X E D

TEXT EDITOR FOR THE COLOR COMPUTER USING OS9

by Eric Dokken

Microtech Consultants Inc.

Installation

Installing XED consists of copying the actual program XED onto your OS9 system disk, and also copying one of the XCODES modules supplied onto the system disk.

To copy XED to your system disk do the following:

For multi drive systems

Put the XED disk into drive 1 and your system disk into drive 0, and enter the following:

```
COPY /D1/XED /D0/CMDS/XED
```

For single drive systems

With your system disk in drive 0 type the following:

```
CHX /D0/CMDS
LOAD COPY
```

Now put the XED disk into drive 0 and type:

```
COPY /D0/XED /D0/CMDS/XED -S
```

Follow the prompts, putting the XED disk in drive 0 when the prompt asks you to insert the source disk, and putting your system disk in the drive when asked to insert the destination disk.

In addition to XED itself, you must also copy one of the XCODES modules onto your system disk. XED requires an XCODES module to define the screen that you are using. Four XCODES modules are supplied with XED. These are for the different types of screens. Modules included with XED include:

XCODES - Module to be used with the standard text screen. Sets screen for 31 characters per line by 16 lines

XCODES.OPAK - Module to be used with O-PAK high res screen. XED automatically adjusts to the number of characters per line and number of lines that O-PAK is currently set to.

XCODES.XS - Module to be used with XSCREEN. XED automatically adjusts to the number of characters per line that XSCREEN is currently displaying.

XCODES.WP - Module to be used with WORDPAK 80 column board. (Note: You must be using the WORDPAK OS9 driver)

When XED is executed it looks for a file named XCODES, so you must copy one of the xcodes modules into your execution directory under the name of XCODES. For example, if you will be using XED with WORDPAK, you must copy the XCODES.WP file under the name XCODES by typing the following:

For multi-drive systems

Put the XED disk in drive 1 and your system disk in drive 0 and type:

```
COPY /D1/XCODES.WP /D0/CMDS/XCODES
```

For single drive systems

With your system disk in drive 0 type the following:

```
LOAD COPY
```

Now put the XED disk into drive 0 and type:

```
COPY /D0/XCODES.WP /D0/CMDS/XCODES -S
```

Follow the prompts, putting the XED disk in drive 0 when the prompt asks you to insert the source disk, and putting your system disk in the drive when asked to insert the destination disk.

If you will be using XED with more than one screen type, you can copy all of the xcode modules you will be using under their original names (eg. XCODES.XS, XCODES.WP) and LOAD the module you want before executing XED. This will make XED use that module because all of these modules have module names of XCODES, and XED will use the XCODES module that is in memory if it is already there.

Execution

The command line used to execute XED is of the following form:

1) XED <newfile> [#mem]
2) XED <oldfile> [<newfile>] [#mem]

Form 1) is used when you want a file to be created. <newfile> is created and all edited text will be put into that file.

Form 2) is used to update an existing file. If <newfile> is specified, then the edited text from <oldfile> is put into a new file named <newfile>. If a file already exists with the name <newfile>, it is deleted. <oldfile> is not altered, so after the editing session you will have two files: <oldfile> with the original text, and <newfile> with the edited text.

If <newfile> is not specified, a scratch file called SCRATCH..XED# is created (the '#' character represents a number which is the process number of the process) and the edited text from <oldfile> is put into this file. After you quit editing <oldfile> is deleted, and SCRATCH..XED# is renamed to <oldfile>. When executing XED with this form of command, RENAME must be on the system disk in order to do the renaming. If RENAME is not on the system disk, an error will occur, and the result will be that <oldfile> will be deleted and the edited text will be in the file SCRATCH..XED#. If this should happen, just put in a system disk with RENAME on it, and rename SCRATCH..XED# to <oldfile>.

The '#mem' option is the standard OS9 option to increase the data size used for the process. If a '#mem' is not specified, then the default data area size will have about a four and a half k buffer area. If you need a larger memory buffer, you may specify a '#mem'. For example, including "#10k" on the command line will give you a memory buffer of about 8.5k, or specifying "#20k" will give a memory buffer of about 18.5k.

If the file being edited is larger than the buffer size, then the buffer will be filled only to within 400 bytes of its capacity. This will leave some room for insertions. To edit the remaining part of the file not in the memory buffer, you must use the **<CLEAR> C,M** command (explained later in the Command Mode chapter) to write out part of the buffer to the output file and read in more of the file.

Using XED

The Status Line

When using XED, the top line of the screen is used as a horizontal index line and as a status line. Every ten characters on the line there is a digit indicating the horizontal location, 1=column 10, 2=column 20, ... , 7 = column 70. Also every five positions there is a '+'. Tabs are indicated on this line by an 'X'.

The last three positions of this index/status line are reserved for indicating the status of XED. The third position from the end will have either an 'O' or an 'I'. The 'O' signifies you are currently in overwrite mode, while the 'I' indicates you are in insert mode.

The second position from the end indicates whether or not you are currently in automatic indent mode. If this character is a 'D', automatic indent mode is on. If this character is a space ' ', automatic indent mode is off.

The last position on the index line indicates whether wordwrap mode is on. If this character is a 'W', wordwrap mode is on. If its a space ' ', wordwrap mode is off.

Turning these modes on and off will be described in the section on command mode commands.

Displayed Carriage Returns

XED handles carriage returns just as any other character. Because of this they are also displayed on the screen. This makes it easier to insert and delete them because you can see where the are.

Carriage returns will look different depending on what screen you are using. With the normal text screen, carriage returns are displayed as a graphics block. With XSCREEN and WORDPAK carriage returns are displayed as a graphics character filling the bottom half of a character. And on O-PAK carriage returns are displayed as a shaded character (ASCII code 127).

2. EDIT MODE

In the normal text editing mode special control keys are used to move the cursor, insert text, delete text, and go to other command modes. These keys are the arrow keys and control keys obtained by pressing one of the letter keys (a..z) while holding down the **<CLEAR>** key.

These control or **<CLEAR>** keys, may be kind of difficult to remember their functions of at first because they are not very mnemonically oriented. This is because XED uses the standard keyboard driver supplied with the Color Computer OS9, and many of the control keys are the same as the special function keys. For example, it would have been nice to use **<CLEAR> I** as the control key for inserting text, but this has the same code as the **<RIGHT ARROW>** key.

Even though the control keys are not very mnemonic, it should not be too difficult to use once you have used it for a while. All the control keys for inserting and deleting text are pressed with the left hand, along with the right hand pressing the **<CLEAR>** key, making them all easy to press.

The control keys are grouped into five groups:

1. Moving the cursor. Using the arrow keys.
2. Inserting text. Using **<CLEAR> A,W,** and **T.**
3. Deleting text. Using **<CLEAR> D,F,G,** and **V.**
4. User defined keys. Using **<CLEAR> K,N,O** and **U.**
5. Going to other command modes. Using **<CLEAR> C** and **B.**

Moving the cursor

The cursor is moved around the text by using the arrow keys, alone and in combination with <SHIFT> and <CLEAR>. The cursor is moved with the following control keys:

<UP ARROW> - Move the cursor up one line.

<DOWN ARROW> - Move the cursor down one line.

<LEFT ARROW> - Move the cursor one position to the left.

<RIGHT ARROW> - Move the cursor one position to the right.

<SHIFT><UP ARROW> - Move the cursor to the top of the screen.

<SHIFT><DOWN ARROW> - Move the cursor to the bottom of the screen.

<SHIFT><RIGHT ARROW> - Tab right. Move the cursor to the next tab position to the right. If there are no more tabs to the right, go to the end of the line. If the cursor is at the end of the file, spaces are added up to the tab position.

<SHIFT><LEFT ARROW> - Tab left. Move the cursor to the first tab stop to the left of the cursor.

<CLEAR><RIGHT ARROW> - Move the cursor to the right end of the line.

<CLEAR><LEFT ARROW> - Move the cursor to the left end of the line.

<CLEAR><UP ARROW> - Move the screen up to the previous page of text.

<CLEAR><DOWN ARROW> - Move the screen down to the next page of text.

Inserting text

These control keys are used to insert text when you are in overwrite mode. When you are in insert mode, these control codes still work, but they are just not needed.

<CLEAR> A - Insert (Add) character. This control key inserts a space at the current cursor position. The character under the cursor, and all remaining characters of the file are moved up one position.

<CLEAR> W - Insert line. This control key inserts 80 spaces at the current cursor position. All text from the cursor on is moved up 80 characters.

<CLEAR> T - Insert paragraph. This control key inserts 400 spaces at the current cursor position. All text from the cursor on up is moved up 400 characters.

Deleting Text

These control keys are used to delete text from the file:

<CLEAR> D - Delete character. The character under the cursor is deleted.

<CLEAR> F - Delete spaces. Deletes all spaces from the current cursor position up to the first non-space character. This command is used to close up open space created with **<CLEAR> W** or **<CLEAR> T.**

<CLEAR> G - Delete word. Deletes all alphabetic characters (a..z, A..Z) from the current cursor position up to the first non-alphabetic character. If the first non-alphabetic character is a space, then the space is deleted, otherwise the first non-alphabetic character is not deleted.

<CLEAR> V - Delete line. Deletes all characters from the current cursor position up to the next carriage return. The carriage return is not deleted. It is important to note that all characters up to a carriage return are deleted, and this line may span more than one physical line as displayed on the screen.

User Defined Keys

These control keys are definable by the user. They are defined using one of the command mode commands.

<CLEAR>k -User defined key.

<CLEAR>N -User defined key.

<CLEAR>O -User defined key.

<CLEAR>U -User defined key.

Going to other Command Modes

These control keys will bring you out of the edit mode into one of the command modes.

<CLEAR>C -Go to command mode. All of the commands are entered in this mode, such as finding and replacing text.

<CLEAR>B -Go to block command mode. All of the block manipulation commands are entered from this mode.

3. COMMAND MODE COMMANDS

All of the control key commands in the edit mode are for moving the cursor, inserting text, or deleting text. The rest of the functions of XED, other than block manipulations, are executed by entering the command mode.

The command mode is entered by pressing **<CLEAR> C** from the edit mode. An alternative way of entering the command mode is by pressing **<SHIFT><BREAK>**.

After pressing **<CLEAR> C**, you will see the following prompt on the status line:

COMMAND? ('?' for help)?

You are now in the command mode. At the COMMAND? prompt you are to enter a command. A command is a letter (a..z) which can be in either upper or lower case, or a questiong mark '?' for help. If you do not want to do a command, you may press **<BREAK>** to exit the command mode and get back to the edit mode. Any illegal command will also return you to the command mode.

The following are the commands that are available through the command mode:

<CLEAR> C,? - List available command mode commands. This command will display a list of the commands that are available in the command mode and you will be kept in the command mode. You may then examine the list of commands and press the command you want, or you can press **<BREAK>** to get back to the edit mode.

<CLEAR> C,H - Help, list edit mode control key commands. This command will list out the control key commands that are available while in the edit mode, such as **<CLEAR> A** for inserting a character.

<CLEAR> C,T - Top of Text. This command will move the cursor to the beginning of the buffer. You should be aware that the cursor is only placed at the top of the memory buffer. Since XED can edit a file that is larger than the buffer, if you are editing a large file near the end, doing the T command will only bring you to the top of the buffer, which may not be the top to the actual file.

<CLEAR> C,B - Bottom of Text. This command will move the cursor to the bottom of the text buffer. Again, since you can be editing a file that is larger than the buffer, executing B may not bring you to the bottom of the actual file.

<CLEAR> C,F - Find text string. This command will allow you to find a string of text characters in the text buffer. You will be faced with the following prompt:

String to find?

You may enter up to 35 characters. The characters can be any of the standard displayable ASCII characters. In addition, you may also search for carriage returns by pressing the **<ENTER>** key. Carriage returns will be displayed just as they are displayed in the text, except they will be in inverse (except when using the normal text screen).

If you make an error when entering the text string, you may press the **<LEFT ARROW>** to backspace over the previous character.

When you are finished entering your search string, terminate it by pressing either **<BREAK>** or **<CLEAR><BREAK>**.

After entering the text string, XED will search for it in the memory buffer. If an occurence of this text string is found, XED will display the cursor just after it. If no occurence of it is found the message "None found PRESS BREAK" will be displayed on the status line. You must press **<BREAK>** to get back to the edit mode.

XED has kind of a unique way of searching for upper and lower case letters. When typing in letters, lower case letters will match both lower and upper case letters. While any upper case letters entered will only match upper case letters.

For example:

cat will match cat, CAT, Cat, CAt

Cat will match CAT, Cat, CAt
but not cat

CAT will only matchCAT

A special character when searching for text is the question mark ('?'). It is a wildcard character, it will match any character.

For example:
h?t will match hat, hit, hot, and hut
b??k will match book, back, and buck

Because the question mark is a wildcard character, you may not try to search for the question mark character itself.

<CLEAR> C,R - Replace text string. This command will find a specified text string and then replace it with another specified text string.

When you enter this command you will first be given the prompt:

String to find?

You must now enter the search string exactly as described for the F command previously described.

You will then be given the prompt:

Replace with?

You must enter a text string that the first string is to be replaced with. This is entered the same way as the search string, with a limit of 35 characters and by pressing **<BREAK>** or **<CLEAR><BREAK>** to terminate the string.

After entering the replace string, you will be prompted with:

Replace how many times?

You can then enter how many times to do the replacing. You may enter a number from 1 to 32767. Pressing just **<ENTER>** at this prompt will cause the replacing to occur 1 time. If you want to replace all occurences of a string, enter a large number such as 9999.

After specifying how many times to do the replacement, XED will search the buffer for the search string and replace it with the replace string the specified number of times. When it is finished it will display the message "# Replaced: x PRESS BREAK", where 'x' is the number of times the string was replaced. Press **<BREAK>** to get back to the edit mode, and the cursor will be placed after the last string that was just replaced.

<CLEAR> C,V - Verified replace. This command is similar to the replace command as previously described, except you are not asked how many times to do the replacing. Instead, the cursor is placed before each occurence of the search string and the message

Replace it? (y,n,r,q) ?

is printed on the status line. You then have four responses:

Y - Yes, do the replacing and then go on to the next string.
N - No, do not replace the string, go on to the next string.
R - Replace the string and then go on and replace the remaining strings without verification just as if the **R** command had been given to replace all occurences of the search string.
Q - Quit verified replace command without replacing the string, leaving the cursor where it is and going to the edit mode.

<CLEAR> C,A - do find/replace Again. This command will repeat the last Find, Replace, or Verified replace command given, using the same search and replace strings.

<CLEAR> C,X - Set tab at current cursor position. This command will set a tab stop at the cursor position that the cursor was at when going into the command mode.

Tab stops are represented on the status line by an 'X'.

<CLEAR> C,U - Unset tab at current cursor position. This command will clear the tab stop that is at the cursor position that the cursor was at when going into the command mode. If no tab stop was at that position, this command will have no affect.

<CLEAR> C,K - Kill (clear) all tab stops. This command will clear all currently defined tab stops. All of the X's on the status line representing these tab stops will be cleared.

<CLEAR> C,O - Execute OS9 command. This command allows you to execute an OS9 shell command from within XED, and to even start a whole new shell. You will be prompted with the following:

Shell Command?

You can then enter a shell command, the screen will be cleared and the command executed. After the command is finished, the message 'PRESS BREAK' will be displayed. You must press **<BREAK>** to get back to the edit mode.

For example, if you press **<CLEAR> C,O,** and then at the prompt you enter **DIR,** the screen will be cleared and the directory will be dislplayed with the message 'PRESS BREAK' after it.

You can start a whole new shell by just pressing **<ENTER>** at the 'Shell Command?' prompt. A new shell will be started, and you will be faced with a new OS9 shell prompt:

Shell
OS9:

You can then work with OS9 as you normally do, with the XED editor waiting for you to come back. **One very important warning,** do not delete or alter any of the files that XED is currently using, this includes the <input file> and <output file> specified when executing XED. And if you entered XED with no <output file> named and XED used the SCRATCH..XED# file, you must leave this file alone also. You should also not change diskettes, because XED will be expecting everything to be the same when it resumes.

When you are done using the OS9 shell, you can get back to XED by pressing **<CLEAR><BREAK>** at the 'OS9:' prompt. The message 'PRESS BREAK' will be displayed, and pressing **<BREAK>** will return you to XED right at the same point where you left it.

You can even use XED to edit another file while you are in the middle of editing a file with XED. This can be done by starting a new shell (by pressing **<ENTER>** at the "Shell Command?" prompt) and executing XED, or by executing XED directly from the "Shell Command?" prompt. But you must be very careful you do not modify the files previously mentioned.

<CLEAR> C,D - Toggle auto indent mode. This command will toggle the automatic indent mode. When automatic indenting is on, a 'D' will be displayed in the next to the last character of the index/status line at the top of the screen. If it is off, a space will be displayed in that position.

When automatic indenting is on, every time you hit **<ENTER>**, the next line will automatically be indented to line up with the previous line. This is very convenient when typing in structured language programs such as PASCAL or C.

When XED is first started, it comes up with auto indent off.

<CLEAR> C,I - Toggle insert/overwrite modes. This command will toggle between insert mode and overwrite mode. When you are in insert mode, an 'I' will be displayed in the third character from the right end of the index/status line. If you are in overwrite mode an 'O' will displayed in that position.

In overwrite mode, everything you type will overwrite the characters that are currently under the cursor. If you want to insert something you must use the control keys to insert a character, line, or paragraph. If you are at the end of the text, the text will be expanded as you type.

In insert mode, everything you type will be inserted at the current cursor position. The character under the cursor and everything past the cursor will be moved up one character to make room for the character being typed.

XED comes up in overwrite mode, so if you prefer insert mode you must change to it with this command.

<CLEAR> C,W - Toggle Wordwrap mode. This command toggles the wordwrap mode on and off. When wordwrap mode is on a 'W' will be displayed in the last character of the index/status line. If wordwrap mode is off, a space is displayed in that position.

When wordwrap mode is on, words will not be split across two lines. Instead, if a word crosses from one line to the next, it will automatically be brought down to the start of the line. Wordwrap will automatically adjust any lines that are changed by insertions or deletions. Wordwrap does not change the text in any way. It does not insert carriage returns at the end of lines or anything else to physical change the text. It just changes the way the text looks on the screen.

Initially XED starts with wordwrap mode off.

<CLEAR> C,Q - Quit editing. This command will end the editing session. The memory buffer will be written to the output file, and if the file was larger than memory and there is more text in the input file, then that text is copied out to the output file. The files are then closed and if you edited an already existing file and only specified the input file name, the input file is deleted and the output file is renamed to the same name as the input file.

<CLEAR> C,Z - Abort editing. This command will abort the editing session. The memory buffer will not be written to the output file, and no files will be deleted or renamed. The original file that was to be edited will be unchanged. Use this command if you decide you don't want to make any changes to the file.

<CLEAR> C,P - Quit editing and Print file. This command will end the session the same as **<CLEAR> C,Q but instead of exiting to the OS9 prompt,** it automatically executes the XPRINT text formatting program (XP). XPRINT is a full featured text formatter that is a part of the XWORD word processing system and is also available separatly from Microtech Consultants Inc. For this command to work you must have XPRINT installed in your current execution directory.

<CLEAR> C,S - Size of text. This command will tell you the current size of the text in the memory buffer and how much memory is left by printing the following line on the status line:

sz:#### Free:#### PRESS BREAK

The # characters represent the numbers that will actually be printed. You must press <BREAK> to get back to the edit mode.

<CLEAR> C,C - Clear memory buffer. This command will clear the memory buffer from the current cursor position to the end of the buffer.

You will be prompted with:

Clear to end of buffer?

Responding with 'y' or 'Y' will cause the clearing to take place. Any other responce will return you to the edit mode with nothing being cleared.

<CLEAR> C,M - get More text. This command will write out all of the text from the beginning of the memory buffer to the current cursor position to the output file, and read in text from the input file until the memory buffer is full (withing 400 bytes of being full).

This is how text files that are larger than the memory buffer can be edited. You first edit the first portion of the file, then you place the cursor at a point past all of the text that is finished being edited and execute **<CLEAR> C,M**. That text will be written out to the output file and more text will be read in. This is done until all of the file is edited.

This command only goes forward through the file. If you execute this command, you can not go back and re-edit the text that has been written out. To do this you must quit and execute XED all over again.

<CLEAR> C,L - change Line length. This command will allow you to change the length of the lines displayed on the screen. Executing this command will give the following prompt:

Line width?

You may enter a value to set the line length. This length is limited by the screen size. For example, if you are using the WORDPAK 80 column board, you can select a maximum line length of 79 characters, with the normal text screen the maximum is 31 characters. The minimum allowed is four.

By pressing **<ENTER>** at the prompt you can find out what the current line length is set at.

<CLEAR> C,N - get liNe number of cursor. This command will display the line number of the line that the cursor is currently on.

<CLEAR> C,G - Get file. This command enables you to read a text file into the middle of the file being edited. You will be prompted to enter a filename. The file will be taken from the current data directory unless a full pathlist is specified.

The file must be smaller than 32000 bytes long, and there must be enough free space in the memory buffer to hold the entire file. If either of these cases are not true, the file will not be loaded.

There is a corresponding command to write out a portion of the text to a file. This command is a block command, and is explained in the next section.

<CLEAR> C,E - dEfine programmable keys. This command enables you to program the four programmable control keys: **<CLEAR> K,N,O,** and **U.** You will be prompted with:

Program what key? (K,N,O,U)?

After specifying your choice you will be prompted with:

?

You can now enter the keystrokes that define your programmable key. You may enter up to 30 keystrokes, these may be any of the standard ASCII characters and any of the control keys (except for **<CLEAR> <BREAK>**). When you are finished entering, terminate it by pressing **<CLEAR> <BREAK>.**

All non-displayable codes will be displayed as '.'. You may not use the **<LEFT ARROW>** to backspace, since the **<LEFT ARROW>** is also a valid keystroke that may be programmed. If you make a mistake you must terminate it by pressing **<CLEAR><BREAK>** and re-program it.

4. BLOCK COMMAND MODE

All of the block manipulation commands are entered from the block command mode. The block command mode is entered by pressing **<CLEAR>B** from the edit mode.

Once in the block command mode, you will see the following prompt:

Block Command? ('?' for help)?

Pressing '?' will list out the valid block commands, which are:

T - Mark top of block
B - Mark bottom of block
U - Unmark block
C - Copy block
D - Delete block
M - Move block
P - Print block
S - Save block to disk

Pressing **<BREAK>**, or an invalid response will exit the block command mode and put you back in the edit mode.

When a block is defined it will be displayed in inverse characters (except when using the standard 32 character/line text screen). When a block is defined, you can still edit freely, both inside the block and outside of it. Any insertions inside the block will automatically expand the block, and any deletions inside the block will automatically contract the block. Any insertions or deletions outside of the block will adjust the block keeping it correctly defined. One exception to this is that you can not make deletions that cross the block boundaries (top or bottom). For example, you can not delete a line (using the **<CLEAR>V** command), that goes from outside the block to inside the block, or from inside to outside.

<CLEAR> B,T - Mark top of block. This command will mark the top of the block at the current cursor position. If no block is currently marked, or the current cursor position is below the bottom of the current block, then the bottom will also be marked at this position.

<CLEAR> B,B - Mark bottom of block. This command will mark the bottom of the block at the current cursor position. If no block is currently marked, or the current cursor position is above the top of the current block, then the top will also be marked at this position.

<CLEAR> B,U - Unmark block. This command will unmark the current block.

<CLEAR> B,C - Copy block. This command will copy the block that is currently defined into the place where the cursor is at. The text will be opened up to allow this copy of the block to fit in.

If there is not enough space in the memory buffer to hold the copy of the block, you will get an 'Out of memory' error. Trying to copy a block within itself will cause a 'BLOCK ERROR'.

After the execution of this command, the original block will remain marked.

<CLEAR> B,D - Delete block. This command will delete the block that is currently defined. After executing this block command, you will be prompted with:

Delete block?

Pressing 'y' or 'Y' will delete the block. Any other response will abort the delete command and return you to the edit mode.

Trying to delete an undefined block will cause a block error.

<CLEAR> B,M - Move block. This command will move the currently defined block to the current cursor position. This is done by first copying the block to the current cursor position, and then deleting the original block. There must be enough free memory in the buffer to do the copy or an 'Out of memory' error will occur. If no block is currently defined, a block error will occur.

After the execution of this command the block will be unmarked.

<CLEAR> B,P - Print block. This command will output the currently defined block out to the printer. It does no formatting when doing this, it just sends one character after another, so the output to the printer may not look like the text on the screen.

This command sends the block to the device descripter /P, so /P must currently be in memory.

<CLEAR> B,S - Save block to disk. This command will save the currently defined block to a specifed file. You will be prompted for the filename. The file will be put in your current data directory unless a full pathlist is specified.

Trying to save an undefined block will cause a block error.

INDEX