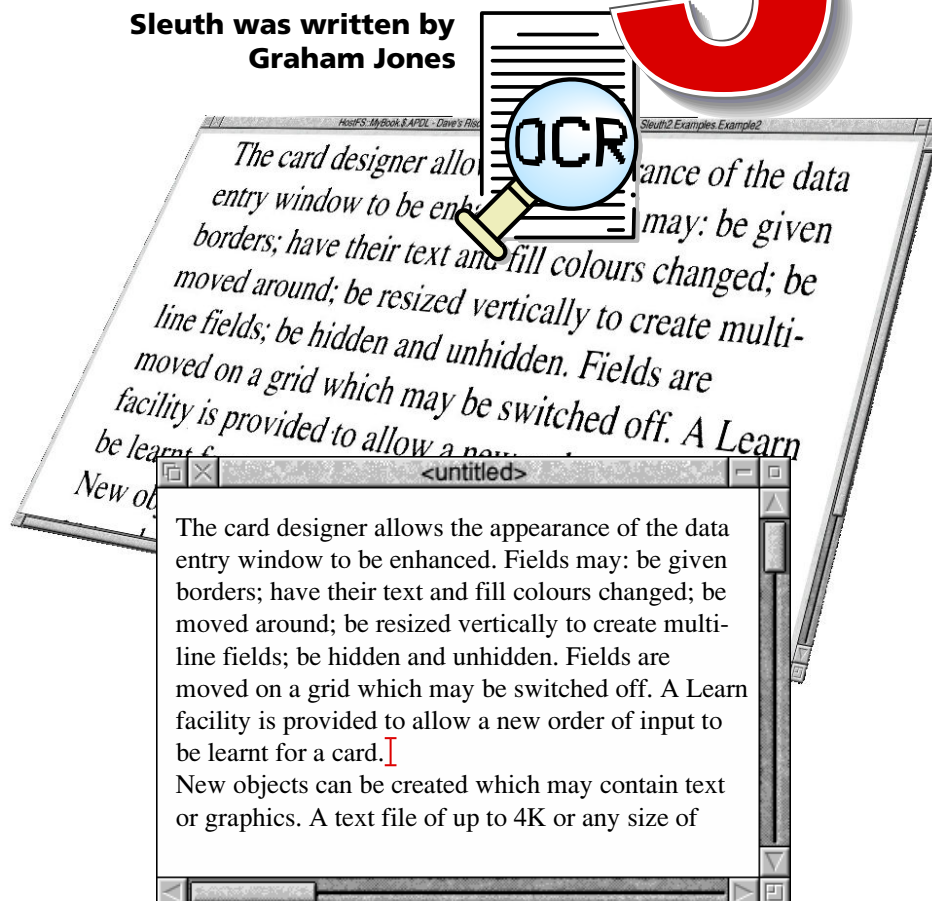


SLEUTH3

Sleuth was written by
Graham Jones



Sleuth 3

Optical Character Recognition for RISC OS

ProAction

3QD
DEVELOPMENTS LTD

APDL

Sleuth 3

© 3QD Developments Ltd 2026
All rights reserved.

No part of this product may be reproduced in whole or part by any means without written permission of the publisher. Unauthorised hiring, renting, lending, public performance or broadcasting of this product or its parts is prohibited.

While every care is taken, the publisher cannot be held responsible for any errors in this product, or for loss of data or consequential effects from the use of this package.

Software and Documentation supplied by
APDL
a part of
3QD Developments Ltd
Single User version downloaded from the free of charge software section of
www.riscos.com

Please note that any free of charge downloads from RISCOS.com are supplied without any technical support package as per the licence agreement.

This document last edited on 29/09/26



If you have found this application and this manual useful please consider making a donation to help cover the costs of cataloguing, documenting, updating and hosting the APDL applications. Scanning the QR code will take you to our donations page.

Contents

Introduction	5
Installing the program	5
Floppy disc version	5
Loading Sleuth	5
Loading a Sprite	5
Greyscale sprites	5
Scanning using TWAIN	5
Configuring Sleuth	6
Output preferences	6
Input and processing preferences	7
The Icon Bar menu	9
The User dictionary	9
Minimise memory	9
Reload image	9
Stop OCR	9
Quit	9
Manipulating the sprite	10
The OCR process	10
Using zones	11
Automatic zone creation	11
Changing the order of zones	11
Manipulating zones	12
Ignore zones	12
Single column zones	12
Saving zones	12
Table zones	12
Zones and TWAIN	12
Guided editing	13
Inserting and deleting paragraph breaks	13
Saving the output	14
Style information	14
Batch processing	14
Getting the best results from Sleuth	15
Important points to remember	15
Problem solving	16
Sleuth limitations	16
How Sleuth works	17
Training	17
Finding lines of text	17
Character recognition	17
Reading a line of text	17
Final steps	18
Why is Sleuth 3 slower than Sleuth 2	18
Appendix 1 - Supported Fonts	19
Appendix 2 - Keyboard shortcuts	21

Copyright notice

Sleuth must not be employed where its use would infringe copyright laws. Most documentation that is under copyright will have a notice to that effect. The publisher will not accept any responsibility for the consequences arising from the use of the software in this manner.

Introduction

OCR is an acronym for Optical Character Recognition. This is a process whereby text is extracted from a scanned image and converted into a plain text file. OCR has been available for some time on other machine platforms and accuracy has been steadily improving. Sleuth 3 can be used with the majority of scanners available for RISC OS computers.

Installing the program

Sleuth 3 is normally supplied in a compressed archive. Run the !INSTALL program to install Sleuth 2 and Sleuth 3 onto your hard drive. This will install just the actual applications. In addition to these there is other material in the compressed archive including various example files and an HTML version of this manual. These are not installed by !INSTALL as not everyone will want them copied to their hard drive. You can either use these directly from the CD, or just copy them to your hard drive by dragging the files or directories containing them in the usual RISC OS way.

Once Sleuth is installed on your hard drive it is ready to use.

Loading Sleuth

To load Sleuth, double-click on the application's icon and it will be installed on the icon bar. If you click Select over the icon bar icon an empty window will be opened. The application now requires a sprite to process.

Loading a sprite

To load a sprite drag the sprite's icon onto the icon bar icon or into the open window. Only one sprite can be loaded at a time. If a further sprite is loaded it will replace the current one. A new sprite cannot be loaded if the package is already converting a sprite. If a sprite file contains more than one sprite only the first will be displayed. Most scanners, especially those using a TWAIN driver (see below), can produce monochrome or greyscale sprites suitable for use with Sleuth.

Sleuth can automatically load sprites that have been compressed using !Squash.

Greyscale sprites

Sleuth will accept greyscale sprites and convert them into monochrome sprites. Using the Input and processing preferences dialogue box, the user can determine the levels of grey converted to white in this process. A Pre-sharpening facility is provided that will generally improve the quality of degraded text during conversion.

Scanning using TWAIN

Sleuth supports TWAIN to allow direct scanning into the package. To use this facility you must have a copy of TWAIN, an appropriate scanner and Scanner Driver. TWAIN and scanner drivers can be downloaded from:

<https://www.davidpilling.com/wiki/index.php/TWAIN>

TWAIN supports the Select and Acquire options on the Scan submenu. Choosing Select opens a dialogue box that allows you to choose the scanner driver source that you require. Choosing Acquire lets you set up the scanner before scanning. Sleuth will only accept monochrome images, so make sure the BW option is selected. Full information about these options is provided with TWAIN.

The Scan and OCR... option opens the same dialogue box as the Acquire option, but the OCR process will start as soon as the image is acquired. This is useful when you want to scan the same area from more than one page. When using the Scan and OCR... option you must do a full scan rather than a preview.

The Save option allows you to save the scanned sprite.

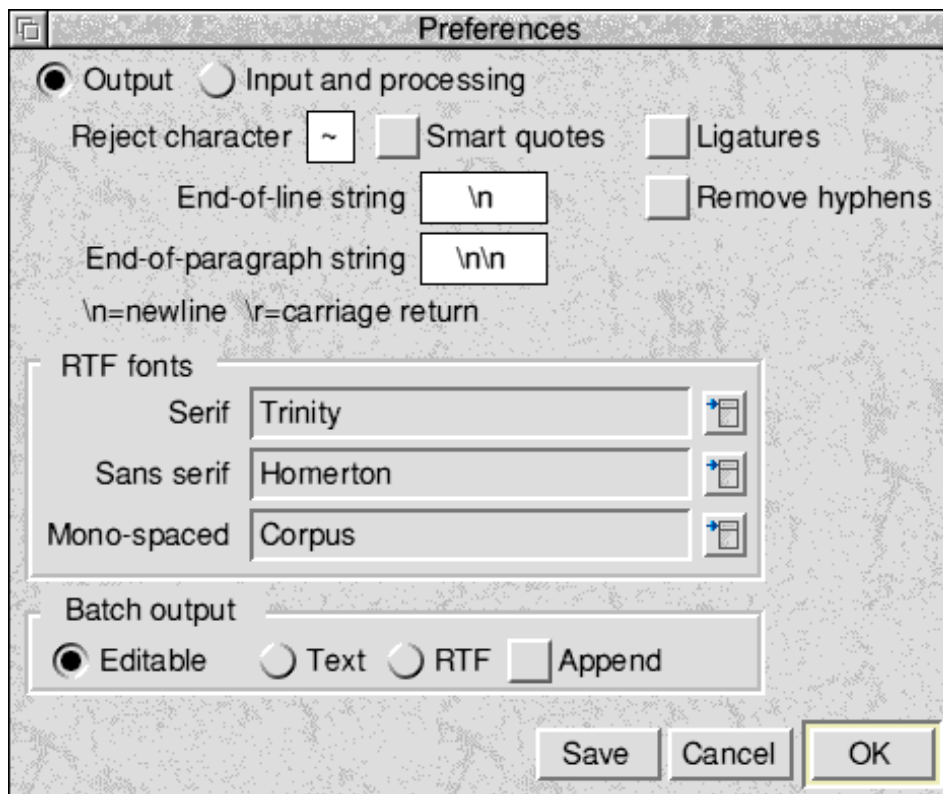
Please note that !TWAIN must be 'seen' by RISC OS before Sleuth can use these options.

Configuring Sleuth

Before beginning the OCR process it is important that Sleuth is configured correctly. The Preferences dialogue box is made up of two separate sets of option accessed through the Output and Input and processing options.

Output preferences

The Reject character is the character used by Sleuth when, in its estimation, there is no equivalent character in the set of characters that it knows, or it is unsure what the correct character is. The Reject character should be set to an infrequently used character. The End-of-line string, End-of-paragraph string, Smart quotes, Ligatures and Remove hyphens options relate to the how the output is saved if it is saved as text.



End-of-line string can be set to any character. A carriage return is represented by \r escape sequence, a line feed by \n. These can be used separately or in combination e.g. \n\r. You could enter a space if you want to reformat the text in another package. These characters will be inserted at the end of each line of text converted by Sleuth.

End-of-paragraph string uses the same escape sequences as the End-of-line option. Sleuth will insert these characters at the end of each paragraph.

Smart quotes will make Sleuth recognise smart quotes " " ' ' if they are present in the image. Otherwise they will be converted into ordinary quotes " and '.

Ligatures replaces the pairs of characters fi and fl with the single character ligatures fi and fl

Remove hyphens will remove hyphens at the end of lines. It will also remove the End-of-line string so that hyphenated words are joined up correctly.

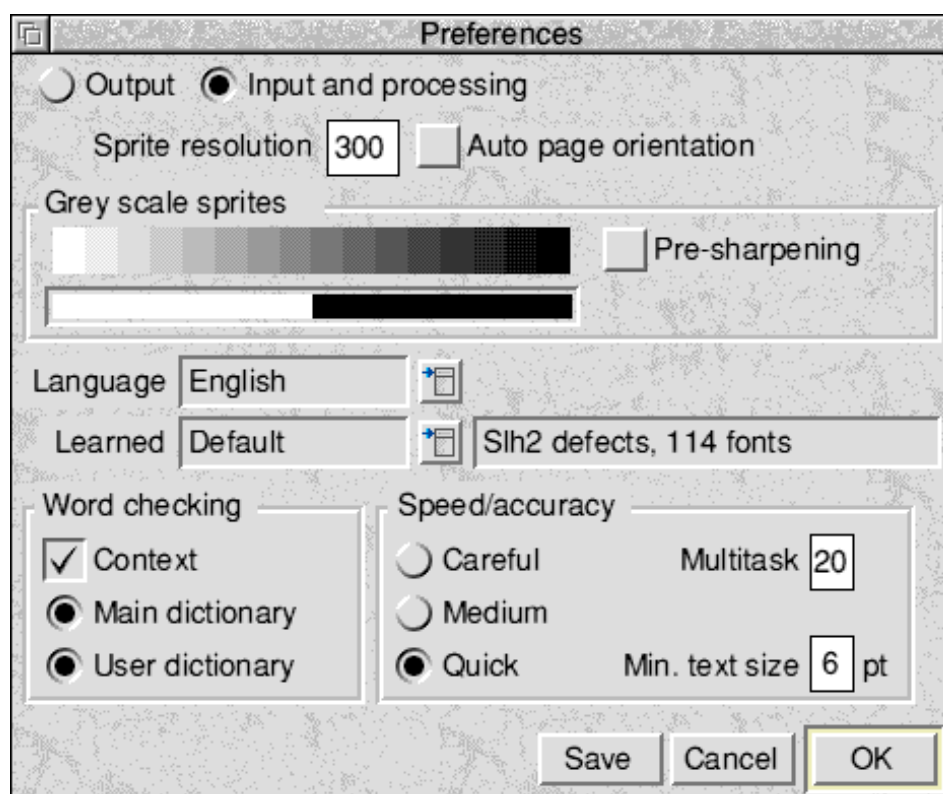
As these settings are read just before the converted text is saved it is possible to change them after the image is converted to text. When saving in RTF format the Remove hyphens process is done automatically and the End-of-line string and End-of-paragraph string are ignored. The settings for Ligatures and Smart quotes will be honoured.

RTF fonts lets the user map fonts present on the system to the generic fonts types (Serif, Sans serif and mono-spaced) recognised by Sleuth. These mappings only apply when the output is saved in RTF format.

When using Sleuth Batch mode you can set what file format is used for the output files. The types are Editable (allows the results of the OCR to be edited at some later date), Text or RTF. A further option, Append, allows you to concatenate the results of OCRing the files in the batch. These options must be set before the batch is processed.

Input and processing preferences

It is very important to set Resolution before starting the OCR process if you have loaded a sprite into Sleuth and you want to output in RTF format. The value refers to the resolution in dpi of the scan you have loaded. This setting will determine what size and spacing text will be given in the RTF file. This setting cannot be altered after the OCR process has begun. Users of TWAIN do not need to set this value as it is read directly from TWAIN.



Auto page orientation makes Sleuth check the sprite file before displaying it to ensure that it has not been scanned sideways or upside down. If it has the program will automatically rotate the sprite by 90, 180 or 270 degrees and then display it. Note that the original image on disc will not be affected. This option is off by default because of the slight time penalty incurred when an image is loaded.

Sleuth will accept greyscale sprites and convert them into monochrome. The user can choose up to which level of grey is converted to white in this process using the Grey scale sprites options. A Pre-sharpening facility is also provided that will generally improve the quality of degraded text during the conversion of the sprite from greyscale to monochrome.

Sleuth has foreign language support built-in. You can choose which language is to be used from the Language menu which selects the resources for that language. You will need to save the preferences and reload the program for the change to take effect.

The Learned menu allows the user to select one of the files that has been provided that best suits the current use of the package. A short description of the learned file is displayed in the icon to the right of the menu icon.

Word checking lets you force Sleuth to limit how it checks whether a word is valid or not. If you choose Context checking is used to distinguish similar characters from one another, for example, an I (an upper case i) and an l (a lower case l) in a sans serif font. Context checking should only be disabled when you are getting a lot of similar errors, where a lower case l (ell) is being interpreted as a 1(one) incorrectly for example. When Context is selected you can choose which dictionaries you want Sleuth to use.

Sleuth 3

The Speed/Accuracy options are fairly self-explanatory. Medium is the default and recommended setting and will give the best overall results in the majority of cases. Quick will improve the speed of output, but generally at the cost of accuracy. Careful will generally improve the accuracy, but at the cost of speed of output.

Multitask lets you define how much time Sleuth uses each time it gets control of the processor. Increasing this value will make Sleuth work faster as it will get a bigger share of processing time but will slow down any other tasks currently running.

Min. text size tells Sleuth what size of text to ignore in the image. You should set this value if you are OCRing documents which include graphics containing text. Sleuth normally ignores text in graphics, but sometimes it will try to convert it. The text is usually smaller than the main body text so by setting this threshold it can be ignored.

Clicking on OK will set the options for the current session. Choosing Save will also set the options and save the settings to a file so that they will be reloaded when the application is next run.

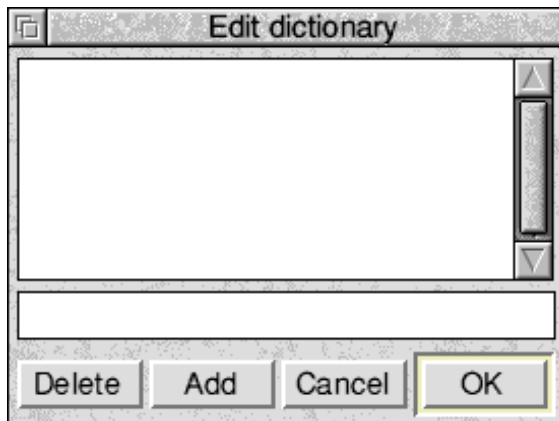
The icon bar menu

These are the other options available from the icon bar menu:

The user dictionary

Sleuth has a user dictionary which lets the user tell Sleuth about words not in the built-in dictionary. This improves the accuracy in reading these words.

To edit the user dictionary choose Edit dictionary... from the icon bar menu which opens the Edit dictionary dialogue. You can enter words one at a time or you can drop a text file onto the dialogue box and the program will read all the words and place them in the dictionary. Any duplicate words will be

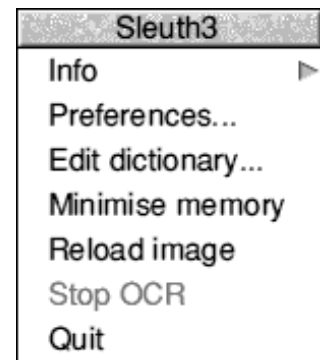


discarded. Entering words already in the built-in dictionary will not cause problems as they will not be added. However having a large user dictionary will slow down the program and use more memory.

Use lower case to add normal words. Adding 'color' will make Sleuth recognise 'color', 'Color' and 'COLOR', whereas 'John' effectively adds 'John' and 'JOHN' but not 'john'. 'SCSI' adds just 'SCSI'.

To delete a word click on it to highlight it and then click on Delete.

Cancel closes the dialogue box without saving the changes made. Choosing OK will save the changes into the user dictionary.



Minimise memory

This reduces the memory required by Sleuth to the minimum. Use it when you want to free some memory but retain Sleuth on the icon bar.

Reload image

Once you have started an OCR you may want to alter some settings described above. You can stop the OCR, change the settings and then click on Reload image to allow you to restart OCRing the image. This only works with the last image you were OCRing.

Stop OCR

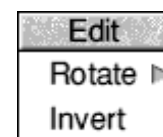
Use this to stop the OCR process. No output will be shown if you do this.

Quit

This quits the program. You will be warned if there is unsaved output from an OCR.

Manipulating the sprite

Before starting to convert a sprite you may need to alter it. The sprite must have black text on a white background and the text has to be upright. Sleuth allows the sprite to be rotated, inverted or zoomed. To rotate or invert the sprite choose the Edit option from the main menu and this submenu will be displayed:



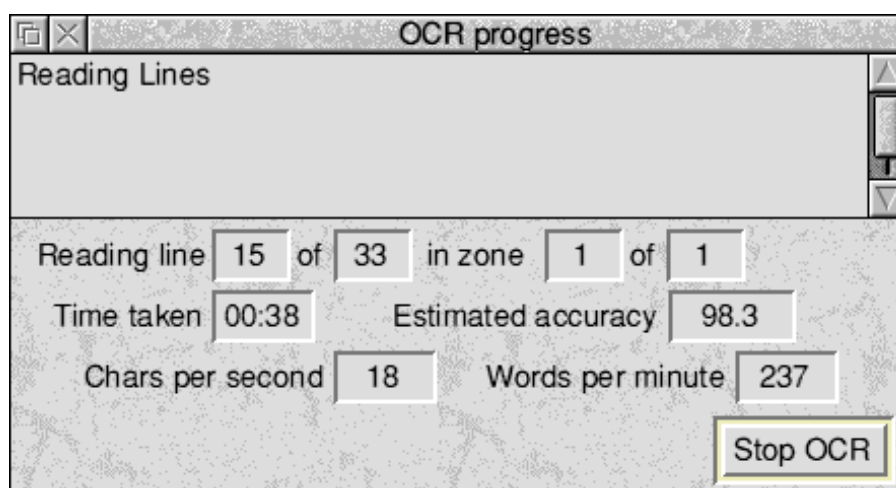
Rotate offers a submenu from which you can choose to rotate the sprite by 90, 180 or 270 degrees. Choosing 90 degrees will rotate the sprite in an anti-clockwise direction; 270 degrees will rotate it in a clockwise direction.

Invert will swap the black and white elements of the sprite.

To zoom the sprite choose Zoom from the main menu and the zoom dialogue box will open.

The OCR process

Sleuth will deal with complex page layouts automatically. A complex page layout might consist of text in columns and/or include graphics. Sleuth should extract text from the scan in a sensible order. You begin the OCR process by clicking Menu over the sprite window and choosing OCR. The sprite window will close and the OCR progress window will open.



This shows information about the progress of the OCR. The top section shows messages telling you what the program is doing. Initially it will show 'Finding lines'. This means that the program is in the process of extracting the text from the scan. Once Sleuth has found all the lines it will begin converting them to text and the window will show 'Reading lines'. If you close the OCR progress window clicking on the icon on the iconbar with Select will re-open it. As the OCRing progresses the statistics are updated showing which line of the total in the current zone is being read, which zone is being processed of the total number plus the time taken, estimated accuracy, words per minute and characters per second. You should take note of the estimated accuracy. While this value is not particularly accurate it will give an indication of how the program is doing. If it is below 95% it is probably worthwhile stopping the OCR, reloading the image and using a zone to OCR a smaller portion. This will save you having to wait to discover if there is a problem with OCRing your image.

If you experience problems or are unhappy with the results you obtain from Sleuth please refer to the problem solving sections on pages 22-25.

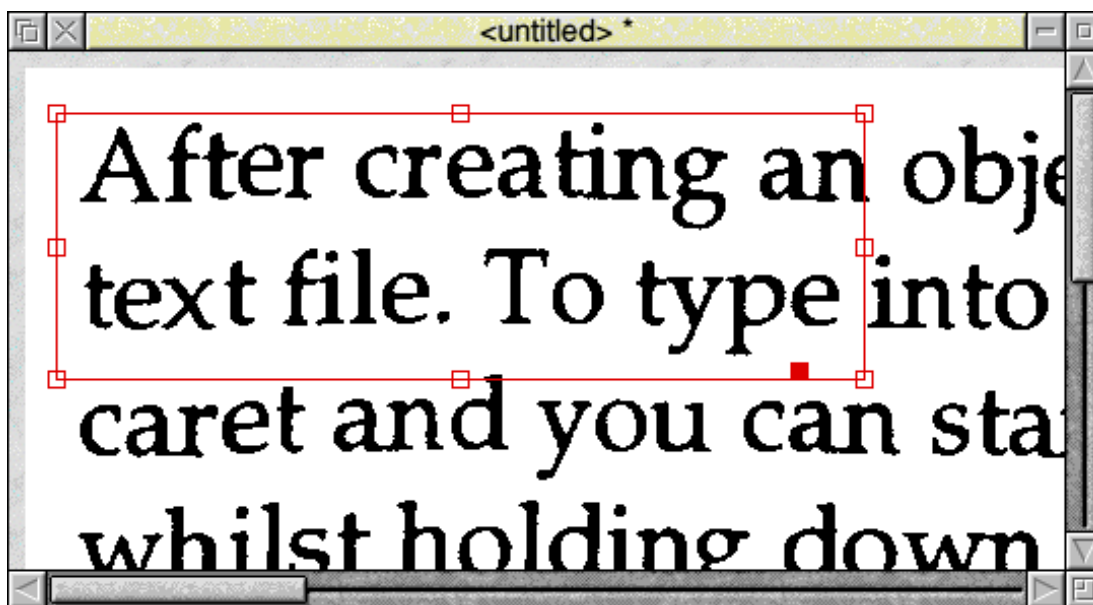
Using zones

Sleuth will be able to deal with many complex page layouts automatically, but some could make it produce text in the wrong order. It is difficult to anticipate the page layouts that could cause problems, but examples are pages where text in columns runs around a graphic, pages containing tables or where text in graphic captions appears in the main text.

To overcome these problems zones can be created. A zone is a user-defined area from which Sleuth will extract text. To create a zone drag with Select over the sprite window. As soon as you start dragging a zone rectangle will be drawn with eight handles and the pointer will be positioned in the bottom right-hand corner handle as shown below:

As you continue to drag the zone will be resized. The window will scroll automatically if necessary. Release Select to finish drawing the zone. The zone can be resized by dragging with Select over one of the handles and moved by dragging with Select inside the zone.

You can create as many zones as you need to enclose the text you want to convert. The order of output from the zones will follow the order in which the zones were created, so text from the second zone created will follow on from the text from the first zone. You cannot draw an ordinary zone on top of another, although zones can overlap. To have overlapping zones, draw a zone and then move it on top of another zone. If any text is common in the overlapping zones it will be output twice.



By default the program will treat the area inside a zone as it would a page without zones, i.e. it will try to find columns or graphics within the text. See the section Single column zones below for details of how to stop this.

Automatic zone creation

Sleuth can find areas of text and create the zones around them for you automatically. Choose the Find zones option to try this. The zones that are created can be manipulated in exactly the same way as zones you draw yourself.

Changing the order of zones

You will notice that all the zones you create are linked by a yellow line that extends from the bottom of one zone to the top of another. These lines show the order of the zones and therefore the order of the text that will be output from them. To change this order, drag over the red (or grey if the zone is not selected) square at the bottom of a zone. A line will be drawn to follow the pointer. Drag the line to the zone that you want to be next in the order. You can drop the line anywhere inside the zone. The program will automatically update the order of the zones, deleting and making links as necessary. This process can be repeated until the order of the zones is correct.

Manipulating zones

Choosing Zone from the main menu will open the menu shown.

This lets you manipulate zones. First a zone must be selected. To select a zone, click in it with Select. An ordinary selected zone is drawn in red. It is possible to have more than one zone selected. To select another click in it with Adjust. All zones can be selected by choosing Select all from the menu. Selected zones can be moved together by dragging inside any zone in the selection.

To deselect a zone click Adjust over it. All zones can be deselected by choosing Clear from the menu. To delete the selected zone(s) choose Delete. The links between zones will automatically be remade after a deletion based on the current order of the zones. Zones will be deleted when a new sprite is imported or the current sprite rotated.

Zone	
Find zones	
Select all	^A
Clear	^Z
Copy	^C
Delete	^X
Ignore	^I
Single column	^S
Table	^T
Save zones	F3 ►

Selected zones can be copied by choosing Copy. Links are automatically made between the old and new zones. Zones are resized to fit inside the grey border inside the sprite window.

Ignore zones

'Ignore Zones' can be created to stop the program trying to OCR a section of a scan. This is useful if a page contains a graphic which contains some text. To create an ignore zone draw an ordinary zone and then select Ignore from the menu. Alternatively you can draw a zone with the SHIFT key held down. Using this method you can draw an Ignore Zone on top of another zone. Ignore Zones contain a cross to distinguish them from other zones.

If you use Ignore Zones on a complex page layout and allow the rest of the image to be automatically processed Sleuth might output the text in the wrong order. This is because the program regards the Ignore Zone as white space. If this occurs it is recommended that if you use ordinary text zones to define where the text is and its order.

Single column zones

Single column zones are provided to let you force the program to treat the area inside a zone as a single column. This is useful if the scanned text is widely spaced or tabulated as often such spacing would make the program output text in the wrong order. To create a single column zone draw an ordinary zone and choose Single column from the menu. Single column zones will be drawn in green when selected.

Saving zones

If you repeat an OCR operation regularly on the same page layout you can save the zones that you use so you don't have to recreate them each time. To do this choose Save zones. The zone file can then be reloaded after another sprite file has been loaded. You need to load the zone file after each sprite as zones are discarded after a sprite is loaded. If you scan using TWAIN the zones will be maintained between scans.

Table zones

Selecting Table on the Zone submenu will tell the program to interpret the data within that zone as a table. The text will be output in the Editor in CSV format. You need to extract this data manually and give a CSV filetype after saving the output as a text file. The program will only recognise simple tables with a single line in each row.

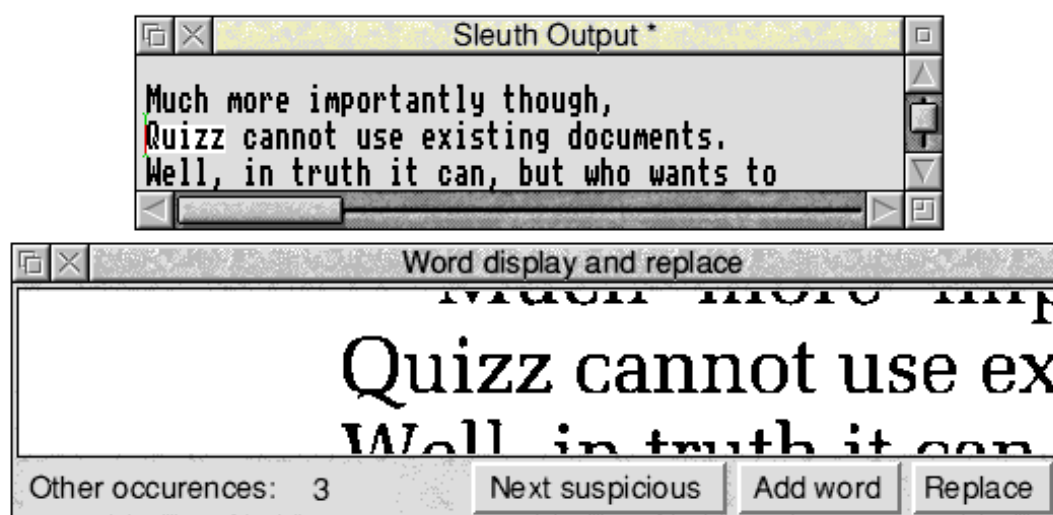
Zones and TWAIN

After using the Preview option, or any other scan from TWAIN, you can create zones to determine which parts of the scan you want to re-scan at a higher resolution for OCRing. Sleuth will automatically calculate what the total area of the scan will be from the zones you create and pass this information to TWAIN when you choose the Scan option. The zones will then be scaled so that they cover the same area on the new scan as they did on the preview scan.

Zone files should not be loaded during the preview stage, unless they were saved at this stage, otherwise they will be scaled after scanning the final image.

Guided editing

As text appears in the text window you will see that some words may be highlighted in yellow. This means that the program thinks they may not be correct. In most cases the word highlighted will have one or more errors. To step through suspicious words place the caret in the text and press F1 or click on Next suspicious. This will move the caret to the first suspicious word and highlight it in white. A word is considered corrected once the caret enters a word and the highlighting is removed. Pressing F1 again will move onto the next word. If you decide to edit a word further down the page pressing F1 will always take you back to the first uncorrected suspicious word. A word may not be highlighted and yet still be wrong. This would only happen when it spell checks correctly.



While editing you can add the currently highlighted word to the user dictionary by pressing F2 or clicking on Add word. A small dialogue box will open for a short time telling you that the word has been added. This will not have any effect on the currently highlighted words.

As you step through dubious words Other occurrences shows how many times this word appears further on in the text. If you find that the same word is read incorrectly or has been incorrectly highlighted as suspicious when it is correct you can correct all occurrences in one go. To do this, edit the first occurrence so it is correct and press F4 or click on Replace. Doing this when the highlighted words are correct will remove highlighting from them all.

While the caret is in the text window the word display window is open showing the word containing the caret from the original sprite.

A list of keypresses that can be used when editing text is given at the end of the manual.

Inserting and deleting paragraph breaks

It is possible to insert and delete paragraph breaks in the text. To insert a break place the caret at the start of the first line of the paragraph or at the end of the previous line and press Return. To delete a paragraph break place the caret at the start of the first line of the paragraph and press Delete or at the end of the previous line and press Copy (End).

Insertion and deletion of paragraphs will affect the way styles are assigned if the output is saved as an RTF file. When inserting and deleting a paragraph the new paragraph will inherit the style of the previous paragraph. Any local styles will be maintained.

Saving the output

Once the OCR process has finished you will want to save the resulting output. You have three choices; you can save the text as plain ASCII text, as a Rich Text Format (RTF) file or in an editable file format. Saving a RTF file will allow you to import data directly into packages that accept RTF files maintaining any style information that Sleuth has managed to extract from the scanned image.

To save in a particular file format choose the relevant option from the main menu over the text window. This will open a standard save as dialogue box. Drag the icon to the filer or application where you want to save the text.

Editable file format

When you save the output in this format you can reload it at a later date and still be able to edit the file using the guided editing features described above. To see the relevant portions of the image you must have saved the image and it must still exist in the same location as it was when it was OCR'd.

Style information

Before you save the output as an RTF file you can see some of the information that will be included in it by choosing Show styles. This will switch the display to use outline fonts to mimic the styles in the original document. In the text editor the styles that will be shown are: normal, bold, italic and bold italic variants of the fonts that are specified in the Output preferences dialogue.

When the file is output as a RTF file these styles will be augmented with information on font size, justification, line spacing and paragraph spacing. Sleuth will create base styles if the same style is used in two or more paragraphs and overlay local styles on top of these. So, for example, you can get a paragraph with a base style of normal serif text with a single word in italic serif within it.

Sleuth can have problems in identifying styles if there is insufficient information for it to analyse. Cases where this can occur are, for example, where only a small amount of text is in a particular style or where the serifs of a font have been lost during the printing/scanning process. Sleuth can also experience difficulty in distinguishing bold words in normal text if the variants are quite similar.

Batch processing

Sleuth can process a batch of scanned images while you do something else. To do this place the scanned images within a directory. The images can be squashed or normal mono or greyscale sprites. You can only process 75 sprites at a time. Output from the batch will be placed in a sub-directory named SlthOutput. The output files will have the same name as the input image files. By default the output file format will be Editable so that the user can edit the files using the guided editing features, but this can be changed using the Output preferences dialogue box. If the Append option is chosen all the output will be written into one file with the same name as the first image processed. This option can only be used if Text is chosen as the output file type.

Zones can be used with the images as they are processed. To use a zone file create a sub-directory named SlthZones and place the zone files that you want to use within it. The zone file should have the same name as the image it is to be used for. If you want to use the same zone file for every image in the batch, call the zone file All.

The Progress window will show relevant information as the batch is processed. If an error occurs during the batch processing, Sleuth will attempt to move onto the next image. The Progress window will show which images have been processed successfully.

The batch process can be halted at any time by clicking on the Stop OCR button or by choosing the Stop OCR option from the icon bar menu.

Getting the best results from Sleuth

There are many factors which affect the accuracy of the output from Sleuth. We recommend that the following suggestions are followed where possible to get the best results.

Use at least 300 or 400 dpi resolution scanning. Scan the original document where possible, not a photocopy. Scan a portion first and convert it to ensure that the scan is not too light or too dark. If you are using a hand-held scanner try to make the scan as straight as possible.

Here are some suggestions to help you get the best results:

(1) Review the document and plan the best method of scanning. Do not waste time scanning graphics that will be discarded in the OCR process.

(2) Scan a portion of the document and convert it to check that the contrast setting of the scanner is correct. Time spent getting this right will be saved later by having fewer errors in the output to correct.

(3) If you are using a hand-held scanner first run the scanner over the document without scanning to check there are no obstacles that will impede progress during scanning and that the scanner is running as straight as possible. Try to line up the body of the scanner with a straight line on the page or the left edge of the text being scanned.

(4) Scan the document carefully. If you are scanning to the end of a page you may find that a hand-held scanner will jump slightly as the body of the scanner rolls off the edge. This will almost certainly affect the output from Sleuth when it reaches this point and can have further detrimental effects if the scanner is tilted. Often in this case the image has smudges which degrade the text beyond recognition. One way to alleviate this scan the top half of the page and then turn the document round and scan the bottom half. The second half can be rotated in Sleuth before conversion.

(5) When you have finished scanning load the images one at a time into Sleuth for conversion or use Batch Processing.

(6) If using a flat-bed scanner make sure that the document sits flat on the scanner. If it is not flat smudges will probably appear on the scan.

(7) When scanning books keep the book flat and use the side of the scanner that has the least gap between the scannable area and the edge of the scanner. This will minimise problems with scanning to the inner margin.

Important points to remember

(1) Check that the image is not too bright or too dark and has a uniform contrast

(2) Check that the image is not overly skewed.

(3) Use a high resolution scan unless the text is large. 300-400 dpi should be sufficient.

(4) Sleuth has only been trained on the fonts shown latter.

(5) There is a limitation on the size of text that can be converted.

(6) Sleuth has been trained on a specific set of characters, other characters will not be converted correctly.

(7) Sleuth will automatically deal with most complex page layouts automatically.

(8) Set the Resolution setting in the Preferences box if you want to output in RTF format.

(9) Style information can be incorrect if there is only a small amount of text in a particular style or the scan quality is poor.

Problem solving

These are possible problems that may occur when using Sleuth with explanations and possible solutions.

(1) I'm getting output but it is all punctuation or apparently random characters.

The scan is either too light, too dark or contains a font that Sleuth has not been trained on. Re-scan the image to lighten or darken it. If there is a font that is not recognised by Sleuth in the image it will not be able to accurately convert it.

(2) The output is good but the occasional word comes out badly.

It is likely that there is some defect in the scan where the conversion accuracy breaks down or the word is in a font unknown to Sleuth.

(3) The output is good at the beginning of the page, but gradually deteriorates.

Check that the contrast of the image is consistent. Uniform contrast over the image is important to get good results. Inconsistent contrast is likely to occur if the document has not been printed well or is a photocopy. Try re-scanning the document in portions adjusting the contrast of the scanner to compensate for the change in contrast.

(4) The output is good, but the order of the text is wrong.

The page layout is probably complex or contains text separated by large amounts of white space. If the page output is very poor or complex try OCRing again using zones. If the page contains a lot of white space in a single column of text place a single column zone around the text.

(5) The output has a lot of the words highlighted in the editor, but they are spelt correctly.

The words are probably not in the spell check dictionary. Either you can ignore the highlighted words or you can un-tick the Context option from the Preferences dialogue box when you OCR this sort of scan. Normally the words will be of a specialised or technical nature.

(6) The output is fair but I'm sure it could be better.

Try using the Careful option from the Preferences dialogue box.

(7) The output is excellent but it is rather slow.

Try using the Quick option from the Preferences dialogue box.

(8) I get text contained in graphics converted.

Use an ignore zone to ignore the graphic or set the Min. font size in the Input preferences dialogue box to a larger value. If the page layout is complex you may need to use text zones to keep the order of output correct.

Sleuth limitations

Although Sleuth has only been trained on the following fonts it will recognise other similar fonts without further training. It will recognise the following characters :

! ? £ \$ % & ' " % + = _ ^ * # () [] < > , -
.:;0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"'"'fifl•—

These fonts were generously provided by The Electronic Font Foundry, we would like to thank them for their efficiency and help in this matter.

Sleuth will convert characters between 9 and 24 point in size. This equates to approximately 1/8th to 1/3rd of an inch depending on the font. The actual limits are between 10 and 80 pixels measured from the top of an upper-case letter, like an 'A', to the bottom of a lower-case letter with a descender, like a 'g'. To convert 24 point text use a 200 dpi scan.

Sleuth will convert text at speeds between 80 to 400 words per minute depending on the machine used, the size and quality of the text and the resolution of the sprite.

Sleuth will automatically cope with slightly skewed scans (generally less than 2 degrees of skew, depending on line spacing) and lines of text that are slightly wavy.

How Sleuth works

This is a brief description of the way Sleuth 3 works.

Training

Before it can recognise letters the OCR program must be taught their shapes. The main problem is that there is a lot of variation for each letter because of different fonts, sizes, and various defects introduced by the printing and scanning processes. The training program for Sleuth 3 (not supplied) incorporates a sophisticated algorithm which starts with an outline font and mimics the processes of printing, faxing and scanning to make many examples of each letter. The training program uses about 1,000,000 examples in all. Each of these shapes is converted into a list of numbers (a "feature vector") which represents the important aspects of the shape. As each feature vector uses about 100 bytes it would require about 100Mb to store this information directly so it is "summarised" in a special way.

Finding lines of text

The first thing Sleuth 3 does is to find all the black shapes in the image. Halftone areas (lots of small dots close together) are detected and these are stored along with large shapes as graphic areas. These are used later when the lines of text are found and ordered. Very small shapes that do not seem to be part of a half-tone area or a letter are weeded out at this stage.

Next the program estimates the skew in the image and removes it. It does not rotate the actual shapes (which would introduce further distortion) but moves them so that lines are horizontal and gutters (the space between columns) are vertical. Then the shapes are gradually merged together to form lines. Care is taken to avoid jumping over graphics and gutters or joining dropped capitals onto a line. Special processing is used for table zones.

The lines of text are tentatively ordered, but unlike Sleuth 2, the final ordering is not fixed until they have been read. This means that Sleuth 3 is less likely to be confused by non-text that looks a bit like text which could make Sleuth 2 get the order of the lines wrong. It can use information about text size to join lines of text into blocks, and it can use sentence structure to order blocks of text where this is ambiguous. Unfortunately this also means that Sleuth 3 cannot output text until the whole page has been read.

Character recognition

During reading shapes are extracted from the image, converted into feature vectors and compared with information about known shapes acquired during training. The result is a shortlist of possible letters, with scores representing how close the matches are, and also information about the style (bold/italic/serif) of the letter.

Although character recognition is at the heart of any OCR program it is only one part of a complex system. Letters are often joined to their neighbours, and sometimes split into two or more parts, so these problems must be dealt with before character recognition can take place. Some characters (eg p,P,c and C) can only be identified once the scale and position is known from the rest of the line. Others (eg l,l,1,e and c) can often only be correctly identified by using information about the likely identities of other shapes in the same word.

Reading a line of text

Sleuth 3 uses a technique called 'oversegmentation' to locate the letters in a line of text. The image of the line is split into many small fragments by making vertical cuts at likely points. These are then ordered left to right, and the program considers various ways of joining them together again to make letters.

Sleuth 2 used a similar technique, but only when it failed to recognise a word using a simple way of dividing up words into letters. Sleuth 3 uses oversegmentation on every line and has a special module to recognise monospaced text. When text quality is poor Sleuth 2 may fail to find enough letters in the line to make a reliable estimate of scale and position. In the picture above Sleuth 2 would see the whole shape as a reject at its first attempt and would fail to interpret the text correctly if much of the line was joined up into clumps of letters or if many letters were split. Sleuth 3 is much better at getting more letters right first time and so can deal with lower quality text. The disadvantage is that it is somewhat slower.



Here is part of a word that has been oversegmented

A shortlist of possibilities is obtained for each shape in the line and the best recognised letters used to find the baseline and scale of the text in the line and therefore remove letters from the shortlists if they have the wrong size or position. Next word breaks are found and each word is visited in turn. If it is possible to choose letters from the shortlists so that the word spell checks and without any letters being poorly recognised the word is considered done. Here's how the word "excellent" from a low quality scan might look like at this stage. There are no split or joined letters, but the correct interpretation is far from obvious.

```
cxcc11cnt  <- first choices
e eelle [  <- second choices
  ll       <- third choices
  !!       <- fourth choices
  ii       <- fifth choices
  }        <- sixth choice
```

If the word is not recognised at the first attempt, it is examined more closely. The program considers various ways in which the fragments can be put together to make new possibilities for the shapes in the word. Each time a new set of pieces is combined to make a shape the shape must be recognised and each time a new set of shortlists for a word is produced the results must be spell checked. The time the program spends on this depends on the Quick/Medium/Careful settings.

For example, in the 'tous' image earlier, the pieces would probably be combined so that 1 was on its own as a 't', 2 and 3 would be combined to make a reasonable 'c', then 4 would be added to make the 'o', 4 and 5 could make an italic 'n', 5 and 6 both make bad i's, but together make a good 'u', it is possible that 6, 7 and 8 could make a bad 'w', but that would be discarded for 7, 8 and 9 to be combined to make the 's'.

Final steps

Once all the (supposed) text lines have been read, any ones that were unreadable are rejected, and the remainder are grouped into blocks and ordered. Then paragraph breaks are found and for each paragraph the text size, line spacing, justification and main style (bold/italic/serif/monospaced) is found. The program then finds any words which do not match the main style. Finally, the text and formatting information are sent to the text editor, or to a file if in batch mode.

Why is Sleuth 3 slower than Sleuth 2?

The main reasons are:

1. Sleuth 3 uses a more careful initial segmentation of each line into letters as described earlier.
2. Sleuth 3 has a more sophisticated spell checker which uses information about word frequencies (eg it knows that 'the' is more likely than 'tho'). It also incorporates a user dictionary.
3. Sleuth 3 does more processing of graphics, which aids the division into text and graphics and the segmentation of text areas, eg using vertical rulings.
4. Sleuth 3 is trained on more fonts and a wider variety of text qualities, including symbols that have been artificially 'faxed'.

Appendix 1 - Supported fonts

We would like to thank The Electronic Font Foundry for their help in supplying the fonts used in training Sleuth.

American.Typewriter.Bold
American.Typewriter.Bold.Italic
American.Typewriter.Medium
American.Typewriter.Medium.Italic

AvantGarde.Gothic.Book
AvantGarde.Gothic.Book.Oblique
AvantGarde.Gothic.DemiBold
AvantGarde.Gothic.DemiBold.Oblique

Benguiat.Bold
Benguiat.Book

Bob
Bob.Bold
Bob.Bold.Italic
Bob.Italic

Bodoni
Bodoni.Bold
Bodoni.Bold.Italic
Bodoni.Italic

Bookman.DemiBold
Bookman.DemiBold.Italic
Bookman.Light
Bookman.Light.Italic

Cheltenham.Bold
Cheltenham.Bold.Italic
Cheltenham.Book
Cheltenham.Book.Italic

Corpus.Bold
Corpus.Bold.Oblique
Corpus.Medium
Corpus.Medium.Oblique

deGaulle
deGaulle.Bold
deGaulle.Bold.Italic
deGaulle.Italic

Garamond.Light
Garamond.Bold
Garamond.Bold.Italic
Garamond.Light.Italic

English
English.Bold
English.Bold.Oblique
English.Oblique

Eric
Eric.Bold
Eric.Bold.Italic
Eric.Italic

Europe
Europe.Bold
Europe.Bold.Oblique
Europe.Oblique

Eve
Eve.Bold
Eve.Bold.Italic
Eve.Italic

Evenletter
Evenletter.Bold
Evenletter.Bold.Slanted
Evenletter.Slanted

Evenwriter
Evenwriter.Bold
Evenwriter.Bold.Italic
Evenwriter.Italic

Frederic.OldStyle
Frederic.OldStyle.Bold
Frederic.OldStyle.Bold.Italic
Frederic.OldStyle.Italic

Griffo
Griffo.Bold
Griffo.Bold.Italic
Griffo.Italic

Homerton.Bold
Homerton.Bold.Oblique
Homerton.Medium
Homerton.Medium.Oblique

Michael
Michael.Bold
Michael.Bold.Italic
Michael.Italic

New.Baskervill
New.Baskervill.Bold
New.Baskervill.Bold.Italic
New.Baskervill.Italic

NewSchbook
NewSchbook.Bold
NewSchbook.Bold.Italic
NewSchbook.Italic

Oxford
Oxford.Bold
Oxford.Bold.Oblique
Oxford.Oblique

PL	Trinity.Bold
PL.Bold	Trinity.Bold.Italic
PL.Bold.Italic	Trinity.Medium
PL.Italic	Trinity.Medium.Italic
Souvenir.Bold	Typewriter
Souvenir.Bold.Italic	Typewriter.Bold
Souvenir.Medium	Typewriter.Bold.Italic
Souvenir.Medium.Italic	Typewriter.Italic
Swiss.Monospaced	Veronica
Swiss.Monospaced.Bold	Veronica.Bold
Swiss.Monospaced.Bold.Italic	Veronica.Bold.Italic
Swiss.Monospaced.Italic	Veronica.Italic

Appendix 2- Keyboard shortcuts

The following keyboard shortcuts are available when using Sleuth 3:

Up Arrow	Move the caret to the previous line
Down Arrow	Move the caret to the next line
Right Arrow	Move the caret to the next character
Left Arrow	Move the caret to the previous character
Shift + Right arrow	Move the caret to the next word
Shift + Left Arrow	Move the caret to the previous word
Ctrl + Right Arrow	Move the caret to the end of the line
Ctrl + Left Arrow	Move the caret to the start of the line
Page Up	Scroll the text window up
Page Down	Scroll the text window down
Delete	Delete the character before the caret
Copy	Delete the character after the caret
Shift + Delete	Delete the word at the caret
Ctrl + Delete	Delete the line at the caret

Here are just a few of the other products you will be able to
download from RISCOS.com free of charge
(You can make a donation to help support future product releases but this is entirely voluntary)



This Documentation is © 3QD Developments Ltd 2026
Some of the products shown above might not be available immediately.
Not all products will run on all versions of RISC OS, check the product information page for details.

Other Products list last updated 29/2/26