

Maximus-CBCS Version 2.00 Technical Reference Manual
Copyright 1990, 1991 by Scott J. Dudley. All rights reserved.
Created November 3, 1991.

Documentation produced by Bob Davis and Scott Dudley,
with Don Dawson and Hubert Lai.

TABLE OF CONTENTS

LICENCE	1
CONTROL FILE REFERENCE	6
MAX.CTL Reference	6
SILT Directives	7
System Section Keyword Listing	9
Equipment Section Keyword Listing	16
Matrix/EchoMail Keyword Listing	20
Session Section Keyword Listing	24
LANGUAGE.CTL Reference	49
Language Section Keyword Listing	49
READER.CTL Reference	50
Reader Section Keyword Listing	50
COLOURS.CTL Reference	52
Colours Section Keyword Listing	52
MSGAREA.CTL Reference	58
Alphabetical Keyword Listing	58
FILEAREA.CTL Reference	65
Alphabetical keyword listing	65
Barricades and Extended Barricades	67
Using Privilege Level Modifiers	68
MENUS.CTL Reference	70
Global Menu Options	70
Menu Option Modifiers	72
Menu Option Format	75
Alphabetical menu option listing	76
Linking Menu Options	89
PROTOCOL.CTL Reference	91
Alphabetical Keyword Listing	91
Examples	94
MECCA LANGUAGE REFERENCE	96
Usage Guide	96
Colour Token Listing	99
Cursor Control and Video Tokens	102
Informational Tokens	104
Questionnaire Token Listing	109
Privilege Level Controls	112
Lock and Key Control	116
Conditionals and Flow Control	117
Multi-Line Token Listing	125
Miscellaneous Token Listing	126
EVENT FILE CONFIGURATION	133
APPENDICES	134
Appendix A: Common Problems	134
Appendix B: Error Messages	136
Appendix C: Operating with DoubleDOS	141

Appendix D: Command Line Switches	142
Appendix E: Local Keystrokes	147
Appendix F: User Editor Keystrokes	150
Appendix G: List of AVATAR Colours	152
Appendix H: Sample Batch Files	153
INDEX	159

LICENCE

Copyright 1991 by Scott J. Dudley. All rights reserved.
COMMERCIAL DISTRIBUTION AND/OR USE PROHIBITED WITHOUT WRITTEN
CONSENT FROM THE AUTHOR.

Noncommercial distribution and/or use is permitted under the
following terms:

- 1) You may copy and distribute verbatim copies of the Maximus source, documentation, and executable code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy a valid copyright notice "Copyright 1991 by Scott J. Dudley"; keep intact the notices on all files that refer to this Licence Agreement and to the absence of any warranty; PROVIDE UNMODIFIED COPIES OF THE DOCUMENTATION AS PROVIDED WITH THE PROGRAM; and give any other recipients of the Maximus program a copy of this Licence Agreement along with the program. You may charge a distribution fee for the physical act of transferring a copy, but no more than is necessary to recover your actual costs incurred in the transfer. Under no circumstances is Maximus to be distributed in such a way as to be construed as "value added" in a sales transaction, such as, but not limited to, software bundled with a modem or CD-ROM software collections, without the prior written consent of the author.
- 2) You may modify your copy or copies of Maximus or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:
 - a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change;
 - b) cause the executable code of such modified version to clearly identify itself as such in the course of its normal operation;
 - c) if the modified version is not a "port", but operates in the same hardware and/or software environment as the original distribution, make the original version equally available, clearly identifying same as the original, unmodified version;
 - d) cause the whole of any work that you distribute or publish, that in whole or in part contains or is a

derivative of Maximus or any part thereof, to be licensed at no charge to all third parties on terms identical to those contained in this Licence Agreement (except that you may choose to grant more extensive warranty protection to some or all third parties, at your option); and:

e) send the complete source code modifications to Scott Dudley at the addresses listed below, for the purpose of evaluation for inclusion in future releases of Maximus. Should your source code be included in Maximus, Scott Dudley retains all rights for redistribution of the code as part of Maximus and all derivative works, with appropriate credit given to the author of the modification;

f) You may charge a distribution fee for the physical act of transferring a copy, but no more than is necessary to recover your actual costs incurred in the transfer, and you may at your option offer warranty protection in exchange for a fee;

g) when distributing modified versions of Maximus, you must not change the name of the program or the official version number, except to append an identifier which indicates that modifications have been made. For ports to other operating systems, the following convention must be followed:

Maximus v<v>.<os>.R<r>

...where <v> is the official Maximus version number, <os> is the name of the operating system which the port runs under, and <r> (optional) is the OS-specific revision number. For example, the second OS/2 revision of Maximus 1.02 must have a version string in this format: `Maximus v1.02.OS/2.R2`

Similarly, modifications to Maximus which are designed to run under MS-DOS must also follow a naming convention. The version string must read:

Maximus v<v>.<i>.<r>

where <v> is the official Maximus version number, <i> is three initials (indicating your first, middle and last names), and <r> (optional) is the revision number of your modifications.

For example, a version of Maximus 2.00 modified by Joe T. SysOp must have a version string in this format: `Maximus v2.00.jts.1`

- 3) Mere aggregation of another unrelated program with this program and documentation (or derivative works) on a volume of a storage or distribution medium does not bring the other program under the scope of these terms.
- 4) You may copy and distribute Maximus and its associated documentation (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

- a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

- b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal shipping charge) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

- c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs.

- 5) You may not copy, sublicense, distribute or transfer Maximus and its associated documentation except as expressly provided under this Licence Agreement. Any attempt otherwise to copy, sublicense, distribute or transfer Maximus is void and your rights to use the program under this Licence agreement shall be automatically terminated.

However, parties who have received computer software programs from you with this Licence Agreement will not have their licences terminated so long as such parties remain in full compliance, and notify Scott Dudley of their intention to comply with this Agreement.

- 6) You may not incorporate parts of Maximus into a program which is NOT completely free for ALL users. If you wish to use Maximus in such a way, you must obtain written

permission from Scott Dudley before using any of the Maximus code.

- 7) You may not incorporate parts of Maximus into a FUNCTIONALLY SIMILAR program, including other bulletin board packages or tossers/scanners. If you are writing another BBS or remote host package, you must contact Scott Dudley before using any of the Maximus code.
- 8) The privileges granted above apply only to noncommercial users of the Maximus software. You are a NONCOMMERCIAL user only if you are running Maximus a private individual with no "sponsors", "backers", and only if your BBS is not making (or helping to make) a profit. You are a COMMERCIAL user if you make a profit from running your BBS. You are also a COMMERCIAL user if your BBS is being run by (or for) any corporation, government, company, foundation, church, school, or any other organization. You are also a COMMERCIAL user if your system is used to advertise such a commercial organization for the purposes of making a profit.

This licence only governs NONCOMMERCIAL users. If you are a COMMERCIAL user, you are not licensed to use or distribute this software without the prior written consent of Scott Dudley. If you wish to run Maximus as a commercial user, please see the section below on site licences.

- 9) This licence may be revoked by Scott Dudley without prior notice.

NO WARRANTY

BECAUSE MAXIMUS-CBCS IS LICENSED FREE OF CHARGE, WE PROVIDE ABSOLUTELY NO WARRANTY. EXCEPT WHEN OTHERWISE STATED IN WRITING, SCOTT DUDLEY AND/OR OTHER PARTIES PROVIDE MAXIMUS-CBCS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF MAXIMUS-CBCS, AND THE ACCURACY OF ITS ASSOCIATED DOCUMENTATION, IS WITH YOU. SHOULD MAXIMUS-CBCS OR ITS ASSOCIATED DOCUMENTATION PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT WILL SCOTT DUDLEY BE RESPONSIBLE IN ANY WAY FOR THE BEHAVIOUR OF MODIFIED VERSIONS OF MAXIMUS-CBCS. IN NO EVENT WILL SCOTT DUDLEY AND/OR ANY OTHER PARTY WHO MAY MODIFY AND REDISTRIBUTE MAXIMUS-CBCS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY LOST PROFITS, LOST MONIES, OR OTHER

SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS) MAXIMUS-CBCS, EVEN IF SCOTT DUDLEY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

You can contact the author at any of the addresses listed below:

FidoNet: 1:249/106
IMEXnet: 89:487/106
Internet: sjd@f106.n249.z1.fidonet.org
Cserve: >INTERNET:sjd@f106.n249.z1.fidonet.org
BBS: (613) 389-8315, 14.4K/HST

Surface mail:

777 Downing St.
Kingston, Ont.
Canada K7M 5N3

The author can also be reached through the EchoMail conferences called MUFFIN (Max support) and TUB (Squish support).

Sending correspondence via electronic mail is strongly preferred. However, if you expect to receive a reply via surface mail, please enclose a self-addressed, stamped envelope. Maximus users outside of Canada should include an international postal reply coupon instead of a stamp.

DO NOT ATTEMPT TO CONTACT THE AUTHOR BY TELEPHONE! VOICE SUPPORT WILL NOT BE PROVIDED FOR NONCOMMERCIAL USERS!

Please feel free to contact the author at any time to share your comments about this software and/or licensing policies.

Our thanks to Richard Stallman at the Free Software Foundation, Inc. and BBS Co. for most of the wording of this licence.

CONTROL FILE REFERENCE

MAX.CTL Reference

The main Maximus control file, MAX.CTL, is the most important part of your BBS. This section of the documentation is a reference for the control file; all of the keywords you can use in MAX.CTL are listed here in alphabetical order, subdivided by control file section. A sample control file is included in the Maximus distribution kit, and it's heavily commented, so you should not have any problems with making simple modifications. However, should you find some items in the control file to be unclear, you can turn to this chapter of the manual for more information.

In addition to the 11 main sections (System, Equipment, Matrix/EchoMail, Session, Messages, Files, Off-line Reader, Languages, Colours, Protocols and Menus), you can also define custom blocks of the control file for use by external programs.

You can start a custom section by placing the word "Section" on the first character of a line, followed by a one-word identifier for the section. The identifier should relate to the name of your program: for example, you might use the statement `Section MyRen' to start a custom section for a message-renumbering program. After you have defined a custom section, it can continue until the end-of-file, or until SILT encounters `End' as the first word on a line.

This provides for a convenient way to store an external application's information inside the main Maximus control files, without having to create another control file for each external application which is added. If you need to just sprinkle items throughout the control files, and place information inside of other area, menu, or section portions of the configuration file, then the `App' or `Application' keywords (see below) may be what you are looking for.

However, what follows is a description of all the standard SILT directives and sections which are required for ALL Maximus systems.

SILT Directives

Directives are commands which are interpreted only by the SILT compiler: they don't produce any output on their own, and simply modify SILT's operation. SILT can interpret several directives, in addition to all of the normal keywords, which are described later in this documentation. These directives can be used globally, in any of the configuration files. As of this date, SILT supports the following directives:

```
App <appname> [<appargs>...]
Application <appname> [<appargs>...]
```

Placing an `App' or `Application' statement into any of the control files will allow an external program to imbed application-dependent information inside the Maximus control files. <appname> should be no more than one word, but can include underscores. <appname> should also be a one-word name to identify the program which uses your identify the name of the external program which the information was designed to be used by. You can have zero or more <appargs>, which can indicate what your external program should do. For example, the following line:

```
App MyRen KillByDay 20
```

...might instruct an external message renumbering program called `MyRen' to delete messages from the current area, by date, and to keep only those messages which are less than 20 days old.

See also the topic on `Custom Sections', above.

```
Include <filespec>
```

The `Include' keyword is a SILT directive, and will not affect Maximus' behaviour. When SILT spots an `Include' keyword, it will try to open a file called <filename>, and process it just as if it were part of the current control file. This method is used to read in MSGAREA.CTL, FILEAREA.CTL and MENUS.CTL. You could easily incorporate all of the above into a single MAX.CTL, but that would be very large and unwieldy. Using this keyword, you can break MAX.CTL up into smaller, more manageable parts.

One limitation is that the `Include' keyword cannot be used inside any section in the control file. (In other words, it must be part of the main control file, outside of the

system, equipment, matrix, session, message and menu sections.) However, most sections can be broken down into parts, so if you use the following in MAX.CTL:

```
Session Section
    % Insert main session information here
End Section
```

```
Include Session.Ctl
```

...and set up SESSION.CTL to look like this:

```
Session Section
    % Other session information here.
End Section
```

...you'll be able to split the session section information (or any other section, for that matter) into two separate files.

```
Version14 <filespec>
Version17 <filespec>
```

These two keywords are provided for compatibility with previous and current Opus control files. SILT will generate these older .PRM files when writing the main MAX.PRM file, which makes it possible to run older Opus utilities. The `Version14' and `Version17' keywords will cause SILT to generate an Opus v14 PRM file and a v17 PRM file, respectively.

System Section Keyword Listing

Dos Close Standard Files

This keyword tells Maximus that it should close all files which are open before running any external programs. These files include the AREA.* area data files, and MAX.LOG. Unless you use this option, other utilities and programs will NOT be able to write to these open files. For example, if you wish to have your external program log information to MAX.LOG use this keyword.

File Password <filespec>

This keyword defines the location of the user file, which is where Maximus stores everything that it knows about each user. If you are just starting out with Maximus and do not yet have a user file, then start Maximus using the '-c' command line switch, and Maximus will create one for you.

Log File <log_name>

This keyword tells Maximus what you want your system log file to be called. The log file contains a record of WHO called your system, WHEN they called your system, and WHAT they did while on-line. If you don't specify a log file, then NO log will be generated. Note: You can also specify an override log file name on the command-line, via the '-l' switch, which is useful in a multitasking/multinode environment.

Log Mode <log_type>

This keyword tells Maximus how much information you would like to have inserted in the log file. Using 'Trace' for <log_type> will give you a detailed account of everything that the user did on your system, including information on the message areas which were entered, the files which were downloaded, etc. Since 'Trace' records quite a lot of information, the log file can become very large very quickly if your system is a busy one. Using 'Verbose' for <log_type> will still put a large amount of information in the log file, but not quite as much as 'Trace'. Using a log mode of 'Terse' specifies that you only want a bare-bones log kept, and only major events will be recorded, such as a user logging on and off, any errors encountered, etc.

Multitasker <type>

This keyword is only required if you are running Maximus under a DOS-compatible multitasking environment. <type> can be any of the following:

- DESQview
- DoubleDOS
- PC-MOS
- MSWindows
- NONE

If type NONE is selected, Maximus will attempt to autodetect the current multitasker. Maximus supports DESQview, DoubleDOS, PC-MOS, Windows 386/Enhanced Mode, MultiLink, or any other program which hooks the int 28H vector.

When running under OS/2, simply specify "none" for the multitasker. OS/2 is always loaded when running MAXP, so this option can be left alone.

Name <bbs_name>

This keyword tells Maximus what the name of your BBS is. It will be displayed when you start up Maximus, and it will also be used as a default for your EchoMail origin lines.

No SHARE.EXE

This keyword tells Maximus that you are running MS-DOS version 3.3 (or prior), and that you do NOT have SHARE.EXE loaded. (This keyword is only needed when you have enabled the 'Path IPC' keyword, and are using the multi-node chat.)

This keyword instructs Maximus to disable file and record locking, since some earlier DOS versions don't include it in the system core. (DOS versions 4.0 and above have record locking built in, and therefore SHARE.EXE is not required.) However, since SHARE can cause certain problems with other programs, Maximus allows you to use this command, as a way to use the multi-line features without loading SHARE.

WARNING This keyword should be used with EXTREME caution, since it disables the safety mechanism for the inter-process communications area. If you absolutely cannot run your system with SHARE loaded, then upgrading to DOS 4.0 is probably a better answer. This keyword should only be used as a last resort. NO GUARANTEES ARE MADE, AND YOU MUST USE THIS KEYWORD AT YOUR OWN RISK!

Path IPC <path>

This keyword defines the path to use for the inter-process communications area. This directory will contain several temporary files, used for inter-task communication between different copies of Maximus. (If you are running several copies of Maximus, and you want all of them to be able to talk together, then you must set this to point to the SAME directory, for all nodes.)

If you are running only a one-line system, then you can leave this path blank, or not declare it at all. However, if you are running two or more lines, and wish to enable the multi-line features, it is STRONGLY recommended that you set this to point to a RAMdisk. Maximus will write several files to this directory during the course of execution, and there will be an especially large number of disk accesses to this path when using the multi-node chat. Setting the IPC path to a normal directory will probably cause the chat to become sluggish, and will cause unwanted disk activity during a normal session.

It is suggested that the size of your RAMdisk be a minimum of about five kilobytes, multiplied by the number of nodes you wish to run. For example, if you are running a five-line system, then the size of your RAMdisk should be about 25 kb.

Also remember to load SHARE.EXE when you have the inter-process communications path enabled. See above for more details, in the description of 'No SHARE.EXE'.

Finally, you should read through the Multi-Line Operation Guide in the Maximus Operations Manual, before attempting to use any of the multi-line features.

Path Language <path>

This keyword tells Maximus where your language files are located. See the Language Section keyword listing.

Path Misc <path>

This keyword tells Maximus where your miscellaneous files are located. Aside from the *.BBS files, this is also where the F*.BBS, CF*.BBS, AF*.BBS, and SF*.BBS files are located. (Please refer to the 'Local Keystrokes' appendix for more information on these files.)

Path Outbound <path>

This outbound path is never used by Maximus itself. The only reason for having this option in the control is so that it can be inserted into the Opus v14 or v17 *.PRM files for compatibility with older programs. If you do not need the v14 or v17 *.PRM files, or if you do not have an outbound path, then you do NOT need to bother with this keyword.

Path System <full_path>

This keyword tells Maximus where the main system files are located. This is also the directory in which SILT will place the optional SYSTEM??.BBS files, if necessary. Please note that <full_path> must be a FULLY-QUALIFIED PATH, including the drive specifier and leading backslash. To illustrate, both of the following are okay:

```
C:\Maximus
D:\Bbs\Maximus
```

But neither of these are acceptable:

```
\Maximus
D:Bbs\Maximus
```

Path Temp <path>

This keyword gives Maximus the name of a directory which it can use as a temporary workplace. This is used for batch uploads and the 'Local Editor' option, among other things. It is NOT advisable to put anything in this directory, since Maximus can and will fiddle with anything it finds there.

Reboot

This keyword tells Maximus that you want the FOSSIL 'watchdog' feature to be activated when external programs are being executed. This basically means that if a user drops carrier (hangs up) while running an external program, the FOSSIL will reboot your computer. Presumably, your batch file will re-load your mailer or start Maximus in WFC mode. If you are using well-behaved external programs (i.e. those specifically written to run under a BBS), then you won't need this option, since the external program itself will detect when a caller has hung up. This option has no effect on Maximus' internal carrier checking, which is on all the time. If a user hangs up while Maximus is running,

Maximus will simply end and the batch file can restart the front-end mailer.

Snoop

This keyword controls the default state of the `Snoop` setting. If you do not use this keyword, snoop is turned off by default. When snoop is turned ON, then the local console will look exactly like what your user is seeing. But when snoop is OFF, then the local screen will be just a status-line display of events. Turning snoop off will greatly increase performance on a multitasking system, since console output chews up a lot of system time.

SysOp <name>

This keyword specifies your name as it will appear to users. Please note that placing your name here doesn't actually confer any SysOp powers upon you. Only users with a privilege level of `SysOp` can obtain SysOp capabilities. The name used in this keyword will be used for display purposes only, such as when Maximus asks the user if he/she wants to leave a log-off comment to <name>.

Task <task_no>

This keyword specifies the task number for the copy of Maximus which uses this control file. If you are not running more than one copy of Maximus at a time, then you have nothing to worry about, and can just comment this keyword out. However, on a multi-line system, several conflicts may occur with filenames and naming conventions, which is why a separate task number is required for each copy of Maximus that will be running concurrently. As long as you specify a different task number for each copy of Maximus, you should not have any problems running in a multi-line environment.

Note! You can also specify an override task number from the command-line, using the `-n` switch.

Finally, if you are running a multi-line system, make sure that ALL of your task numbers are non-zero. If you set the task number of a particular node to zero, the node will assume that it is running in single-line mode, and won't be able to communicate with the rest of the nodes.

Video <mode>

This keyword specifies the video mode that is to be used for the local display. Maximus supports six options for <mode>:

FOSSIL. Maximus will send all local output through the normal communications FOSSIL. This is the slowest method, but the most portable. (Maximus neither needs nor uses a "Video FOSSIL".)

DOS. Maximus will send all local output through the DOS stdout hooks. This is reasonably fast, and is reasonably portable.

FAST. Maximus will send all local output through the DOS 'fast character output' hook, which is interrupt 29h. This function is usually much faster than 'Video DOS'; however, some alternate operating systems (such as PC-MOS) do not support this option. On the other hand, almost all normal versions of MS-DOS and PC-DOS do support this option.

BIOS. Maximus will send all output via the PC's video BIOS. It is almost as fast as the IBM mode, but will work on some PCs where the IBM mode will not. Maximus window and status line support is available with this mode.

IBM. Maximus will write directly to the video hardware. This is lightning-fast and is the quickest way to run Maximus, but it may not be compatible with some hardware or multitaskers. This mode has been tested and works with DESQview, TopView, and DoubleDOS. Whether or not it works with other multitaskers depends on how they handle the screen. Using this option also means that you do not have to load ANSI.SYS to run Maximus, although it may still be required in order to use your mailer or other programs which may rely on it. Please see the DoubleDOS appendix for more information on using this video mode with DoubleDOS. Maximus window and status line support is available with this mode.

IBM/snow. This mode is identical to 'Video IBM', except that Maximus will wait for the vertical retrace before writing to the video buffer. This option is only needed with some older IBM CGA cards, since most third-party CGA cards (and all EGA/VGA cards) do not have this problem. Try using 'Video IBM' before using this option, and if you see 'snow' near the edges while Maximus is writing to the screen, then using this option is necessary. If you see no snow, then using the straight IBM video mode should be okay.

Maximus window and status line support is available with this mode.

Equipment Section Keyword Listing

Answer <cmd>

This command is used if you wish for Maximus to answer the phone and you want it done by software, instead of using the modem hardware. It must be used in coordination with the Ring and Init commands. It is the command that would be sent to the modem to tell it to answer an incoming call. The typical command would be:

```
Answer  ATA|
```

See the Busy command for special characters that are available for use here.

Baud Maximum <speed>

This keyword specifies the maximum baud rate that your modem supports. <speed> can be any of 300, 600, 1200, 2400, 4800, 9600, 19200, or 38400.

Busy <cmd>

This keyword defines the string which Maximus will send to the modem after a caller logs off. This string is sent as-is, with the exception of the following special characters:

```
v:  Sets DTR low
^:  Sets DTR high
~:  This causes Maximus to pause for one second
`:  This causes a pause for 1/20th of a second
|:  This translates into a carriage return
```

Connect <string>

This command defines the string that will be sent by the modem to indicate that a successful connection has been established after the modem has answered the phone. This is only used if you wish for Maximus to answer the phone. A typical Connect command is:

```
Connect
```

Init <cmd>

This command is only used if you allow Maximus to answer the phone. It is the string that will be sent to the modem when Maximus is started and after every caller. The rules for the composition of this command are identical to those for the Busy command. A typical Init command is:

```
Init      ~v~`~`~`~`~`|~^`~`ATH0|
```

If you wish for the modem to answer the phone, use this string:

```
Init      ~v~`~`~`~`~`|~^`~`ATH0S0=1|
```

If you wish for Maximus to answer the phone, set the dip switch or volatile ram to disallow the modem to answer the phone (S0=0), and enable the Ring and Answer commands in MAX.CTL.

Mask Carrier <mask>

This keyword tells Maximus which one of the MSR (modem status register) bits is used to signify that there is a caller on-line. For almost all modems in North America, the value for <mask> should be 128. Other modems in other countries will probably vary, so consult your modem manual for specifics. NOTE: <mask> should be specified in DECIMAL, and not in hexadecimal as some other programs require.

Mask Handshaking <flow_control>

This keyword tells Maximus which kind of flow control you would like to use. Supported values for <flow_control> are `XON', `CTS', and `DSR'. If you want your users to be able to use ^S and ^Q to stop and restart output, then using `XON' is necessary. Also, if you are using a high-speed (9600 bps or faster) modem, then it is necessary to use either CTS, DSR, or both. You can use more than one type of handshaking by adding additional `Mask Handshake' statements.

No Critical Handler

Use this command to disable Maximus's critical error handler, which is turned on by default. The critical error handler will instruct DOS to fail the operation after printing one of two error messages through the BIOS:

Critical error reading/writing drive X:

or

Critical error accessing device COMx

This should prevent problems such as disk-left-in-drive or others from displaying an Abort/Retry/Ignore error message and waiting for keyboard input.

Output <location>

This keyword tells Maximus to which port modem output ought to be sent. Supported values for <location> are `Com1' through `Com8', and `Local'. Although Maximus supports up to eight COMx: ports internally, your FOSSIL and modem hardware must also support the port that you wish to use. Check with your FOSSIL documentation and modem manual for details on this. `Local' tells Maximus that you want local output. However, this option is fairly redundant, since you can achieve exactly the same effect by using the `-k' command line switch when executing Maximus. Note: You can specify an override COM port at runtime, via the `-p' command line switch.

Ring <string>

This command is only used if you allow Maximus to answer the phone. It defines the string that will be sent by the modem when the phone rings. It should be used in coordination with the Answer string in MAX.CTL. Also see the Init command. A typical Ring command is:

Ring Ring

Send Break to Clear Buffer

This statement tells Maximus that you are using a modem (such as the USR HST or the Hayes V-Series) which has an internal buffer, and that Maximus must send a BREAK signal whenever it needs to clear the modem's internal buffer. If you do not use this statement and your modem DOES have an internal buffer, then users may have problems using ^C to halt output from your BBS. IMPORTANT: If you are using a U.S. Robotics HST modem, make sure to specify the '&Y0' parameter in your mailer/front-end's initialization string! The factory HST configuration may not always work properly with this option enabled, so you should make sure to use &Y0 before using this keyword.

WARNING! Some HSTs may not be compatible with this option. Some systems started spewing out garbage when users hit ^c or used hotkeys, when this option was enabled. If this is the case, then you'll have to disable this option.

Matrix/EchoMail Keyword Listing

Address [zone:]<net/node>[.point]

This keyword specifies the network address (or `matrix address') which is used by Maximus. This address is used for sending NetMail messages, and is also placed on the origin line of EchoMail messages. You can specify up to sixteen matrix addresses, but the first address you specify will be your primary address. The primary address will be used on all NetMail, and will be the default address used in origin lines. You can specify secondary addresses by adding additional `Address' lines below the primary `Address' line.

IMPORTANT: If you are using Maximus to run your point, then you must follow some special guidelines for this statement. For the FIRST `Address' statement, you must put your FULL network address, including the point number, as you wish it to appear in your EchoMail origin lines. As your FIRST SECONDARY ADDRESS, you must use your fake network address, which should have been assigned to you by your bossnode. This is what will be used for the SEEN-BY lines in EchoMail.

After EchoMail Exit <errorlevel>

This statement specifies which errorlevel Maximus will exit with if a user has entered one or more EchoMail messages. This exit errorlevel will supersede the `After Edit' errorlevel, so Maximus will exit with this errorlevel, even if a user enters BOTH NetMail and EchoMail. <errorlevel> MUST be more than (or equal to) 5, and less than 254.

After Edit Exit <errorlevel>

This statement specifies which errorlevel Maximus will exit with after a user enters a matrix (or `NetMail') message. This exit is superseded by the `After EchoMail Exit' errorlevel, so if a user enters BOTH NetMail and EchoMail, the `After EchoMail' is the exit which will be taken. <errorlevel> MUST be more than (or equal to) 5, and less than 254.

After Local Exit <errorlevel>

This statement specifies which errorlevel Maximus will exit with after a user enters a local message. This exit is superseded by both the 'After EchoMail Exit' and the 'After Edit Exit' errorlevels, so if the user enters either matrix or echomail, in addition to a local message, then the 'After Local Exit' errorlevel will not be used.

FidoUser <filespec>

This statement specifies the location of a user/address list in standard FIDouser.LST format generated by nodelist compilers such as XlaxNode, SysNL, ParseLst, Qnode, etc. The format of the list is a plain ASCII text file exactly 60 columns wide in the format shown here:

Davis, Bob	1:106/114
Doe, John	2:123/4
Dudley, Scott	1:249/106

Gate Netmail

This statement will tell Maximus to route (or 'gate') interzone netmail through the standard ZoneGate. (For example, if you are sending a message from zone 1 to zone 2, the message will get addressed to 1:1/2, although the internal message address, which is processed by the ZoneGate, will indicate the correct destination of the message.) If this keyword is commented out, and if your packer (such as oMMM or Squish's SQUASH function) is not configured otherwise, the message will be sent directly to its destination.

Log Echomail <filespec>

This statement specifies the name of a log file which will be written when a user enters an EchoMail message. This log is written only when the user logs off, and it will contain the echo tag (specified by the 'MsgName' keyword in each area description) of each area in which the user entered EchoMail. This log is compatible with QM and ConfMail's '-f' command line switch. You may use Max's percent-sign translation characters in the filename for this command.

Message Edit <action> <attribute> <priv>

The 'Message Edit' series of keywords tell Maximus what to do when a user enters a NetMail message. <action> can be either 'Ask' or 'Assume'. If <action> is 'Ask' and the user's privilege level is <priv> or greater, Maximus will prompt the user to specify whether he/she wants <attribute> ON or OFF. If <action> is 'Assume' and the user's privilege level is <priv> or greater, Maximus will automatically toggle <attribute> ON. Valid values for <attribute> are:

Private

- Crash
- FileAttach
- KillSent
- Hold
- FromFile*
- FileReq
- UpdateReq

*) 'FromFile' can only be used with an <action> of 'Ask'. For this option, <priv> specifies the privilege level required to use the Forward/Bomb Run feature.

Message Send Unlisted <priv> <cost>

This command controls how Maximus treats messages which are addressed to non-existent nodes. If the user's priv is LESS than <priv>, then s/he will not be able to send a message to such as node. (In other words, Maximus will report "Unlisted node", and re-prompt the user for the destination address.) However, if the user's priv is greater than or equal to <priv>, then the message can be entered, and the user will be assessed a matrix charge of <cost> cents. If you wish unlisted messages to be free, then set <cost> to 0. Likewise, if you don't want anyone to be able to send messages to an unlisted node, set <priv> to HIDDEN.

Message Show <item> to <priv>

This keyword tells Maximus whether or not to display certain items to users which are embedded within messages. <item> can be any of 'Ctl_A', 'Seenby', or 'Private'. 'Ctl_A' defines the priv required to see the imbedded IFNA kludge lines inside a message, and to quote kludge lines inside the editors. 'EchoMail' defines the priv level required to view SEEN-BYs in EchoMail, and also to quote SEEN-BYs. 'Private' defines the minimum priv required to see a private message to ANYONE, in any area. Normally, users can only see

private messages which are to or from themselves. However, this option will let you set the priv level required to see all private messages, no matter who they are to or from. (Setting this option to below AsstSysOp is not recommended.) <priv> specifies the minimum privilege level necessary to see the specified item. If you don't want anyone (including yourself) to see these, then use the word 'Hidden' for <priv>. The default value for 'Ctl_A' and 'Seen-By' is Hidden, while the default priv for 'Private' is SysOp.

Nodelist Version <version>

This keyword tells Maximus the format of the nodelist which it uses. <version> can be either '5' or '6': Maximus will use a version 6 nodelist by default, but you can tell it to use the older version 5 nodelist for compatibility with other programs. It is also possible to use Maximus for matrix/netmail operations without a nodelist; to do so, you must first uncomment the 'Message Send Unlisted' option in MAX.CTL, and set the priv level appropriately. Then change to the directory specified by 'Path NetInfo', and create a zero-length file called NODELIST.IDX. The following DOS command can be used to accomplish that task:

```
REM > NODELIST.IDX
```

Once this has been done, Maximus will be able to run without a nodelist. Although you won't be able to see the system names of netmail messages' destinations, you'll at least be able to use the matrix area.

Path NetInfo <path>

This keyword tells Maximus in which directory to find the nodelist files.

Session Section Keyword Listing

After Call Exit <errorlevel>

This keyword tells Maximus which errorlevel to use after a caller logs off, if the user did not enter either EchoMail or NetMail. <errorlevel> MUST be more than (or equal to) 5 and less than 254.

Alias System

This keyword tells Maximus that a user's alias, should be used instead of that user's real name. This setting only affects the default; you can specify that a user's real name should be used in any given message area via the "Use Realname" keyword.

This keyword only affects the default; if Alias System is turned on, the user's alias will be used for entering messages, in the userlist, and in the Who is On command. If Alias System is turned off, the user's real name will be used by default in all of the above.

The "Ask Alias" keyword should also be used in conjunction with "Alias System" to prompt new users for an alias at log-on. This chart describes the various combinations of the two keywords:

ASK ALIAS

		YES	NO
ALIAS	YES	New users prompted for alias at log-on. By default, messages entered will use the alias unless "Use Realname" is used for that area. Users show up on W)ho is On as alias. Alias field is searched and alias field is displayed when doing a userlist.	New users not prompted for alias at log-on. If user gives self alias at the change menu, this functions identically to YES/YES. Otherwise, this functions identically to NO/NO
SYSTEM	NO	New users prompted for alias at log-on. By default, messages entered will use the real name unless "Use Alias" is used for that area. Users show up on W)ho is On as real name. Userlist displays and searches real name.	No alias use whatsoever.

Area Change Keys <keylist>

This keyword allows one to change the sequence of keys used for the A)rea Change mini-menu. <keylist> should be a sequence of three keys: the FIRST key in the sequence defines the key which takes the user to the PRIOR area. The SECOND key defines the key which takes the user to the NEXT area. Finally, the THIRD key defines the key which displays the area menu. Although this command will allow you to change the default keys, and in fact add new ones, all of the old area-changing commands will ALWAYS work. For example, even if you have changed the definition to "PN=" (for Prior/Next/=list), users will still be able to use `<',

`[' or `-' to switch to the prior area. Likewise, `>', `]' and `+' will always change the user to the next area, just as `?' and <enter> will always display the area listing.

Area Data <filespec>

This keyword specifies the location of the compiled area data file. This file contains a compiled version of the information in MSGAREA.CTL and FILEAREA.CTL.

Area Index <filespec>

This keyword specifies the location of the area index file. This file contains an index of the data contained in the 'Area Data' file. Note that Maximus will keep two area indices. This keyword specifies the location of the "old-style" index, typically called AREA.IDX. It is retained for compatibility with older Maximus utilities. The new style index will be located and named as a derivative of this keyword. For example, if you have:

Area Index C:\MAX\AREA.IDX

...the new style index will be named C:\MAX\AREA.NDX. Silt will simply substitute .NDX for the file extension.

Arrow Keys to Read

This keyword enables the use of the left and right arrow keys for reading messages at the local console. Use the left arrow to move to the previous message and the right arrow to move to next message. Control-left can be used to read the prior message in a thread, and control-right can be used to read the next message in a thread. <Alt-R> and <Alt-Q> can be used to reply to a message, <Alt-E> can be used to enter a message, and <Alt-K> can be used to kill a message.

Note! To use this option, your menus must use the same general keyboard layout as the distribution Max menus. As long as "E" is used to enter a message, "R" is used to reply, and so forth, this command will work as expected.

Ask Phone

This keyword tells Maximus to ask new users for their telephone number after they log on. This information is stored in the user file, and can be displayed either through the user editor, on the status line, or by pressing ENTER while a user is on-line.

Ask Alias

This keyword tells Maximus to prompt new users to enter an alias at log-on. See the "Alias System" keyword for more details.

Charset Swedish

Enable internal support for other Swedish 7-bit character set.

Note: To properly implement the Swedish 7-bit format, it is necessary to edit ENGLISH.MAD and uncomment the appropriate definitions for LBRACKET and RBRACKET, since the default left/right brackets conflict with the Swedish 7-bit character set.

Charset Chinese

Enable internal support for the Chinese character set. The Chinese character set provides for the "BIG5" two-byte codes used by most Chinese programs.

Chat Capture On

The chat capture buffer can be turned on automatically by uncommenting this command. This causes Max to open the chat log as soon as <Alt-C> is pressed, with no further SysOp intervention required.

Chat External <prog_name>

This keyword tells Maximus to use an external program for chat instead of the internal CHAT routine. Be sure to specify a full path for <prog_name>, or else Maximus may not always be able to find the external chat program. Include command line parameters for the chat program, if necessary.

Compatible LocalBaud

This keyword tells Maximus that you want the LASTUS*.BBS structure produced by Maximus to be compatible with the old Opus 1.03 structure. Normally, Maximus fills in a few of the fields in this structure in a more sensible fashion than Opus. For example, it uses 0 as the baud rate for a local caller, instead of 9600. However, this may confuse some programs (notably, C. Holten's `DRLINK') that relied on Opus' quirks. Enabling this keyword will allow you to use such programs.

Comment Area <area>

This keyword specifies the message area to use for SysOp comments. (If this keyword is not specified, Max will place all comments in area 0.) The area number specified with this token will be used for log-off comments, and it will also be used for comments left through the [leave_comment] MECCA token.

Define <priv> <attribute> <value>

This keyword is used to control the various limits and restrictions for various users, based on <priv>. <attribute> can be any one of the following items:

- Time
- Cume
- Logon Baud
- File Baud
- File Limit
- File Ratio

`Time' specifies the maximum length of time, in minutes, for which a user with <priv> privilege level can stay on-line in a single session. The day begins at 00:00 and ends at 23:59.

`Cume' specifies the maximum amount of time, in minutes, for which a user with <priv> privilege level can be on-line for the entire day.

`Logon Baud' specifies the minimum baud rate at which a user who has a privilege level of <priv> must be calling at in order to log on. (Note that this is in ADDITION to the `Min Logon Baud' rate specified elsewhere in the control file.)

`File Baud' specifies the minimum baud rate at which a user who has a privilege level of <priv> must be calling in order to download or upload a file.

`File Limit' specifies the maximum number of kilobytes which a user can download in a single day.

`File Ratio' enforces a download ratio for users of <priv> privilege level, in the form of <value>:1. In other words, for every <value> kilobytes downloaded in total, the user must upload at least one kilobyte. Setting <value> to zero means that users with a privilege level of <priv> will not have a download ratio enforced.

Edit Disable <option>

Using this keyword will tell Maximus to disable certain editor options. <option> can be one of the following items:

MaxEd
UserList

Normally, a user may use either the full-screen editor or the line-oriented editor to enter messages. Specifying `MaxEd' tells Maximus to disable the use of the MaxEd full-screen editor, and force users to use the line-oriented BORED editor.

Also, when entering private messages in local message areas, users can type `?' to get a listing of other users on the BBS, so that he/she can remember the spelling of the intended recipient's name. `UserList' tells Maximus to ignore `?' when used while a caller is entering a message. This is useful if you don't want users to be able to obtain a listing of all users on your system.

External Protocol <filespec> [desc]
External BatchProtocol <filespec> [desc]
External ExitProtocol <filespec> [desc]
External BatchExitProtocol <filespec> [desc]

These keywords are obsolete. See the new documentation for PROTOCOL.CTL.

External Protocol Errorlevel <erl>

This command instructs Max to exit with an errorlevel of <erl> when executing an external protocol which has the 'Type Errorlevel' flag set. For more information, please see the section on PROTOCOL.CTL.

File Date <type> [format]

The 'File Date' command tells Maximus which way to display dates inside the file areas. You can choose from one of several formats, including U.S.A., Canadian/British, Japanese, and scientific. In addition, you can tell Maximus to get the files' dates and sizes directly from their directory entries, or you can enter the dates into FILES.BBS itself, for greater speed when displaying file catalogues on CD-ROMs and WORMs.

<type> can be either of 'Automatic' or 'Manual'. 'Automatic' means that Maximus will look at the file's directory entry to determine both the file's size and date. If <type> is 'Manual', then Maximus won't look at the directory entry at all, and will assume that the size and date information is imbedded in FILES.BBS itself, as ASCII text.

[format] specifies the format to use for file-entry date stamps, and can be any of the following options:

```
mm-dd-yy (U.S.A., default)
dd-mm-yy (Canada/England)
yy-mm-dd (Japanese)
yymmdd   (Scientific)
```

If <type> is 'Automatic', then the format above will be used when DISPLAYING files' directory entries. In other words, it will be generated at runtime. However, if <type> is 'Manual', then Max will insert the date, in the format specified, into the FILES.BBS catalog when the file is UPLOADED, and will get the date from FILES.BBS from that point on. You must manually insert dates for any preexisting files in the file areas.

The format specified by [format] will also be used when prompting the user for a date while doing a new-files check.

Examples:

File Date Automatic dd-mm-yy

Using the option would tell Maximus to automatically determine the dates and sizes of files from their directory entries, and also to display dates using a Canadian format.

File Date Manual yy-mm-dd

This option tells Maximus to expect to find file sizes and dates inserted directly into FILES.BBS, and when inserting upload descriptions into the file catalog, to insert dates using the Japanese date format.

FileList Margin <col>

This keyword causes long FILES.BBS descriptions to be indented by a particular offset for the second and third lines. This may be useful if an external program is being used to add a "download counter" to file descriptions, and you want to have the file description kept in its own column.

First File Area <area>

This keyword defines the file area into which all new users will be placed. The area number specified should be a valid file area, and should be accessible to new users. If this keyword is not given, then Maximus will default to placing users into area 1.

First Menu <menu>

This keyword tells Maximus which menu file to display after displaying the WELCOME files. If no <menu> is specified, then the menu called `MENU' will be displayed by default.

First Message Area <area>

This keyword defines the message area into which all new users will be placed. The area number specified should be a valid message area, and should be accessible to new users. If this keyword is not given, then Maximus will default to placing users into area 1.

Format Date <date_format>

The 'Format Date' option controls the format of in which dates are displayed by Maximus. (Be sure to include only the date portion in this command. Another command, 'Format Time', controls the format in which times are displayed.) This format is used for displaying the dates in message headers, and also in various other places throughout the system. The date format is output to the user exactly as specified, except for several special two-character sequences, which are started with a percent-sign. The following case-sensitive sequences are supported:

- %A: Either 'am' or 'pm', as appropriate.
- %B: The month, as a decimal
- %C: The month, as an abbreviated text string
- %D: The day-of-month, as a decimal
- %E: The hour, in the range of 1 to 12
- %H: The hour, in the range of 0 to 23
- %M: The minute
- %S: The second
- %Y: The year, without the century
- %%: A single percent sign

Examples:

%E:%M%A

This would translate to the time in a 12-hour format. An example time shown with this format would be '08:23pm'.

%H:%M:%S

This would translate to the time in a 24-hour format, including seconds. An example time shown with this format would be '20:23:15'.

%B-%D-%Y

This would translate to the current date, in a numeric format. This will translate to '10-01-91', assuming a date of October 1st, 1991.

`%B %C %Y`

This translates into the date, with a numeric day-of-month, an alphanumeric month, and a numeric year. An example date using this format might be `29 Dec 90'.

```
Format FileFormat <format>
Format FileHeader <format>
Format FileFooter <format>
Format MsgFormat <format>
Format MsgHeader <format>
Format MsgFooter <format>
```

These options were created to help the sysop get out of designing all those MSGAREA.BBS and FILEAREA.BBS screens, while still allowing him/her to maintain some originality in the message/file area listings.

The MsgHeader and FileHeader are displayed at the top of the message and file area listings, respectively. The MsgFormat and FileFormat are in the body of the message and file area listings, respectively. Finally, the MsgFooter and FileFooter statements are shown after all of the applicable message/file areas have been shown, using the MsgFormat/FileFormat statements.

Each of these strings is made up of straight ASCII text, also with some optional format characters. The percent sign (`%') is used as an initiator for the control sequences. Anything not preceded by a percent sign is treated as normal text, and is passed directly to the user. The format of each control sequence is as follows:

`%[-][min][.max]<format_char>`

Programmers will notice that this is somewhat similar to C's printf() function. However, it is not the same function, and you should refrain from using control characters other than those listed below).

Everything except for <format_char> itself is optional and may be omitted.

[-], if present, specifies that the data field produced by this sequence is to be left justified, instead of right-justified as the data field normally would. This will only have an effect if the [min] parameter is specified.

[min], if present, specifies the minimum width of the data field produced by this sequence. If the [-] parameter is specified, then the field will be padded with blanks on the RIGHT. Otherwise, the data field will be padded with blanks on the LEFT.

[.max], if present, specifies the MAXIMUM width of the data field produced by this sequence. If the data field is longer than [max], then the field will be truncated. Also, be sure to include the period in [.max], so that Maximus can distinguish it from the [min] parameter.

<format_char> is case-sensitive, and specifies the data field to display for the current area. <format_char> can be one of:

- # This translates to the area number of the current area.
- * When used in a message-area format statement, this token will check the user's lastread pointer for that area, and compare it to the number of messages which exist in the area. If there are more messages than the user has read (ie. there are new messages), then this token will cause an asterisk ("*") to be displayed. If not, then a blank space will be displayed instead. This is especially useful when your users read a large number of echomail areas, since it quickly points out areas which have NEW mail, without the need to enter each area individually.

This token is required, if you wish to enable the message area tag function and have the tags show up on the message area display.

This token can only be used for message area listings.

- c This token is special, and not like any of the others described in this section. Instead of DISPLAYING something, this token causes Maximus to selectively SKIP one or more of the following characters in the sequence. If you place a sequence in the form "%#.\$c", where "#" and "\$" are numbers, then Maximus will SKIP the display of the next "\$" characters, after every "#"th area processed. For example, the sequence "%2.3cABC" would cause Maximus to display the sequence "ABC"

for every second area. (The "3" tells Maximus to skip the three following letters, which are "ABC" in this case.) If the number-of-characters-to-skip part of the format is omitted, then Maximus will skip only the next character. For example, the sequence `%2cZ' would cause Maximus to display the character `Z' after every second area. PLEASE NOTE: for the purposes of this command, a `character' counts as one letter, number, OR one of the other tokens. That means that the sequences `%x0a', `%45.45n', etc., all count as one character.

- f This option causes a *.BBS file to be displayed, as part of the format statement. The filename should directly follow the "%f specifier, and the end of the filename is delimited by a single space. The file can contain anything that you would normally place in a .BBS file; however, note that the file will be displayed BEFORE the rest of the format string is processed. In other words, if you use the following format string: `Format MsgHeader Abcd %fD:\PATH\FILENAME.BBS 1234', then the user will be shown the contents of D:\Path\Filename.Bbs, and THEN `Abcd' and `1234'.
- l This translates into the last `leaf' in the path of the current area. For example, the last leaf of the `D:\Msg\Abcd' subdirectory would be `Abcd'.
- n This translates into the long name of the message or file area, as specified in the area's `MsgInfo' or `FileInfo' statements.
- t This translates into the `echo tag' of the current message area, as specified in the area's `MsgName' statement.
- x This specifies that the next two characters in the sequence are hexadecimal digits which should be inserted directly into the output as one ASCII character. This is useful for inserting otherwise-unprintable control characters (such as the AVATAR colour controls) into the area display.

Examples:

%-15.15t

This would translate into the `echo tag' for the current message area, which would be left justified, and exactly fifteen character long (padded with spaces).

`%30.30n`

This would translate into the name of the current message area, right justified to make the field exactly 30 characters long, no more, no less.

`%%2# / %-25.25n %2c%x0a`

This statement would cause Maximus to display the area number for each area, followed by a space and a forward slash, another space, and the area name (left-justified), to a maximum length of 25 characters. After every second area, a CR/LF would be printed (that's the `x0a' you see above), effectively creating a two-column area display. Finally, if an area contained new messages, then a "*" would be displayed beside the area number.

For those who don't want to bother figuring all of this out, using the following formats will cause Max to display areas with the one-area-to-a-line style:

```
FileFormat:
%x16%x01%x0d%2#%x16%x01%x07 ... %x16%x01%x03%-n%x0a
FileHeader:
%x16%x01%x0fFile Areas
%x16%x01%x0d-----%x0a%x0a
MsgFormat:
%x16%x01%x0d%2#%x16%x01%x07 ... %x16%x01%x03%-n%x0a
MsgHeader:
%x16%x01%x0fMessage Areas
%x16%x01%x0d-----%x0a%x0a
```

`Format Time <format>`

This statement controls the format in which Maximus will display times. Please refer to the `Format Date' keyword for explanations of the format characters which can be used.

Highest FileArea <area>

This statement specifies the highest file area number which can be accessed using the `A]` (Area Next) and `Locate` commands. If this keyword is not enabled, the Locate and A] commands will automatically scan all file areas.

Highest MsgArea <area>

This is the equivalent of `Highest FileArea`. However, this command defines the "number" of the highest message area which can be accessed with the Browse, Mailchecker, or A] and A[commands.

Input Timeout <mins>

This command specifies the number of minutes of inactivity which Maximus will allow before automatically hanging up on the user. The default value is 4 minutes. You may use this command to specify a range of 1 to 255 minutes. When the input timeout occurs, Maximus will send a warning message "Please respond:", then wait one minute longer. If no input received after that final minute, Maximus will hang up.

Kill Private <when>

This keyword controls Maximus' handling of private messages in local message areas. <when> can be any of the following values:

ALWAYS: Maximus will always kill a private message after it has been read by the recipient.

ASK: Maximus will ask the user whether or not to kill a private message.

NEVER: Maximus will never kill a private message.

Local Editor [!]<editor_cmd>

This keyword tells Maximus that when you are logged on locally, you want to use an external editor instead of MaxEd to edit and enter messages. With this keyword enabled, Maximus will execute <editor_cmd>, instead of calling MaxEd to enter the message. If the message is a reply to another, Maximus will quote the original and place it in a file called `MSGTMPxx.\$\$\$` before invoking the editor, where `xx` is the current task number, in hexadecimal. Whether or not your message is a reply, Maximus expects to find your final

message in MSGTMPxx.\$\$\$, so if you can specify a file to edit on the command-line when invoking your editor, it would be best to specify `MSGTMPxx.\$\$\$', using the current task number for `xx'.

Also, be sure to specify the full path to your editor (including the .EXE or .COM extension), or else Maximus may not always be able to find it.

Normally, the `Local Editor' command only affects messages entered locally. However, if you wish to allow on-line users to also access this editor instead of MaxEd, then insert an exclamation mark (`!') just before the name and path of your editor. After doing so, this editor will be invoked instead of MaxEd for any user who has a privilege of at least AsstSysOp, and who has the full-screen editor option turned on.

Finally, if you place the sequence `%s' inside the <editor_cmd> string, then Maximus will replace the `%s' with the name of the temporary file to edit. This can be useful in a multitasking situation, if you don't want to hard-code a task number into the control file.

For example, the command:

```
Local Editor C:\Util\Q.Exe %s
```

...would invoke the program C:\Util\Q.Exe, and pass the name of the temporary file as the first command-line argument.

Local Input Timeout

Placing this statement in MAX.CTL will cause the inactivity timer to be active for local log-ons. Normally, without this statement enabled, the input timer is DISABLED for local log-ons, meaning that you can log on, walk away, have a nice, cold Canadian beer, etc., and have Maximus still be waiting for you when you come back. However, if this statement is enabled, Maximus will time out, and automatically log you off (which returns you to the calling batch file) after five minutes of inactivity.

Logon Level <priv>
Logon Preregistered

The `Logon Level' statement specifies the privilege level to assign to new users who log onto your system. If you use the `Logon Preregistered' statement instead, then no new users will be allowed on the system, and Maximus will hang up on new users after displaying the `Uses Application' file.

Logon Timelimit <time>

This keyword specifies the maximum amount of time, in minutes, that the user is allowed to complete the process of logging on, reading through the `Uses Application' file, entering their password, etc.

Mailchecker Kill <priv>
Mailchecker Reply <priv>

These two commands control the priv level required to access certain commands in the B)rowse command and in the internal mailchecker. The `Mailchecker Kill' option specifies the priv level required to delete a message through Browse, and the `Mailchecker Reply' option controls the priv level required to reply to a message through the Browse command. Both of these options also apply to the mailchecker.

Menu Path <path>

This keyword tells Maximus where it can find all of the default *.MNU files. NOTE: This path can be changed at run-time, by the `[menupath]' MECCA token.

Min Logon Baud <speed>

This keyword specifies the minimum baud rate at which any user must be calling in order to log on to the system. This is in addition to the rates specified by the `Define <priv> Logon Baud' statements.

Min NonTTY Baud <speed>

This keyword specifies the minimum baud rate at which any caller must be calling in order to use a terminal mode other than TTY.

No FilesBBS Download <priv>

This keyword specifies the minimum privilege level that a user must possess in order to download a file which is NOT contained in the FILES.BBS listing but which does exist in the subdirectory for a particular file area.

No RealName Kludge

This keyword tells Maximus to NOT insert the `^aREALNAME:' kludge into messages entered in an `Anonymous OK' message area. Normally, this kludge aids in tracking down users who try to abuse the ability to leave anonymous messages. However, there are circumstances when you want to assure the confidentiality of a user's identity, and in those cases you should use this keyword. Unless you have special needs, it is highly desirable to NOT use this keyword..

Ratio Threshold <k>

This option specifies the MINIMUM number of kilobytes a user must download before Maximus will start harassing the user about upload/download ratios. If the user has downloaded LESS than the number of kilobytes specified with this option, then Maximus will disregard the ratio, and let the user download anyway. See also `Define <priv> File Ratio' for more details.

Save Directories <drives>

This keyword specifies the drives for which Maximus is to save the current directory when executing an external command. The <drives> statement should normally include all drives on your system, except for removable (i.e. floppy) disks.

StatusLine

This keyword instructs Maximus to place a status line at the bottom of the screen, whenever a remote user calls. The status line cannot be used while calling in local mode. NOTE! This keyword can only be used if the current video mode is `Video IBM' or `Video BIOS'.

Upload .BBS Priv <priv>

This keyword specifies the minimum priv needed to upload a file with a .BBS or .GBS extension. Normally, this should be set to SysOp or AsstSysOp, since anyone who can upload a .GBS file could conceivably upload a FILES.GBS containing MECCA commands which would invoke a DOS shell. Thus, an unauthorized user could gain unrestricted access to your system at the DOS command level, if the priv for this command is set too low. Maximus will also check files uploaded through external protocols, and adjust their filenames as well.

Upload Check Dupe Upload Check Dupe Extension

These keywords specify whether Maximus should check the upload file specification for duplication of an already existing file on the system. `Upload Check Dupe' means to only the filename. `Upload Check Dupe Extension' means to check the filename and extension. Note that in order for you to use this feature, you must utilize the Maximus utility FB.EXE. See the Maximus Operations Manual on how to use FB.EXE.

Upload Check Virus <batchfile>

This keyword instructs Maximus to call a batch file for each uploaded file. <batchfile> simply specifies the name of the batch file to run. For example:

```
Upload Check Virus vircheck.bat
```

This tells Maximus to call VIRCHECK.BAT every time a file is uploaded. Max will call the batch file using the following format:

```
vircheck D:\File\Upload\ THISFILE .TXT D:\Max\Misc\
```

where "D:\File\Upload\" is the path to the uploaded file (including trailing backslash), where "THISFILE" is the root of the filename, where ".TXT" is the file extension, and where "D:\Max\Misc\" is the path to the Max miscellaneous directory (including trailing backslash).

The batch file can perform any amount of processing, including scanning for viruses, refusing files with bad extensions, and so on. After the batch file returns, Maximus will look to see if the original uploaded file still

exists. If the file DOES exist, Max will display \MAX\MISC\FILE_OK.BBS. Normally, this file should contain a message informing the user that the file contained no viruses. Maximus will then ask for an upload description and credit the user's account.

If the uploaded file does NOT exist (presumably because VIRCHECK.BAT renamed it to something else), Maximus will display \MAX\MISC\FILE_BAD.BBS to the user, which should mention that the virus check failed. The uploaded file will not be added to FILES.BBS, nor will the user be credited for the upload.

This feature was designed for an automated virus-checking program to use, but certain things can be done with batch files as well. The file's extension can be tested separately, so it can be used to block uploads of files with certain extensions. \MAX\MISC\FILE_BAD.BBS can also be swapped by VIRCHECK.BAT for another file, so a different FILE_BAD.BBS can be displayed for virus checks and archive corruption checks. FILE_BAD.BBS could also be used to display the log of the virus checking program, thereby giving the user more information about the virus itself.

Upload Log <log_name>

This keyword specifies the name of a log file which will be written to whenever Maximus receives any uploads. One line will be written to this file for every upload received. This line will contain the name of the file uploaded, the name of the user who uploaded it, the size of the file, and the date/time when the file was uploaded. This makes it very easy to keep track of who uploaded what, in case you need to contact the user for some reason.

Upload Reward <percent>%

This keyword specifies the percentage of time which is given back to the user for uploading a file. A reward of 100% gives one second back to the user, for every second spent uploading. In other words, the user loses no time for uploads. You can also `reward' a user for uploading by setting the number to higher than 100%. If you set this reward to 0%, the time spent uploading will not be returned to the user.

Upload Space Free <amount>

This keyword tells Maximus not to allow users to upload, unless there is more than <amount> kilobytes of free space on the upload drive. If there is less than <amount>, then Maximus will display the `Uses NoSpace' file. In addition, when receiving file headers with SEALink and Zmodem, Maximus will automatically abort the upload if disk space is limited.

UserList Maximum <priv>

UserList Minimum <priv>

These two commands specify the range of privilege levels (inclusive) for which users are displayed when the UserList command is executed.

Uses Application <filespec>

This defines the file which is displayed to a new user after they answer YES to `First Last [Y,n]?', but before prompting the user for their city, phone number, etc.

Uses BOREDhelp <filespec>

This is the file displayed to first-time callers who have the help-level set to NOVICE, when they first enter the BORED editor.

Uses BadLogon <filespec>

As a security feature, this file is displayed to users whose last logon attempt failed because of a wrong password. This is useful for alerting users to the fact that someone may be attempting to hack into their account.

Uses Barricade <filespec>

This file is displayed to users after they enter a barricaded message or file area, but before they are prompted for the password.

Uses BeginChat <filespec>

If specified, this file is displayed to the user when the sysop enters CHAT mode. This is a good place to put something like, 'Hi [user], this is the SysOp speaking.' If no 'BeginChat' file is specified, then Maximus will default to displaying the standard 'CHAT: start'.

Uses ByeBye <filespec>

This is the file which is displayed to users when they are on their way out the door (logging off) after selecting the Goodbye command.

Uses Cant_Enter_Area <filespec>

If specified, then this file will be displayed to the user when he/she picks an area which does not exist, or which is not accessible to the user. If this statement is NOT specified, then Maximus will default to displaying, 'That area doesn't exist!'

Uses ContentsHelp <filespec>

This is the name of the file which will be displayed if the user asks for help when using the Contents command.

Uses DayLimit <filespec>

This file is displayed to a user when he/she tries to log on if he/she has already overrun his/her daily time limit.

Uses EndChat <filespec>

If specified, this file is displayed to the user when the sysop exits CHAT mode. If no 'EndChat' file is specified, then Maximus will default to displaying the standard 'END CHAT'.

Uses FileAreas <filespec>

If specified, this file will be displayed to the user when he/she asks for a listing of file areas, instead of the semi-canned listing generated by the 'Format FileHeader' and 'Format FileFormat' statements.

Uses Filename_Format <filespec>

This specifies the name of the file which will be displayed if the user enters an invalid filename when attempting to upload.

Uses InquireHelp <filespec>

This file will be displayed to the user when they ask for help when using the message-area Inquire command.

Uses Leaving <filespec>

If specified, this file will be displayed when Maximus exits to run an external program from a menu option. (Nothing will be displayed if the file is run from a *.BBS file.)

Uses ListHelp <filespec>

This file will be displayed to the user when they ask for help in using the message-area List command.

Uses LocateHelp <filespec>

This file will be displayed to the user when they ask for help in using the file-area Locate command.

Uses Logo <filespec>

This keyword specifies the file to display immediately after Maximus connects with the user. This should normally be a fairly short file describing your BBS, the type of computers it supports, its hours of operation, etc.

Uses MaxEdHelp <filespec>

This keyword specifies the file to display to the user when they ask for help (by pressing '^k?') from within the MaxEd editor.

Uses MsgAreas <filespec>

If specified, this file will be displayed to the user when he/she asks for a listing of message areas, instead of the semi-canned listing generated by the 'Format MsgHeader' and 'Format MsgFormat' statements.

Uses NewUser1 <filespec>

This is the file which is displayed to a new user right before he/she is asked to enter the password which he/she wishes to use.

Uses NewUser2 <filespec>

This is the file which is displayed to a new user, in lieu of the `Uses Welcome <filespec>' file.

Uses NoMail <filespec>

This file is displayed when a user uses either the [msg_checkmail] MECCA command or the "SR" (Scan/Read) command and there was no mail waiting for him/her.

Uses NoSpace <filespec>

This keyword defines the name of the file to display when the amount of space free on the upload drive is less than that specified by the `Upload Space Free' option.

Uses NotFound <filespec>

This file is displayed to a new user AFTER he/she enters his/her name, but BEFORE the `First Last [Y,n]?' prompt is displayed.

Uses ProtocolDump <filespec>

This file will be displayed to the user instead of the standard, `canned' list of protocol names. This file will be displayed for both the U)pload and D)ownload commands.

Uses Quote <filespec>

This keyword specifies the name of a straight ASCII text file which can contain quotes, random pieces of wisdom, etc. Each separate quote in the file should be separated by a single blank line. This file can be accessed using the MECCA [quote] token.

Uses ReplaceHelp <filespec>

This is the name of the file that is displayed to the user when he/she selects the Edit option on the BORED menu. This file should describe the search and replace feature of the editor.

Uses Returning <filespec>

If specified, this file will be displayed to the user when he/she returns from an external program invoked via a menu option.

Uses Rookie <filespec>

This is the name of the file which is displayed to a user who has called between two and eight times, in lieu of the 'Uses Welcome' file.

Uses ScanHelp <filespec>

This is the name of the file which is displayed to the user when they ask for help with the message-area Scan command.

Uses Shell_Leaving <filespec>

This file is displayed to the user after the sysop presses Alt-J to shell to DOS, but before the sysop is actually let out to the DOS prompt.

Uses Shell_Returning <filespec>

This file is displayed to the user AFTER the sysop returns from an Alt-J DOS shell.

Uses TimeWarn <filespec>

This keyword defines the file which is displayed to a user who has already made more than one call on the current day, just before displaying the main menu.

Uses TooSlow <filespec>

This file is displayed to users whose baud rate is lower than the baud rate defined in 'Min Logon Baud', or if it is lower than the minimum baud rate specified for their particular privilege level.

Uses Tunes <filespec>

This option specifies the name and location of the Maximus tunes file. This tune file can be used to play simple melodies on the PC speaker. For more information on the format of this file, please see the comments in the distribution version of TUNES.BBS.

Uses Welcome <filespec>

This file is displayed to normal users who have called more than eight times, after they enter their password but before they reach the main menu.

Uses XferBaud <filespec>

This file is displayed to the caller if his/her baud rate is less than the speed given in the `Define <priv> File Baud' statement.

Yell Off

This disables the Yell function altogether.

Yell <schedule>

This keyword is obsolete and is no longer used. Please see the section on EVENTS.BBS for more information.

LANGUAGE.CTL Reference

LANGUAGE.CTL is the default name of the Maximus language control file. This file is used to define the languages supported via Max's multilingual system, and it also allows the sysop to select a default language through placement of the "Language" tokens in the control file.

Language Section Keyword Listing

Language <filename>

This specifies the filename of a Max-specific language file. For example, "Language English" indicates that a file called ENGLISH.LTF exists in the 'Path Language' directory. You may specify up to a total of eight language files. Remember, language names must conform to DOS filename specifications, so they must be less than eight characters long.

The first language listed will be used as the default for both new users and the sysop's log file.

READER.CTL Reference

READER.CTL is the default control file for the Maximus off-line reader. If you do not wish to use the off-line reader, this control file may be commented out.

Reader Section Keyword Listing

Archivers <filespec>

This file contains definitions for external compression utilities. The file is identical in format to the COMPRESS.CFG as used by the Squish mail processor. If you are running both Max and Squish, this keyword can point to the same compression control file as used by Squish.

Packet Name <filename>

This command defines an eight-character identifier for your system. Max will use this identifier when building QWK packets; downloaded packets will be called <filename>.QWK, and uploaded replies will be called <filename>.REP. This keyword should normally be an abbreviation of your BBS name. The abbreviation must be eight characters or less and can't include any spaces. Only valid DOS filename characters are permitted. (A-Z / 0-9 / !@#\$%&()_)

Work Directory <path>

This keyword specifies the name of the directory in which Max will create QWK packets. Max will create subdirectories off the path you specify, one for each node. Packing for each node will be performed in the individual node directory, but any files contained in the specified work directory will be INCLUDED IN EACH QWK PACKET. For more information on what you can do with this feature, please see the section entitled "QWK Mail Packer" in the Maximus Operations Manual.

Max Messages <num>

This keyword specifies the maximum number of messages that Maximus will pack in one download session. If you do not wish to limit the number of messages, specify `Max Messages 0`.

Phone Number <phone number>

This keyword should contain the phone number of your BBS, to be placed in downloaded mail packets. The number SHOULD be in the "(xxx) yyy-zzzz" format, since some off-line readers depend on the phone number looking like this. Max itself doesn't care, and it will copy this string out to CONTROL.DAT verbatim. However, some off-line readers may have problems if the number is entered in a different format or if the number is longer than 15 characters.

COLOURS.CTL Reference

COLOURS.CTL is the default colour control file. Max uses this file to select colours for certain Maximus prompts. (The colours for other prompts are imbedded in the language file.) If a keyword is omitted, Max will use the version 1.02 colour defaults.

Colours Section Keyword Listing

FSR Address <colour>

Colour for the to/from addresses. The default colour is yellow on blue.

FSR Attribute <colour>

Colour for the message attributes. The default colour is yellow on blue.

FSR Border <colour>

Colour for the top/bottom border. The default colour is lightcyan on blue.

FSR Date <colour>

Colour for the message date. The default colour is white on blue.

FSR MsgInfo <colour>

Colour for the message area name. The default colour is yellow on blue.

FSR MsgLinks <colour>

Colour for the FSR up/down reply links. The default colour is yellow on blue.

FSR MsgNum <colour>

Colour for the message number in the full-screen reader. The default colour is lightred on blue.

FSR Static <colour>

Colour for the static information in the full-screen reader, such as "To:", "From:", etc. The default colour is white on blue.

File Date <colour>

The colour of the date in a file listing. The default colour is green.

File Desc <colour>

The colour of the description in a file listing. The default colour is yellow.

File Find <colour>

The colour of the search text in a L)ocate. The default colour is yellow.

File Name <colour>

The colour of a filename in a file listing. The default colour is yellow.

File New <colour>

The colour for new files in a file listing. The default colour is blinking green.

File Offline <colour>

Colour for "offline". The default colour is red.

File Size <colour>

The file's size. The default colour is magenta.

Hotflash Bar <colour>

The colour of the main hotflash bar. The default colour is black on white.

Hotflash Clear <colour>

The colour to use for the rest of the bar. This should always be <background> on <background>. The default colour is gray on gray.

Hotflash More <colour>

The colour of the 'SPACE>' (more) prompt. The default colour is red on white.

Menu Highlight <colour>

The first letter of each option for novice menus. The default colour is yellow.

Menu Name <colour>

The menu's name, such as "MAIN:" or "MESSAGE:". The default colour is yellow.

Menu Text <colour>

The option text for each option on novice menus. The default colour is gray.

Message Address <colour>

The colour for the address of a message ("Orig:"). The default colour is cyan.

Message Attrib <colour>

The colour for the message attributes ("Pvt Crash"). The default colour is lightgreen.

Message Body <colour>

The colour for the message body. The default colour is cyan.

Message Date <colour>

The colour for the message date. The default colour is lightgreen.

Message From <colour>
Message FromTxt <colour>

The attributes for the "From: xxxx" prompt. The default colour for "From" is cyan. The default colour for "Fromtxt" is yellow.

Message Kludge <colour>

Colour for kludge lines within a message. The default colour is lightmagenta.

Message Locus <colour>

The colour for the city of the sender: "Kingston, ON (1:249/106)". The default colour is green.

Message Quote <colour>

Colour for quoted text within a message. The default colour is grey.

Message Subj <colour>
Message SubjTxt <colour>

The attributes for the "Subj: xxxx" prompt. The default colour for "Subj" is cyan. The default colour for "Subjtxt" is yellow.

Message To <colour>
Message ToTxt <colour>

The attributes for the "To: xxxx" prompt. The default colour for "From" is cyan. The default colour for "Fromtxt" is yellow.

Popup Border <colour>

Colour for the border of a pop-up window. The default colour is yellow on blue.

Popup Highlight <colour>

Colour used for highlighted text inside pop-up windows. The default colour is yellow on blue.

Popup LSelect <colour>

Colour for selected items on pop-up pick lists. The default colour is grey on red.

Popup List <colour>

Colour for standard items on pop-up pick lists. The default colour is black on grey.

Popup Text <colour>

Colour for the text in a pop-up window. The default colour is White on blue.

Status Bar <colour>

Colour for the main status bar. The default colour is Black on white.

Status Chat <colour>

Colour for the chat-request indicator ("C"). The default colour is blinking black on white.

Status Key <colour>

Colour for the rest of the status-line key flags, such as "K". The default colour is black on white.

WFC Activity <colour>

Colour for the activity window on the "Waiting for Caller" screen. The default colours is white on blue.

WFC ActivityBor <colour>

Colour for the activity window border on the "Waiting for Caller" screen. The default colour is lightcyan on blue.

WFC Keys <colour>

Colour for the keys window on the "Waiting for caller" screen. The default colour is yellow on blue.

WFC KeysBor <colour>

Colour for the keys window border on the "Waiting for caller" screen. The default colour is white on blue.

WFC Modem <colour>

Colour for the modem window on the "Waiting for caller" screen. The default colour is gray on blue.

WFC ModemBor <colour>

Colour for the modem window border on the "Waiting for caller" screen. The default colour is lightgreen on blue.

WFC Line <colour>

Colour for the bar at the top of the "Waiting for caller" screen. The default colour is white.

WFC Name <colour>

Colour for the Maximus name on the "Waiting for caller" screen. The default colour is yellow.

WFC Status <colour>

Colour for the status window on the "Waiting for caller" screen. The default colour is white on blue.

WFC StatusBor <colour>

Colour for the status window border on the "Waiting for caller" screen. The default colour is yellow on blue.

MSGAREA.CTL Reference

MSGAREA.CTL describes the message areas on a Maximus system. Although this file is not quite as complex as the main MAX.CTL, there are still enough options to devote an entire section to this control file.

Alphabetical Keyword Listing

Access <priv>[/<lock>]

This keyword is obsolete. See also `MsgAccess`.

Alias

Specifies that a user's `Alias` will be used from his/her user record instead of that user's real name. This option has no effect unless the Maximus alias system has been enabled in MAX.CTL.

Area <area>

This keyword is a SILT directive which tells SILT that the current area definition begins here. <area> must be no longer than nine characters, and can be composed of any alphanumeric character. For example, `1`, `2`, `A`, `B`, `AA`, `H1` `NETMAIL`, and `CHATTER` are all acceptable. An area definition is ended by the `End Area` directive.

Anonymous

By placing this keyword in an area definition, Maximus will allow users to enter anonymous messages in the specified area. When users enter or reply to messages, Maximus will prompt the user for the name to use in the `From:` field, but Max will also provide the user's real name as a default.

Barricade <filespec>

This keyword is obsolete. See also `MsgBarricade`.

Conference <path>

This keyword tells Maximus that the current area is accessible as a Conference message area. If the area uses the *.MSG format, <path> should specify the directory for storing the message files. If the area uses the Squish format, <path> should specify the path and root filename of the area. For example, a Squish-format area could be defined as `C:\MAX\MSG\MUFFIN`. Ensure that each *.MSG type message area uses a separate subdirectory, or else message areas may become crosslinked (mixed up). A conference area is a shared area similar to echomail, but conferences use PIDs and have no tear lines. Conference areas are suitable for use in a GroupMail environment.

EchoMail <path>

This keyword tells Maximus that the specified area is an EchoMail area. If the area uses the *.MSG format, <path> should specify the name of the message directory. If the area uses the Squish format, <path> should specify the path and root filename of the area.

End Area

This keyword is a SILT directive; this tells SILT that the current area definition ends here.

High Bit Allowed

This command tells Maximus that messages entered in this area can use characters with the "high bit" set, more commonly known as IBM Extended ASCII, which is normally used for non-English characters and accents. Note: if you wish to use this command in an echomail area, please check the rules of that specific area first. Some conferences strictly prohibit extended ASCII, while others promote its use. Also, unless you use this keyword, you will be unable to use high-bit characters in the origin line for the current area.

Local <path>

This keyword tells Maximus that the specified area is accessible as a local message area. If the area uses the *.MSG format, <path> should specify the name of the message directory. If the area uses the Squish format, <path> should specify the path and root filename of the message area. This statement CANNOT be used in conjunction with either `EchoMail`, `Matrix` or `Conference`.

Matrix <path or path/filename>

This keyword tells Maximus that the specified area is accessible as a NetMail message area. If the area uses the *.MSG format, <path> should specify the name of the message directory. If the area uses the Squish format, <path> should specify the path and root filename of the message area. This statement CANNOT be used in conjunction with either `EchoMail` or `Local` or `Conference`.

MsgAccess <priv>[/<lock>]

This keyword specifies the access level required to access (or to even see) this message area.

MsgBarricade <filespec>

This keyword specifies the barricade file to be used for the current message area.

MsgInfo <desc>

This keyword specifies the name and description of the current message area, as you wish it to appear to the user.

MsgMenuName <name>

The MsgMenuName token instructs Maximus to use a different *.MNU file when a user enters the specified message area. <name> should be just the root of the menu name, without a path, and without the .MNU extension. For example, the statement `MsgMenuName MSG2` would cause Maximus to use the menu `MSG2.MNU` while the user is in the given area.

MsgName <tag>

This keyword tells Maximus the name of the `area tag' for the current message area. This tag is used for the Echo TossLog and should be the same tag as specified in AREAS.BBS or SQUISH.CFG.

MsgOverride <cmd> <priv>[/<lock>]

By using the MsgOverride command, you can alter the priv level required to access a specific message-menu option, for only one specific area, without going to all the bother of defining a separate menu. <cmd> is simply the first word (or even first letter) of the command whose priv level you wish to alter. <priv>[/<lock>] indicates the override priv level which you want to assign to the command while in this area. Unlike other<tm> programs, priv-level overriding is not limited to only a fixed set of commands. You can override the priv level required for ANY command, up to sixteen per area.

Examples:

MsgOverride Enter AsstSysOp

This command would set the priv level for the ENTER command to AsstSysOp.

MsgOverride K Normal/25

This command would set the priv level for KILL (or whichever command started with the letter `K') to Normal, and would require that the user has keys #2 and #5 to access the command.

No Realname Kludge

This keyword TOGGLES the setting of the `No Realname Kludge' option in MAX.CTL. If the MAX.CTL keyword has been used to disable the ^aREALNAME kludge, then adding this option to a definition will enable the kludge for the current area only. Similarly, when the keyword in MAX.CTL has NOT been enabled, adding this option to an area definition will disable the ^aREALNAME kludge for that area only.

Origin <aka_num> [origin]

This keyword tells Maximus to use an origin line in this area OTHER THAN THE DEFAULT. <aka_num> specifies which address number to use in the origin line; an <aka_num> of `0' being the FIRST address specified in MAX.CTL, an <aka_num> of `1' being the SECOND address specified in MAX.CTL, and so on. [origin] is the optional custom origin line for this area. If this is not specified, the default origin line in MAX.CTL will be used.

For example, the following statement:

```
Origin          0      Fowl Weather Post * We fly!
```

...would cause the origin line to appear like this, assuming that the first address declared in MAX.CTL was `1:249/106':

```
* Origin: Fowl Weather Post * We fly! (1:249/106)
```

Private and Public

This keyword instructs Maximus to allow users to enter BOTH private and public messages in this area. This keyword can be modified by an optional access level. See the section on privilege level modifiers for details.

Private Only

This keyword instructs Maximus to allow only PRIVATE messages to be entered in this area. This keyword can be modified by an optional access level. See the section on privilege level modifiers for details.

Public and Private

This keyword is identical to `Private and Public', and allows users to enter BOTH public and private messages in this area. This keyword can be modified by an optional access level. See the section on privilege level modifiers for details.

Public Only

This keyword instructs Maximus to allow only PUBLIC messages to be entered in this area. This keyword can be modified by an optional access level. See the section on privilege level modifiers for details.

Read-Only

This keyword tells Maximus that NO messages can be entered in this area, unless the user's privilege level is at least AsstSysOp. If the user's privilege is not at least AsstSysOp and if the file `READONLY.BBS` exists in the message subdirectory, then Maximus will display it to the user when he/she tries to enter a message. If READONLY.BBS does not exist, then Maximus will default to displaying, `This area is READ-ONLY`.

Renum Days <num>

Sets the maximum number of days of messages to keep on line in a given area. For *.MSG areas, this keyword is only used by the MR renumbering program. For Squish areas, this setting causes SQPACK to delete all messages older than <num> days old.

NOTE! If you are using Squish areas, this option has to be set individually for each area. The SquishMail message processor is also capable of setting this field, through the -\$d switch in SQUISH.CFG; however, you only need to specify this renumbering information for each area ONCE; "Renum" settings made in MSGAREA.CTL will automatically take effect when Squish is writing to the message base, and vice versa.

Renum Max <num>

Sets the maximum number of messages that are kept on line in a given message area. If no limit is specified, message areas will keep on growing and growing. At this point in time, the number specified for Renum Max is used only by MR. If you are using another renumbering utility, you may need to specify the maximum number of messages another way.

For Squish-style areas, this controls the maximum number of messages to keep in that area at any one time. Messages will be dynamically purged as the area grows, so you don't need a separate renumbering program. (However, you may wish to obtain a copy of SQPACK from the Squish distribution

archive, since it allows you to get rid of any "holes" in Squish-style message areas.)

NOTE! If you are using Squish areas, this option only needs to be set in one place. You can either use "Renum Max" for all areas in MSGAREA.CTL, or "-\$m" for all areas in SQUISH.CFG; specifying this value in both is unnecessary.

Type <format>

This keyword instructs Maximus to use a particular storage format for the current message area. <format> can be either "*.MSG" (FTSC-0001, the default) or "Squish" (the Maximus/Squish format message base).

Use Realname

This keyword tells Maximus to insert the user's real name, if applicable, when messages are entered in this area. Some EchoMail area policies may dictate that all users must use real names; if you run where users are allowed to use aliases (or 'handles') when they log on, this keyword can be enabled for the above-mentioned areas, so that the user's real name is placed in the message header.

FILEAREA.CTL Reference

FILEAREA.CTL is used to control all of the file areas on a Maximus system; to define a new file area, simply add an area entry to this file.

Alphabetical keyword listing

Access <priv>[/<lock>]

This keyword is obsolete. See also `MsgAccess`.

Area <area>

This keyword is a SILT directive which tells SILT that the current area definition begins here. <area> must be no longer than nine characters, but it can be composed of any alphanumeric character. For example, `1`, `2`, `A`, `B`, `AA`, `H1`, `NETMAIL`, and `CHATTER` are all acceptable. An area definition is ended by the `End Area` directive.

Barricade <filespec>

This keyword is obsolete. See also `FileBarricade`.

End Area

This keyword is a SILT directive; this tells SILT that the current area definition ends here.

Download <path>

This keyword tells Maximus that this area is accessible as a file area, and that the download directory for this area is <path>.

FileAccess <priv>[/<lock>]

This keyword specifies the access level required to access (or to even see) this file area.

FileBarricade <filespec>

This keyword specifies the barricade file to be used for the current file area.

FileInfo <desc>

This keyword specifies the name and description of the current file area, as you wish it to appear to the user.

FileList <filespec>

This command allows you to specify an alternate file, other than FILES.BBS, to be displayed when a user selects the Files command from the current. This is useful if you have a CD-ROM or WORM on-line, since you may need to keep the FILES.BBS-format files on a different drive. FB will also use this as the "base" filename for creating compiled file information. FB will remove the extension from the file you specify, and add .DAT, .IDX and .DMP extensions to hold the compile file information. For example, if you specified "FileList D:\Areal.Txt", FB would create binary files called D:\Areal.Dat, D:\Areal.Idx and D:\Areal.Dmp.

FileMenuName <name>

The FileMenuName token allows you to select a *.MNU file to be used while in the current file area only. <name> should be the root filename of the menu file to display. For example, the statement `FileMenuName FILE_ALT' would instruct Maximus to use the menu FILE_ALT.MNU when the user enters the file area in question.

FileOverride <cmd> <priv>[/<lock>]

The FileOverride command allows you to alter the priv level for a specific option on the file area menu, for the current area only. For more information on the use of FileOverride, please see the description for the corresponding MsgOverride command in the MSGAREA.CTL section.

Upload <path>

This keyword tells Maximus where to put files that were uploaded while the user was in this area.

Barricades and Extended Barricades

Using the `MsgBarricade' and `FileBarricade' keywords informs Maximus that you wish the current area to be protected by a password, or that you want to grant certain users a different access level when in this area (without a password). When a user enters a barricaded area, the `Uses Barricade' file will be displayed, and the user will then be given three tries to enter the correct access code. (If the user stumbled in by mistake, they can return to the area menu by pressing ENTER.)

These access codes/passwords are contained in a `barricade file', which is what the `Barricade' keyword should point to. A barricade file is a straight ASCII text file containing a list of passwords, each followed by the privilege level that is to be granted to a user who enters that password. Each line of the barricade file is in the format, ` <priv>'. The following are the contents of a sample barricade file:

```
helloworld      SysOp
kentucky        Privil
cleanse         Normal
scum            Twit
```

If a user typed the word `helloworld' at the password prompt, s/he would be let into the area, and his/her privilege level would be temporarily altered to `sysop' while s/he was in that area. If the user typed `kentucky', his/her privilege level would be temporarily altered to Privil, while inside that area. In addition, if you specify a priv level of `NoAccess', then a user who enters the password will be completely denied access, and told that the area doesn't exist.

However, there is still more. Maximus supports an `Extended Barricade File' which allows you to selectively promote specific users' access levels without requiring the user to enter a password and without using up a lock/key. Before displaying the `Uses Barricade' file, Maximus will quickly check through the barricade file to see if it using the extended barricade syntax. If so, then Maximus will skip displaying the `Uses Barricade' warning and will instead process the file directly. An extended barricade file has the following format:

```
!<user_name>   <priv>
!All           [priv]
```

The `!' in the first column of each line is NOT optional, as it is used to distinguish between normal and extended barricade files.

<user_name> specifies the name of the user whose access level you wish to raise. Since no spaces can be used in <user_name>, you must replace spaces with underscores like this: `Joe_SysOp'

If Maximus finds a specific match for the user trying to enter the barricaded area, then that user's privilege level will be altered to <priv>, with no questions asked. However, the `!All' keyword comes in handy if the user does NOT have a specific entry in the barricade file. By using `!All' by itself, Maximus will let other users into the area using their real privilege level. If you specify an optional privilege level after the `!All' keyword, then Maximus will let all other users into the area using [priv], instead of their real privilege level. The `!All' statement must be at the very END of the barricade file to function properly. Finally, you can even use the `NoAccess' pseudo-priv level, mentioned in the normal barricade description, with extended barricades. This allows you to make an area with an extended barricade to be totally invisible to certain users.

Example extended barricade file:

```
!Jesse_Hollington    Privil
!Hubert_Lai          Privil
!Steven_Bonisteel   Extra
!All Twit
```

This file would assign the privilege level of `Privil' to the first two users, give Steven Bonisteel the privilege level of `Extra', and assign everyone else the privilege level of `Twit'. If you didn't want anyone else to be allowed into the area, then you could replace the `Twit' on the last line with the word `NoAccess'.

Using Privilege Level Modifiers

By placing an optional privilege level in front of an attribute keyword (such as `Private Only', `Private and Public', etc.), you can specify different attributes for each area, based on the user's privilege level. The only restriction on using these keywords is that if you use an attribute statement WITHOUT a privilege level, it must come before any that do. In other words, the following will work as expected...

```
% Only DISGRACE and above can write messages
Public Only
Twit Read-Only
```

while this will NOT work:

```
% Don't do this!
Twit Read-Only
Public Only
```

A prime application for this feature is on a free BBS which offers EchoMail access. You would probably want to allow all callers to READ messages in these conferences, so you can attract new users. However, you also might not want to allow new users to WRITE messages, until they have learned about EchoMail etiquette.

MENUS.CTL Reference

The menus control file is the heart and soul of Maximus. It supports an extremely in-depth menu system which allows you to completely change the look and feel of your system. You can have a one-menu-does-everything system, a la PCBoard or RBBS, or you can go with the structured Opus/Fido layout. Maximus also supports the display of customized *.BBS file instead of the 'canned' menu that would otherwise be generated. The following text describes each command in MENUS.CTL in details, including examples where necessary.

Global Menu Options

'Global Menu Options' are commands which can be placed anywhere in a menu definition. These options usually effect the operation of the entire menu and are not related to any specific command or menu option.

HeaderFile <filespec>

This command defines the name of a custom *.BBS file to display when ENTERING a message/file area. This file will be displayed AFTER the built-in MenuHeader (if any), but before the MenuFile is displayed. This is useful for designing custom menus for message and file areas, since this file is only displayed when the user enters a message area, and can be used to simulate the "real" MenuHeader, even if it has been disabled. This is also an ideal location to insert the '[message]' or '[file]' MECCA tokens.

Menu <filestem>

This directive starts the definition of a new menu. <filestem> specifies the name of the menu file to be written. <filestem> should NOT include an extension, because Maximus will automatically add the '.MNU' file extension.

MenuColour <attr>

This command is only needed if using the 'MenuFile <filespec>' keyword and also using the HOTFLASH help mode. When hotkeys are turned on, and a custom *.BBS file is displayed, then Maximus will behave as QuickBBS and other systems do. Namely, if the user presses a key while the menu is still being displayed, Maximus will not display the remainder of the file but will instead immediately process

the keystroke. Since there may be some strange colour combinations in your menu (such as a non-black background), were Maximus to simply jump to the requested option, the screen would look ugly. This keyword is a cure for that, since Maximus will change the colour to <attr> before displaying the hotkeyed keystroke which the user typed. <attr> is an AVATAR colour code, a list of which can be found in the appendices.

MenuFile <filename> [helplevel...]

This instructs Maximus to display <filename>.BBS, instead of displaying a 'canned' Maximus-generated menu. If you are using this option for the message menu, it is strongly advised that you also use the 'MenuLength <length>' keyword in conjunction with this command.

By default, a MenuFile will be used for all help levels. However, if you only wish to use a MenuFile for certain levels (such as NOVICE and REGULAR), you can add zero or more qualifiers after the filename. [helplevel] can be any of the following keywords:

- Novice
- Regular
- Expert
- Hotflash

For example, this:

```
MenuFile Misc\MenuMain
```

would display the menu to everyone. However, the following:

```
MenuFile Misc\MenuMain Novice Regular
```

would only display the menu to callers with a help level of NOVICE or REGULAR. This command:

```
MenuFile Misc\MenuMain Novice Regular Expert Hotflash
```

shows the help file to all help levels, which is the equivalent of this:

```
MenuFile Misc\MenuMain
```

MenuHeader <header_type>

This keyword tells Maximus what to display when this menu is first displayed. <header_type> can be any of the following values. Each of the separate MenuHeaders will display something different:

Change: All of the user's statistics are displayed.

Chat: A list of on-line users, and the current user's chat availability.

File: The name of the current file area is displayed.

Message: The standard message-area information is displayed, including the area name, current message number, etc.

None: Nothing is shown before the menu is displayed.

MenuLength <length>

This command is only needed if using the `MenuFile <filespec>' keyword. This option tells Maximus how many lines long your custom *.BBS file is, so that messages and other displayed items will not scroll off the screen while the custom menu is being displayed. Currently, this keyword is only required on message-area menus.

SilentMenuHeader <header_type>

In all respects but one, this command is identical to the MenuHeader keyword. By using this keyword, Max will still go through all of the standard message and file-area entry routines, but nothing will be displayed to the user. If you have defined a new look and feel for your message menu, you should use "SilentMenuHeader Message" instead of turning off the MenuHeader completely.

Title <name>

This keyword defines the name of the current menu, as it will appear to the user.

Menu Option Modifiers

These commands are simple `flags', which can be placed in front of another menu option, to modify its operation in some respect.

Some modifiers can be used only with certain menu options; if this is the case, then it will be indicated in the text below.

Ctl <option>

This is used as a modifier for menu options, which are described below this alphabetical keyword list. The `Ctl` keyword is only useful when running Opus-compatible external programs. If you have a program that can ONLY be run from the old Opus `External_XXX_Mgt` option, then you should insert the `Ctl` keyword on the same line, BEFORE the `Xtern_XXX` token.

Conf <option>

This is a modifier which can be inserted at the beginning of a menu option. If this keyword is specified, then the following command will only be visible in a CONFERENCE message area.

Echo <option>

This is another modifier which can be inserted at the beginning of a menu option. If this keyword is specified, then the following command will only be visible in an ECHOMAIL message area.

Local <option>

This keyword is identical to `Echo` except that in this case, the command will only be visible in a LOCAL message area. Make sure not to confuse this option with `UsrLocal`, which causes the current line to be displayed only if a user is calling locally.

Matrix <option>

This keyword is identical to `Local`, except that the command will only be visible in a MATRIX message area.

NoCLS <option>

This keyword is only useful when used with the `Display_Menu` option. When using the `NoCLS` keyword, Maximus will NOT clear the screen before displaying the specified menu.

NoDsp <option>

This keyword is another menu option modifier. If you place `NoDsp' before a menu option, then the option will NOT be displayed to the user, but will still be accessible from the menu. This is useful for having hidden commands, and also for using `linked' menu options. (See below for details.)

ReRead

This modifier is only useful when used in conjunction with either of the `Xtern_Dos' or `Xtern_Run' menu options. If this flag is used, then Maximus will re-read the LASTUS*.BBS file for the current task, after returning from the external command. This is useful in situations where the external program modifies the user's statistics by changing LASTUS*.BBS, and you want Maximus to recognize those changes, by re-reading that file upon return. Note that this operation is implied for the Xtern_Erlvl and Xtern_Chain menu options, so specifying `ReRead' for either of the two is superfluous. Also, you can achieve the same effect through the `[dos]', `[xtern_dos]', and `[xtern_run]' MECCA tokens, by using a `@' character as the first character of the program name to run.

Stay

This modifier can be used for a Display_Menu command. Normally, when Max executes a Display_Menu, it "leaves" the current message or file area. (If you are using the standard menuheaders, this would cause the header to be redisplayed.) However, "Stay" instructs Maximus to stay in the current message/file area when displaying the new menu, which is ideal for a "READ" sub-menu or other menus which hang off the main message menu.

UsrLocal

The `UsrLocal' modifier tells Maximus that the specified option should ONLY be displayed (and be selectable) by a local caller. This is useful if you have several menu commands which cannot be run from remote, such as those which use direct screen writes for output. Make sure not to confuse this with the `Local' modifier, which acts on the type of the current message area, not the user's local/remote status. See also `UsrRemote'.

UsrRemote

This modifier tells Maximus to make the current command only available to REMOTE users. This can be useful for hiding commands which you don't want to be seen or used locally, such as a call-back verification program, or others which deal exclusively with the modem. See also `UsrLocal`.

Menu Option Format

Aside from the compiler directives and handling commands described above, the menus are comprised of Maximus' menu options. The menu option definitions each consist of one line, and have several optional parameters. You can have up to 127 options on one single menu.

The general format of a menu option is:

```
[modifier] <option_name> [arg] <priv> "<desc>" ["key_poke"]
```

[modifier] can be any of the menu-option modifiers, described in the previous section.

<option_name> is the operation which is performed when a user executes this command. <option_name> be any of the keywords specified in the next section, the alphabetical menu option listing.

[arg] is not needed for some options, and is mandatory for others. If it is not specifically stated that the command needs an argument, then you may not use one.

<priv> is the priv level required to access this menu option. If you want to place a lock on this option, you can follow the priv level with a slash, followed by the number(s) of the locks to close for this option. ie. An option with a <priv> of "Disgrace/127" would only be accessible to users with a priv of Disgrace or above, and had keys #1, #2 and #7. Note there are NO spaces between the privilege, slash, and keys.

<desc> should be the description of the command, as you wish it to appear on the menu. Since the FIRST letter of the description is what activates the option, be sure to not use the same first letter more than once.

["key_poke"] is optional, and can contain a sequence to insert into the keyboard buffer, before this command is executed. This sequence is identical to that used by the Key_Poke menu option, except for the fact that it must be surrounded by quotes. Please

see the description for Key_Poke in the MENUS.CTL reference, for more details on this feature.

Alphabetical menu option listing

Area_Change <type>

Invokes the Area Change command. <type> can be either `Message' or `File', and controls which list of areas is displayed to the user when they press `?'.

Chat_CB

This function invokes the multi-node chat, in the group chat (CB simulator) mode. Please see the section on Multi-Line Operations, in the Maximus Operations Manual.

Chat_Page

This command will send a message to the user on the specified node, indicating that the current user wants to initiate a chat. After the message is sent, Maximus will deposit the user inside the multi-line chat, where s/he can wait for the other user to respond to the page. Please see the section on Multi-Line Operations, in the Maximus Operations Manual.

Chat_Pvt

This function invokes the multi-node chat, in the private chat mode. Please see the section on Multi-Line Operations, in the Maximus Operations Manual.

Chat_Toggle

This function toggles the current user's chat availability. Please see the section on Multi-Line Operations, in the Maximus Operations Manual.

Chg_Alias

This allows the user to change the data which is stored in USER.BBS regarding his/her alias.

Chg_Archiver

This allows the user to specify the default compression method they wish to use for downloading their QWK packets for their offline reader. By default, they are asked each time they download, what compression method they want. Once the default is specified by this menu option, the compression question will not be asked again.

Chg_City

This allows the user to change the data which is stored in USER.BBS regarding his/her city and state/province.

Chg_Clear

This toggles the user's clearscreen setting.

Chg_Editor

This toggles the user's setting between the BORED line-oriented editor, and the MaxEd full-screen editor.

Chg_FSR

This allows the user to toggle the full screen message reader option. By enabling this option with ANSI or AVATAR graphics capabilities is presented an attractive header for each message as it is read, or when a new message is to be created. All changeable fields in the header are presented and the user can move to these fields, instead of answering interactive questions for the various fields.

Chg_Help

This changes the user's help level to any of four settings.

Chg_Hotkeys

This command toggles the user's hotkeys setting.

Chg_IBM

This toggles whether Maximus will send IBM-specific 'extended ASCII' characters directly to the user or whether it will translate them to their ASCII equivalents.

Chg_Language

This allows the user to specify which language will be their default for all Maximus menus and prompts.

Chg_Length

This command lets the user change his screen length setting.

Chg_More

This toggles whether or not the user wishes to receive `More [Y,n,=]?' prompts.

Chg_Nulls

This option allows the user to change the number of NULLs sent after every line transmitted.

Chg_Password

This option allows the user to change his password.

Chg_Phone

This allows the user to change the data which is stored in USER.BBS regarding his/her telephone number.

Chg_Protocol

This gives the user the option to specify a default protocol for all file transfers to and from Maximus. This includes file upload, file download, offline reader message download, offline reader upload, and single message upload. Once a default is selected, the user is not asked to specify a protocol when a file transfer is requested.

Chg_Realname

This menu item is retained for Maximus 1.02 compatibility, however its meaning has been changed. It is now equivalent to the Chg_Alias menu selection. Refer to the section on Maximus's alias system in the Operations Manual.

Chg_Tabs

This option allows the user to tell Maximus whether it should transmit tab characters (which is faster) or whether it should translate the tab characters into the equivalent number of space characters.

Chg_Userlist

This option toggles whether or not the current user can be displayed in the userlist. If this option is set to NO, then the user will never be displayed in the userlist. If this option is set to YES, then the user WILL be displayed, as long as his/her privilege level falls inside the range defined in MAX.CTL by UserList Minimum/Maximum.

Chg_Video

This option lets the user select a new video mode.

Chg_Width

This command lets the user change his/her screen width.

Clear_Stacked

This command will clear the user's command-stack buffer, and kill any commands which the user has stacked previously.

Contents

This command will allow a user to view the contents of a compressed files. The supported archive types are ARC, ARJ, PAK, ZIP, and LZH.

Display_File <filespec>

This command will display the file <filespec>.BBS to the user. If you wish to use external program translation characters, you can still do so. However, instead of using a percent sign ('%') as a lead-in sequence, you should use a plus sign ('+') instead. For example, to display a file called `Bps', where is the current baud rate, you would do something like this:

```
Display_File D:\Bps+B Disgrace "Bps-specif. file"
```

Display_Menu <name>

This command will cause <name>.MNU to be displayed. Do NOT add a path or extension to this name, since Maximus will do that for you. Also, `Display_Menu' calls are NOT nested. In other words, if the callee menu wants to be able to return to the caller, it must have a corresponding `Display_Menu' statement with the caller's name.

Download

This command allows the user to download a file from the current file area.

Edit_Abort

This command will only work from the line editor. Selecting this command will abort the entry of the current message.

Edit_Continue

This command will only work from the line editor. Selecting this will allow a user to continue entering his/her message.

Edit_Delete

This command will only work from the line editor. Selecting this will allow a user to delete a line from his/her message.

Edit_Edit

This command will only work from the line editor. Selecting this command will allow a user to edit a line in his/her message.

Edit_From

This command will work from either the line editor or the full-screen editor. Selecting this will allow a user to edit the `From:' field of his/her message.

Edit_Handling

This command will work from either the line editor or the full-screen editor. This command will let the user toggle the message attribute bits, including the private bit, crash bit, file attach bit, etc.

Edit_Insert

This command will only work from the line editor. Selecting this will allow a user to delete a line from his/her message.

Edit_List

This command will only work from the line editor. Selecting this will allow a user to list his/her message currently being entered.

Edit_Quote

This command will only work from the line editor. Selecting this will allow a user to quote the message to which he is replying.

Edit_Save

This command will only work from the line editor. Selecting this will allow a user to save his/her message to disk.

Edit_Subj

This command will only from either the line editor or the full-screen editor. Selecting this will allow a user to edit the `Subject:` field of his/her message.

Edit_To

This command will work from either the line editor or the full-screen editor. Selecting this will allow a user to edit the `To:` field of his/her message.

Enter_Message

Selecting this will invoke the message-entry routine, and let the user enter a message.

File

This command will cause Maximus to act just as though it had encountered a `MenuHeader File' statement, but without the associated display.

File_Hurl

Selecting this command allows a user to move a file from one file area to another.

File_Kill

Selecting this command allows a user to delete a file from the current file area.

File_Tag

The File_Tag command is used to "tag" a specified file, and place it in a queue of filenames to be downloaded. The Tag command works across different areas, so a user may tag a file, switch to a different area, tag another file, and then download both at once.

File_Titles

When this option is selected, Maximus will display a list of all of the files in the current file area, including their names, date stamps, and descriptions. This file listing is contained in FILES.BBS.

Forward

This option allows a user to forward a copy of the current message to another user.

Goodbye

Selecting this command will log off the current user.

Key_Poke

This command will insert the specified keystrokes into the user's type ahead buffer, just as if the user had entered them manually. Make sure to replace any spaces with underscores.

Example:

```
Key_Poke m;lv* Disgrace "*List new msgs"
```

The above command would, if executed from the message menu, display a list of all new messages in the current area. You can also use outside program translation characters in the key-poke sequence, and have the translation inserted into the keyboard buffer. For example, from the main menu, you could use the sequence `@%u' to invoke the user editor, and display the current user.

Note! Keys can also be implicitly poked in the keyboard buffer by placing an extra set of quotation marks after the option name. For example, the following menu option:

```
Msg_Browse Disgrace "Download" "tnp"
```

would automatically place 'T', 'N' and 'P' in the keyboard buffer before executing the specified command.

Finally, if you wish to keypoke an <enter> keystroke, a ";" or a "| " will work in most (but not all) places.

Leave_Comment

This command puts the user in the message editor, and allows them to edit a message addressed to the sysop. The message will be saved in Area 0 or the area defined by "Comment Area" in MAX.CTL.

Locate

This command is used to search all of the file areas on the system for a file which has a certain filename or description.

Message

This command will cause Maximus to act as though it had encountered a `MenuHeader Message' statement, but without the associated display.

Msg_Browse

This command invokes the Browse function which allows a user to selectively read, list, or pack (for offline reader) messages by area selection and message header or message body search criteria.

Msg_Change

The Msg_Change command will allow the user to CHANGE a message that was previously entered, as long as the message has not been received by the addressee, scanned as EchoMail, or packed and sent as a netmail message. (Although Maximus won't allow a user to change one of the above-mentioned message types, it will allow the SysOp to do so, since it's assumed that the system operator knows what s/he is doing.) This command works with both MaxEd, BORED, and also with any local editor you may be using.

Msg_Checkmail

This command invokes the built-in mailchecker, and is identical to the MECCA `[msg_checkmail]' command.

Msg_Current

This command displays the current message to the user.

Msg_Edit_User

This command is used in the message section and will invoke the user editor for the user listed in the FROM: field of the current message.

Msg_Hurl

This command lets a user move a message from one area to another.

Msg_Kill

This command allows a user to delete a message. If the user's privilege level is lower than `sysop`, then he can only delete messages which are addressed to or written by himself/herself. If his/her privilege level is `sysop`, then he/she can delete any message.

Msg_List

This command allows a user to display a short list of messages in the current area, optionally with subject headings.

Msg_Reply

This allows the user to reply to the current message in the current area.

Msg_Tag

This command allows a user to "tag" (preselect) specific message areas of interest for use with the B)rowse command. Note that the Msgformat display must be enabled in MAX.CTL with the `%*' formatting character, in order for the tag characters to show up with this command.

Msg_Upload

This allows a user to enter a message, but instead of invoking one of the editors, Maximus will allow the user to upload an ASCII text file containing the body of hi/her message, using their choice of protocols. Maximus will then save the message normally. High bit characters are allowed, but only in the areas in which the `High Bit Allowed' option is enabled.

Msg_Upload_Qwk

This command invokes the upload function for uploading a user's "reply" packet from their QWK offline reader.

NewFiles

This command is identical to the `[newfiles]' MECCA token. This menu option causes Maximus to search for new files in all file areas.

Other

This command will cause Maximus to act as though it had encountered a `MenuHeader None' statement, but without the associated display.

Override_Path

This command will allow a user to temporarily change the path which the current file area points to, for uploads and downloads. Any path can be changed to when using this command, including paths on another drive, so it is not advisable to set the priv for this command below SysOp. This command is only usable from inside a file area.

Press_Enter

This option will set the text colour to white, and prompt the user to press <Enter>. This is most useful in conjunction with linking menu options.

Raw

This allows a user to display a raw directory of all files in the current area, whether or not they are contained in FILES.BBS.

Read_DiskFile

This command will work from either the line editor or the full-screen editor. Selecting this option allows the user to import an ASCII text file from the local disk and incorporate into his/her message.

Read_Individual

This command allows the user to jump directly to and read a specific message, specified by its number, in the current area.

Read_Next

This will allow the user to read the message after the current message.

Read_Nonstop

This will consecutively display all of the messages after the current message, without stopping.

Read_Original

This will allow the user to read the message (if it exists) to which the current message is a reply.

Read_Previous

This will allow the user to read the message before the current message.

Read_Reply

This will display the message (if it exists) which is the reply TO the current message.

Same_Direction

This will allow the user to read the message either before or after the current message, depending on whether 'Read_Previous' or 'Read_Next' was last selected.

Statistics

The 'Statistics' command will display the user's statistics, including the amount of time he/she has spent on-line, the amount he/she has downloaded, etc.

Type

This command will allow a user to display an ASCII text file in the current file area.

Upload

This will allow a user to upload a file.

User_Editor

This will allow a user to invoke the user editor, which can be used to edit the user records of every user on the system. Generally, the only person who should have access to this option is the sysop.

Userlist

This will display a list of all users on the system who have a privilege level at least as high as that specified by the 'Userlist Minimum' statement and no higher than that specified by the 'Userlist Maximum' statement. Both of these two statements are located in MAX.CTL. The user's city and date of last call will also be displayed.

Version

This will display the version number (and some other information) about the revision of Maximus that is being executed.

Who_Is_On

On a multi-line system, this displays the names, task numbers, and status of users on other nodes.

Xport

This will export a message to an ASCII text file on disk.

Xtern_Chain <cmd>

This will cause Maximus to run the external program <cmd>. See the chapter on external programs for an explanation of the differences between the four types of external commands. If <cmd> has any arguments, make sure to replace any SPACES with an UNDERSCORE, or else the arguments will not be passed correctly.

Xtern_Dos <cmd>

This will cause Maximus to run the external program or batch file called <cmd>. See the chapter on external programs for an explanation of the differences between the four types of external commands. If <cmd> has any arguments, make sure to replace any SPACES with an UNDERSCORE, or else the arguments will not be passed correctly.

Xtern_Erlvl <errorlevel>[_<cmd>]

This will cause Maximus to terminate with an errorlevel of <errorlevel> to run an external program. See the chapter on external programs for an explanation of the differences between the four types of external commands. If <cmd> has any arguments, make sure to replace any SPACES with an UNDERSCORE, or else the arguments will not be passed correctly.

Xtern_Run <cmd>

This will cause Maximus to run the external program called <cmd>. See the chapter on external programs for an explanation of the differences between the four types of external commands. If <cmd> has any arguments, make sure to replace any SPACES with an UNDERSCORE, or else the arguments will not be passed correctly.

Yell

Selecting this command will generate some noise on the SysOp's console, informing the SysOp that the user would like to chat.

Linking Menu Options

Maximus' menu system is very flexible, as it allows you to place as many options as you want, anywhere you want. You can therefore imitate the style of a lot of the more popular BBS systems. In addition to that, the menu system supports one more important feature: command linking.

When processing a user's menu choice, Maximus will read through the entire menu looking for a match. As expected, when Maximus finds the first match for the menu option it will perform the command. However, unlike other BBS programs, after Maximus executes the command, it **KEEPS ON SEARCHING**. Thus, you can have Maximus execute as many commands on a single keystroke as you want it to. You can use the `NoDsp <option>' keyword in

conjunction with this to keep the rest of the linked commands invisible on the menu.

This may not seem very useful at first, but consider this scenario:

A user selects a message area, and begins to read messages. Since there are over a dozen or so options in the message area, the menu can get fairly large, and it will leave less space on-screen for messages. If you use something like this in your message-area menu...

	Read_Next		Twit	"Next Message"
	Read_Previous		Twit	"Previous Message"
NoDsp	Display_Menu	ReadMsg	Twit	"N"
NoDsp	Display_Menu	ReadMsg	Twit	"P"

...you can then have the `ReadMsg' menu displayed after the user has read a message. This could be a very short menu, with perhaps one or two options, such as <C/R> to keep reading in the same direction, and `Q' to quit back to the message menu. If you implement a customized menu for the `ReadMsg' menu, you can see that the possibilities for command linking are almost endless, and can greatly enhance the look and feel of your BBS.

PROTOCOL.CTL Reference

Max is capable of directly using external protocols such as DSZ, MPT, Kermit, and others. Max has a configurable, control-file-drive protocol system which permits the use of almost any external protocol.

In addition to "standard" protocols such as DSZ, Max also supports "Opus-compatible" protocols, such as OKermit, OASCII and others. These protocols must also be defined in this control file, although with a slightly different format. (See the "Examples" subsection for details.)

Alphabetical Keyword Listing

ControlFile <filespec>

This keyword defines the name of a control file to create for this protocol. The text contained within this control file has a definable format, including support for both DSZ and Opus-compatible protocols. If you run multiple nodes, be sure to specify a task number as part of the control file name (through the "%K" token) to ensure that the same control file is not used by multiple tasks.

DescriptWord <num>

When parsing the upload log, this keyword defines the "word number" of the upload description. Max will search for the <num>th word after the `UploadKeyword`; everything from that word on will be interpreted as the description for the uploaded file. If the upload log doesn't include descriptions, use 0 for <num>. For example, if the upload log looked like this:

```
= 10 Sep 14:10:10 FROG Got \upl\maxdocs.zip Maximus docs
```

you would specify `DescriptKeyword 2`, since the description of `Maximus docs`, begins two words after the `Got` UploadKeyword.

DownloadCmd <cmd>

This keyword specifies the command to execute when a user downloads a file using the current protocol. For example, an Opus-compatible protocol normally uses the following format:

```
DownloadCmd <n>.Exe <n>%K.Ctl -p%p -b%b -t%k -m%d -f%D -r%t
```

where <n> is the name of the external protocol, such as "ASCII" or "Kermit".

DownloadKeyword <keyword>

When parsing the download log created by the external protocol, this keyword identifies the "user downloaded a file" string. For example, with Opus-compatible protocols, the 'Sent' keyword indicates that a file was downloaded. To search for a string containing spaces, use double quotes around the keyword.

DownloadString <cmd>

This keyword specifies a command to place in the download control file. This command is written once for each file that the user requested to download. If a "%s" is included in the command string, that "%s" will be translated to the name of the file to be sent. For Opus-compatible protocols, a DownloadString of "Send %s" should be used.

End Protocol

This keyword marks the end of a protocol definition.

FilenameWord <num>

When parsing the download log, this keyword defines the "word number" of the uploaded filename. Max will search for the <num>th word after the 'UploadKeyword'; everything from that word on will be interpreted as the name of the uploaded file. For example, if the upload log used this format:

```
= 10 Sep 14:10:10 FROG Got \upload\maxdocs.zip Maximus docs
```

you would specify a FilenameWord of 1, since '\upload\maxdocs.zip' is the first word after the 'Got' (the UploadString).

LogFile <filespec>

This keyword defines the location of the external protocol's log file. "%" translations characters can be used in this command. This file will be scanned for both download and upload file information, as specified by the DownloadString and UploadString.

Protocol <name>

This keyword signifies the beginning of a protocol definition. Since <name>, is used to identify the protocol on the protocol option menu, the first letter of the protocol name should be unique.

Type Batch

Type Errorlevel

Type Bi

Type Opus

These optional modifiers are used to modify the operation of an external protocol. Any or all of these keywords may be used in one protocol definition:

- Batch: The specified protocol accepts more than one file at a time, and filenames are sent along with the files themselves.
- Errorlevel: To perform an upload or a download, Max will exit with the "External Protocol Errorlevel", as given in the session section of MAX.CTL. A file called ERRORLxx.BAT will be created containing the name of the protocol to run, and Max can then be reloaded with the -r switch. See the "Errorlevel Batch Files" section of the Maximus Operations Manual for more information.
- Bi: The protocol can transfer files both ways at the same time. Max will scan the log for both upload and download entries at the same time.
- Opus: Maximus will generate Opus-compatible information at the beginning of the protocol control file.

UploadCmd <cmd>

This keyword specifies the command to execute when a user uploads a file using the current protocol. Percent translation characters may be used. The following command string is normally used for an Opus-compatible protocol:

```
UploadCmd <n>.Exe <n>%K.Ctl -p%p -b%b -t%k -m%d -f%D -r%t
```

where <n> is the name of the protocol to execute.

UploadString <cmd>

This defines the string which indicates that a file is to be uploaded. A "%s" in <cmd> will be translated to the name of the file to receive. Currently, "%s" will either translate to the filename entered by the user (for a non-batch protocol), or a filespec in the "D:\Path*.*" format for batch transfers. External protocols should place uploaded files in the specified directory, log those filenames in the protocol log file, and return to Maximus.

UploadKeyword <keyword>

This keyword defines the word in the upload log which specifies that a file was uploaded. For example, Opus-compatible protocols use an UploadKeyword of 'Got'. To search for a string containing spaces, use double quotes around the keyword.

Examples

Sample protocol entries for BiModem, DSZ (Zmodem MobyTurbo), OASCII, OKermit and MPt are contained in the distribution version of PROTOCOL.CTL. However, these protocol entries are commented out by default; to enable a protocol, simply uncomment all of the lines belonging to that protocol.

If you are using an Opus-compatible external protocol, the entry in PROTOCOL.CTL should have the following form:

```

Protocol <name>
  Type           Batch
  Type           Opus
  LogFile        <name>%K.Log
  ControlFile    <name>%K.Ctl
  DownloadCmd    <name>.Exe <name>%K.Ctl -p%p -b%b -t%k -m%d
                 -f%D -r%t (cont'd from line above)
  UploadCmd      <name>.Exe <name>%K.Ctl -p%p -b%b -t%k -m%d
                 -f%D -r%t (cont'd from line above)
  DownloadString Send %s
  UploadString   Get %s
  DownloadKeyword Sent
  UploadKeyword  Got
  FilenameWord   1
  DescriptWord   4
End Protocol

```

Replace each instance of <name> with the name of the external protocol.

MECCA LANGUAGE REFERENCE

This chapter serves as a reference guide to the MECCA language. The MECCA compiler itself is not documented here; see the chapter on Maximus Utilities to learn more about the compiler.

Usage Guide

The MECCA language gives the SysOp a large amount of flexibility when designing screens which are to be displayed to the user. MECCA allows you to imbed personalized information about each user in text screens, change the screen colour, run external programs, and a variety of other features.

The input file for the MECCA compiler, which is usually a file ending with a *.MEC extension, is composed of straight ASCII text. The file can also optionally contain special tokens which are interpreted and translated by the MECCA compiler. MECCA tokens are delimited by a pair of SQUARE BRACKETS. Anything outside of square brackets is treated as straight text, and will be displayed exactly as-is to the user. (A list of usable tokens follows in the alphabetical token listing.)

This makes it possible to use the names of the MECCA tokens themselves in a display, without having the MECCA compiler translate it into the code which Maximus wishes to see.

For example, entering the following example sentence in a *.MEC file:

```
This is your [usercall] call.
```

might be displayed by Maximus as follows, after being compiled with MECCA:

```
This is your 14th call.
```

Since MECCA will only compile tokens contained inside of square brackets, it left the second `times' as normal text.

Also, if you wish to include a left square bracket directly into a display, simply use two left brackets instead of one. Only the left square bracket needs to be doubled, since that is the starting character of a MECCA token, and is the only character checked when scanning for MECCA tokens.

In other words, if you wanted to display the following line to a user:

Want to check for your mail [Y,n]?

...you would have to enter it like this inside a .MEC file:

Want to check for your mail [[Y,n]?

Now, on to using MECCA's tokens. Although an alphabetical token listing follows below, there are a few things of which you should be aware of before using these tokens:

- * Tokens are NOT case-sensitive. That means that using `[user]', `[USER]', and `[UsEr]' will all produce identical results.
- * Spaces are ignored. That means that `[user]', `[user]' and `[user]' are also equivalent. You can use a space, tab or return inside a MECCA token, and still have it compile normally.
- * MECCA allows you to place more than one token inside a set of square brackets, as long as the token is separated from the others by a space. In other words, `[lightblue blink user]' is equivalent to the older fashioned way of writing `[lightblue][blink][user]'. It also makes creating *.MEC files faster, because it reduces the number of characters needed to perform the same action.

MECCA also allows you to enter ASCII codes into the compiled .BBS file without going to great lengths. By placing the number of the ASCII code inside a pair of square brackets, MECCA will insert the character representing that code into the compiled .BBS file. For example, using the token "[123]" would place ASCII code 123 into the output file.

In addition to the hard-coded MECCA tokens, MECCA also allows you to define your own reserved words, otherwise known as labels. When combined with the `[goto]' token, labels allow you to jump to different places in your *.MEC file based on user input, or on whether or not the user has graphics turned on, or on some other condition. A label definition looks just like an ordinary token, except that it is prefixed by a forward slash. You can use anything you want for the name of a label, as long as it is unique, composed of alphanumeric characters, and is not one of the reserved words listed in the alphabetical token listing. An example of an acceptable label definition is `[/mylabel]'.

However, after you have DEFINED a label, you must still use the name of the label itself, but without the preceding slash. The slash should ONLY be used to mark the location in the file which

the label represents. But when you are using a label with the `[goto]' token to cause Maximus to jump to the previously-marked spot, you want it to represent the LOCATION of the other spot, and therefore should not use the slash. MECCA also supports forward-referenced labels, so it is possible to have a `[goto]' instruction to reference a label before you define it.

Example #1: This will display the question `Do you want to play more games [Y,n]?' to the user. If the user answers `Y', then Maximus will display the file `C:\Maximus\Misc\Games', and go back and ask the question again. Otherwise, Maximus will quit the current file. Don't worry if you don't yet understand all of the commands used here. Concentrate on the label definitions:

```
[/askgames]Do you want to play more games? [[Y,n]? [menu]YN
[choice]Y[link]C:\Maximus\Misc\Games
[choice]Y[ goto askgames ]
[choice]N[ quit ]
```

Example #2: This demonstrates a forward-referenced label:

```
Want a "Zippy the Pinhead" quotation? [[y,n]? [menu]YN
[choice]Y[goto zippy]
[choice]N[ quit ]
[/zippy]
Okay, here's the quote!
[quote quit]
```

For more examples, have a look at the *.MEC files in the \MAX\HLP\ and \MAX\MISC\ directories.

Maximus allows you to use any of the 128 separate colour combinations on the PC for displaying text, and MECCA fully supports all of them. To specify text with a coloured background (text for which you do not specify a background colour is assumed to have a black background), instead of just placing the name of the foreground colour inside a token (i.e. `[cyan]'), add the word `on', and the name of the background colour. For example, to display text with a light green foreground and a blue background, you would enter: `[lightgreen on blue]', and MECCA would generate the appropriate code. Warning! You can use only the first eight colours (see list below) for the background colour. You cannot use any of the colours beginning with `light', nor can you use `white' or `yellow'.

MECCA also supports blinking text. To make text blink, simply insert the `[blink]' token AFTER the colour token. Text that follows this token will blink. If you forget to do the above, and put the `[blink]' BEFORE the colour token, the colour token will

override the prior blink token, and the text will not be blinking.

For example, the following will create blinking green text:

```
[green blink>Hello, world!
```

...while this will only create non-blinking text:

```
[blink green>Hello, world!
```

If the user does not support ANSI or AVATAR graphics, then Maximus will strip out any colour and cursor-movement codes before transmitting the screen to the user. Thus, it is possible to display the same file to users who have TTY, ANSI, or AVATAR terminals.

Colour Token Listing

Colours can be defined by simply entering the name of the colour inside a pair of square brackets. ie. `[lightgreen]` will set the current colour to light green. You can also specify a background colour, by doing the same as before, but adding the words `on XXX` to the end of the colour, where `xxx` is a valid background colour. For example, `[red on blue]` would set the current foreground colour to red, and the current background colour to blue. To make the colour blink (for both foreground-only, or background and foreground combinations), simply append the word `blink` to the end of the token. For example, `[lightgreen blink]` sets the current colour to blinking light green.

Colours: `[xxx]` or `[xxx on yyy]`. Blinking colours: `[xxx blink]`

	<code>[black]</code>		<code>[darkgray]</code>
	<code>[blue]</code>		<code>[lightblue]</code>
Foreground	<code>[green]</code>	Foreground	<code>[lightgreen]</code>
and	<code>[cyan]</code>	only:	<code>[lightcyan]</code>
Background:	<code>[red]</code>		<code>[lightred]</code>
	<code>[magenta]</code>		<code>[lightmagenta]</code>
	<code>[brown]</code>		<code>[yellow]</code>
	<code>[gray]</code>		<code>[white]</code>

Other tokens relating to colours are:

[BG <c>]

This token is a SILT directive which sets the current background colour to <c>, without modifying the foreground colour. For example, the sequence: `[red on blue]Hello, [BG green]user` would display the text `Hello,` in RED ON BLUE, while it would display `user` in RED ON GREEN.

[blink] - ^v^b

This causes text that follows the `[blink]` token to blink. Note that the blinking is only effective up to the NEXT colour token.

[bright]

This token is a MECCA directive, which sets the intensity bit of the current colour. The following sequence:

```
[red]Is it not a [lightred]BEAUTIFUL DAY?
```

...can be replaced with this simpler form:

```
[red]Is it not a [bright]BEAUTIFUL DAY?
```

[dim]

This token is a MECCA directive, which turns OFF the intensity bit of the current colour. Instead of adding a token which changes to the lower-intensity version of a colour, you can simply add a `[dim]` token instead. For example, this:

```
[lightgreen]H[dim]e[bright]l[dim]l[bright]o[dim]!
```

Would display the word "Hello!", with each character alternating between normal green and lightgreen.

[FG <c>]

This token is a MECCA directive which tells it to set the current foreground colour to <c>. Unlike just using one of the tokens in the above chart, the background colour is NOT altered.

Example:

```
[lightred on blue]Hi, [FG yellow]Scott[FG lightgreen]...
```

The above line would display the text `Hi,' using LIGHTRED ON BLUE, the word `Scott' in YELLOW ON BLUE, and finally display three periods with LIGHTGREEN ON BLUE.

[load]

This token is a MECCA directive, which instructs it to restore the colour which was previously saved using the `[save]' token. These two tokens can be especially useful when creating screens with backgrounds, since typing `[lightgreen on cyan]' can get repetitive. If you want to alternate between two colours, you can save the first by using `[save]', and after inserting the appropriate command to change to another colour, you can restore the first colour by simply inserting a `[load]' token.

Example:

```
[yellow on blue save]This is yellow on blue.[cleol]
[lightred on green]This is lightred on green.[cleol]
[load]This text is also yellow on blue.[cleol]
```

[on]

This is a MECCA directive which tells MECCA to interpret the next token as a background colour. See the introduction above for more information.

[save]

This keyword is a MECCA directive - it tells MECCA to save the current colour (plus blink status), and to store it for later retrieval by the `[load]' token.

[steady]

This keyword is also a MECCA directive. It instructs MECCA to turn off a previous `[blink]' command.

Example:

```
[yellow]This doesn't blink. [blink]This does.
[steady]However, this text IS non-blinking.
```


Cursor Control and Video Tokens

This section describes several video and terminal control commands, which can be used to manipulate the user's (and the local) video screen.

[bell] - ^g

This causes a beep (ASCII 07) to be generated on the user's terminal.

[bs] - ^h

This causes a backspace (ASCII 08) to be generated, which moves the cursor back one column.

[cleol] - ^v^g

This causes Maximus to send a clear-to-end-of-line command, if the user has graphics turned on.

[cls] - ^l

This clears the user's screen.

[cr] - ^m

This sends a carriage return to the user.

[down] - ^v^d

This tells Maximus to move the cursor down one line, if the user has graphics support turned on.

[left] - ^v^e

This moves the cursor one space to the left, if the user has graphics support turned on.

[lf] - ^j

This sends a linefeed to the user.

[locate <r> <c>] - ^v^h<r><c>

This command moves the cursor to the <r>th row, and the <c>th column, if the user has graphics turned on. (The top left corner of the screen is row 1, column 1.)

[tab] - ^i

This command sends a tab to the user.

[right] - ^v^f

This moves the cursor one space to the right, if the user has graphics turned on.

[sysopbell] - ^w^g

This will cause a bell to ring on the LOCAL console. The bell will NOT be transmitted to the on-line user.

[up] - ^v^c

This will move the cursor up one line, if the user has graphics turned on.

Informational Tokens

Maximus supports a large variety of tokens, which can display selected information about the user, about the system, etc. The currently supported informational tokens are:

[addr]

Retained for compatibility with OECC; this token does nothing.

[alist_file] - ^rlF

Displays file area listing, either building it on the fly according to fileformat in MAX.CTL or displays filearea.bbs, if defined in MAX.CTL.

[alist_msg] - ^rlM

Displays message area listing, either building it on the fly according to msgformat in MAX.CTL or displays msgarea.bbs, if defined in MAX.CTL.

[city] - ^f^c

This causes Maximus to display the user's city.

[date] - ^f^d

This will display the current date, in the format `dd mmm YY'.

[dl] - ^f^x

This displays the user's TOTAL number of kilobytes downloaded, including today's statistics.

[expiry_date] - ^wyD

This displays the expiration date of the current user's subscription, or "None" if the user has no expiration date.

[expiry_time] - ^wyT

This displays the time left in the current user's subscription. If the user has time remaining, this token will display "x minutes", where "x" is the number of minutes remaining in the user's subscription. If the user has no timed subscription, this token will display "None".

[file_carea] - ^wGA

This displays the area NUMBER of the user's current file area.

[file_cname] - ^wGN

This displays the NAME of the current file area. This can only be used while the user is in the files section.

[first] - ^f^f

This displays the user's first name.

[fname] - ^f^f

This displays the user's first name.

[lastcall] - ^w^a

This displays the date of the user's last call.

[length] - ^f^l

This displays the duration of this user's call, in minutes.

[minutes] - ^f^k

This displays the number of minutes for which the user has been on-line during the last 24 hours.

[msg_carea] - ^w^mA

This displays the message number of the current area.

[msg_cmsg] - ^w^mL

This displays the current message number in the current message area. This command will only work while the user is in the message section.

[msg_cname] - ^w^mN

This displays the name of the current message area. This command will only work while the user is in the message section.

[msg_hmsg] - ^w^mH

This displays the number of the highest message in the current message area. This command won't work unless a user is in a message area.

[msg_nummsg] - ^w^m#

This displays the number of messages in the current area. This command will only work while the user is in the message section.

[netbalance] - ^w^nB

This displays the current user's matrix balance (credit minus debit), in cents. See also `[netcredit]` and `[netdebit]`.

[netcredit] - ^w^nC

This displays the current user's matrix credit, in cents. See also `[netbalance]` and `[netdebit]`.

[netdebit] - ^w^nD

This displays the current user's matrix debit, in cents. See also `[netbalance]` and `[netcredit]`.

[netdl] - ^f^r

This displays the user's net downloads for today. (In other words, today's downloads minus today's uploads.)

[phone] - ^wP

This command causes Maximus to display the current user's phone number.

[ratio] - ^f^y

This will display the current user's download ratio, in the format of UL:DL.

[realname] - ^wR

This will display the real name (if applicable) of the current user.

[remain] - ^f^o

This displays the number of minutes the user has left for the current call.

[response] - ^w^e

This will display the last line entered by the user, using the `[readln]` token. This command will even work across files - if you have a `[readln]` token in one file, you can use a `[response]` token in a separate file to display the result. See also `[ifentered]`.

[syscall] - ^f^q

This displays the total number of calls your BBS has received, as an ordinal number.

[sys_name] - ^r^c

This displays the system name to the caller as defined in MAX.CTL

[sysop_name] - ^r^d

This displays the sysop's full name to the caller as defined in MAX.CTL.

[time] - ^f^t

This displays the current time, in the format `hh:mm:ss`.

[timeoff] - ^f^p

This displays the time by which the user must be off the system by. NOTE! This string also includes a carriage return for compatibility with OECC. Therefore, you should not include a carriage return after this token, unless you want a blank line after this command.

[ul] - ^f^w

This will display the user's total amount of kilobytes uploaded, including today's statistics.

[user] - ^f^b

This will display the user's full name.

[usercall] - ^f^e

This will display the number of times the current user has called your system, as an ordinal number.

Questionnaire Token Listing

The tokens in this section are useful for designing an on-line questionnaire, or for logging user information to a specific file. Although the questionnaire tokens are covered in more detail below, most questionnaires will follow the same general format, which is outlined below.

One of the first tokens in a questionnaire file should be the `[open]` token. This will open the specified questionnaire output file, to which Maximus will log the answers. (This file is human-readable, so it is possible to TYPE the answer file from DOS.)

Following the `[open]` token is usually a `[post]` token, which writes the current user's name, city, and the current time/date to the questionnaire file. (This step can be omitted, if you want the questionnaire to be anonymous.)

After the `[open]` and `[post]` tokens, the main portion of the questionnaire begins. Any responses the user gives to `[readln]` tokens will be written to the questionnaire file, as will any `[store]` commands, when used in conjunction with `[menu]` tokens. There is no limit to the number of questions you may place in a questionnaire, and you can also use these questionnaire tokens in any .MEC file which is displayed to the user.

The following tokens may be useful when designing questionnaires:

[ansopt] - ^f^v

This causes Maximus to NOT require an answer for all `[menu]` and `[readln]` tokens. See also `[ansreq]`.

[ansreq] - ^f^u

This causes Maximus to require an answer for all `[menu]` and `[readln]` tokens. See also `[ansopt]`.

[choice]<c> - ^oU<c>

This causes Maximus to process the current line ONLY if the response to the last `[menu]` choice is equal to the character <c>.

[leave_comment] - ^wK

This puts the user in the message editor, and allows him/her to enter a message to the SysOp. The message will be saved in Area 0 or the area defined by "Comment Area" in MAX.CTL. This token also allows you to do some semi-intelligent branching ([goto]s), based on what the user entered in his/her message. If the message was aborted, or the message was blank, then Maximus will SKIP displaying the rest of the line which contained the `[leave_comment]` token. A construct such as this will allow you to determine whether or not the user entered a message, and react accordingly:

```
Please leave a comment to the SysOp, [fname].  
[enter]  
[/Do_Comment leave_comment goto Successful]
```

```
You didn't leave a real message! Try again...  
[enter goto Do_Comment]
```

```
[/Successful]Thanks for leaving a comment, [fname].
```

[menu]<k> - ^oR<k>

This command will cause Maximus to prompt the user to press a key, whose value can be later manipulated using `[choice]`, `[store]`, and some of the other questionnaire commands. <k> should be a list of valid keys that the user can enter to respond to the command. (If the `[ansopt]` token has been used, then the user will be allowed to press <enter> to skip the option. On the other hand, using `[ansreq]` will force the user to enter a response.) If the user enters a key which is not listed in <k>, then Maximus will tell the user so, and prompt the user to try again. Valid characters for <k> are anything between, but not including, ASCII 32 (the space) and ASCII 127 (DEL). Even if `[ansreq]` has been used, you can still use a vertical pipe ("|") to make <enter> a valid response.

[open]<f> - ^oO<f>

This command tells Maximus to open a questionnaire answer file called <f>. See also `[post]`, `[store]` and `[readln]`. Maximus honors "%" style translation characters for creating files that are unique to node number, etc.

[post] - ^oP

This command will cause the current user's name, city, and current time/date to be written to the questionnaire answer file.

[readln]<d> - ^oN<d>

This will get a line of input from the user, and then write it to the questionnaire answer file, placing the optional one-word description <d> beside the user's answer. NOTE: by default, the `[readln]' token allows stacked commands. In other words, if a user enters `x This is the response' at a prior prompt (and presuming that `x' is a valid response for the previous prompt), Maximus would automatically assume that the `This is the response' is the response to the `[readln] token. If you wish to DISABLE this behaviour, simply include a `[clear_stacked]' token before the `[readln]'. In other words, using `[clear_stacked readln]' instead of `[readln]' will ensure that the user is forced to enter his/her response when the `[readln]' statement is encountered, and not before.

[sopen]<f> - ^oo<f>

This command tells Maximus to open a questionnaire answer file called <f>. It is identical in function to the [open] token, but is included for compatibility with other BBS support software. See also `[post]', `[store]' and `[readln]'.

[store]<d> - ^oM<d>

This writes the user's response to the last `[menu]' command into the questionnaire answer file, placing the optional one-word description <d> beside the user's answer.

[write]<l> - ^wW<l>

This command will write the line <l> directly to the questionnaire answer file, interpreting any possible external program translation characters. (See the section on running external programs for more details.)

Privilege Level Controls

[?below] - ^pB?

This command causes the rest of the line to be displayed only if the user's privilege level is BELOW or EQUAL TO `?'. `?' should be one of the following characters.

H	-	Hidden
S	-	SysOp
A	-	AsstsysOp
C	-	Clerk
E	-	Extra
F	-	Favoured
P	-	Privil
W	-	Worthy
N	-	Normal
L	-	Limited
D	-	Disgrace
T	-	Twit

[?equal] - ^pQ?

This command causes the current LINE to be displayed only if the user's privilege level is equal to the privilege level specified. See `[?below]' for possible values for `?'. `?' can also be a number from 1 to 8, and can be used to test whether or not a user has a particular key number. To test for alpha keys (A-X), see the [ifkey] token under "Lock and Key Controls". To make sure that a line is displayed to no one, use "[hequal]".

[?file] - ^p?

This command causes the rest of the FILE to be displayed only if the user's privilege level is `?' or greater. See `[?below]' for the possible values for `?'. `?' can also be a number from 1 to 8, and can be used to test whether or not a user has a particular key number.

[?line] - ^pL?

This command causes the rest of the LINE to displayed, only if the user's privilege level is `?' or greater. See `[?below]' for the possible values for `?'.

[?xclude] - ^pX?

This causes the current line to be displayed only to users whose privilege level is NOT `?'. See `[?below]' for the possible values for `?'. `?' can also be a number from 1 to 8, and can be used to test whether or not a user does NOT have a particular key number. For alpha keys (A-X), please see the [notkey] token under "Lock and Key Controls".

[priv_down] - ^wpD

This lowers the priv of the current user by one level. (ie. If the user's priv is NORMAL, the user's priv would be lowered to LIMITED.) See also `[priv_up]' and `[setpriv]'.

[priv_up] - ^wpU

This raises the priv of the current user by one level. (ie. If the user's priv is NORMAL, then the user's priv would be raised to WORTHY.) See also `[priv_down]' and `[setpriv]'.

[setpriv <priv>] - ^ws?

The [setpriv] token adjusts the current user's priv level to a certain value. <priv>, which must be inside the same set of square brackets as the `[setpriv]' token, can be any valid privilege level, or even just the first letter of a priv level name. For example, both `[setpriv Favoured]' and `[setpriv F]' would set the user's priv level to Favoured. See also `[priv_up]' and `[priv_down]'.

In addition to the above tokens, MECCA supports an alternate format for conditionally displaying (or not displaying) items, based on priv levels. Some may find this method easier to use than the above four tokens, since it allows for additional flexibility, and doesn't have as many tokens to remember.

The format for the new priv tokens is as follows:

```
[<operator> <priv> <action>]
```

<operator> specifies the operation to perform, if the specified condition is met. Valid operators are:

EQ or EQUAL	Priv is EQUAL to
NE or NOTEQUAL or UNEQUAL	Priv is NOT EQUAL to
LT or BELOW	Priv is LESS THAN
GT or ABOVE	Priv is MORE THAN
GE or AE	Priv is MORE THAN OR EQUAL to
LE or BE	Priv is LESS THAN OR EQUAL to

<priv> can be the name (or first letter) of any standard Maximus priv level, and for the EQ and NE tokens ONLY, you can also specify a key number.

<action> specifies which action you wish Maximus to perform, if the specified condition is met. <action> can be any of the following verbs:

SHOW, SEE or nothing	DISPLAY rest of line
SKIP	DON'T display rest of line
QUIT	QUIT entire file

If no <action> is specified, then "SHOW" is implied by default.

Examples:

[Equal SysOp]This line will be displayed only to SysOps!

The above line would only be displayed if the user's priv was equal to SysOp.

[LE Disgrace Show]Show only to DISGRACE!

The above line would only be displayed to callers whose priv was less-than-or-equal-to Disgrace.

[GE E Skip]Don't show to EXTRA or above!

The above line would only be displayed to callers whose priv was NOT Extra or above. (In other words, it would be displayed to those whose priv was Favoured or below.) Note that you don't have to specify the FULL priv level; it is permissible to use only the first letter, if you so desire.

[LT Normal Quit]Rest of file won't be displayed to < Normal

The above line would cause the rest of the FILE to be skipped if the user's priv level was less than Normal. (ie. If the user's priv is Limited or Twit, then the rest of the file will be skipped.)

[Above Privil]Display line only to users above Privil.

The above line would only be displayed to users whose priv level is ABOVE Privil. Note that the no <action> was specified, so `Show' was used by default.

[NE SysOp Show]This is only for non-SysOps.

This line will only be displayed to all users EXCEPT the SysOp.

Lock and Key Control

[ifkey]<keys> - ^wkI

If the specified keys are set, then the rest of the line will be displayed. You can specify as many keys as you like, but they must be separated from the rest of the line with a space.

ie. [ifkey]123a You have keys 1-3 and A set.

[notkey]<keys> - ^wkN

Same as above, except that the line will be displayed only if the specified keys are NOT set.

ie. [notkey]8b You don't have either key 8 or key b.

[keyon]<keys> - ^wkO

This command turns ON the specified keys. <keys> must be separated from the rest of the line by a space.

ie. [keyon]6abc User, you have keys A-C and 6.

[keyoff]<keys> - ^wkF

This command turns OFF the specified keys.

ie. [keyoff]fgh User, keys F, G and H have been removed

In addition, the OACOMP-like priv tokens can also be used to test for keys:

[EQ /1234abc]Keys 1-4 and keys a-c are set.

[EQ SysOp/1234abc]Keys 1-4 and keys a-c are set, and you are the sysop.

Note that NO extra space is required if you are using the OACOMP-like tokens.

Conditionals and Flow Control

[b1200] - ^w^b1

This tells Maximus to skip the current line, if the user is NOT at 1200 bps or above. You can also use the `[b1200]` token to display a line only to those UNDER 1200 bps (ie. 300 baud callers), by using a construct such as this:

```
[b1200 goto FastUser]
    You are a 300 baud user!
[goto Done]

[/FastUser]
    You are a 1200-bps or above user!
[/Done]
```

[b2400] - ^w^b2

This tells Maximus to skip the current line, if the user is NOT at 2400 bps or above.

[b9600] - ^w^b9

This tells Maximus to skip the current line, if the user is NOT at 9600 bps or above.

[col80] - ^w8

This causes Maximus to display the current line only if the user's screen is 79 columns wide or over.

[color] - ^oE

[colour] - Canadian spelling of above

This causes Maximus to display the following text (up to the next `[endcolor]` or `[endcolour]` token), only if the user has ANSI or AVATAR graphics. See also `[nocolour]`.

[endcolor] - ^oe

[endcolour] - Canadian spelling of above

This signifies the end of a sequence for colour callers only. See also `[color]`.

[expert] - ^wHE

This causes the rest of the line to be displayed only to those whose help level is EXPERT. See also `[hotflash]', `[novice]' and `[regular]'.

[exit] - ^wE

This causes Maximus to quit ALL linked files. See also `[link]' and `[quit]'.

[filenew]<f> - ^wf

This token will cause the rest of the current line to be displayed, only if the date on the file <f> is greater than that of the user's last log-on. (In other words, if the file has been updated since the last time the user called, then the rest of the line will be displayed.) The filename <f> should be separated from the rest of the line by a space.

ie. [filenew]D:\Path\Bulletin.Bbs The file has been updated!

[goto <l>] - ^oV

This tells Maximus to jump to the label <l> in the current file.

[hotflash] - ^wHH

This command causes Maximus to display the current line to only those users who have their help mode set to HOTFLASH. See also `[expert]', `[novice]' and `[regular]'.

[hotkeys] - ^rh

This command causes Maximus to display the current line to only those users who have hotkeys enabled.

[ifentered]<s> - ^we<s>

This token will compare what the user last entered through the `[readln]', to the string <s>. If the two are equal, then the rest of the line will be displayed. (<s> should be separated from the rest of the line by a single space.)

For example, given the following sequence:

```
What kind of yogurt do you like best? [readln]
```

```
[ifentered]peach You are a real peach, y'know!  
[ifentered]lemon Well, you are what you eat!
```

if the user entered `peach' at the prompt, then Maximus would display `You are a real peach, y'know!' If the user entered `lemon', then Maximus would display `Well, you are what you eat!' If the user entered neither, then Maximus would display nothing.

```
[ifexist]<filename> - ^wi<filename>
```

If the specified filename exists, then the rest of the line will be displayed to the user. The filename must be separated from the rest of the line by a space.

```
[iflang X] - ^wBX
```

If user's language is set to X, display the rest of the line. Note that language number is 0-based. A zero means the first language listed in MAX.CTL, one means the second language, etc.

```
[iftask]<tasknum> - ^wb<tasknum>
```

If the specified task number (interpreted in decimal) is equal to the current task number, then the rest of the line will be displayed. The task number must be separated from the rest of the line by a space.

```
[iftime <op> <hh>:<mm>]
```

The [iftime] token allows decisions and statements to be executed in a .MEC file, based on the current time of day. If the condition specified is met, then the rest of the current line will be displayed. However, if the condition is NOT met, then the rest of the line will be skipped.

<op> is a token which specifies the type of comparison operation to perform on the specified time. The keywords are identical to those used in the priv level controls - see page 119 for more details. However, the acceptable operators are, in short:

EQ or EQUAL
NE or NOTEQUAL
LT or BELOW
GT or ABOVE
GE or AE
LE or BE

<hh> and <mm> tell Maximus which hour and minute to compare the current time to, in 24-hour time.

Examples:

```
[iftime GE 20:00]It's after 8PM!  
[iftime GE 20:00 iftime LE 21:00]Between 8 and 9.  
[iftime LT 20:00 iftime NE 12:00]Before 8 & not 12.
```

[incity]<s> - ^wR

This token will display the rest of the line, if the string <s> can be found in the city field, of the current caller's user record. <s> should be separated from the rest of the line by a single space.

Example:

```
[incity]Kingston Hi, [first]. You are a Kingstonian!
```

[islocal] - ^wIL

This command causes Maximus to display the current line to only LOCAL users.

[isremote] - ^wIR

This command causes Maximus to display the current line to only REMOTE users.

[jump] - ^oV

This token is provided for OACOMP compatibility only. It is identical to [goto], in all respects.

[label <l>]

This token is also provided for OACOMP compatibility, and allows an alternate way to define a label. The sequence `[label <l>]'` is identical to `[<l>]'`.

[maxed] - ^wm

The [maxed] token causes the rest of the line to be displayed ONLY if the user is currently in the MaxEd editor. This can be useful when designing a custom menu for the edit menu.

[msg_conf] - ^w^maC

This causes the rest of the current line to be displayed ONLY if the current area contains Conference messages. This command won't work unless the user is in a message area.

[msg_echo] - ^w^maE

This causes the rest of the current line to be displayed ONLY if the current area contains EchoMail messages. This command won't work unless the user is in a message area.

[msg_local] - ^w^maL

This causes the rest of the line to be displayed ONLY if the current area contains local messages. This command won't work unless the user is in a message area.

[msg_matrix] - ^w^maM

This causes the rest of the current line to be displayed ONLY if the current area contains matrix/NetMail messages. This command won't work unless the user is inside a message area.

[msg_next] - ^w^miN

This will cause Maximus to display the rest of line, only if the current message-reading direction is FORWARDS. This token won't work unless the user is inside a message area.

[msg_nomsgs] - ^w^mnM

This token will cause Maximus to display the current line, only if there are NO messages in the current message area. This token won't work unless the user is inside a message area.

[msg_nonew] - ^w^mnN

This token will cause Maximus to display the current line, only if there are NO new messages in the current message area. This token won't work unless the user is inside a message area.

[msg_noread] - ^w^mnR

This token will cause Maximus to display the current line, only if the user has not read ANY of the messages in the current message area. This token won't work unless the user is inside a message area.

[msg_notenter] - ^w^mnE

This token will cause the current line to be displayed only if the user has NOT just entered the message area. (In other words, if the MsgHeader was just displayed, and the user has not executed a command, then the current line will NOT.) This token will only work inside a message area.

[msg_prior] - ^w^miP

This will display the rest of the line to be displayed only if the current message-reading direction is BACKWARDS. This token won't work unless the user is inside a message area.

[no_keypress] - ^wG

This command tells Maximus to skip the current line, if there are any keystrokes waiting in the input buffer. See also `[nostacked]`. This differs from `[nostacked]`, in that `[no_keypress]` only checks for currently-pending input which has not been processed, while `[nostacked]` checks for commands entered at a previous prompt, which were buffered for further processing.

[nocolor] - ^o^^ (Control-O and a carat)

[nocolour] - Canadian spelling of above

This command tells Maximus to skip any following text, up to the next ``[endcolor]'` or ``[endcolour]'` token. See also ``[colour]'`.

`[nostacked]` - `^wS`

This tells Maximus to display the current line only if NO stacked commands are waiting to be executed.

`[notontoday]` - `^wQ`

This tells Maximus to only display the line if the user was NOT on the system previously in the day.

`[novice]` - `^wHN`

This tells Maximus to display the current line only if the user's help level is set to NOVICE. See also ``[expert]'`, ``[hotflash]'` and ``[regular]'`.

`[ofs]`

This token for compatibility with the program OACOMP. It does nothing.

`[permanent]` - `^wq`

This command tells Maximus to display the current line, only if the current user is marked as being permanent. The ``permanent'` flag in the user file is used to indicate that a particular user is ``special'`, and should not be deleted by any `delete-users-by-date-of-last-call`, or similar routines. Since the internal user editor doesn't have any `kill-by-date` functions, it also doesn't allow you to directly toggle the ``permanent'` flag. However, if you are using an external user editor which DOES use `kill-by-date` functions, chances are that it too supports the toggling of the permanent user flag, and you'll be able to create permanent users from that program. The permanent flag has no other effect, other than to tell the purging program not to delete this user automatically.

[regular] - ^wHR

This will cause Maximus to display the current line only if the user's help level is set to REGULAR. See also `[expert]', `[hotflash]', and `[novice]'.

[top] - ^oT

This tells Maximus to go to the top of the current file, and to start displaying from there.

Multi-Line Token Listing

This section gives several MECCA tokens which may be useful for systems with more than one phone line.

[apb] - ^wA

This will send a message to all users currently on-line, assuming that you have the IPC feature enabled. Examples:

```
[apb][yellow bell]%!User [lightgreen]%n [yellow]just
logged on the system%!
```

...or even enter a message to sent to all users:

```
> [readln]
[apb][yellow bell]%!User %n says "%J"!
```

[chat_avail] - ^wCA

This token will cause Maximus to display the current line, only if the user is AVAILABLE for paging by other users.

[chat_notavail] - ^wcN

This token will cause Maximus to display the current line, only if the user is NOT available for paging by other users.

[who_is_on] - ^ww

This token will cause Maximus to execute the Who_Is_On menu command, and display a list of all users who are currently logged on to other nodes of a multi-line Maximus system.

Miscellaneous Token Listing

The following is an alphabetical list of all of the tokens that the MECCA compiler uses, which code they translate into, and the effect of each token.

[ckoff] - ^b

This causes Maximus to turn OFF ^C and ^K checking, and not to allow the user to use those keys to abort the current file. See also `[ckon]'.

[ckon] - ^c

This causes Maximus to turn ON ^C and ^K checking, and to allow the user to use those keys to abort the current file. See also `[ckoff]' and `[onexit]'.

[clear_stacked] - ^w0

This command causes Maximus to clear the user's command stacking buffer, and to eliminate any previously-stacked commands. This command is identical to the `Clear_Stacked' menu option.

[comment <c>]

This is a comment which can be placed in a *.MEC file, and is NOT copied to the compiled *.BBS output. <c> can contain any characters, but the comment ends when a `]' is encountered.

[copy <f>]

This is a command which is processed by the MECCA compiler. When MECCA encounters this token, it will copy the file <f> directly into the output file, without performing any translations. Also, make sure that the <f> is INSIDE the square brackets, or else the command will not work. See also `[include]'.

[decimal] - %d

This command is not needed by Maximus, and is included for OECC compatibility only.

[delete]<f> - ^wD<f>

This will delete the file <f> from disk, if it exists. You can use external program translation characters in the filename, if you wish. For example, the command ``[delete]c:\max\%u.bbs'` would delete the custom welcome file for the current user.

[display]<f> - ^oS<f>

This will display the file <f>.BBS. Control is NOT returned to the current file after <f> has finished displaying. See also ``[link]'` and ``[quit]'`.

You can also use external program translation characters inside the filename specification; however, instead of using a percent-sign (``%'`), you must use a plus sign (``+'`) instead. For example, the command ``[dos]dir c:\max\+u.bbs'` would display a directory listing of the custom welcome file for the current user.

[dos]<c> - ^oC<c>

This tells Maximus to run the dos command <c>, which can include arguments. See also the `[xtern_???]` options.

[enter] - ^a

This displays ``Press ENTER to continue'`, and then waits for the user to press enter (or return) before displaying the rest of the file.

[file] - ^wNF

This command causes Maximus to act as if it had encountered a ``MenuHeader File'` statement, but without the associated display.

[hangup] - ^f^n

This immediately disconnects the current user.

[hex] - %x

This command is not needed by Maximus, and is included for OECC compatibility only.

[ibmchars] - ^wd

This token will cause the rest of the line to be displayed only to those who have the IBM Characters option enabled.

[include <f>]

This command causes MECCA to read in the file <f>, and process it as part of the current file, including interpreting any MECCA tokens contained within. See also '[copy]'.

[key?] - ^wT?

This command causes Maximus to toggle the setting of the user's key number '?'. '?' can have a value from 1 to 8. Please note that this TOGGLES the specified key. To force a specific key ON, use, '[?xclude key?]', and to force a specific key OFF, use '[?equal key?]'.

[key_poke] - ^wP

This command inserts a command into the keyboard command-stack buffer, just as if the user had typed it in manually. This is identical to the 'Key_Poke' menu option, and can be used to automatically guide a user through several commands at once.

[language] - ^oL

This token will invoke the Chg_Language menu option.

[link]<f> -^wL

This displays the file <f>.BBS. Control IS returned to the current file after <f> has finished displaying. However, you can have up to a total of 8 nested [link] commands. See also '[display]'.

[log]<s> - ^wA<s>

This adds the statement <s> to the system log. The FIRST CHARACTER of <s> should be the character you wish to insert in the far left-hand column of the log, while the rest of <s> should be the string to insert. For example, this statement:

```
[log]+User's name is "%n"
```

Would create a log entry of "User's name is '<name>'", with a priority type of "+".

[menu_cmd <s>]<arg> - ^rr

Invokes the menu command, defined by <s> from a displayed file. For example, [menu_cmd goodbye] would invoke the 'Goodbye' menu option. <arg> is the optional argument for the menu command. For example, "[menu_cmd area_change]message" could be used to select a new message area. See the menu section of this document for a list of valid menu options.

[menupath]<p> - ^wM<p>

This sets the current path for the *.MNU files to <p>.

[message] - ^wNM

This command causes Maximus to act as if it had encountered a 'MenuHeader Message' statement, but without the associated display.

[more] - ^d

This displays a 'More [Y,n,=]?' prompt.

[moreoff] - ^k

This turns OFF the automatic 'More [Y,n,=]?' prompting, which would normally occur at the end of each screen displayed. See also '[moreon]'.

[moreon] - ^e

This turns ON the automatic `More [Y,n,=]?' prompting, which would normally occur at the end of each screen displayed. See also `[moreoff]'.

[msg_checkmail] - ^wC

This invokes the internal mail checker. This is identical to the `Msg_CheckMail' menu option.

[newfiles] - ^wF

This invokes a new-files scan, identical to entering an "L*" at the file area menu.

This token is normally used in a display file called from the files menu. However, [newfiles] can be used in other display files (such as WELCOME.MEC) if you observe one restriction. To use [newfiles] from elsewhere, you must have a menu called "FILE" which contains the 'File_Tag' option. Normally, you'll already have this menu and this option on the file menu as part of your normal menu structure. However, if you are using a different menu configuration, you might not have a file menu. [newfiles] needs to know the required priv level to tag files, so it looks for the File_Tag option on the FILE menu to find out. If you are using a different menu structure, there doesn't need to be anything ELSE on the `FILE' menu, as long as a File_Tag is there somewhere. This is a concern to less than 1% of all Max systems, since almost everyone has a menu called FILE of some form or another.

[onexit]<f> - ^oF<f>

This sets the `On Exit' filename for the current file. When the file has finished displaying (through either end-of-file, a `[quit]' command, user pressing ^c, etc.), then the specified file will be displayed.

[other] - ^wNO

This command causes Maximus to act as if it had encountered a `MenuHeader None' statement, but without the associated display.

[pause] - ^f^g

This command causes Maximus to pause for half a second.

[quit] - ^oQ

This will quit the display of the current file immediately. This will only exit the current file, and will return to the previous file if a `[link]` command was used. See `[exit]` and `[link]` for more details.

[quote] - ^f^a

This will display the next quote from the file defined by the `Uses Quote` statement file in MAX.CTL. Please see the control file reference section for more information on the format of this file.

[repeat]<c>[<n>] - ^y<c><n>

This command tells Maximus to output a sequence of repeated bytes. Maximus will output the character <c>, <n> times. Note that this command normally is not needed, since MECCA will compress repeated byte sequences automatically. Please note that the number of times to repeat the character should be inside square brackets... i.e. `[repeat]=[15]` would repeat the character `=` 15 times.

[repeatseq <len>]<s>[<n>] - ^v^y<len><s><n>

The [repeatseq] token is similar to the [repeat] token, in that it will repeat the specified text a certain number of times. However, the [repeatseq] token allows an entire STRING to be repeated, up to 22 characters long. The string can even include AVATAR colour and cursor-movement codes. <len> specifies the length of the string, in bytes. <s> is the string itself. (Make sure to place the string OUTSIDE of the token's square brackets!) <n>, which MUST be enclosed in square brackets, specifies the number of times to repeat the string <s>.

Example:

```
[repeatseq 7]Hello! [10]
```

The above sequence would display the phrase "Hello!" followed by a space, for a total number of ten times.

[string] - %s

This command is not needed by Maximus, and is included for OECC compatibility only.

[subdir]<d> - ^oD

This will set the current directory to <d>.

[unsigned] - %u

This command is not needed by Maximus, and is included for OECC compatibility only.

[tune]<name> - ^wu<name>

This will cause Max to play the specified tune from the TUNES file specified in MAX.CTL. For example:

```
[tune]Yell1
```

[xtern_chain]<c> - ^wXC<c>

[xtern_dos]<c> - ^wXD<c>

[xtern_erlvl]<c> - ^wXE<c>

[xtern_run]<c> - ^wXR<c>

These commands run the external program <c>. If you wish to pass arguments to the external program, then simply add them after the name of the program to execute, separating each by a space. Please see the section on running external programs for more details, in the Maximus Operations Manual.

EVENT FILE CONFIGURATION

Maximus 2.0 includes an internal event file manager. This manager allows the SysOp to select Yell paging hours, and if using the WFC subsystem, external events can also be run at predefined times. The events section can only be used in WFC mode, since external events are ignored when starting Max with a caller on-line.

All events are defined in an ASCII file called EVENTSxx.BBS, where 'xx' is a task number (in hex). If you are running a one-line system, then this will be called EVENTS00.BBS.) In addition, MAX.EXE will automatically compile this into an file called EVENTSxx.DAT whenever a user calls in. By default, Max will always load the event file specified by the '-n' (task number) command line switch. However, one event file can be used for an entire multi-node system, as long as the event file contains only yell events. To override the 'xx' in 'EVENTSxx.BBS', use the '-e' to set the event "task number" (in hex).

For more information on the contents of EVENTSxx.BBS, please see the comments contained in the distribution version of EVENTS00.BBS.

APPENDICES

Appendix A: Common Problems

Although Maximus was designed to be both flexible and easy-to-use, there is a chance that you may have trouble either in installing Maximus and setting it up, or with a particular feature which was recently implemented. This section is an attempt to deal with some of the more predictable problems.

PROBLEM: Maximus is not adding an origin line or a tear line to messages originating from my system, but this only seems to be happening in certain areas.

SOLUTION: You probably specified the wrong area type in MSGAREA.CTL, by using `Local' or `Matrix' instead of `EchoMail'.

PROBLEM: Whenever I try to run an external program, Maximus tries to access one of my floppy drives.

SOLUTION: You forgot to edit the `Save Directories' statement in MAX.CTL. Maximus will try to save the current directory on all of the drives specified in that command. If you accidentally specify a floppy drive, then Maximus will try to find the current directory on that drive. This causes problems if there is no floppy disk in the drive.

PROBLEM: Whenever I try to look at a file with ANSI graphics locally, it is garbled and I can see all of the ANSI commands.

SOLUTION: Do not use ANSI graphics. Use the supplied ANSI2BBS utility to convert your ANSI screens into a Maximus-format *.BBS file. No great harm will come if you do not use the ANSI2BBS utility, but you will then have to live with this inconvenience.

PROBLEM: One of my users is complaining that their screen is being overwritten near the bottom when using HOTFLASH mode.

SOLUTION: Your caller has their screen length set incorrectly. When working in the full-screen HOTFLASH mode, Maximus needs to

have the exact dimensions of the caller's screen to work correctly.

PROBLEM: When one of my users tries to use the AVATAR graphics mode, they report that everything has changed colours, the full-screen editor doesn't work, etc.

SOLUTION: Your user is probably using Telix 3.1x, which has a bug in the AVATAR emulation code. Use a program which supports AVATAR correctly, such as TinyTerm or FDterm.

PROBLEM: Whenever a user presses ^C, he/she cannot see anything else, although the display looks fine locally.

SOLUTION: Try turning off the 'Send Break to Clear Buffer' command in MAX.CTL. If some modems, such as the HST, are improperly configured, then enabling this command may cause problems.

PROBLEM: I have specified a file for Maximus to display/use/view/etc. in one of the control files, or in a *.BBS file. However, Maximus acts as though the file does not exist.

SOLUTION: Make sure to specify full paths EVERYWHERE, including drive specifiers and leading backslashes. Maximus changes the current directory as it executes, so you cannot assume anything about the current path.

PROBLEM: Callers sometimes don't see the end of my BYEBYE.BBS file, and they report that Maximus hangs up before it is finished displaying.

SOLUTION: Place several '[pause]' MECCA tokens at the end of BYEBYE.BBS. Modems which have a transmit buffer (such as USR's HST and Hayes' V-Series) make no effort to empty the buffer before hanging up on remote callers. Once Maximus sends the BYEBYE.BBS file to the modem, it assumes that the file has also been received by the caller, and hangs up right away. Unfortunately, most modems with a buffered transmit buffer refuse to flush the buffer before hanging up, which makes the use of the above-mentioned [pause] tokens necessary.

Appendix B: Error Messages

Hopefully, you won't ever have to see any of these error messages in your system log. But just in case you do, we have prepared a list of all the possible error messages that Maximus produces, what they mean, and how to fix them (if possible).

ANSI sequence found, area XX msg YY

This warning is generated by Maximus' security system to let you know that it found an ANSI command embedded in the header of a particular message. Since some ANSI commands can be dangerous, Maximus blocks all of them out.

Barricade file priv, `XXX'?

This means that you misspelled or used an invalid privilege level in the barricade file for that area.

Can't find `XXX'

This means that Maximus was looking for a certain file, but was not able to find it. Read the message closely to determine which file Maximus wants, and correct the situation.

Can't find barricade file XXX

This means that a barricade file is missing, as specified in the MsgBarricade/FileBarricade in MSGAREA.CTL/FILEAREA.CTL.

Can't find class record

This means that Maximus was unable find the statistics for a particular user's privilege level. This usually means that either the person's user record or the *.PRM file is garbled. Try recompiling the *.PRM file, and if that does not correct the problem, use the user editor to reset the user's privilege level to what it is supposed to be. Also check to make sure that you have the appropriate `Define' statement for that user's privilege level in the control file, since Maximus needs this information to allow a user to log on.

Can't open `XXX'
Can't read `XXX'
Can't write `XXX'

These three messages indicate that Maximus was looking for a particular file, but was unable to open/read/write to it for some reason. Either replace the file if it was missing, or determine why Maximus was not able to write to the file. It may have been because the file being written to has a read-only attribute set, or because the disk was full.

Can't open log file `XXX'

This message means that Max was unable to open the standard log file. Chances are that you specified an incorrect path or filename.

If you are running a multiline system, you may see this error when running two copies of Max with the same log file. Each task must use a separate log, so you'll have to specify a different log name when starting each node. To override the log name given in MAX.CTL, see the `-l' command line switch.

Critical error reading/writing drive X:
Critical error accessing device COMx

Errors put out via the BIOS when Maximus's critical error handler detects a drive or com port error respectively.

Err: Lastread ptr xlinked, usr#nnn

A user's last-read pointer has become crosslinked. This usually indicates that an external utility has damaged your user file. To fix this problem, run "CVTUSR -1".

Exec error

This is an undefined error returned when a child process could not be started, for no obvious reason. Check to make sure that you have enough memory to load the program, and check to make sure that the executable is not damaged.

Invalid RESTARxx.BBS version number

This means that the RESTARxx.BBS file Maximus was reading was somehow corrupted by an external program. There's no fix, except for the user to hang up and call back.

Invalid UL path, area XX

This means that the upload path specified for area `XX' does not exist.

Invalid current pwd `XXX'

This means that the user tried to change their password in the C)hange Setup section, but failed to correctly enter their current password.

Invalid custom cmd: `X'

This means that you had an invalid character in a `Format XxxFormat' sequence in MAX.CTL. Fix the sequence, and recompile.

Invalid outside cmd: `X'

This means that you had an invalid character in a percent-sign translation sequence. Such a sequence is usually used for external programs or for the `[write]' MECCA token.

Invalid outside errorlevel

This means that you specified an invalid errorlevel for an errorlevel exit. Valid errorlevels are 5 through 254 inclusive.

MEM:ndir
MEM:nmsga
MEM:nmsgb

These messages are displayed when Maximus is critically short of memory, and does not have enough to save some information it needs. Although these errors are not fatal, it certainly means that more memory is needed.

Max nest lim. exceeded, XXX aborted

This message is displayed when you have tried to [link] a *.BBS file more than 8 levels deep, which is the maximum supported by Maximus.

No mem for delete buf
No mem for lastread scan
Not enough mem

These also mean that Maximus is short on memory. See `MSG:ndir`.

Null ptr/XXX

This means that there was a critical error in the Maximus code, and memory is really messed up. Please report this to the author, along with the circumstances under which the `Null Ptr` message was generated.

OA-MEMOVFL

See `MEM:ndir`.

Too many messages in XXX (max 2000)

This means that you have too many messages in the specified area; Maximus can handle up to 2,000 messages in one area. If you have more than this number, you should seriously consider deleting some of the messages, since message access will slow down greatly when there is anything more than 500 or 600 messages in one area.

Unknown option type `XXX`

This means that Maximus found an invalid option number in a menu file that it tried to access. The menu file is probably corrupt, and should be replaced.

Upload `ABC.BBS` renamed to `ABC.BBX`

This means that a user (whose privilege level was below the `Upload .BBS file` privilege level) tried to upload a file with an extension of `.BBS`. Maximus renamed the file to .BBX as a security feature.

User gave device/path `XXX`
User supplied path `XXX`

These messages are generated by Maximus' security system, when a user specifies an explicit path or device. For example, if the user typed `C:\Maximus\Virus.Com' at the `File to upload?', Maximus would generate a log entry of: `User supplied path `C:\Maximus\'. This message is only a reminder; Maximus will always strip off the path before allowing a file transaction.

If you are still having trouble and none of the above suggestions help, try posting a message describing your problem in the MUFFIN echomail area, or contact the Maximus Help Node, which is FidoNet address 1:1/119.

Appendix C: Operating with DoubleDOS

Since Maximus was originally designed on a machine running DoubleDOS, support for the environment is built in and a few tips and tricks have been discovered to allow Maximus and DoubleDOS to live happily with each other.

A lot of time is used up when displaying output on the local console. In order to minimize the slowdown when a user is on-line, you should take one of two steps:

- a) Keep snoop turned OFF by default, and only turn it on when you need it. That will ensure that no timeslices are wasted writing output to the local console, especially when nobody is watching it.
- b) Add the line ``DISPLAY=TEXT'` to your DDCONFIG.SYS file. This will allow Maximus to use the Video IBM video mode, and write directly to the screen buffer. The direct video output package is compatible with DoubleDOS, if the above statement is added to DDCONFIG.SYS, and it's also very fast. This will minimize the delay when another user is on-line, and will also allow you to keep snoop on permanently, since the delay is very slight while using Video IBM. If you can't live with the disadvantage of not being able to use graphics programs with the ``DISPLAY=TEXT'` option, then you might instead try adding the word ``REFRESH'` to DDCONFIG.SYS. This will tell DoubleDOS to look out for direct screen writes, and hopefully correct them before they become noticeable. However, this method is not foolproof, and using ``DISPLAY=TEXT'` is the recommended way of using Video IBM with DoubleDOS. If neither of these options work for you, then you'll have to use either ``Video FAST'`, ``Video DOS'`, or ``Video FOSSIL'` instead. Please see the control-file reference section for more details.
- c) Do not use any of the COM port assignments in DDCONFIG.SYS. This will only confuse Maximus and your FOSSIL.
- d) Also, for optimum performance, it is best to set the partition priority to EQUAL. You will still retain a decent speed in your partition, and the opposite partition should still be fast enough to keep up with a 14.4K HST.

Appendix D: Command Line Switches

The following is a list of all the command line switches which are supported by Maximus. In addition to these switches, you can also specify the name an alternate *.PRM to be used; simply including the name of the *.PRM file on the command line, either before or after any of the optional switches. If no switches are specified, Maximus will default to local mode. (At least the '-b' switch is required to handle a remote caller from an external mailer (such as BinkleyTerm or FrontDoor), and at least the '-w' switch is required to handle a caller using the internal WFC module.)

These are the switches which are currently supported:

-b<x> This parameter tells Maximus of the baud rate at which the user is calling. Normally, this number will be passed to Maximus by your mailer, or whatever program answers the telephone. (If you are using the internal WFC feature, this switch can be used to set the baud rate for communicating with the modem; Maximus will automatically detect the user's baud rate and gear down as necessary.)

Maximus also supports a locked baud rate selectable from the command-line. See the documentation on the '-s' switch for more information. When using the '-s' parameter for the locked baud rate, the speed given for -b will be used only for calculating file transfer times only.

-c This parameter instructs Maximus to create a USER.BBS file. When Maximus is executed with this parameter, a new USER.BBS file will be created, and the FIRST user who logs on will be given sysop privileges. Normally, you will only need to use this parameter once when you set up your system for the first time.

-e<x> This parameter specifies a decimal number which can be used to override Maximus's default event file. By default, Maximus will use EVENTSxx.BBS, where xx is the hexadecimal task number. This parameter allows several Maximus nodes that have the same events to utilize the same event file. See Appendix I.

-j<x> This parameter can be used to jam in a sequence of

keystrokes into the keyboard buffer, just as if a user had entered them manually. This option is useful for logging a user in automatically, such as you might want to do for yourself locally, or for some specialized mailer, such as a packet radio interface. If you wish to imbed spaces in the jam command, you must enclose the entire parameter in double quotes. For example, to insert the sequence "Joe SysOp;y;Password" into the keyboard buffer, the command line you would use is:

```
MAX -k "-jJoe SysOp;y;Pwd"
```

This would automatically log on Joe SysOp, using a password of "Pwd".

A slight modification of the above command, ``-j-'`, can be used to completely clear the keyboard buffer for a local log-on. By using ``-j-'`, the SysOp's name is NOT inserted in the keyboard buffer (which happens by default), and LOGO.BBS will be displayed, just as if there were a remote caller.

`-k` This parameter tells Maximus that you want to log on in local mode, and that there is no remote caller on-line.

`-l<x>` This selects the name of the log file to be used for this session. This parameter overrides anything specified in MAX.CTL. If ``<x>'` is blank, then no log file will be used.

`-m<x>` This commands selects the type of multitasker support that should be used for the current session. This parameter overrides anything specified in MAX.CTL. ``<x>'` can have any of the following values:

d	DoubleDOS
q	DESQview
p	TopView
t	TaskView
m	PC-MOS
w	MS-Windows
n	No multitasker

If no multitasker is specified, Max will attempt to autodetect the current multitasker at runtime.

`-n<x>` This selects the task number that Maximus should use

for the current session. This parameter overrides anything specified in MAX.CTL.

-p<x> This selects the port number to use for the current session. This parameter also overrides anything specified in MAX.CTL.

-q Please note that this command was NOT intended for normal use, and should be avoided if possible.

This instructs Maximus to try to assign a unique task number to the current copy of Maximus. If you use this parameter, then you must make sure that ALL copies of Maximus are using this parameter, and that you have not given a specific task number (through the '-n' command) to any task.

The only real use for this is on a LAN, when there may be multiple local users logging on at the same time, from different workstations, and each user may not know what their individual task number is. LAN users may also find the '-xj' switch convenient, which allows the disabling of the Alt-J (local shell-to-dos) command.

N.B. If you use this switch, then you must NEVER use either of the 'Xtern_Erlvl' or 'Xtern_Chain' options. When returning from an outside task via the '-r' parameter, you would need to tell Maximus which task to re-start via with the '-n' parameter, which is impossible to determine when using '-q'.

-r This tells Maximus to restart a previous session that was ended by either an 'Xtern_Erlvl' or 'Xtern_Chain' command. Maximus will read the information it needs to restart from RESTAR*.BBS (where '*' is the current task number), and then pick up from where it left off. If you are using a NON-ZERO task number, you MUST accompany the '-r' option with the '-nXX' (set task number) option. Maximus needs to know which RESTARxx.BBS file to restart from, and therefore needs to know the task number. For more information on this option, please see the Maximus Operations Manual, in the 'Running External Programs' section, on the subtopic of 'Restarting after Chain/Errorlevel'.

-s<x> The -s parameter tells Max to use <x> as a locked baud

rate. The rate specified here will be used when communicating with the FOSSIL, regardless of the rate set with the -b switch. If you choose to not lock your FOSSIL's baud rate (usually performed through a command line switch when installing your FOSSIL), or if you wish to use the "sometimes locked" feature in some of the newer HST modems, this switch can be used to select a locked baud rate at runtime.

-t<x> This tells Maximus the maximum amount of time to let the current user stay on-line, regardless of his/her time limit. This command is usually passed to Maximus by your mailer to make sure that a user does not overrun any events.

-u
-uq
-uh

These two commands automatically run the user editor locally, without waiting for the local operator to log on. It also causes Maximus to terminate as soon as the local operator exits from the user editor. The '-uh' command causes the editor to be invoked with HOTFLASH mode enabled. The '-uq' command-line parameter will invoke the user editor with hotkeys enabled, but still using the NOVICE help level.

-vo
-vd
-vf
-vb
-vi

These switches can be used to select Max's video mode from the command line. This setting overrides the video mode specified in MAX.CTL. The primary difference between these modes is speed and availability of the Maximus windowed interface and status line.

Switch	Mode	Speed	Window and Status Line
-----	----	-----	-----
-vo	Fossil	Slowest	No
-vd	Dos	Slow	No
-vf	Fast	Medium	No
-vb	Bios	Fast	Yes
-vi	IBM	Fastest	Yes

-w This switch instructs Max to run in 'waiting for call'

mode. This means that Maximus will answer the phone itself and accept callers without a front end.

- xj This switch disables the local shell-to-dos function. This may be useful as a LAN security feature, if it is undesirable to have users shelling to DOS while inside Maximus.

- xz This command can be used to disable Max's internal Zmodem protocol. When using this command-line switch, you can have a separate external protocol defined with the `Z' key, including Zmodem variants such as Zmodem-90 MobyTurbo.

Appendix E: Local Keystrokes

Maximus will respond to a variety of keystrokes while a user is on-line. Through these keystrokes, you can modify the user's privilege level and/or access keys, initiate chat mode, increase the user's time limit, and more. The keys that you can use with a user on-line are:

- <esc> Abort the current SysOp operation. This key will dismiss a pop-up window, abort a file transfer, and exit chat mode.

- <space> The <space> key displays the user's statistics, including his/her time on-line, UL/DL statistics, phone number, etc. To remove this window, either wait for the 20-second timer or press <esc>.

- A This key turns ON the local keyboard. If you don't press `A' to turn on the local keyboard, characters that you type will be interpreted as some of the special commands listed below, and not as input for the Maximus prompts that the user sees. To turn off the local keyboard mode, so that you can execute some of the commands below, just press <esc>.

- L The `L' key locks the user's privilege level to its current setting. You can then change the privilege level using the `+' or '-' keys, but the user's privilege level will be restored when the user logs off, or when you press `U'.

- N The `N' key turns ON `Snoop Mode', which means that you will be able to see what the user is doing. (Snoop can be either ON or OFF by default, as defined in MAX.CTL.)

- O The `O' key turns OFF `Snoop Mode'. Turning off snoop causes Maximus to use a status-line display on the local console instead of displaying the user's antics. This will also make other tasks run faster if you are running your BBS under a multitasking environment.

- S The `S' key causes a "priv window" to pop up on the

screen. This window allows you to change the user's privilege level and key settings. To toggle a key, simply enter that key's number/letter. To modify the user's priv level, use the up and down arrows to point to the appropriate access level. Once you have finished making changes, press <enter> to save these changes in the user's profile.

- U This command restores a user's privilege level to the level at which it was when his/her privilege level was locked.
- Z The `Z' key zeroes the user's cumulative on-line time. This key is useful if the user has been on for almost his entire time period previously in the day, but you want to allow him/her to be able to call back again and still have the full time allotment.
- 1..8 The number keys at the top of the keyboard toggle the user's specified key number, from one to eight.
- + The `+' key bumps the user's privilege level UP by one level.
- The `-' key bumps the user's privilege level DOWN by one level.
- ! The `!' key toggles the noise that the Yell command creates. Although you will normally want to keep this on, you can toggle the noise on or off at will, especially for those idiot users who insist on yelling five times in a row on Sunday morning. (See also the <Alt-N> key.)
- = Pressing the `=' key will display the current user's password. This is useful if you see a caller logging on and mistyping his/her password, or if you want to see what the caller's password is, without entering the user editor.
- ? Pressing a question mark does everything that pressing <space> does, except that it also turns SNOOP off. This is useful in cases where the user is displaying a long text file, and the statistics would have scrolled off the screen before you had a change to read them.
- <Up>
<PgUp> The <Up> key will ADD one minute to the user's

remaining time. The <PgUp> key will add five minutes, instead of just one.

<Down>

<PgDn> The <Down> key will SUBTRACT one minute from the user's remaining time; the <PgDn> key will subtract five minutes, instead of just one.

<Alt-C> The <Alt-C> key will initiate CHAT mode with the user who is currently on-line. Both of you can type, and if the user is using a non-TTY video mode, the text you and the user type will be in different colours. Press <Esc> to exit CHAT mode.

<Alt-J> Pressing <Alt-J> while a user is on-line will cause Maximus to shell-to-DOS., and allow the SysOp to perform DOS commands. A The 'Shell_Leaving' file will be displayed to the user after this keystroke is hit, but before shelling to DOS. Once in the shell, type 'exit' to return to Maximus. This keystroke will also work when running in local mode.

<Alt-N> Pressing <Alt-N> will toggle the "Nerd" setting for the current user. When the nerd flag is set, that user's yells will make no noise on the local console. The nerd flag remains turned on until it is reset by the SysOp.

<Alt-D> Pressing <Alt-D> performs the same function as <Ctrl-X>, except that it generates a bit of fake line noise beforehand, so the user will think that they were disconnected because of noise on the telephone line.

<Ctrl-X> Pressing <Ctrl-X> will cause Maximus to immediately hang up on the caller, without any warning.

<Fx> Pressing a function key while a user is on-line will cause the file F*.BBS to be displayed, where 'x' is the number of the function key you pressed. For example, pressing <F5> would cause the file F5.BBS to be displayed to the user. Similarly, pressing <Ctrl-Fx>, <Shift-Fx> and <Alt-Fx> will display CF*.BBS, SF*.BBS, and AF*.BBS, respectively.

Appendix F: User Editor Keystrokes

The internal user editor supports a few keystrokes which may not be obvious. These are:

- 0-9 Typing in a number on the keyboard will cause Maximus to jump directly to the specified user number.
- = Toggles the display of the user's password. The password is replaced with dots by default, but you can use the '=' key to hide/unhide the real password.
- > Toggles the 'Nerd' setting for this user. The nerd setting will cause Yell commands issued by this particular user to emit no noise. This allows you to prevent particularly bothersome users from disturbing you via the Yell command.
- ~ Pressing a tilde invokes the Find-User function. Maximus will prompt you for the username to find, and will search the user file for that name, or part thereof. You may also use this command to search for a particular phone number. If you type a string beginning with a number, Maximus assumes you are searching for a phone number.
- ` Find NEXT user. This will search for another match on the specified username, after using the `~' command.
- ^ Delete the current user. Users will be marked for deletion, but they won't be actually deleted until a manual purge is performed (see below). If you change your mind, you can use the '^' key again to undelete a user.
- | Purge users. This will delete all users that had the delete flag set in their record by the delete key or by a Maximus compatible external user editor.
- " Undo last change. If you decide to abandon the changes to the current user record, just select this key to restore the user record as it was prior to your changes. Undo will only work if you are still looking at the same user record while in the user editor.
- + Display the next user.

- Display the previous user.
- / Redraw the screen.
- ? Display help on the commands which are available.
- A Add a user. A new user record will be appended to the end of the user file. This record can then be edited.

Appendix G: List of AVATAR Colours

The following is a list of AVATAR colour codes which can be used when entering AVATAR commands into a *.BBS file, or as an argument to the `MenuColour' command. To use this chart, first look in the far left column to find the foreground colour you want. (If you want the high-intensity shade of the colour, then look at the `hi' row. Otherwise, look at the `low' row.) Then look to the top of the chart and decide which background colour you want to use. Finally, look down the chart from the background colour column, and find where it meets with the row you determined earlier. This is the number which represents the colour you selected.

		Background							
		Blck	Blue	Grn.	Cyan	Red	Mgnt	Yllw	Wht
F o r e g r o u n d	Black low	0	16	32	48	64	80	96	112
	hi	8	24	40	56	72	88	104	120
	Blue low	1	17	33	49	65	81	97	113
	hi	9	25	41	57	73	89	105	121
	Green low	2	18	34	50	66	82	98	114
	hi	10	26	42	58	74	90	106	122
	Cyan low	3	19	35	51	67	83	99	115
	hi	11	27	43	59	75	91	107	123
	Red low	4	20	36	52	68	84	100	116
	hi	12	28	44	60	76	92	108	124
	Mgnta low	5	21	37	53	69	85	101	117
	hi	13	29	45	61	77	93	109	125
	Yellow low	6	22	38	54	70	86	102	118
	hi	14	30	46	62	78	94	110	126
	White low	7	23	39	55	71	87	103	119
	hi	15	31	47	63	79	95	111	127

Appendix H: Sample Batch Files

These batch files are meant to illustrate how to use batch files to integrate your mailer to Maximus. They will require some revision, depending on how you have your mailer set up, which errorlevels you tell it to use, etc. In any case, they should take care of the bulk of the work in constructing your own batch files.

```
=====
=====          Sample WFC Batch File          =====
=====
```

```
Echo Off
rem * Insert your time zone here
set TZ=EST05

:Loop
cd\Max
max -w
if errorlevel 50 goto event
if errorlevel 12 goto scan
if errorlevel 11 goto pack
if errorlevel 5 goto after
if errorlevel 4 goto error
if errorlevel 3 goto error
if errorlevel 2 goto after
if errorlevel 1 goto done
goto after

:event
rem * Run external maintenance program here.
goto Loop

:scan
rem * This command should invoke your scanner. For example:

squish out squash -fEchoToss.Log
scanbld user.bbs area.dat local matrix @echotoss.log
goto loop
```

```

:pack
rem * This should invoke your mail packer.  For example:

squish squash
scanbld user.bbs area.dat local matrix
goto Loop

:after
rem * Insert after-caller utilities here.
goto Loop

:error
ECHO A fatal error occurred!

:done
ECHO Maximus down
exit

```

```

=====
=====          Sample FrontDoor Batch File          =====
=====

```

```

Echo Off
REM * Insert your time zone here
SET TZ=EST5

```

```

:loop
cd\FD
FD
if errorlevel 100 goto Local
if errorlevel 40  goto Maint
if errorlevel 34  goto UnpackMail
if errorlevel 33  goto B2400
if errorlevel 32  goto B1200
if errorlevel 31  goto B300
if errorlevel 10  goto Done
goto loop

```

```

:Local
rem * A local log-on to Maximus
cd \Max
Max -k
goto after_Max

```

```

:B2400
cd \Max
Max -b2400 -p1

```

```

goto After_Max

:B1200
cd \Max
Max -b1200 -p1
goto After_Max

:B300
cd \Max
Max -b300 -p1
goto After_Max

:After_Max
if errorlevel 12 goto scan
if errorlevel 11 goto pack
scanbld user.bbs area.dat local
goto loop

:unpackmail
rem * This should invoke your mail unpacker.

squish in out squash link -fEchoToss.Log
scanbld user.bbs area.dat @echotoss.log matrix
goto Loop

:scan
rem * This should invoke your mail scanner.

squish out squash -fechotoss.log
scanbld user.bbs area.dat local matrix @echotoss.log
goto loop

:pack
rem * This should invoke your mail packer.

squish squash
scanbld user.bbs area.dat local matrix

goto Loop

:maint
rem * Daily maintenance routine goes here
goto Loop
:done
ECHO FrontDoor ... down
exit

```

```

=====
=====          Sample BinkleyTerm BINKLEY.BAT          =====
=====
Echo Off
rem * Insert your time zone here!
Set TZ=EDT5

rem * Now Let's get Bink up and running
rem * Fossil and Video fossil are unloaded
rem * and then reloaded.

:Top
VFOS_DEL
BNU -U
BNU
VFOS_BIO
BT unattended
Rem * Checking ErrorLevels for types of calls, etc.
If ErrorLevel 255 goto Top
If ErrorLevel 96 goto BBS ; 9600 bps
If ErrorLevel 54 goto BBS ; 19200 bps
If ErrorLevel 30 goto Mail ; Incoming ARCmail/pkt/file
If ErrorLevel 24 goto BBS ; 2400 bps
If ErrorLevel 14 goto Maint ; Daily maintenance routine
If ErrorLevel 12 goto BBS ; 1200 bps
If ErrorLevel 3 goto BBS ; 300 bps
If ErrorLevel 2 goto Top
If ErrorLevel 1 goto End

:Mail
rem * Execute TOSS or IMPORT function here

squish in out squash link -fechotoss.log
scanbld user.bbs area.dat matrix @echotoss.log
goto Top

:Scan
rem * Execute SCAN and PACK functions here

squish out squash -fechotoss.log
scanbld user.bbs area.dat local matrix @echotoss.log
goto Top

:Pack
rem * Execute PACK functions here

squish squash
scanbld user.bbs area.dat local matrix
goto Top

```

```

:Maint
rem * Insert daily maintenance routine here
Goto Top

:BBS
Rem * A human caller is here and wants into the BBS. Bink
Rem * will create BBSBATCH.BAT which calls SPAWNBBS.BAT with
Rem * the proper parameters (baud, time 'till next event,
Rem * and port) which in turn calls Maximus... whew!
c:
cd \binkley
bbsbatch.bat
goto top

:End
Rem * I exited Bink and back to DOS.
c:
cd \binkley
Rem * This is the end...
echo Binkley ... Down

=====
=====          Sample BinkleyTerm SPAWNBBS.BAT          =====
=====

echo OFF
cd \Max
Max -b%2 -p%3 -t%4

:ELoop
If ErrorLevel 255 goto End
If ErrorLevel 65 goto Outside
If ErrorLevel 12 goto Export
If ErrorLevel 11 goto Mash
If ErrorLevel 10 goto End
If ErrorLevel 5  goto Acall
goto End

:Outside
rem * Replace the following line with `COMMAND /C
rem * ERRORLVL.BAT' if using rem * DOS 3.2 or previous.

call ERRORLVL.BAT
Maximus -r
goto ELoop

:Export
squish out squash -fechotoss.log

```



```
scanbld user.bbs area.dat local matrix @echotoss.log  
goto end
```

```
:Mash  
squish squash -fechotoss.log  
scanbld user.bbs area.dat local matrix  
goto end
```

```
:Acall  
scanbld user.bbs area.dat local  
goto end
```

```
:end  
Binkley
```

INDEX

*.BBS	11, 45, 70, 72, 126, 134, 135, 139, 152	Hotflash More	54
*.MEC	96-98, 126	Menu Highlight	54
*.MNU	39, 129	Menu Name	54
*.PRM	12, 136, 142	Menu Text	54
^aREALNAME	40	Message Address	54
\MAX\HLP\	98	Message Attrib	54
\MAX\MISC\	98	Message Date	54
300 baud	117	Message From	55
AF*.BBS	11, 149	Message FromTxt	55
Alias	58	Message Kludge	55
ANSI	99, 117, 134, 136	Message Locus	55
ANSI.SYS	14	Message Quote	55
ANSI2BBS	134	Message Subj	55
ARC	79	Message SubjTxt	55
ASCII	46, 77, 86-88, 96, 97	Message To	55
AVATAR	35, 71, 99, 117, 135, 152	Message ToTxt	55
Barricade	67	Msg Body	54
Barricades	67	Popup Border	55
Beer	38	Popup Highlight	55
BORED	43, 47, 77	Popup List	56
Canada	30	Popup Lselect	56
Canadian	30, 31, 38, 117, 122	Popup Text	56
CD-ROM	30, 66	Status Bar	56
CF*.BBS	11, 149	Status Chat	56
CGA	14	Status Key	56
Colour	99	WFC Activity	56
COLOURS.CTL		WFC ActivityBor	56
File Date	53	WFC Keys	56
File Desc	53	WFC KeysBor	57
File Find	53	WFC Line	57
File Name	53	WFC Modem	57
File New	53	WFC ModemBor	57
File Offline	53	WFC Name	57
File Size	53	WFC Status	57
FSR Address	52	WFC StatusBor	57
FSR Attribute	52	ConfMail	21
FSR Border	52	Cursor	102
FSR Date	52	Custom menu	121
FSR MsgInfo	52	Custom welcome	127
FSR MsgLinks	52	DDCONFIG.SYS	141
FSR MsgNum	52	Default Protocol	78
FSR Static	53	DESQview	14, 143
Hotflash Bar	53	DoubleDOS	14, 141, 143
Hotflash Clear	54	Download	78, 82
		DRLINK	28
		EchoMail	10, 20, 21, 24, 59, 60, 69, 84, 134

Editor	37, 38, 43, 45, 77, 135	MAX.CTL	6, 7, 62, 88, 131, 134, 138, 143, 144, 147
Equipment	16	Address	20
Equipment Section	16	After Call Exit	24
Errorlevel	20, 21, 24, 89, 138	After EchoMail Exit	20, 21
Extended ASCII	59, 77	After Edit Exit	20, 21
Extended Barricades	67	After Local Exit	21
F*.BBS	11, 149	Alias System	24
FB	66	Answer	16
FB.EXE	41	App	6, 7
FDterm	135	Application	6, 7
FIDUSER.LST	21	Area Change Keys	25
File Transfers	78	Area Data	26
FILE_BAD.BBS	42	Area Index	26
FILE_OK.BBS	42	Arrow Keys to Read	26
FILEAREA.BBS	33	Ask Phone	27
FILEAREA.CTL		Ask Real Name	27
Access	65	Baud Maximum	16
Area	65	Busy	16
Barricade	65	Charset Chinese	27
Download	65	Charset Swedish	27
End Area	65	Chat Capture On	27
FileAccess	65	Chat Program	27
FileBarricade	66	Comment Area	28, 83, 110
FileInfo	66	Compatible LocalBaud	27
FileList	66	Connect	16
FileMenuName	66	Define	28, 39, 40
FileOverride	66	Dos Close Standard Files	9
Upload	66	Edit Disable	29
FILES.BBS	40, 66, 82, 86	External	
FOSSIL	12, 18, 141	BatchExitProtocol	29
garbage	19	External BatchProtocol	29
Hayes	19	External ExitProtocol	29
High Bit	59	External Protocol	29, 146
HST	19, 135, 141	External Protocol	
Joe SysOp	143	Errorlevel	29
Kill	37	Fidouser	21
Label	97, 98	File Date	30
LAN	144, 146	File Password	9
LANGUAGE.CTL		filearea.bbs	104
Language	49	fileformat	104
LASTUS*.BBS	28, 74	FileList Margin	31
LinkMenuOpt	89	First File Area	31
Local	60, 134	First Menu	31
Locked baud rate	144	First Message Area	31
LOGO.BBS	143	Format Date	32
LZH	79	Format FileFooter	33
Matrix	20, 134	Format FileFormat	33
		Format FileHeader	33

Format MsgFooter	33	Ratio Threshold	40
Format MsgFormat	33	Reboot	12
Format MsgHeader	33	Ring	18
Format Time	36	Save Directories	40, 134
Gate Netmail	21	Send Break to Clear	
Highest FileArea	37	Buffer	19, 135
Highest MsgArea	37	Shell_Leaving	149
Include	7	Snoop	13
Init	17	StatusLine	40
Input Timeout	37	SysOp	13
Kill Private	37	Task	13
Local Editor	12, 37, 38,	Upload .BBS Priv	41
	84	Upload Check Dupe	41
Log Echomail	21	Upload Check Dupe	
Log File	9	Extension	41
Log Mode	9	Upload Check Virus	41
Logon Level	39	Upload Log	42
Logon Preregistered	39	Upload Reward	42
Logon Timelimit	39	Upload Space Free	43, 46
Mailchecker Kill	39	UserList Maximum	43, 79,
Mailchecker Reply	39		88
Mask Carrier	17	UserList Minimum	43, 79,
Mask Handshake	17		88
Mask Handshaking	17	Uses Application	39, 43
Menu Path	39	Uses BadLogon	43
Message Edit	22	Uses Barricade	43, 67
Message Send Unlisted	22,	Uses BeginChat	44
	23	Uses BOREDhelp	43
Message Show	22	Uses ByeBye	44
Min Logon Baud	39, 47	Uses Cant_Enter_Area	44
Min NonTTY Baud	39	Uses ContentsHelp	44
mmsgarea.bbs	104	Uses DayLimit	44
Msgformat	85, 104	Uses EndChat	44
Multitasker	10	Uses FileAreas	44
Name	10	Uses Filename_Format	45
NewUser2	46	Uses InquireHelp	45
No Critical Handler	18	Uses Leaving	45
No FilesBBS Download	40	Uses ListHelp	45
No RealName Kludge	40, 61	Uses LocateHelp	45
No SHARE.EXE	10, 11	Uses Logo	45
Nodelist Version	23	Uses MaxEdHelp	45
Output	18	Uses MsgAreas	45
Path IPC	10, 11	Uses NewUser1	46
Path Language	11	Uses NoMail	46
Path Misc	11	Uses NoSpace	43, 46
Path NetInfo	23	Uses NotFound	46
Path Outbound	12	Uses ProtocolDump	46
Path System	12	Uses Quote	46, 131
Path Temp	12	Uses ReplaceHelp	47

Uses Returning	47	[?below]	112
Uses Rookie	47	[?equal key?]	128
Uses ScanHelp	47	[?equal]	112
Uses Shell_Leaving	47	[?file]	112
Uses Shell_Returning	47	[?line]	112
Uses TimeWarn	47	[?xclude key?]	128
Uses TooSlow	47	[?xclude]	113
Uses Tunes	48	[addr]	104
Uses Welcome	46, 48	[alist_file]	104
Uses XferBaud	48	[alist_msg]	104
Version14	8	[ansopt]	109, 110
Version17	8	[ansreq]	109, 110
Video	14	[apb]	125
Video DOS	141	[b1200]	117
Video FAST	141	[b2400]	117
Video FOSSIL	141	[b9600]	117
Video IBM	40	[bell]	102
Yell	48	[black]	99
Yell Off	48	[blink]	98, 100, 101
MAX.LOG	9	[blue]	99
MaxEd	37, 38, 45, 77	[bright]	100
MAXP	10	[brown]	99
MECCA	46, 74, 84, 86, 96-98, 101, 126, 128, 131, 138	[bs]	102
leave_comment	28	[chat_avail]	125
MECCA token	114	[chat_notavail]	125
ABOVE	114	[choice]	98, 109, 110
AE	114	[city]	104
BE	114	[ckoff]	126
BELOW	114	[ckon]	126
Conditional Tokens	117	[clear_stacked]	111, 126
EQ	114	[cleol]	102
EQUAL	114	[cls]	102
GE	114	[col80]	117
GT	114	[color]	117
Informational Tokens	104	[colour]	117
LE	114	[comment]	126
LT	114	[copy]	126, 128
Miscellaneous Tokens	126	[cr]	102
Multi-Line Tokens	125	[cyan]	99
NE	114	[darkgray]	99
NOTEQUAL	114	[date]	104
Questionnaire Tokens	109	[decimal]	127
QUIT	114	[delete]	127
SEE	114	[dim]	100
SHOW	114	[display]	127, 128
SKIP	114	[dl]	104
UNEQUAL	114	[dos]	74, 127
		[down]	102
		[endcolor]	117, 123

[endcolour]	117, 123	[load]	101
[enter]	127	[locate]	103
[exit]	118, 131	[log]	129
[expert]	118, 123, 124	[magenta]	99
[expiry_date]	104	[maxed]	121
[expiry_time]	104	[menu_cmd]	129
[file_carea]	105	[menupath]	39, 129
[file_cname]	105	[menu]	98, 109-111
[filenew]	118	[message]	70, 129
[file]	70, 127	[minutes]	105
[first]	105	[moreoff]	129, 130
[fname]	105	[moreon]	129, 130
[goto]	98, 117, 118, 120	[more]	129
[gray]	99	[msg_carea]	105
[green]	99	[msg_checkmail]	46, 84, 130
[hangup]	128	[msg_cmsg]	106
[hex]	128	[msg_cname]	106
[hotflash]	118, 123, 124	[msg_conf]	121
[hotkeys]	118	[msg_echo]	121
[ibmchars]	128	[msg_hmsg]	106
[ifentered]	107, 118	[msg_local]	121
[ifexist]	119	[msg_matrix]	121
[ifkey]	116	[msg_next]	121
[iflang]	119	[msg_nomsgs]	122
[iftask]	119	[msg_nonew]	122
[iftime]	119	[msg_noread]	122
[incity]	120	[msg_notenter]	122
[include]	126, 128	[msg_nummsg]	106
[islocal]	120	[msg_prior]	122
[isremote]	120	[netbalance]	106
[jump]	120	[netcredit]	106
[key?]	128	[netdebit]	106
[key_poke]	128	[netdl]	106
[keyoff]	116	[newfiles]	86, 130
[keyon]	116	[no_keypress]	122
[label]	121	[nocolor]	122
[language]	128	[nocolour]	117, 122
[lastcall]	105	[nostacked]	122, 123
[leave_comment]	110	[notkey]	116
[left]	102	[notontoday]	123
[length]	105	[novice]	118, 123, 124
[lf]	102	[ofs]	123
[lightblue]	99	[onexit]	126, 130
[lightcyan]	99	[on]	101
[lightgreen]	99	[open]	109-111
[lightmagenta]	99	[other]	130
[lightred]	99	[pause]	131, 135
[link]	98, 118, 127, 128, 131, 139	[permanent]	123

[phone]	107	Chat_CB	76
[post]	109-111	Chat_Page	76
[priv_down]	113	Chat_Pvt	76
[priv_up]	113	Chat_Toggle	76
[quit]	98, 118, 127, 130, 131	Chg_Alias	76, 78
[quote]	46, 98, 131	Chg_Archiver	77
[ratio]	107	Chg_City	77
[readln]	107, 109-111, 118	Chg_Clear	77
[realname]	107	Chg_Editor	77
[red]	99	Chg_FSR	77
[regular]	118, 123, 124	Chg_Help	77
[remain]	107	Chg_Hotkeys	77
[repeatseq]	131	Chg_IBM	77
[repeat]	131	Chg_Language	78
[response]	107	Chg_Length	78
[right]	103	Chg_More	78
[save]	101	Chg_Nulls	78
[setpriv]	113	Chg_Password	78
[sopen]	111	Chg_Phone	78
[store]	109-111	Chg_Protocol	78
[string]	132	Chg_Realname	78
[subdir]	132	Chg_Tabs	79
[sys_name]	107	Chg_Userlist	79
[syscall]	107	Chg_Video	79
[sysop_name]	107	Chg_Width	79
[sysopbell]	103	Clear_Stacked	79
[tab]	103	Conf	73
[timeoff]	108	Contents	79
[times]	96	Ctl	73
[time]	108	Display_File	79
[top]	124	Display_Menu	80
[tune]	132	Download	80
[ul]	108	Echo	73
[unsigned]	132	Edit_Abort	80
[up]	103	Edit_Continue	80
[usercall]	108	Edit_Delete	80
[user]	44, 108	Edit_Edit	80
[white]	99	Edit_From	80
[who_is_on]	125	Edit_Handling	81
[write]	111, 138	Edit_Insert	81
[xtern_chain]	132	Edit_List	81
[xtern_dos]	74, 132	Edit_Quote	81
[xtern_ervl]	132	Edit_Save	81
[xtern_run]	74, 132	Edit_Subj	81
[yellow]	99	Edit_To	81
MENUS.CTL	7, 70	Enter_Message	82
Area_Change	76	File	82
		File_Hurl	82
		File_Kill	82

File_Tag	82	Read_Reply	87
File_Titles	82	ReRead	74
Forward	82	Same_Direction	87
Goodbye	82	SilentMenuHeader	72
HeaderFile	70	Statistics	87
Key_Poke	76, 83	Stay	74
Leave_Comment	83	Title	72
Local	73	Type	87
Locate	83	Upload	88
Matrix	73	User_Editor	88
Menu	70	Userlist	88
MenuColour	70	UsrLocal	74
MenuFile	70-72	UsrRemote	75
MenuHeader	70, 72	Version	88
MenuHeader Change	72	Who_Is_On	88
MenuHeader Chat	72	Xport	88
MenuHeader File	72, 82, 127	Xtern_Chain	88
MenuHeader Message	84, 129	Xtern_Dos	89
MenuHeader None	72, 86, 130	Xtern_Erlvl	89
MenuLength	71, 72	Xtern_Run	89
Message	84	Yell	89
Msg_Browse	84	Message Upload	78
Msg_Change	84	MS-Windows	143
Msg_Checkmail	84	MSGAREA.BBS	33
Msg_Current	84	MSGAREA.CTL	7, 58, 134
Msg_Edit_User	84	Access	58
Msg_Hurl	84	Alias	58
Msg_Kill	85	Anonymous	58
Msg_List	85	Area	58
Msg_Reply	85	Barricade	58
Msg_Scan	85	Conference	59
Msg_Upload	85	EchoMail	59
Msg_Upload_Qwk	85	End	59
NewFiles	86	High Bit Allowed	59
NoCLS	73	Local	60
NoDsp	74, 89	Matrix	60
Other	86	MsgAccess	60
Override_Path	86	MsgBarricade	60
Press_Enter	86	MsgInfo	60
Raw	86	MsgMenuName	60
Read_DiskFile	86	MsgName	61
Read_Individual	86	MsgOverride	61
Read_Next	87	No Realname Kludge	61
Read_Nonstop	87	Origin	62
Read_Original	87	Private and Public	62, 68
Read_Previous	87	Private Only	62, 68
		Public Only	62, 63
		Read-Only	63
		Renum Days	63

Renum Max	63	UploadCmd	94
Type	64	UploadKeyword	94
Use Realname	64	UploadString	94
MSGTMPxx	37	QM	21
multi-line	13, 76, 125	QuickBBS	70
Multi-node	76	Quote	22
multi-node chat	10, 11	RAMdisk	11
NetMail	20, 24, 60	RBBS	70
NoAccess	67, 68	READER.CTL	
OECC	97, 127, 128, 132	Archivers	50
OFFLINE READER	77, 78, 84, 85	Max Messages	50
compression method	77	Packet Name	50
QWK	77	Phone Number	51
OMMM	21	Work Directory	50
Opus	8, 12, 28, 70, 73, 97	READONLY.BBS	63
origin line	10, 20, 59, 62, 134	RESTAR*.BBS	144
PAK	79	SEEN-BY	22
PC-Board	70	Session	24
PC-MOS	14, 143	Session Section	24
Permanent	123	SF*.BBS	11, 149
Priv	112	SHARE.EXE	10, 11
Priv Level Controls	114	Shell-to-dos	144, 146, 149
Private	22	SILT	7, 8, 58, 59, 65
Privilege level	23, 113	snow	14
AsstsysOp	112	SysOp	13
Clerk	112	System	9
Disgrace	112	System Section	9
Extra	112	Task number	144
Favored	112	TaskView	143
Limited	112	Telix	135
Normal	112	Terse	9
Privil	112	TinyTerm	135
SysOp	23, 112	TopView	14, 143
Twit	112	Trace	9
Worthy	112	Translation characters	79, 83, 111, 127
Problems	134	TTY	99
PROTOCOL.CTL		Upload	78
ControlFile	91	User editor	150
DescriptWord	91	user file	9
DownloadCmd	92	USER.BBS	76-78, 142
DownloadKeyword	92	Userlist	79
DownloadString	92	V-Series	135
End Protocol	92	Verbose	9
FilenameWord	92	Video	102
LogFile	93	DOS	14
Protocol	93	FAST	14
Type	93	FOSSIL	14
		IBM	14

IBM/snow	14
Virus-checking	42
WORM	30, 66
Yogurt	119
ZIP	79