Apple® IIGS

ChewBagger Reference Manual

Photo: Courtesy of Mike Maginnis

Dedicated to the memories of Joe Kohn 1947-2010
and Ryan Suenega 1967-2011

spectrumdaddy@speccie.uk

Disclaimer

All characters appearing in ChewBagger are fictitious.
Any resemblance to real persons, living or dead contained in any movie titles from LucasArts, or Industrial Light and Magic, is purely coincidental.

Credits

My thanks go to all who originally helped me develop ByteBagger, the suggestion from Hugh Hood for a Disassembly function, and the suggestion for the Compare Files function from Andrew Roughan.
My thanks also go to Hugh Hood, David Schmidt, and the rest of the beta testers, for their invaluable help and suggestions.

Preface

Welcome to ChewBagger

Background

Way back, in the dim mists of the early days of the Apple II, I wrote a sector editor for 5.25" floppies. Since then, as a programmer, I have from time to time needed to be able to look into files, and if necessary, alter their bytes on the fly.

The original ByteBagger NDA arose from this idea, and was well received. However I was not happy about the actual user interface when editing the two data fields, and after a suggestion from Hugh Hood, also wanted to add a disassembly feature. To have added more features to ByteBagger itself would have resulted in a bloated, memory hungry NDA, which would have defeated the whole idea of what an NDA should be.

Thus the need for a stand alone version of the ByteBagger NDA arose, one in which many new and useful features could be added as time went on.

So ByteBagger has now grown up, and ChewBagger has been born...

The ChewBagger Application

Requirements

ChewBagger is a stand alone application, and requires no fonts or tools other than those already in System 6.0.1 and the Undo Manager. ChewBagger uses custom fonts that are built into the application itself. To optionally label a disassembly, the supplied 'Tool2.Calls' and 'File2.Calls' files must be in the same folder as the application.

What is ChewBagger

ChewBagger lets you view the raw data of either the data or resource fork of a file, search for target strings within the fork, using either Hex or ASCII data, and optionally allows you to edit the data, and write the changed data back to the file. You can also load an OMF file into memory, or examine a section of memory. You can also compare two files for any differences there may be between them.

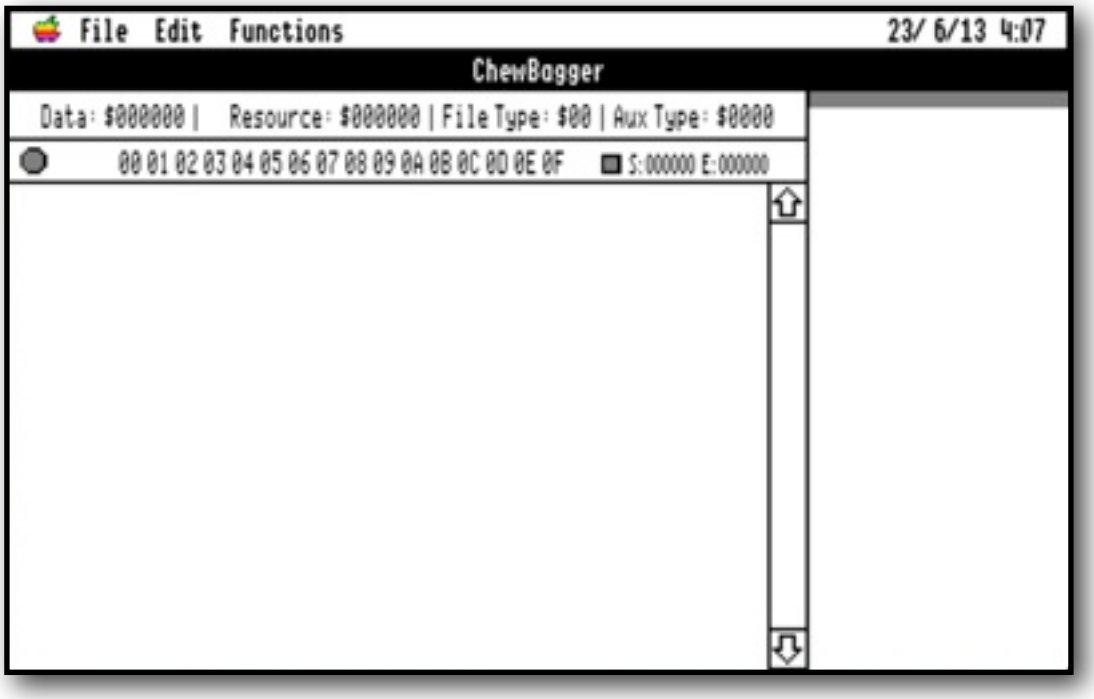
In addition, you can disassemble, print, or dump to file, a selection or range of data from any of the viewing screens.

Warning: ChewBagger will not let you alter the length of a file, as with many IIgs files this can be a very dangerous operation. You should also know exactly what you are doing before you alter the data in a file, as you could easily corrupt an application and make it unusable! ChewBagger will not be held responsible for any damage that you may cause to your files...

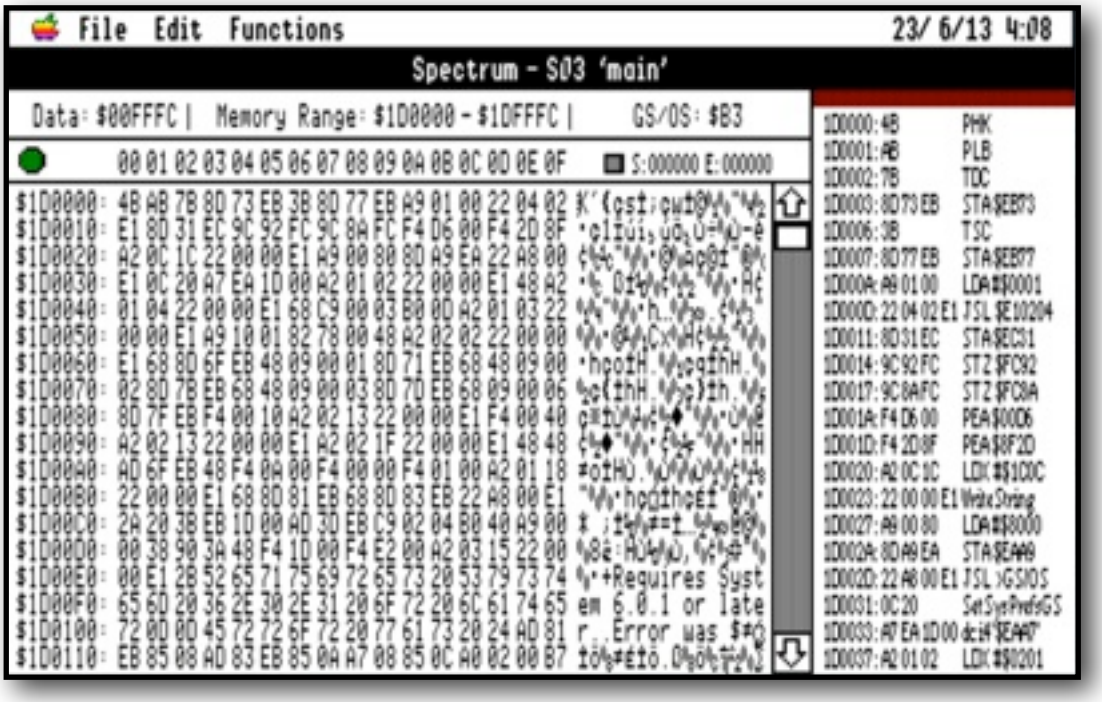
Legal

Using ChewBagger

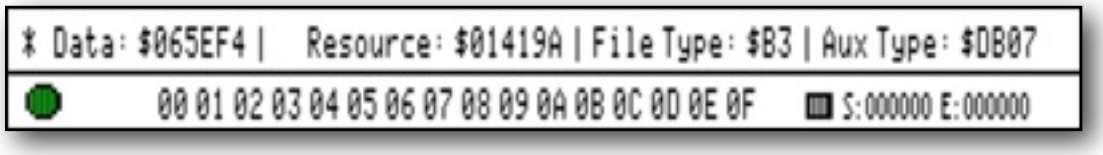
To use the application, open ChewBagger and you will see this window:



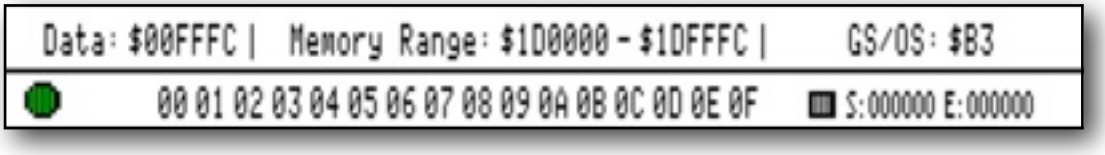
From the File Menu, you can choose 'Open File', or 'Load File' to select a file to examine, choose 'Examine Memory' and enter a memory range, or choose 'Compare Files' to find differences between them:



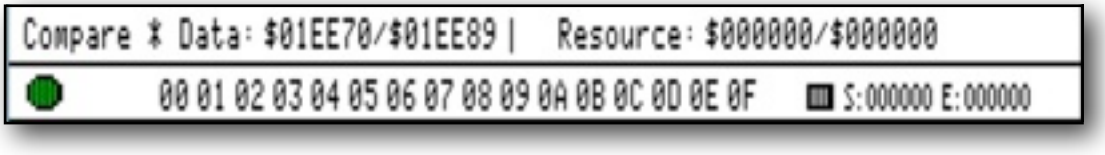
The main window has a Title bar showing the name of the current file, and an upper 'Info' bar showing the 'Data' size, 'Resource' size, 'File Type', and 'Aux Type' for the file. A lower 'Info' bar displays flags and other information:



Information bar for an 'Opened' file.



Information bar for a 'Loaded' file, or when 'Examining Memory'.



Information bar for 'Compare Files'.

Reading from left to right, the upper 'Info' bar shows the size of the 'Data' and 'Resource' forks, the 'File Type', and the 'Aux Type'. If you have 'Loaded' a file, this will show the data size and memory range, and whether it is a 'Native GS/OS' file or a 'ProDOS 8' file, and its 'File Type'. If you are 'Examining Memory', it will show the size and range or memory you are examining. If you are 'Comparing' two files, the 'Info' bar will show the sizes of the two files.

If you have 'Opened', or are 'Comparing' files, an asterisk will show to the left of the 'Data' or the 'Resource' fork label, to show which fork you are viewing.

Reading from left to right, the lower 'Info' bar shows a traffic light icon, which indicates the state of the data, a byte counter for the columns of data, a square icon which shows if you have set 'Markers' or not, and the values of these 'Markers'. Refer to the relevant sections for more details.

The main Data area has three columns. The left column shows where in the file you are currently positioned, to its right is a Hex data field, giving a Hex representation of the data in the file, and to its right is an ASCII field, giving an ASCII representation of the data. To the extreme right is a column where disassembled data will be displayed.

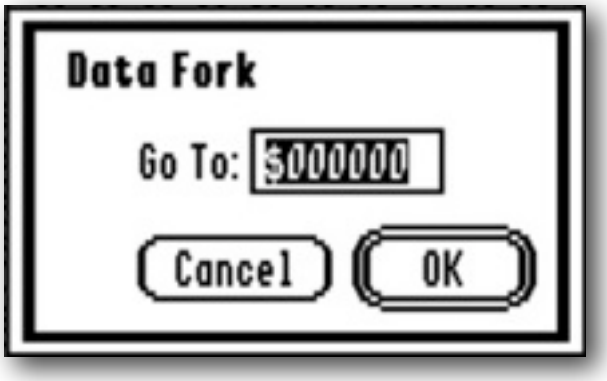
Note: Although the Hex data field has all values correctly displayed in their Hexadecimal equivalents, the ASCII data field may have some control characters replaced with a period.

Both fields have also been formatted with CRs to break the displayed lines.

To reduce memory overhead, and to allow very large files to be opened quickly, ChewBagger only 'Opens' a 288 byte window into a file. You can though move smoothly through the file using the scrollbar or Arrow keys, but if you have made any changes to one of the fields, and the changed data is valid, you will be prompted to save the changed data before you can scroll any further.

If you are 'Comparing Files', the window is split horizontally, with the top 144 byte window into the file showing the 'Old' file, and the bottom 144 bye window into the file showing the 'New' file.

If you 'Load' a file, then either the entire file, or if it is a large application with multiple segments, a segment from the file, will be loaded into memory. You will still be able to move through the data in the same way as if you 'Opened' a file.



To switch forks, or jump to a fixed point in one of the forks, either click on the 'Data:' or if the file has one, the 'Resource:' fields in the Information bar, or choose 'Data Position' or 'Resource Position' from the Functions Menu. When comparing files, you cannot switch directly to view the other fork.

If you have 'Opened' a file, or are 'Editing' when 'Comparing Files', you can edit data in either the 'Hex Data' or the 'ASCII Data' field. Any changes you make in one field will immediately display in the other. You cannot directly edit data if you have 'Loaded' a file, are 'Examining' memory, or are 'Comparing Files'.

File Menu

Open File...

'Opens' a file, and displays its data in both Hexadecimal and ASCII format. You can choose to open either the 'Data' fork, or the 'Resource' fork.

You can view and scroll through the data of the file, 'Find' target strings in either Hex or ASCII values, 'Edit' the data, or 'Disassemble' a range of data.

Load File...

Loads the data fork of an Object Module Format file (OMF) of File Type $B1 - $BD, a data file of File Type $06, or a P8 application of File Type $FF, and displays its data in both Hexadecimal and ASCII format. The advantage over 'Open File', is that if you then disassemble the data, all relocatable links within an OMF file will have been resolved correctly. The first segment of an OMF file will be loaded first, and if there is more than one segment, use the 'Next Segment' menu option to display the other segments.

P8 applications, and $06 data files, will be loaded at the virtual address that you have specified. Refer to the 'Preferences' section for more information.

Next Segment

If you have 'Loaded' an OMF file with multiple segments, this will display the next segment in sequence. When you reach the end of the segments, it will display the first segment again. The segment number, and segment name, are displayed to the right of the filename in the Title bar.

Examine Memory...

Displays a range of the IIgs memory in both Hexadecimal and ASCII format.

Compare Files...

Compares two files, stopping and highlighting the first difference that is found.

Edit File...

Switches into 'Edit' mode for the file that is currently active. Refer to the 'Comparing Files' section for more information.

Next Difference...

Jumps to, and highlights, the next difference between the two files.

Close...

Closes the current file. If 'Editing' a file from 'Comparing Files', closes the 'Edit' window, and returns you to the split window displaying the two files.

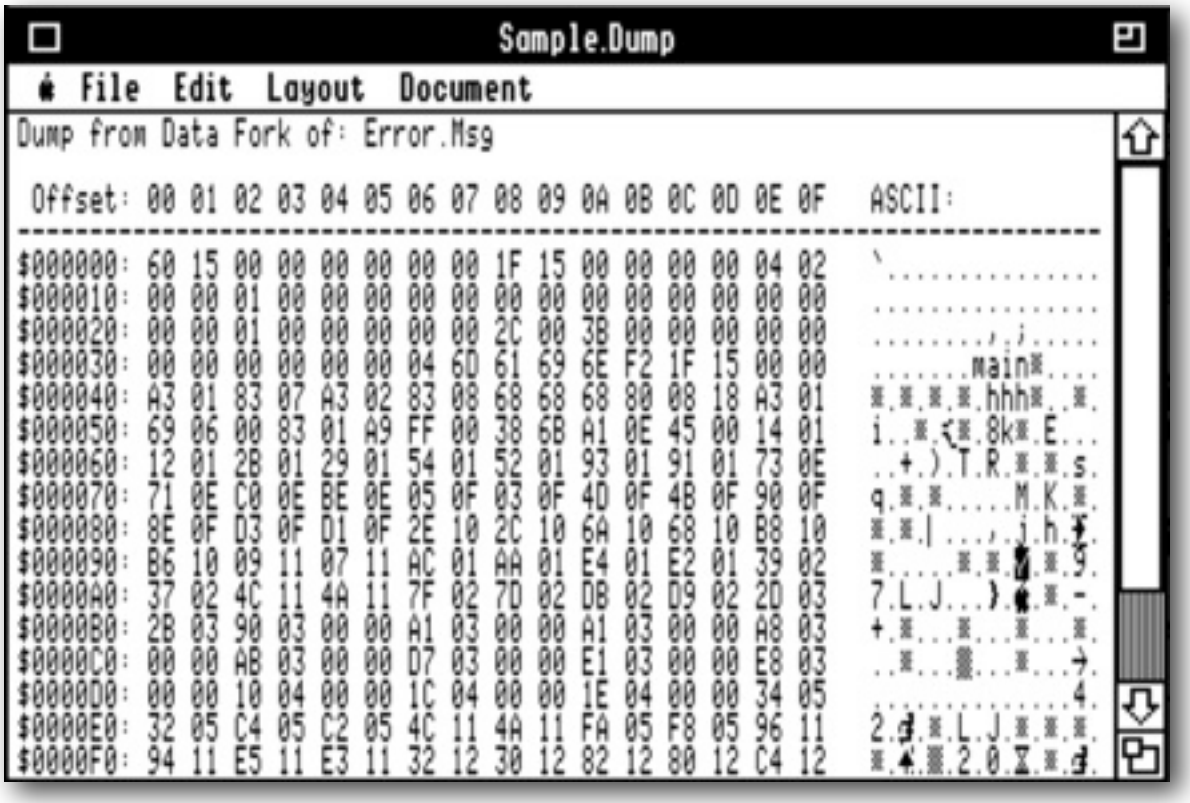
Revert...

If you have made any changes and saved them back to disk, this option will allow you to 'Revert' the changes you have made, and restore the original file data.

Save...

If the active cursor is currently in either the 'Hex' or 'ASCII' data fields, and you have made any changes to the data, this will save the changed data back to the original file on disk. To avoid disasters, you will be warned before the data is actually saved back to the original file.

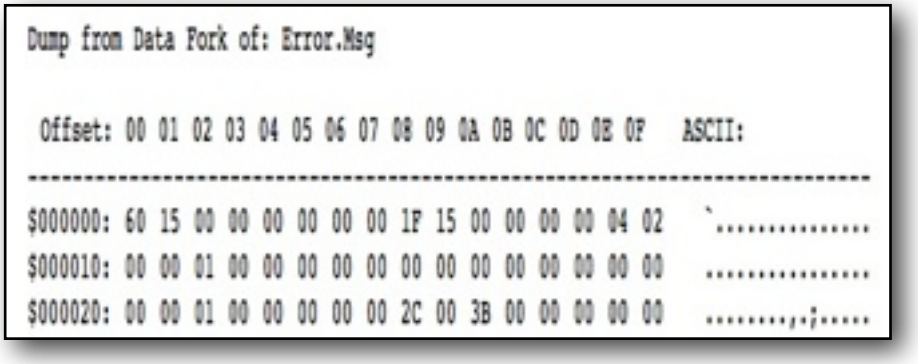
You can in an emergency, later choose to 'Revert' any changes that you have made.



Dump To File...

Saves a selected range of the data to a text file. You first choose the range of the selection to dump by entering values for the start and end of the data range, or by setting 'Markers' to define the range. Refer to 'Selecting Data' for more information on how this works.

The formatted text will be saved to a file in this format, with a descriptive legend at the head of the file:



Printer Setup...

Opens the standard Printer Setup dialog. In order to print, you must first set up your printer from the 'DirectConnect' Control Panel.

Print...

Prints a range of data to the selected printer, in the same format as 'Dump To File'. You first choose the range of the selection to print, by entering values for the start and end of the data range, or by setting 'Markers' to define the range. Refer to 'Selecting Data' for more information on how this works.

The formatted text will be sent to the printer, with a descriptive legend at the head of the first page.

Quit

Quits ChewBagger.

If you have not 'Saved' any changes you have made to the data fields, you will be prompted to do so. If the data fields are not of the correct length, you will be warned, but will be unable to go back and correct the data.

Edit Menu

Undo...

If you have made any changes in the current 'window', and have not 'Saved' them to disk, this will undo the changes that you have made.

If you are 'Comparing Files', and have set up 'Offsets' between the two files, this will undo all the 'Offsets' you have set, and redraw the data.

Find...

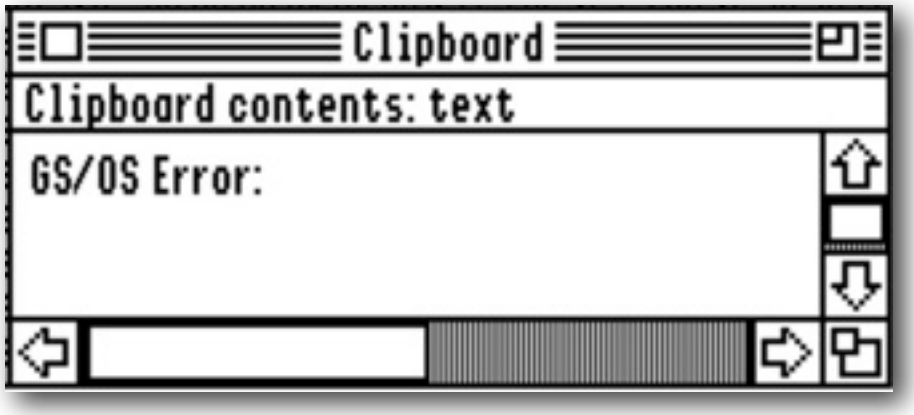
Lets you search for, and optionally replace, either Hex or ASCII data. Refer to 'Find & Replace' for further details.

Find Again

If you have already searched for a target string, this will repeat the search from the current cursor position. Refer to 'Find & Replace' for further details.

Show Clipboard...

Opens the standard Clipboard window.



Functions Menu

File & Aux Type...

Allows you to change the File Type, or Aux Type of the file you have 'Opened'. A dialog window will open where you can change the values of each. Alternatively, click on the 'File Type:' field in the Information bar. to open the same dialog. You must enter Hex values.

Data Position...

This opens a 'Go To:' dialog, where you can enter either a Hex value, or a decimal value to jump to that position in the data fork. Alternatively, click on the 'Data:' field in the Information bar to open the same dialog.

You may enter Decimal or Hex values, using a $ sign to indicate Hex values.

Resource Position...

This opens a 'Go To:' dialog, where you can enter either a Hex value, or a decimal value to jump to that position in the resource fork. Alternatively, click on the 'Resource:' field in the Information bar to open the same dialog.

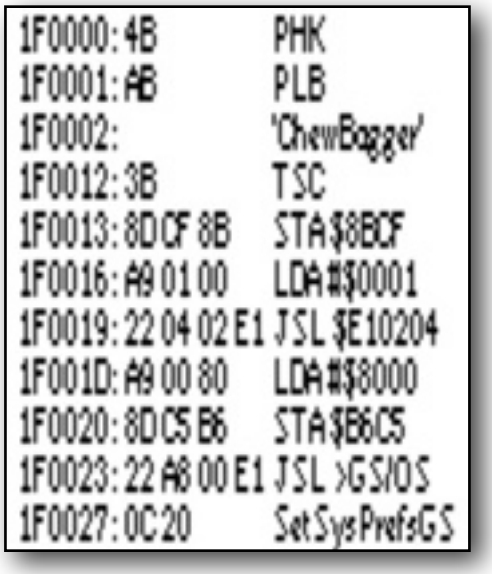
You may enter Decimal or Hex values, using a $ sign to indicate Hex values.

Preferences...

Allows you to change various values that are used for the 'Disassembly' display. Refer to the section on 'Disassembling' for more information.

Disassemble...

Disassembles a range of code, and displays the result in the far right hand column. Enter the range in the dialog box for the start and end of the data to disassemble, or first set up 'Markers' to define the range. Refer to 'Selecting Data' for more information on how this works. Refer to the section on 'Disassembling' for further details on the disassembly process.



Editing

Warning: Be very careful if you decide to edit the data in a file. You should know exactly what you are doing before you change anything at all, as modifying either the data, or the resource fork of a file could have unpredictable consequences!

To keep the memory overhead of the application low for large files on a low memory machine, ChewBagger only displays a 288 byte 'window' into a file. Through scrolling, it will however appear as if you are moving normally through the entire file. If you make any changes to the data in either of the fields, this 288 byte 'window' must be saved before you can scroll further.

ChewBagger uses standard TextEdit windows to display the editable data fields. Entering characters will always 'overwrite' the character to the right of the cursor, regardless of any text that has been selected, and the change will immediately be shown in the other data field.

When Chewbagger Quits, you will get a chance to 'Save' any changes you have made. It is not possible to go back and edit the data, so if you do not 'Save', you will lose any changes.

You can enter any value from $00-FF into the Hex field using Hex notation, but as the ASCII field cannot display some control characters, this means you cannot enter some characters directly into the ASCII field. Any of those characters that are in the file will be displayed as a period, and can be changed to normal printable characters if you wish. To enter any such control, or other non-standard character, you must enter them in the Hex data field in Hex format instead.

Typing the first character in the Hex data field will enter two to maintain the correct length of the Hex value for that character. If you do not move the cursor position, entering a second character will place it in the correct position within the pair. This allows you to easily type a Hex value.

If you have changed any data, and try to scroll through the file, you will be warned, and have a chance to either 'Save', 'Restore', or 'Cancel'. If you 'Cancel', you will get a chance to edit the data once more. If you 'Restore', it will restore the original Data from the file and scroll as requested. If you 'Save', the data will be 'Saved' back to the file, and you will also scroll as requested.

To 'Save' any modified data back to disk without scrolling, select 'Save' from the File Menu. Confirm that you wish to 'Save', or 'Cancel' without saving. You can also choose to 'Restore' to the original saved data.

Note: While viewing data, you can toggle the display of characters in the ASCII field with high bits set. Press OA-H to toggle the mask, and redraw the display. An asterisk will show at top left of the ASCII field in the Lower Status Bar, if the high bits are being cleared.

Traffic Light Icons

At the far left of the lower Status bar, is a circular coloured icon. This changes colour depending on the state of the data fields:


Grey - No data has been loaded.


Green - Data is loaded, and has not been changed.

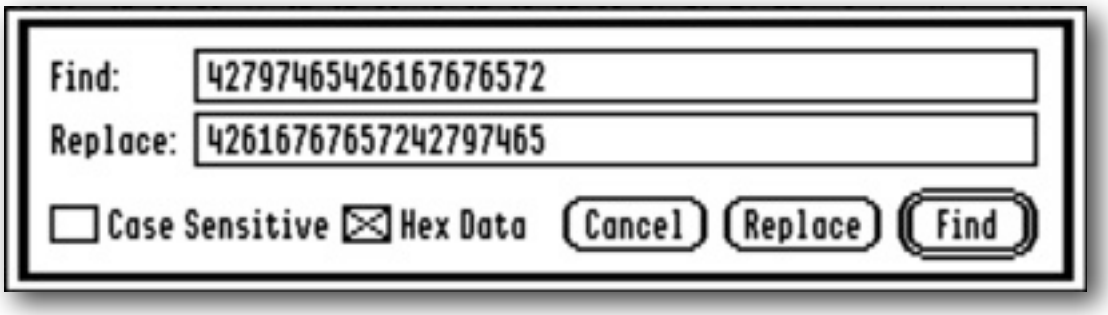

Orange - Data has been changed and can be saved.

Revert versus Undo/Restore

While you are editing a 'screen' of data, and have made some changes, you can choose 'Undo' from the Edit Menu to restore the screen to the original data from the file. If you choose 'Save' from the File Menu, you will also get a chance to restore the screen to the original data.

If you choose 'Revert' from the File Menu, you will be able to revert to the original file, removing all the changes that you have made, thus restoring the original file.

Find & Replace



To find a target string within the file, select 'Find' from the Edit Menu, or press the OA-F keys. You can enter either a string representing Hex values, and check the 'Hex Data' box, or enter a ASCII string, with the 'Hex Data' box unchecked. Click 'OK' to search for that string, starting from the current position in the file. Optionally for ASCII text, you can choose to make a 'Case Sensitive' search.

You can optionally choose to 'Replace' with a second string. This string must be of the same length as the first string, and must be different to it.

Note: If you 'Replace' with a second string, the data will be immediately saved back to disk. You can always 'Revert' to the original file later if you wish.

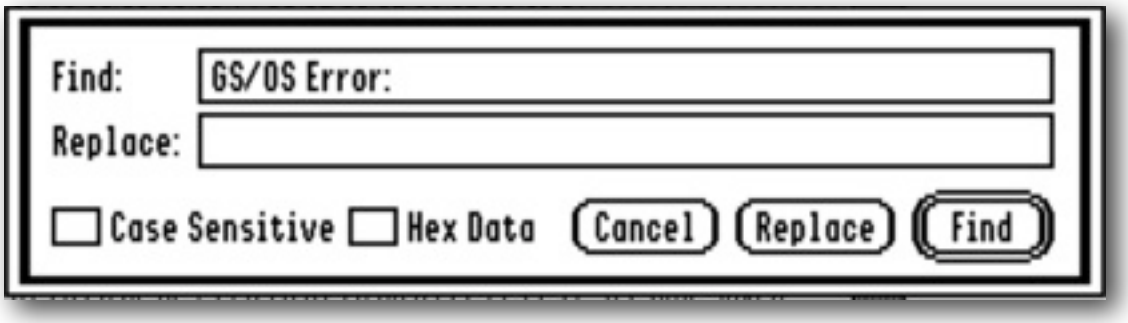
Note: If you have not 'Saved' any changes you have made, you will be asked to do this before you can continue.

If the search string is found, it will be highlighted in the appropriate data field, and a small window opens.

You can then choose to 'Stop' searching, 'Find Again', or if replacing, to 'Replace' with the new string.

Note: If you copy data from either of the two data fields using the OA-C keys, the data will be correctly formatted for the search. The data will be the actual data, with any formatting spaces, periods, and invisible CRs removed. Because ASCII data may contain control characters and other high bit characters, which the custom font used in the ASCII field is able to display, it may look different when you paste it into the Search box.

Note: If you are 'Comparing Files', the active file will be searched. To change the active file, click the cursor into one of the windows for the other file. You will not be able to 'Replace' any data during a search when 'Comparing Files'.



Find Again

Repeats the search, starting from the active cursor point, which unless you have changed the cursor position, will be where the last search left off. You can also press OA-G, instead of selecting from the Edit Menu.

If the search string was not found, you will get the chance to 'Stop' searching or 'Start Over' from the beginning.

Selecting Data

For the 'Dump to File', 'Print', and 'Disassembly' functions, you need to enter the range of data to be dumped or printed. If no data is selected in the two fields, then you will be prompted with the entire range of bytes that is currently being displayed. If you have selected any data, then you will be prompted with that range instead. Click the 'All' button to set the maximum range.

If you wish to dump or print a larger range of data than the current display, then you can either manually make a note of the required range, or use the 'Start' and 'End' marker features. If both markers have been set, which will show by a small green icon, then these 'Start' and 'End' marker values will be used instead.

You can set or clear the markers at any time, and the markers are also cleared when you 'Open', 'Load', or 'Compare' new files to display. A green icon shows if you have set the markers, and you will see the current marker values to its right.

If you have not yet set the markers, and use one of the 'Dump to File', 'Print', or 'Disassembly' functions, the markers will be set to the range you have selected.

Type OA-B to set the Start marker, OA-E to set the End marker, and OA-K to clear them. If the markers have not been set, the icon will show as grey.

Disassembling

ChewBagger can disassemble data from the currently displayed file or memory location, and display the result in the far right hand column. Either manually select a range, or use the 'Start' and 'End' markers, to select the range to disassemble.

ChewBagger can disassemble both ProDOS 8 6502 code, or native GS/OS 65816 code. Refer to the 'Preferences' section on how to select this.

The disassembled code will be displayed in the far right column, where you can copy the text to the clipboard, scroll through it, or 'Print', and 'Dump to File'. If no text is selected, the entire range will be printed or dumped, but if you select a range of text, only that section will be used.

Note: You can disassemble the entire code of a large file, dependant on available free memory. However you need to be patient if you decide to disassemble a lot of code, as the speed of your processor may mean it takes some time to complete!

Note: ChewBagger assumes that what it is dumping is valid code. If you incorrectly identify whether it is 6502 or 65816 code, then the results may not be accurate. Data segments within code cannot be identified, so they too might display incorrectly. This may be especially apparent at the end of a block of data, where the first real bytes of code may be missed by the disassembly process, thus incorrectly displaying the subsequent code. If you can identify the end of a block

of data, then simply make a fresh disassembly from that point onwards to see the code displayed correctly.

Note: P8, GS/OS, and Toolbox calls, will be identified as they occur. They can optionally be shown referenced by their actual call name. See the 'Preferences' section on how to invoke this.

If you have not chosen to 'Display Tool Calls', then for 'Tool' calls, reference the call number in the preceding LDX #$0000 command. A Toolbox call will be displayed with its Tool number like this '>Toolbox $01'. By cross referencing the Tool number and the Call number, you can determine which call is being made.

GS/OS calls may be 'Inline' or 'Stack' calls. 'Stack' calls will be preceded by the required data being pushed onto the stack, and 'Inline' calls will be followed by a two byte call number, and a four byte address for the parameter block. These will be shown as data statements.

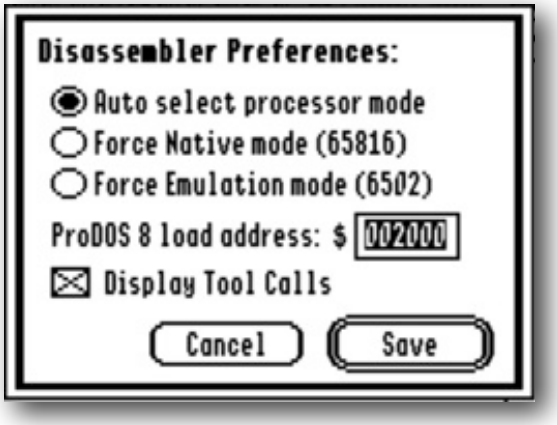
You can extend the range by a further 288 bytes, or to the end of the file, by pressing the down arrow key when you get to the end of the current range.

Note: You can halt a long disassembly by pressing OA-. (Open Apple - Stop).

If you wish to close a disassembly without closing the main display, press the red area above the display. It will change to grey if there is no disassembly currently being displayed.

Note: The OA-Y keys will show the current selection within the Disassembly in the related Data field.

Preferences



To make a meaningful disassembly, ChewBagger needs to know whether it is native IIgs 65816 code, or emulated Apple // 6502 code, that it is processing.

If you select 'Auto select', ChewBagger will check the file type, and set the mode automatically. If you want to force a particular mode, simply choose 'Native' or 'Emulation' accordingly.

ProDOS 8 6502 code is normally 'Loaded' at an address of $2000. If you wish to simulate it being 'Loaded' at a different address, then change the value in the 'ProDOS 8 load address' box. You can enter any load address between $000000-$FFFFFF, with the displayed offset wrapping round when it reaches $FFFFFF.

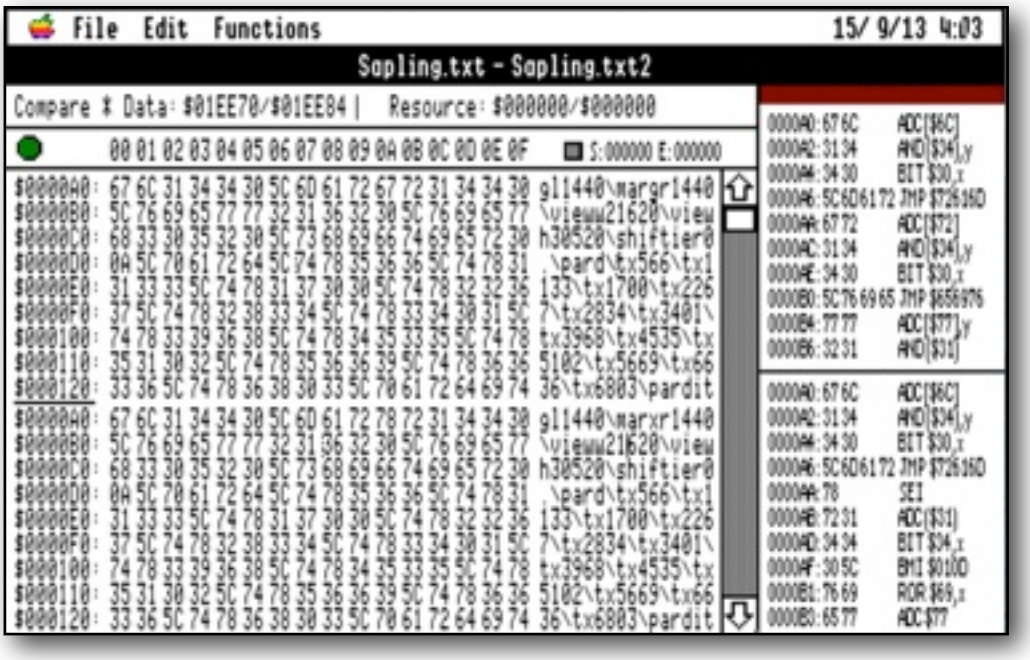
Optionally you can expand the tool and OS calls in the disassembly. To do this, two files, 'Tool2.Calls', and 'File2.Calls', are provided with the archive. They must both be in the same folder as Chewbagger itself. It will take a little longer for ChewBagger to expand the calls.

These two files can be edited if you wish to add new calls, edit calls, or remove redundant calls. Follow the format of the existing files, and add new calls or edit existing ones as necessary.

'REM' statements must start with an asterisk and space, and the call number must precede the tool description itself:

```
* File: M16.Window
590E AlertWindow
1E0E BeginUpdate
240E BringToFront
0A0E CheckUpdate
0B0E CloseWindow
600E CompileText
0C0E Desktop
1A0E DragWindow
550E DrawInfoBar
5B0E EndFrameDrawing
510E EndInfoDrawing
1F0E EndUpdate
620E ErrorWindow
170E FindWindow
150E FrontWindow
540E GDRPrivate
480E GetContentDraw
3E0E GetContentOrigin
2F0E GetContentRgn
```

Comparing Files



You can compare two files, find the differences between them, and optionally edit either of the files. When a difference has been found, the difference will be highlighted in both files.

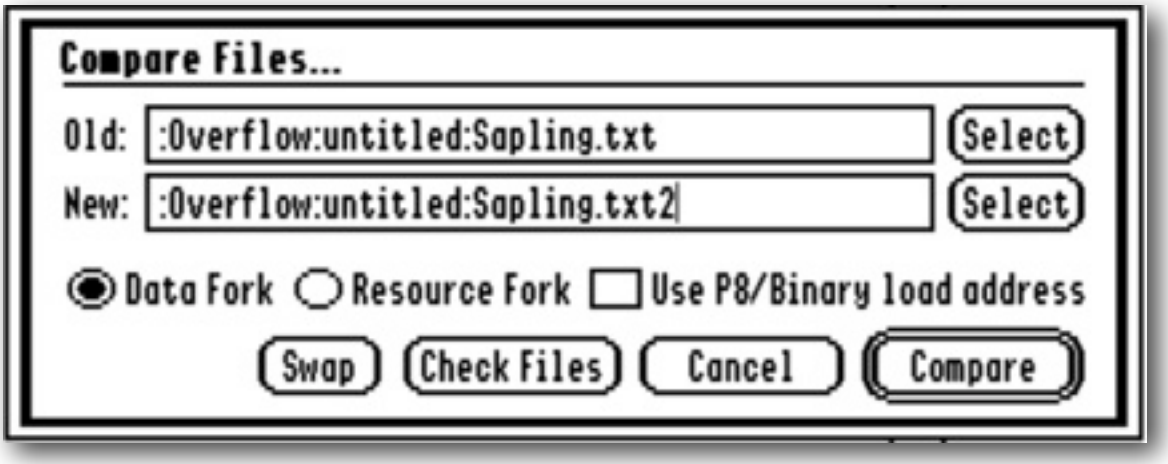
If you move the cursor, any differences to the right of the cursor will always be highlighted.

You can optionally set up an 'Offset' between the two files, so you can find any differences that may occur after a position where the new file may have had data inserted. See 'Setting an Offset' for more information.

The screen is split horizontally into two halves, with the 'Old' file displayed at the top, and the 'New' file displayed at the bottom. The two files will scroll together as a pair.

In most cases, any resulting actions will depend on which field or file is currently active. Simply click the cursor in a field to make it active. For instance, click in one of the two top fields to 'Disassemble' the 'Old' file, and click in one of the bottom two fields to 'Disassemble' the 'New' file.

You will first need to select the two files to 'Compare'. These need not be of the same length, and as long as they are in different folders, can have the same name. When you select 'Compare', the files will first be checked to see if they are identical, and if so, you will not be able to continue and display them.



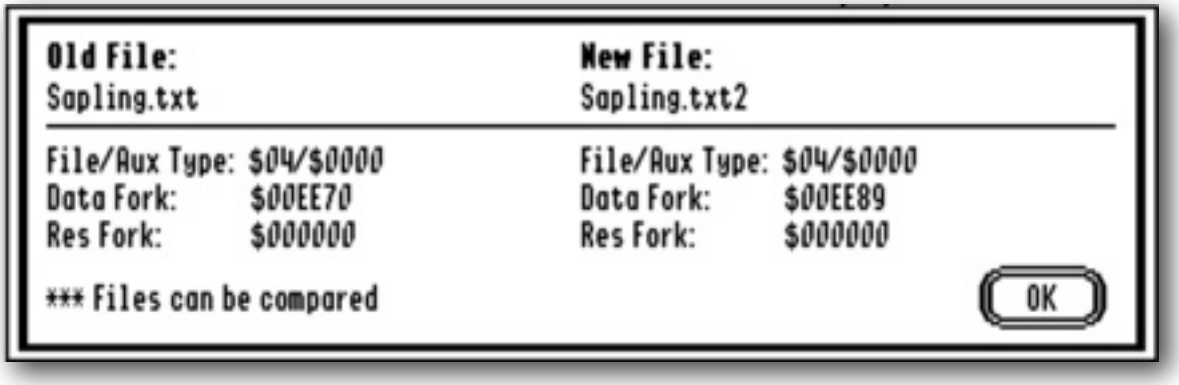
Swap

Swaps the two file entries in the dialog. As only the 'New' file, which displays in the bottom fields, can have an 'Offset' applied to it, a swap can be useful if you need to create an 'Offset' within the other file.

Check Files

Checks whether the two files are identical, whether they have valid data in the selected forks, and whether they can be 'Compared'. A display shows relevant information for the two files.

This same check is made when you select 'Compare', so 'Check Files' is a good way of first seeing whether two files can be 'Compared':



Compare

Compares the two files, displaying them in the two halves of the display, with the 'Old' file in the top half, and the 'New' file in the bottom half. The first difference that is found between the two files will be highlighted in both files.

Next Difference

Finds the next difference between the two files from the cursor point onwards, and highlights the differences.

Edit File

Switches into 'Edit' mode, for the active file, with any highlighted text selected. Click in either the top 'Old' file, or the bottom 'New' file, to make that file active.

Return from Edit

'Closes' the 'Edit' window, and returns to 'Comparing' the two files, with any highlighted text selected.

Setting an Offset

Two differing versions of a file, might mean that one may have bytes that differ from the other, or it could mean that one file may have had data inserted at various points, thus offsetting what may well be matching data later in that file.

To set an 'Offset' between the two files, simply select the range of data you wish to jump over, and press the OA-T keys. The active cursor should be in the bottom field for this function to work correctly. Sometimes the selected data from the 'Next Difference' is all that is needed. The start and end range of the selected data will then be used to calculate an 'Offset', and the bottom 'New' file will now show that 'Offset'. You can then continue to find any differences between the two files. Selecting 'Offsets' is cumulative, so you can continue to add more 'Offsets' as you go along. You can also make a selection in the top field, and if it is shorter than the current 'Offset' in the bottom field, and press OA-T, it will be subtracted from the total 'Offset'.

Holding the Open Apple key down while scrolling with the scroll bar, will freeze the 'Old' file, and create an 'Offset' to the 'New' File in the lower box.

To clear any accumulated 'Offsets', just select 'Undo' from the Edit menu, or press the OA-Z keys. The screen will then be redrawn without the 'Offset'.

Any 'Offset' will be respected for the functions such as 'Editing', 'Disassembling,' 'Dumping', and 'Printing'. Just make sure that the field you want to work on is active, by first clicking in that field.

Appendix

Keyboard Navigation

The editable fields use standard TextEdit controls, and respond to these keystrokes:

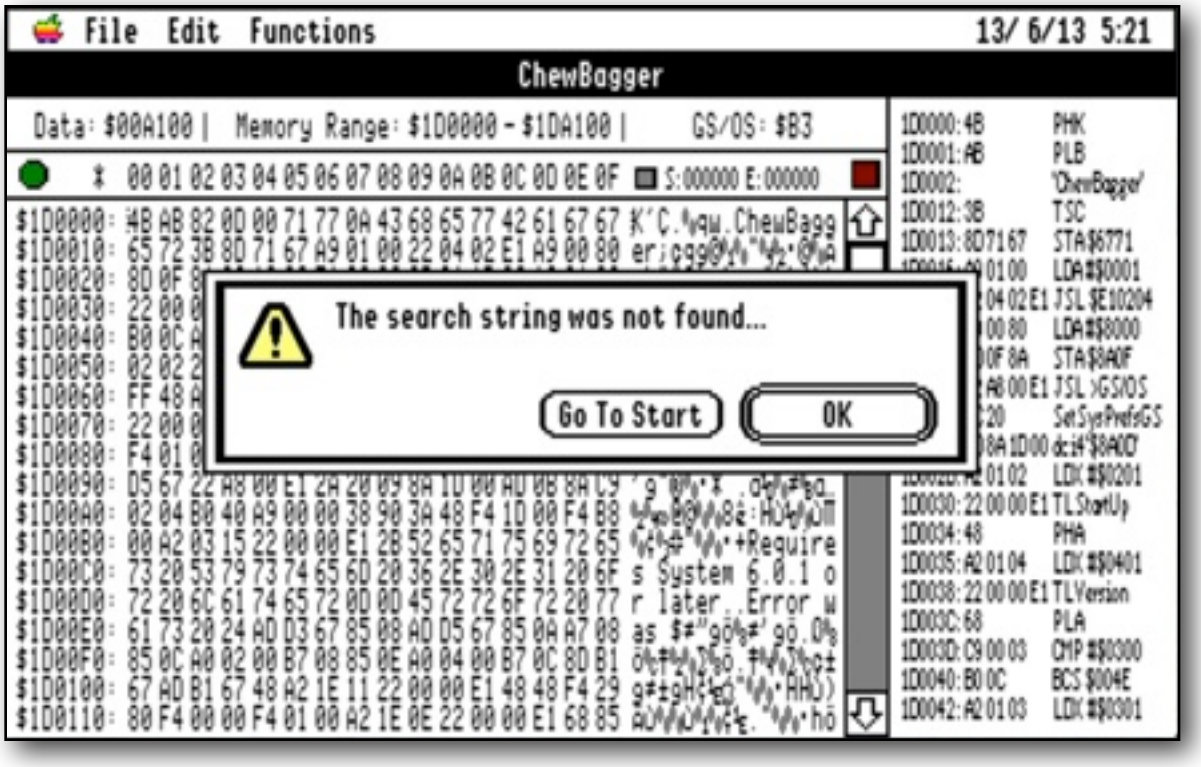
<left-arrow> moves one byte left.
<control><left-arrow> moves one byte left.
<command><left-arrow> moves one byte left.
<option><left-arrow> moves to start of line.
<right-arrow> moves one byte right.
<control><right-arrow> moves one byte right.
<command><right-arrow> moves one byte right.
<option><right-arrow> moves to end of line.
<up-arrow> moves up one line.
<control><up-arrow> moves up one line.
<command><up-arrow> moves to start of page, or if at start, up one page.
<option><up-arrow> moves to start of document.
<down-arrow> moves down one line.
<control><down-arrow> moves down one line.
<command><down-arrow> moves to end of page, or if at end, down one page.
<option><down-arrow> moves to end of document.

In addition, if you are using an extended keyboard:

<home> moves to start of document.
<end> moves to end of document.
<page-up> moves to start of page, or if at start, up one page.
<page-down> moves to end of page, or if at end, down one page.

There are some other keystrokes that do not show on the menus that ChewBagger recognises:

OA-B	Sets the 'Start' marker.
OA-E	Sets the 'End' marker.
OA-K	Clears the 'Start' and 'End' markers.
OA-H	Toggles the display of high bit characters in the ASCII field.
OA-T	Uses the selected text to set an 'Offset' in the Compare window.
OA-Y	Shows the current Disassembly selection in the Data field.
OA-.	Stops a disassembly.



Extras

Problems

Hopefully you will have none, but if you do, and they cannot be answered by reading these notes, please contact me on:

spectrumdaddy@speccie.uk

Other information

If you do not already know about Spectrum™, please drop by my home pages and read more. Apart from all the other wonderful things it does, Spectrum™ offers many useful tools for processing files, such as post processing text files that you have received that may have obstinate formatting.

Spectrum™ is now Freeware, and with all my other applications, is available from my web site:

http://www.speccie.uk

Someone once said to me, 'Spectrum™ does everything!'