choice automatically to save a word processed file as straight ASCII after it has been processed by TagWrite.

ASCII Input Options allows you to select an ASCII input text file, process it with TagWrite, and save it in Microsoft Word

RTF or WordPerfect conventional format or Styles. (See Chapter 6).

- Find Template on your drive
- Set New Default Directory for Templates
- Access the TagWrite Template Editor
- Storage of Templates and the .IDX file
- The Template Manager allows you to copy, rename, and delete Templates. The Template Manager, located under File on the Menu Bar, is discussed in Chapter 4.

Starting and Exiting TagWrite

Start

► To start TagWrite:

1. Start Corel Ventura Publisher.
2. Choose TagWrite from the Ventura Tools menu. It appears in the flyout when you click on the Extensions item.

Exit

► To exit TagWrite:

1. Click on File, Exit TagWrite or press the letter X. You will be returned to the Windows Program Manager.
2. To continue this tutorial, startup TagWrite again if you practiced how to exit.

Choose Template

A Template is a TagWrite application. Each Template is specially designed for a specific application. To use TagWrite, you must choose the correct Template for your application.

The "Choose Template" dialog box displays all of the Template applications on the current drive and directory. Unless you specify a different directory, TagWrite, by default, will look at the \TAGWRITE directory on the current drive.

Template Name

Template names are up to nineteen characters long. They often take the form of:

Techman.WP -> TAG

or

Techman.RTF —> TAG

where "Techman" is the name of the application. WP tells you that this Template is written for WordPerfect files, or .RTF tells you that the Template is written for a Microsoft Word file saved in Rich Text Format. The arrow —> indicates that the file originally saved as WordPerfect or RTF

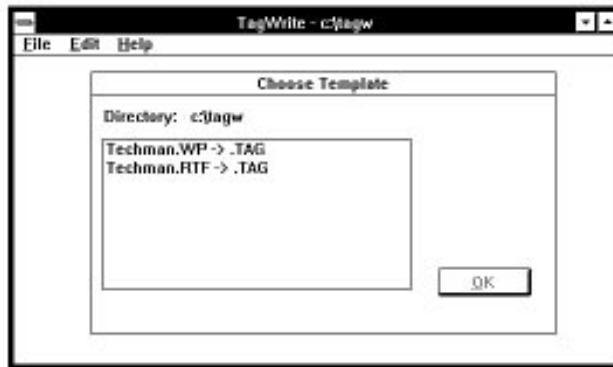
is going to be transformed into a TAG file containing the correct tags for the application.

Note: Templates can have *any* name selected by the Template designer; therefore, the appearance of Template names on your screen may be different from the example above.

Template names cannot be edited from within the Choose Template dialog box. See the TagWrite *Template Designer's Guide* if you must change the name of the Template.

Function of Choose Template

The "Choose Template" dialog box allows you to select a Template for tagging a text file.



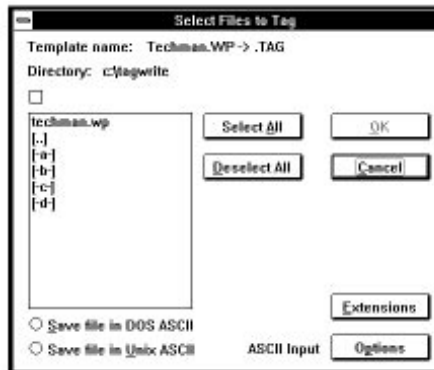
If you are using a Microsoft Word or other RTF Product, select TECHMAN.RTF. If you are using a WordPerfect Product, select TECHMAN.WP

► To choose a Template:

- Highlight the name and either double click on the name or press the OK button. The "Select Files To Tag" dialog box will appear as detailed below.

Select Files To Tag

The Select Files To Tag dialog box is shown below:



► **For the purpose of this tutorial:**

1. Select the TECHMAN.WP file and press OK to start the tagging process. Clicking on Cancel will return you to the TagWrite Main Window and the Choose Template Dialog Box.

A progress meter appears on screen. The progress meter indicates that TagWrite is processing the file.

Note: You can click on the Cancel button to stop the tagging process.

2. When tagging is finished, Minimize or exit TagWrite, and open your word processor.
3. Check the .TAG file. Notice that the .TAG file has all of the original text, and also carries a large number of tagnames.

TagWrite has automatically tagged the TECHMAN file. This concludes the tagging process.

For your further information, the various functions of this dialog box are:

List of Files

The dialog box includes a list box on the left which displays the files in the current directory whose filename extensions match the extensions which have been assigned to the Template. Chapter 5 “Working With Word Processed Files” explains the TagWrite file management system in detail.

Select one, two or more Files

TagWrite can process one file at a time or batch process many files. Files are selected by clicking on one or more files with the mouse, or using the “Select All” command button. You can scroll up and down the list to browse and choose individual files.

Using standard Windows techniques, the list box also allows you to search for files in another drive and/or directory.

Select All/Deselect All

Clicking on the Select All command button selects all files displayed in the list box. Clicking on Deselect All cancels the selection of all files in the list box. Clicking on OK will initiate the tagging process for the selected files.

Optional features of the “Select Files To Tag” Dialog Box

The following sections explain options available to control and modify the tagging process. Please read to the end of the chapter.

Extensions

Input File Extension: Each TagWrite Template is designed to display all text files in the current drive and directory that carry the filename extension associated with that Template.

For example, the Techman Template is designed to list all files in the current drive and directory that carry the filename extension .RTF or .WP depending on which Template you choose.

You can override the default filename extension associated with a Template.

Output File Extension: Each TagWrite Template is designed to save the processed file with an output filename extension that is associated with that Template application.

For example, the Techman Template is designed to produce tagged files with the filename extension .TAG

Override The Default Extensions: You can override the default input and output filename extension associated with a Template as follows:

1. Click on Extensions. Two text boxes appear that allow you to change the input filename extension and the output filename extension.
2. Enter a different Input filename extension if you are seeking to tag files carrying the new extension.
3. Enter a different Output filename extension if you want the tagged file to carry a name different from the default.

Save File In ASCII: To produce DOS or Unix ASCII format

The “Save file in ASCII” options allow you to accept a file in Microsoft Word RTF or WordPerfect format, process it with TagWrite, and automatically save the file in straight DOS ASCII or Unix ASCII format.

This feature is important if you are tagging a Microsoft Word RTF or WordPerfect file into Standard Generalized Markup Language (SGML) or if you are trying to transport the tagged file from word processing format to some other environment which accepts or stores the document in tagged ASCII.

► To save in ASCII:

At the bottom of the “Select Files To Tag” dialog box are two buttons titled “Save file in DOS ASCII” and “Save file in Unix ASCII”. If you select one of these options, all files processed will be saved in ASCII until you undo the selection or exit the dialog box. If you do not make a selection, then the files will be saved in Microsoft Word (RTF) or WordPerfect format.

The Save in ASCII feature preserves all ASCII alphabetic and numeric characters, the space, tab and other characters in the lower ASCII set from 00 to 7F. The text of footnotes is preserved in line.

All other information, including binary or graphic information, is stripped out of the file. Certain information that is generated internally within the word processor will not appear in the ASCII file.

Save in DOS ASCII: preserves the carriage return/line feed combination for subsequent use on computers running the DOS operating system.

Save as Unix ASCII: preserves only the line feed for files that will be transported to computers using the Unix operating system.

See Chapter 6, “Working With ASCII” for additional details.

ASCII Input Options

TagWrite allows you to accept an ASCII input file and save the file process by TagWrite as:

- WordPerfect
- RTF for Microsoft Word, Ami Pro and other products.
- ASCII

Save As Default Microsoft Word or WordPerfect Format: By clicking the ASCII Input Options, TagWrite provides an option to save an ASCII input file as a default WordPerfect or RTF file. This file can be imported into the word processor.

Save With A Specific Style: TagWrite can accept a straight ASCII file and transform it into a Microsoft RTF or WordPerfect file formatted with Styles. TagWrite “pastes” a valid word processor stylesheet onto the untagged ASCII file. When the file processed by TagWrite “With Styles” is brought into your word processor, the file will display fully formatted.

To use this feature, you must have a correct TagWrite Styles Application installed on your computer, and you must make the correct selection under the ASCII Input Options dialog.

This is a powerful use of TagWrite which can vastly simplify document handling. The most common application is to accept an ASCII file tagged with Standard Generalized Markup Language (SGML) or other markup, untag the document and transform the tags to WordPerfect or Microsoft Word RTF styles.

If your application accepts ASCII files and transforms them into Microsoft RTF or WordPerfect, then please read Chapter 6 for details. The ASCII Input Options dialog is explained there.

Location of the Template on your computer

TagWrite Templates normally are stored in the \TAGWRITE\TEMPLATES directory on your hard disk drive. For speed and ease of use, we recommend that you run TagWrite from the hard disk and that you always keep Templates in the \TAGWRITE directory.

Using the TagWrite Copy utility, Templates can be copied and stored to any drive and any directory. (See the Chapter titled “The Template Manager”).

On a hard drive, you may want to store Templates in special directories to suit your needs. We recommend that you become familiar with TagWrite before moving away from the \TAGWRITE\TEMPLATES directory.

On a floppy drive, Templates normally are stored in the root directory. That is, if you wish to copy and store Templates on a floppy disk, you are not required to have a \TAGWRITE directory on the floppy disk.

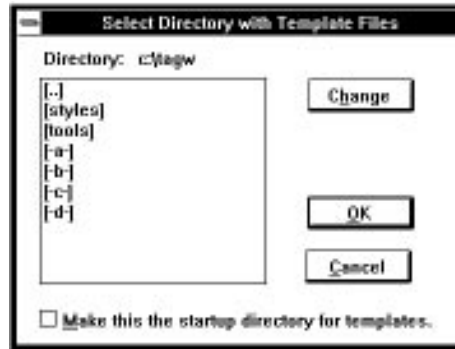
Find Templates

Because Templates can be copied and stored to any drive and any directory, TagWrite provides a Find Template utility. Use “Find Template” to locate and display Templates located on any drive and directory.

Please explore the “Find Template” utility. If you just installed TagWrite, there may not be other Templates right now to find, but you should familiarize yourself with this tool. Follow the steps below.

► To find a Template:

1. Select Find Templates under the TagWrite File menu. The Select Directory With Template Files dialog box appears.



2. Select a new drive and/or directory in the list box of drives and directories.
3. Press the “Change” button and the directory you have selected will become the active directory for Templates.
4. Press “OK” to exit the “Select Directory with Template Files” dialog box. If the path you selected contains one or more TagWrite Templates, the name of the Template or Templates will display in the Choose Template Dialog Box. This directory will remain active during your current work session until you change the directory or until you exit from TagWrite.

Note: If you select a drive and directory that *does not* contain a TagWrite Template and index file, you will receive a message that tells you there are no TagWrite templates in the selected drive and directory.

5. If you get the “No Template...” message, press OK. You will then return to the “Select Directory with Template Files” dialog box.
6. Make the correct drive or directory selection in the list box of drives and directories, and press “Change” followed by “OK”.

or

Press “Cancel” to exit the “Select Directory With Template Files” dialog box. You will then return to the “Choose Template” dialog box.

Setting A New Default Directory

If you want to fix the new directory to be the default start-up directory for TagWrite Templates, then you would click on the box that says “Make this the startup directory for Templates.” The current directory will then become your new default directory, and will automatically display its Template list when you start-up TagWrite.

For the purpose of this tutorial, do **not** attempt to make a new startup directory at this time.

Template Editor

The Template Editor is the workspace where new Templates are created or existing Templates modified. Use of the Template Editor requires study of the TagWrite *Template Designer's Guide*.

► To open a Template:

- Click on Edit at the Menu Bar. A menu drops. You may choose to edit a New Template or Open an existing Template.

Do not modify existing Templates unless you first study the *Template Designer's Guide*.

Information about Template and Index Files

TagWrite Applications are stored with a filename composed of eight random numbers and letters and the suffix .TWW, eg. 0A8C49F2.TWW. The .TWW file is stored in binary format which means that you cannot open the file and try to read or edit it. (TagWrite comes with an easy to use, Template Designer development tool called TEMPLDIR that allows the .TWW files to be displayed in plain English. See the *Template Designer's Guide, Appendix C*.)

An “index file” is always stored in the *same directory* as the .TWW files. The index file has the suffix .IDX. If the Index file is not stored with the .TWW file, then the TagWrite Templates will not display and cannot be copied, renamed, or deleted. Do not delete or edit the .IDX file.

The Template Manager

You must use the TagWrite Template Manager to Copy, Rename, or Delete a Template.

Caution: TagWrite Templates are stored in a special format that requires you use the Template Manager. Do *not* copy templates using the Windows File Manger, DOS Copy, or any other third party file management utility.

Template Manager

Click on File at the Menu Bar at the top of the TagWrite window. A list of choices appears. The first choice is Template Manager.

For convenience, selecting “Template Manager” activates the Copy Template, Rename Template, and Delete Template functions.

Copy A Template

► **To copy a template from the hard drive to the A drive:**

1. Place a blank disk in the A: drive.
2. Click on File at the Menu Bar at the top of the TagWrite Window.
3. Select Copy Template. The Copy Template dialog box appears.

Notice that the Copy Template dialog box has a left and right side. One side will be the source of the file you want to

copy and the other side will be the target. Since you can copy in two directions, which side is the Source and which the Target depends on your intentions.

One side will list existing templates. One side is likely to be blank. The blank side has a button beneath it titled "Open".

4. Click on the "Open" button. The "Select Directory With Template Files" dialog box appears.

This dialog box displays the available disk drives.

By clicking on the "Change" button, you can use standard Windows procedures to select any drive or directory that resides on the new drive.

5. Select the A: drive and click on the OK button. If you forgot to place a disk in the A: drive, you will receive a "Can't read from Drive A" error message. If your disk is blank, a dialog box appears telling you "There are no TagWrite Templates in A:\ (yet). Do you want to use this directory?"
6. Accept the A:\ drive by selecting Yes in the dialog box. The Copy Template dialog box now appears with:

Templates in: A:\

The dialog box will be empty if there are no templates on the A: drive. Notice also that the button beneath the dialog box now says "Change" which allows you to change drive and directory as necessary.

7. Place the mouse pointer on a Template name in the left hand dialog box. The Template name is highlighted and the "Copy" button turns from light grey to black on grey indicating that the copy utility is live.
8. Click on the "Copy" button. The Template you selected automatically copies from the Source to the Target drive.

Note: You may select one, several, or all templates to copy in one session.

As in any standard Windows application, if you try to copy a template from one directory to the other, and the templates have the same name, you will be asked if you want to overwrite the existing template.

Caution: As in all computer applications, if you decide to overwrite an existing file, the old file is not recoverable.

Renaming Templates

► To rename a Template:

1. Click on “File” at the menu bar at the top of the TagWrite window. Select “Rename Template”. The “Rename Template” dialog box appears.
2. Select a Template to rename. A dialog box appears asking you to fill in the new name of the template.
3. Type the new name and click on “Yes”. You are returned to the “Rename Template” dialog box and the new name has replaced the old name.
4. Click on “Quit” to return to the “Choose Template” dialog box.

Caution: If you decide to rename, then the old name disappears from the list of templates. Rename is not the same as the Windows “Save As” function. “Save As” creates a new template with a new name and also preserves the old template and old name. You can “Save As” through the TagWrite Template Editor. See the *Template Designer’s Guide*.

Delete Template

Delete is permanent. If you plan to delete a template from your hard drive, and you think you might need it someday, then first use the TagWrite Copy utility and copy the template to a floppy disk for storage.

To delete a Template:

1. Click on File and select Delete Template. The Delete Template dialog box appears.
2. Click on the Template name you want to delete. A dialog box appears asking if you want to delete the selected template.
3. Click on Yes. The file is deleted.

Caution: Like the Windows File Manager, if you select “Yes”, the file is instantly deleted without further prompt or query from the computer. The deletion is permanent. You may be able to recover the file using an undelete utility in DOS 5.0, Norton Utilities, PC Tools or other software that support undelete. If you need to “undelete”, then exit immediately from TagWrite. Do not save any files until you have run your undelete utility.

Working with word-processed Text files

Overview

The first section of this chapter explains why a TagWrite template application *must* match the text files for that application. TagWrite requires that:

- You use the correct word processor with a specific TagWrite application.
- You know the “keyentry convention” for your document. For a particular TagWrite Template, how should the original document be typed (keyentered) in your word processor? Should you use “styles” to keyenter your document?

The second section of this chapter explains how filename extensions are used to insure that there is a correct match between the template application and the text files.

The third section explains how TagWrite handles Pictures in a text file.

For detailed information on working with straight ASCII files, read the next chapter titled “Working With ASCII”.

Use The Correct Word Processor With a Template

Each Template is designed to work with *either* WordPerfect or Microsoft Word saved as RTF.

You cannot run a WordPerfect file with a Template built for Microsoft RTF. You cannot run a Microsoft RTF file with a Template built for WordPerfect. If your word processor and Template are mismatched, you will receive an error message after you press OK.

Document is wrong type for {Templatename}

In most work settings, the match between a TagWrite application and the word processor file format will be evident.

If you need to run the same TagWrite application on both WordPerfect and RTF files, then you need two different TagWrite Templates, one designed for each word processor. See the TagWrite *Template Designer's Guide*

Templates Support A Specific Document Type

A TagWrite application Template is matched to a document type.

Note: If TagWrite runs very slowly (slower than three seconds per page), then your TagWrite application may not be matched to the document type. Either you are using the wrong template for the document or the document is not correctly keyentered for the template.

A "document type" is a document with a specific, defined structure. The "document type definition" (DTD) is the formal declaration of the components of a specific document.

There is no limit to the kind of document types that can occur in the publishing world. One document type may be a technical manual and have a chapter head, a chapter title and eight levels of headings. Another document type may be a data report composed of a series of tables with a table number, table title, and many rows of information. The variety of possible document types is endless.

The Key Entry (typing) Convention For a Document Type

Depending upon the Template that has been built for your application, you will key enter the document using:

- WordPerfect or Microsoft Word Styles if your application uses word processing styles.
- Conventional key entry (using tab, indent, center and so forth).
- A combination of Styles and Conventional key entry.

You must use the key entry convention that is matched to the Template application.

Match of the Template and the Key Entry Convention

For each TagWrite application, there is a keyentry (typing) convention.

The “convention” is simply an agreed upon way to keyenter text that will be processed by a specific TagWrite application.

There is **not** one standard for key entry. You must know the keyentry convention for your Application.

Styles

If your TagWrite application supports Styles, then the WordPerfect or Microsoft Word style sheet for your document must be installed with your word processing software. You will then style each paragraph.

Usually the WordPerfect or Microsoft Word stylesheet will be prepared by the Template designer and provided for you.

Conventional Typing

If your keyentry instructions tell you to use conventional typing, without styles, then you must know things like:

- Headings — are they centered?
- Text paragraphs — are they entered flush left or tabbed?
- Paragraph headings — are they flush left? Tabbed?

The TagWrite system makes it possible for you to type your document in a very simple format. Using conventional key entry (without styles), a document usually can be created with most text entered flush left. A well designed application will generally not require more than the use of tab, center or indent and the use of character formatting (bold, italic, underline) wherever necessary.

Caution: It is important to repeat that a key entry convention is matched with each TagWrite application. TagWrite relies on the fact that you will observe the key entry conventions prescribed for a specific TagWrite application.

If you are a TagWrite enduser, the person who designed your Template application should provide you with a set of typing instructions or rules on how to enter each structured document into your word processor. If you designed the Template yourself, then you will know how the document should be entered.

If you do not know how to keyenter the document, ask the person who designed your TagWrite template application.

Controlling The Match—Template and Word-processor

TagWrite uses the filename extension as a mechanism to help insure that you match the word processed file with the correct Template application.

The Filename Extension

Each TagWrite Template is assigned to look for files with a specific filename extension. When you select a Template at the “Choose Template” dialog box, the “Select Files To Tag” dialog box automatically displays only those text files in your directory that carry the correct filename extension.

The files displayed are listed only if they carry the filename extension that is related to the active Template.

Save Your Text Files With The Correct Extension

When working in the word processor, save your text files with the filename extension that is associated with its Template.

Caution: If you do not give your input file the correct extension, it will not display automatically in the “Select files to tag” dialog box.

You can always use the “Extensions” button to force the display of any file.

How To Use the Extensions Button

We suggest that you keep file management problems to a minimum. In general, give your text files the filename extension assigned to your application. However, as you become more experienced with TagWrite, you may want to run a Template on files that carry a filename extension that differs from the default.

In that case, you must override the Template’s default input filename extension so that your new files will display in the “Select files to tag” dialog box.

► To override the Template’s default filename extension:

1. Press the “Extensions” button in the “Select Files To Tag” dialog box. Two new text boxes appear above the “Extensions” button.
2. Place the cursor in the “Input File Extension” text box,
3. Delete the current filename extension.
4. Type any input file extension from zero to three characters. Do NOT include the period in the extension name. TagWrite “beeps” if you try to type the period.

Input file extension:

Output file extension:

All text files in your current directory carrying the *new* filename extension will be listed in the “Select Files to Tag” dialog box

5. The filename extension given to the tagged, output file can also be changed by editing the contents of the “Output File Extensions” text box. Do NOT include the period in the output extension name.
6. Click on the “Extensions” button to toggle back and forth between the extension names displayed and no extension names displayed.

Note: The changes to the input and output extension names are temporary and do not “stick” after you exit from the “Select files to tag” dialog box.

Pictures

A picture or “graphic” is a binary file format that instructs software how to display an image. The picture is not truly a part of the “text” file.

WordPerfect 5.0 and 5.1 do not save pictures with the text file. In WordPerfect 5.0 and 5.1 pictures are saved as separate graphic files; therefore, the presence of pictures is not an issue with TagWrite.

Microsoft Word is able to save the picture as part of the text file. If a picture is saved with the RTF file, TagWrite will pass through (preserve) all graphic data to the output document. The pass through is accomplished with TagWrite internal code. No special modification of the TagWrite application is required.

Save File as ASCII — Picture Stripped Out

If you choose to save the TagWrite output file as ASCII, TagWrite strips out all binary data including pictures. Binary information is not straight ASCII and technically does not belong in a file saved as ASCII. This is standard procedure for most document processing software.

Keeping Track Of Pictures Under Windows

In most structured documents, pictures are stored as separate files and are imported into the word processor.

Working with Microsoft Word for Windows or WordPerfect For Windows, the cleanest technique to merge pictures is to use Dynamic Data Exchange (DDE) or WordPerfect Link. With a Link or DDE, the picture is stored in a separate file. The DDE contains the name of the file and the path to the file. The DDE automatically will merge the picture file in the position designated by the DDE.

Working With ASCII Files

TagWrite has powerful ability to import, export and transform ASCII. The TagWrite ASCII capability is especially important if you are working with the Standard Generalized Markup Language (SGML), databases, or any other storage and retrieval system that uses an ASCII format.

Definition of ASCII

ASCII stands for “**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange.” ASCII is an international standard for representing alphanumeric and other characters in machine readable format. In ASCII, each upper case letter, lower case letter, number and punctuation character is represented by a number which can be interpreted by any computer.

Overview of ASCII and TagWrite

Without special preparation, TagWrite can accept an input text file in WordPerfect or RTF format and automatically save the output file in straight DOS ASCII or Unix ASCII format.

With a correctly prepared application, TagWrite can accept a straight ASCII file, process it with TagWrite, and produce an output file in WordPerfect or RTF format including Styles.

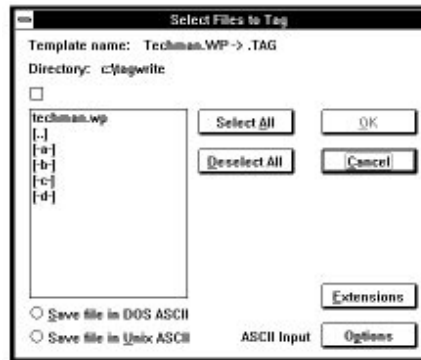
Save Word Processed Input File As An ASCII Output File

If your input file is in WordPerfect or RTF format, you can instruct TagWrite to process the input file (tag or untag it) in its native format and automatically generate an output file as straight ASCII.

This feature is important if you are tagging a Microsoft Word RTF or WordPerfect file into Standard Generalized Markup Language (SGML) or if you are trying to transport the tagged file from word processing format to some other environment which accepts or stores the document in tagged ASCII.

To Save In ASCII

At the bottom of the “Select Files To Tag” dialog box are two buttons titled “Save file in DOS ASCII” and “Save file in Unix ASCII”.



If you select one of these options, all files processed will be saved in ASCII until you undo the selection or exit the dialog box. If you do not make a selection, then the files will be saved in Microsoft Word (RTF) or WordPerfect format.

The Result Of Save As ASCII In DOS Or UNIX Format

From the WordPerfect or RTF file, “Save as ASCII” *preserves* the occurrence of any of the first 128 characters of the ASCII character set including tab, carriage return/linefeed, and alphanumeric and punctuation characters.

Footnote text is preserved at the place in the document where the footnote originally was entered. A TagWrite application that tags footnotes and saves as ASCII will result in tagged footnote text which is contiguous with the footnote’s main text. Footnote text does **not** appear at the bottom of the ASCII document.

Selecting “Save in DOS ASCII” preserves the carriage return/linefeed combination for subsequent use on computers running the DOS operating system.

Selecting “Save as Unix ASCII” preserves only the line feed for files that will be transported to computers using the Unix operating system.

Header, footer, and certain other word processing features that are generated only from within WordPerfect or Microsoft Word are *not* preserved.

All other file format information, including word processing format commands, DDE and Link information, pictures, and other binary information are stripped out of the file.

Accept The Input File in ASCII Format

TagWrite can accept an input file in ASCII format. The principal uses of this feature are:

- To untag a tagged document that exists in ASCII.
- To manipulate files stored in ASCII like database or scanned files that require new formatting in WordPerfect, RTF, SGML or some other tagged format.

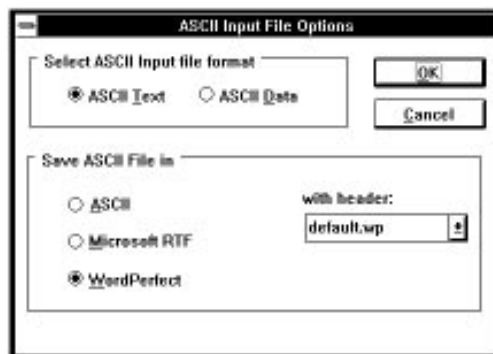
With a correct TagWrite Template application, TagWrite can process an ASCII input file with the following options:

- Process the input file in ASCII and save it as ASCII perhaps adding or removing tags or other information.
- Process the input file in ASCII and save it in WordPerfect or RTF format. Your TagWrite application can be designed to transform ASCII into a document that is formatted with WordPerfect or RTF styles and/or conventional formatting such as tab, indent, center, bold, italic, underline and most other word processing formats including footnotes.

To save an ASCII file in WordPerfect or RTF format, you must use the ASCII Input File Options dialog box.

The ASCII Input File Options Dialog Box

To view the “ASCII Input Options” dialog box, press “Options” in the “Select Files To Tag” dialog box.



The ASCII Input File Options dialog box contains the following components:

Select ASCII Input File Format

At the top of the dialog box, under the heading “Select ASCII Input File Format” are two option buttons named “ASCII Text” and “ASCII Data.”

The difference between “ASCII Text” and “ASCII Data” is the way in which hard returns (carriage return/line feed) are managed in the conversion from ASCII to WordPerfect or RTF.

ASCII Text

Use the “ASCII Text” option if your input ASCII file is formatted with hard returns at the end of every line to simulate a printed page. TagWrite will attempt to anticipate the end of an actual paragraph by removing extra hard returns and converting the extra returns to a space. The end of paragraph hard return is preserved. If the conversion is successful, text will word wrap when imported into your WordPerfect or RTF word processor.

Generally, a text file such as a SGML file will be treated as “ASCII Text”. This will help insure that extra hard returns are converted to a space for correct display in your word processor.

Caution: The ASCII Text conversion is not always perfect in TagWrite or other ASCII conversion utilities. You may find extra hard line breaks in your converted document. On rare occasion, two paragraphs that should be separated by a hard return will be separated only by a space. That is, the paragraphs will be run together. Run on paragraphs can be avoided if paragraphs are separated with two hard returns in the original ASCII file.

ASCII Data

Use the “ASCII Data” option if your input ASCII file contains long, unbroken streams of text or information from spreadsheets or databases saved as ASCII. When “ASCII Data” is selected, TagWrite will convert all hard ASCII hard returns into word processor hard returns.

Save ASCII File In

Below the “ASCII Text” and “ASCII Data” option buttons are three option buttons named “ASCII,” “Microsoft RTF” and “WordPerfect,” and a list box named “With Header.”

The option buttons display the file format which the Template is designed for. The list box allows you to select a word processor header file which will be pasted on to the ASCII file to convert it to RTF or WordPerfect format.

You can save an ASCII file in one of three ways:

- Save ASCII as ASCII
- Save ASCII as RTF
- Save ASCII as WordPerfect

Save ASCII as ASCII (the default mode)

► To tag or untag an ASCII file and produce an ASCII output file:

1. Choose an ASCII Template from the Choose Template dialog box. The Select Files to tag dialog box appears.
2. Select the ASCII file or files that you wish to process in the Select Files to Tag dialog box.

Note: It is not necessary to use the ASCII Input File Options dialog box to save an ASCII file in ASCII format.

Save ASCII as WordPerfect

► To tag or untag an ASCII file and produce a WordPerfect output file:

1. In the Choose Template dialog box, select a Template which is designed for WordPerfect. The Select Files to tag dialog box appears.

Note: You must choose a WordPerfect Template to save an ASCII file in WordPerfect format.

2. Select Options in the Select Files to Tag dialog box. The ASCII Input File Options dialog box appears.
3. Select either the ASCII Text or ASCII Data option button depending on the structure of the input ASCII file. This option is described on page above.
4. Make sure that you have selected a WordPerfect Template: the WordPerfect option button will be blackened, and default.wp will appear in the With Header drop-down list box.
5. Select OK to exit the dialog box and return to the Select Files to Tag dialog box.
6. Select the appropriate files and press OK. The result will be to produce an output file that contains a default WordPerfect header and the text of the document.

You can bring the output file produced by TagWrite into WordPerfect. The file will display as a “native” file using default formats and whatever other formats (like bold, indent, italic) that have been used in your TagWrite Application Template.

Save ASCII as RTF

► To tag or untag an ASCII file and produce an RTF output file:

1. In the Choose Template dialog box, select a Template which is designed for RTF. The Select Files to tag dialog box appears.

Note: You must choose an RTF Template to save an ASCII file in RTF format.

2. Select Options in the Select Files to Tag dialog box. The ASCII Input File Options dialog box appears.
3. Select either the ASCII Text or ASCII Data option button depending on the structure of the input ASCII file. This option is described on page above.

4. Make sure that you have selected an RTF Template: the RTF option button will be blackened, and default.rtf will appear in the With Header drop-down list box.
5. Select OK to exit the dialog box and return to the Select Files to Tag dialog box.
6. Select the appropriate files and press OK. The result will be to produce an output file that contains a default RTF header and the text of the document.

You can bring the output file produced by TagWrite into Microsoft Word. The file will display as a "native" file using default formats and whatever other formats (like bold, indent, italic) that have been used in your TagWrite Application Template.

Save With WordPerfect or RTF Style

Overview

A TagWrite Template application can be written to accept an ASCII input document and produce a WordPerfect or RTF styled output document.

The most common application for this is to untag an ASCII text file and replace the old tags with word processor styles. The styled document is then imported into WordPerfect or Microsoft word formatted and ready for display and printing.

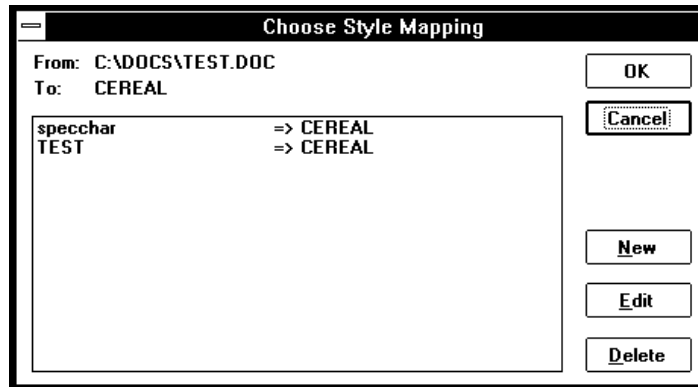
Note: All versions of TagWrite 3.0 can run Style Template Applications. To create a Style Template, your version of TagWrite must support the Styles Module.

How To Use “With Styles”

To use this powerful utility, a correct TagWrite Template must be installed and display at the “Choose Template” dialog box and a “TagWrite Style Header” file must reside in your \TAGWRITE\STYLES directory. Both the Template and the TagWrite Style Header must be prepared by the template designer following the directions in the *Template Designer’s Guide* and “Preparing A Styles Application”.

Style Mapping

When you select “TagWrite with Styles”, the Choose Style Mapping dialog box appears. Use this to choose the style to map the RTF file to.



The functions of the buttons are as follows:

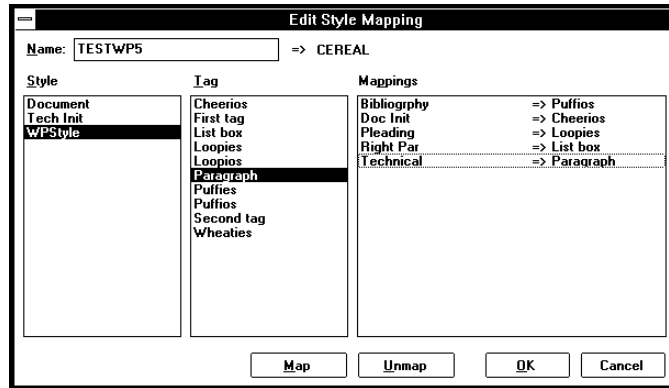
OK - Selects the highlighted mapping. This instructs TagWrite to process the input file using the selected template.

Edit - This calls the Mapping Editor (explained in detail below) with the selected mapping as a starting point.

New - This calls up an empty Mapping Editor.

Cancel - This backs out of the file import one level.

When you select Edit, the Edit Style Mapping dialog box appears, as shown below:



It's functions include:

Name - This field allows you to name the input document stylesheet.

Style - This list shows the style names from the input document.

Tag - This list shows tags from the current (if New) or named (if Edit) Ventura stylesheet.

Mappings - This list shows user-selected mappings from Styles to Tags.

Map - This button enters the highlighted Style and Tag into the Mapping list.

Unmap - This button removes the highlighted Style->Tag Mapping.

OK - This saves the selected mappings in the form of a TagWrite template. The template name will be the concatenation of the "From" edit contents, "->" and the Ventura stylesheet name. The template is made by copying a master template file from a standard directory and replacing the prototype's style rule with an instance for each selected mapping.

Cancel - Cancels the mapping edit, with confirmation if changes have been made.

Mappings are made by selecting an item from both the Style and Tags list and then either clicking the Map button or double-clicking the second selection. Mappings are removed by either double-clicking the Mappings list item or selecting it and clicking the Unmap button.

When a Style has been mapped, it is removed from the Styles list. Tags, however, are not removed since they must remain available for other Styles. Unmapping a Style->Tag selection moves the Style back to the Styles list.

How to Install the TagWrite Style Header

For the "With Style" application to work, the Template designer must provide you with a special WordPerfect (.WP) or Rich Text Format (.RTF) file.

If you install the TagWrite Style Header manually, then use the Windows File Manager or another file management utility. You copy the

.WP or .RTF style header file into the \TAGWRITE\STYLES directory on your hard drive.

We call this special file the “TagWrite Style Header” because it is suited only for use with the TagWrite Styles Module.

Note: The TagWrite Style Header for WordPerfect or RTF is specially prepared according to directions found in the Styles supplement of the TagWrite *Template Designers Guide*.

When the “TagWrite Style Header” is correctly installed in your \TAGWRITE\STYLE directory, the “with header” drop-down list box will display the RTF or WordPerfect document style options that are available for use with a TagWrite Template application. Click on the arrow at the right of the list box to display the complete list of available style headers.

► **To accept ASCII input and obtain Styled output:**

1. The TagWrite Template for the application must be copied onto your computer using the TagWrite Copy utility.
2. The TagWrite Style Header must be provided for you. It is a special file (with either a .WP or a .RTF filename extension) that **must** be copied into the \TAGWRITE\STYLES subdirectory on your computer. Instructions to install the header are found below under the heading “How to Install The TagWrite Style Header”.

Save ASCII as WordPerfect with Styles

► **To tag or untag an ASCII file and produce a WordPerfect output file formatted with styles:**

1. In the Choose Template dialog box, select a styles application Template which is designed for WordPerfect. The Select Files to tag dialog box appears.

Note: You must choose a WordPerfect Template to save an ASCII file in WordPerfect format.
2. Select Options in the Select Files to Tag dialog box. The ASCII Input File Options dialog box appears.
3. Select either the ASCII Text or ASCII Data option button depending on the structure of the input ASCII file. This option is described on page above.
4. Make sure that you have selected a WordPerfect Template: the WordPerfect option button will be blackened, and default.wp will appear in the With Header drop-down list box.
5. Click on the arrow on the right of the “With Header” list box. A list of one or more style headers appears.
6. Select the TagWrite Style Header in the list box that is appropriate to the Template being used.
7. Select OK to exit the dialog box and return to the Select Files to Tag dialog box.
8. Select the appropriate files and press OK.

The result will be to produce an output file that contains the WordPerfect stylesheet, the text of the document, and the styles that are associated with the text.

You can bring the output file produced by TagWrite into WordPerfect. It should display normally formatted according to the specifications in the stylesheet.

Save ASCII as RTF with Styles

► To tag or untag an ASCII file and produce a RTF output file formatted with styles:

1. In the Choose Template dialog box, select a styles application Template which is designed for RTF. The Select Files to tag dialog box appears.
Note: You must choose a RTF Template to save an ASCII file in RTF format.
2. Select Options in the Select Files to Tag dialog box. The ASCII Input File Options dialog box appears.
3. Select either the ASCII Text or ASCII Data option button depending on the structure of the input ASCII file. This option is described on page above.
4. Make sure that you have selected a RTF Template: the RTF option button will be blackened, and default.rtf will appear in the With Header drop-down list box.
5. Click on the arrow on the right of the "With Header" list box. A list of one or more style headers appears.
6. Select the TagWrite Style Header in the list box that is appropriate to the Template being used.
7. Select OK to exit the dialog box and return to the Select Files to Tag dialog box.
8. Select the appropriate files and press OK.

The result will be to produce an output file that contains the RTF stylesheet, the text of the document, and the styles that are associated with the text.

You can bring the output file produced by TagWrite into RTF. It should display normally formatted according to the specifications in the stylesheet.

Raw ASCII Text From Scanners

For raw ASCII text with no embedded tags, such as text that comes from a scanner, a TagWrite application can probably be prepared that substantially formats the file in WordPerfect or RTF. This kind of application is based on ASCII text string searches and searches for ASCII characters that might give the document an intelligible structure. Experience with the TagWrite *Template Designer's Guide* is required to build this kind of application.

Problems and Error Messages

This chapter first describes some common problems. The second section lists all error messages that can appear.

Problems

TagWrite runs very slowly (slower than three seconds per page)

- a. You may be running a Template application with the wrong text files. Each Template is written for a particular document type with its own key entry convention or known structure. If you run a TagWrite application Template designed for one type of document on the wrong files, then TagWrite cannot interpret the file and the tagging or untagging process will be very slow. Read the “Getting Started Tutorial” in this manual for a full explanation.

Solution: Insure that the text files for your application carry the correct filename extension. TagWrite is designed so that by default, each Template associates the Template with the specific text filename extensions.

Solution: Insure that you have keyentered the document correctly. See Chapter 5.

- b. The Template may be correct for the text files, but the Template may not be written correctly. Important Template rules may be missing. The [KEEP] rule may be missing.

Solution: The person who designed the Template must do additional work to insure that all the necessary Template rules are present including the correct Escape Rules.

TagWrite functions endlessly with no apparent progress.

- a. If you have a very long document, the progress meter indicating how much of the document has been processed may move slowly. Give the operation a reasonable amount of time before deciding that there is no progress. If the document is short, then the progress meter should move quickly and visibly. If the document is very short and your computer is fast, the progress meter may simply flash on and off the screen.
- b. If there is no progress, the Template may be “looping”. This is an endless process. Cancel the operation. Click on the Cancel button immediately below the progress meter.

Solution: The Template designer must insure that each Element in the Template has the correct “Escape Rules”. TagWrite has the capability to insure that a properly written Template will not loop.

Strange codes appear in the document.

TagWrite tagging applications are designed to insert (embed) tags or codes into the visible, text document. The presence of these tags in the output document indicates that TagWrite is working correctly. See Chapter 1 of this manual for an explanation of “tags”.

No Templates appear in “Choose Template” Dialog Box

Each TagWrite application requires a Template application to be installed. Each Template has a name and the name displays in the “Choose Template” dialog box.

If no Template names are listed then:

- a. No Templates have been installed.
- b. You must change drive and directory at the Choose Template dialog box to locate the Templates on your hard drive.
- c. The Templates were copied into the \TAGWRITE directory using the Windows File Manager or other copy utility. You must use the TagWrite Copy Utility to install Templates. If you do not use the TagWrite Copy Utility, the Template names will not display in the “Choose Template” dialog box.

Error Message: STOP : Document is wrong type for {templatename}

Each Template is designed for **either** RTF or WordPerfect or straight ASCII. If you try to run a word processed file from another word processor or a word processor that does not match the Template, you will receive this error message.

Solution: Name your files clearly to insure that files prepared for TagWrite carry clear identification. Check the chapter titled “Getting Started Tutorial” for an explanation of how TagWrite matches filename extensions with Templates.

Error Message: Microsoft Word Binary is no longer supported OR Document is not an RTF document

TagWrite supports Microsoft Word For Windows and all Microsoft Word products with files saved as “RTF”.

RTF stands for “Rich Text Format”. RTF is the Microsoft standard for exchanging documents.

Solution: Save your files in RTF Format.

Caution: Do **not** save your Microsoft Word files in “Normal” format. “Normal” format, also called “binary”, is different from RTF format. TagWrite works only with Microsoft Word files saved in RTF format.

If you have saved your file as Normal, Microsoft Word requires that you rename the file before you are allowed to save as RTF. See your Microsoft Word product documentation for instructions on how to save as RTF.

Template names do not display

You may receive an error message saying:

Can't read the template index; must be corrupted.

You may not receive any error message depending on the problem.

- a. Go to the \TAGWRITE directory. Determine if there is an .IDX file in the \TAGWRITE directory. Each Template name is stored in the .IDX (index) file in the \TAGWRITE directory. If the .IDX file has been erased or corrupted, the Template names will not display. The .IDX file is critical to the operation of TagWrite.

Solution: The \TAGWRITE directory contains a file named TAGWRITE.BAK which is a backup of the .IDX file. Rename TAGWRITE.BAK to TAGWRITE.IDX. Restart TagWrite from within Windows. See if the Template names appear.

- b. If the .IDX file is present, to track down the problem, begin by using TEMPLDIR in the \TAGWRITE directory to determine if the Template is installed.

<At the C: prompt, copy TEMPLDIR from \TAGWRITE\TOOLS to \TAGWRITE.

<Type TEMPLDIR for on screen instructions or see Appendix C of the *Template Designer's Guide* for simple instructions to use TEMPLDIR.

If the Template name is not listed after you run TEMPLDIR or if the Template name is listed in TEMPLDIR but does not display, then the Template is not installed correctly.

Solution: You must use the TagWrite Copy utility to install the Template.

Note: If other files seem corrupted, you may be experiencing hard disk problems that are affecting your whole system.

List of Error Messages

Can't read the template index; must be corrupted.

Exit TagWrite. Backup TAGWRITE.BAK. Then, copy TAGWRITE.BAK to TAGWRITE.IDX. Restart TagWrite.

Unable to process document; style definition is too long

The style definition material in the header may exceed available memory. This is a hardware limitation dependent upon the available RAM in your computer.

Document is wrong type for template {name}

You are attempting to tag a file created in one document format with a template designed for another. For example, the template is designed for RTF, but the file is in WordPerfect format.

You may not edit this template

Certain application templates released by ZANDAR Corporation are "locked" and cannot be edited in the TagWrite Template Editor. The commercial release of TagWrite does not contain the lock or the key to the lock.

Out of Memory — close another application!

General Windows error message based on the available memory in your computer. There is no solution to this problem except to close other files, applications, or memory resident programs.

Read file Errors:

Error reading file!
Error opening file!
Can't open input file {name}.
Error opening file!
Can't open input file {name}.

TagWrite cannot read the text file. The reason is unknown, but probably is due to corruption of the file.

Write file Errors:

Error creating file!
Error writing file!
Unable to create output file {name}
Error writing output file {name}

TagWrite cannot produce an output file. Reason is unknown. Hard disk may be full. May be other disk problems or system problems.

Error deleting temporary file!

System problem. Try restarting (rebooting) your computer. Run the application again.

Errors related to the document's file format:

Bad beginning style in input file {name}
Bad regular header in document {name}
Bad special header in document {name}
Bad style packet in document {name}
Bad RTF format: No backslash in control word
Bad RTF format: No backslash in hex value
Bad RTF format: No quote in hex value
Bad RTF format: No backslash in control symbol
Unexpected end of file in input file {name}

In each of the cases above, the WordPerfect or RTF file has been corrupted. The file is not in the correct format.

Solution: Return to the word processor, save the file again before running TagWrite.

Bad RTF control word {name} in CODES

The Codes Column in the Template Editor contains the list of Tokens available for use in an application. The Codes Editor displays the RTF control word for a particular Token. This error message tells you that the RTF control word associated with a Token is invalid. You must remedy the problem using the Codes Editor. To find the valid RTF control Word, consult the *Microsoft Word Technical Reference*, Microsoft Press, see the chapter on RTF.

Unknown code {name} run error check on template

This message at run time is critical and fatal. A Token has been used in the Template that is not listed in the Codes Editor display of codes that are valid for the Template. You must identify the invalid code and change it. Often the prob-

lem occurs because of an editing error in the Template Editor. A Token name has accidentally been changed or part of it deleted in the Template Editor.

Microsoft Word Binary is no longer supported

Binary is the “Normal” or default file format selection when saving Microsoft Word. TagWrite reads Microsoft Word files only if they are saved in RTF format.