Maximus-CBCS Version 2.00 Operations Manual

Created November 3, 1991.

Documentation produced by Bob Davis, Scott Dudley,
Jesse David Hollington and Erik Van Riper,
with Don Dawson and Hubert Lai.

TABLE OF CONTENTS

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-CBCS 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-CBCS 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-CBCS 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-CBCS 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-CBCS 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-CBCS. Should your source code be included in Maximus-CBCS, Scott Dudley retains all rights for redistribution of the code as part of Maximus-CBCS 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-CBCS, 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-CBCS v<v>.<os>.R<r>
```

...where <v> is the official Maximus-CBCS 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-CBCS 1.02 must have a version string in this format: `Maximus-CBCS v1.02.OS/2.R2'

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

```
     Maximus-CBCS v<v>.<i>.<r>
```

where <v> is the official Maximus-CBCS 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-CBCS 2.00 modified by Joe T. SysOp must have a version string in this format: `Maximus-CBCS 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-CBCS 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-CBCS and its associated documentation except as expressly provided under this Licence Agreement. Any attempt otherwise to copy, sublicense, distribute or transfer Maximus-CBCS 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-CBCS into a program which is NOT completely free for ALL users. If you wish to use Maximus-CBCS in such a way, you must obtain written permission from Scott Dudley before using any of the Maximus-CBCS code.

7) You may not incorporate parts of Maximus-CBCS 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-CBCS code.

8) The privileges granted above apply only to noncommercial users of the Maximus-CBCS software. You are a NONCOMMERCIAL user only if you are running Maximus-CBCS 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-CBCS 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.

INTRODUCTION

Notes from the Author

Maximus-CBCS v2.00 is the first major upgrade since Maximus 1.0 was released. Although 1.02 added several new features and functions, it has taken over a year to complete Max 2.0. When writing version 2, major portions of the message and file sections were gutted and rewritten. Both the file and message sections now use compiled database for quick access; indeed, one of the biggest new features in Max 2.0 is the message system. Max now uses a generic message API (Application Program Interface) when dealing with messages. Under OS/2, this means that a new message format can be used by simply replacing the MSGAPI.DLL! Max 2.0 still supports the standard *.MSG format, but Max 2.0 also supports the proprietary "Squish" message format, selectable on an area-by-area basis.

As promised in the 1.02 documentation, Max 2.0 includes full multilingual support, a new message base format, and more. I have some unique ideas planned for the next release; in fact, one of the planned features for the next version has been in development for the past nine months (in parallel to Max 2.0). I can't release any more information at this point, but I did want to say that there are some neat things ahead.

If you are having trouble installing Maximus, in spite of the automated installation program, please have a look in the troubleshooting section of this manual. If you still can't get Maximus to work correctly, then either post a message in the MUFFIN echomail conference, or send NetMail to the Maximus Help node at 1:1/119.

Scott Dudley
October 14th, 1991

For contact information, please see the program licence.

Maximus, Money and You

In general, Maximus is a freeware program. If you are a noncommercial user and you abide by the terms given in the licence, there is NO CHARGE for running Maximus. NONCOMMERCIAL USERS ARE STILL WELCOME TO RUN MAXIMUS WITHOUT PAYING A CENT.

Unlike other software, this program does not have crippled features or "registration incentives". Maximus does not have any "registered only" commands. There is one simple difference between the commercial and noncommercial versions of Maximus: the commercial version entitles you to legally use Maximus in a commercial environment.

If you are a COMMERCIAL USER as defined in point 8) of the licence above, you must obtain a licence before using Maximus. Please see ORDER.FRM for more information. If you did not receive a copy of the order form with this document, contact the author at one of the addresses listed in the licence for more information.

If you are a NONCOMMERCIAL user as defined in the licence above, you are welcome to use Maximus for free. However, Maximus is an extremely large and complex program, and simply maintaining the existing code is consuming a large percentage of my time. Donations of any value (or even just a postcard) will be gladly accepted. However, noncommercial donations are on a VOLUNTARY basis and are NOT REQUIRED.

However, BOTH commercial and noncommercial users must abide by several restrictions before using Maximus. The main points in the licence are:

- You may not distribute Maximus on a CD-ROM, WORM, or as any other form of a "value added" good in a sales transaction. You may not bundle Maximus with other software without prior written permission of the author.

- You may modify Maximus, but only under the conditions given in the licence.

- You may not incorporate parts of Maximus into another BBS package.

- You may not incorporate parts of Maximus into another program which is not completely free for all users.

Other than the above, there are few restrictions on the use of Maximus. However, because additional restrictions or qualifications may apply to you, the licence agreement should still be read carefully.

Credits

Even though Maximus is almost all original code, several other people contributed a great deal to the project and deserve to be acknowledged.

First and foremost, my thanks go to Wynn Wagner III. Without Wynn, there would be no Opus; without Opus, there would be no Maximus, or at least not in the format that it is today. Maximus does not use any of Wynn's code, but he did provide the program after which Maximus was originally modeled. Wynn also wrote a large number of utilities and established the WaZOO handshaking standard.

Secondly, my thanks go to Peter Fitzsimmons, for all of his help and code. Peter is the head of the Max-OS/2 development team, but he has also contributed a good deal of code to the DOS version. Comments can be sent to him at 1:250/628.

As well, I should also thank Bob Hartman, Vince Perriello and Alan Applegate. These three people created the BinkleyTerm front end, and they also contributed all of Max's file transfer protocols.

I would also be remiss if I failed to thank all of my alpha and beta testers. Although the list is too long to include here, my sincere gratitude goes out to all of them, not only for their help in tracking down the many bugs which were visible in earlier Maximus releases, but also for their helpful suggestions and comments. My thanks also go to the rest of the documentation team, as listed on the front page of each manual.

Thanks go to the following people who have contributed to the Maximus source, by donating algorithms, structures, ideas, or even code. (My sincere apologies if someone has been left out.)

Peter Fitzsimmons
Vince Perriello and Bob Hartman
Andrew Farmer
Scott Friedman
Ray Duncan
Thomas Plum
Alan Hughes
Anders Brink
Jim Lin
Gordon Benedict

Finally, I wish to acknowledge several products and trademarks which have been mentioned in this document. Use of these names and trademarks neither constitutes an endorsement nor suggests any affiliation with the specified products.

ARC, SEAlink & SEAdog:	Thom Henderson
BinkleyTerm:	Bob Hartman & Vince Perriello
BNUcomm:	David Nugent, Unique Computing
Courier HST:	U.S. Robotics Inc.
DESQview:	Quarterdeck Office Systems
DoubleDOS:	SoftLogic Solutions Inc.
FEND:	David Luong
Fido & FidoNet:	Tom Jennings, Fido Software
FrontDoor & FDterm:	Joaquim Homrighausen
FView:	Doug Boone
IBM-PC, PC-DOS, OS/2:	International Business Machines
Lharc and LZH:	Haruyasu Yoshizaki
Minimus:	John Cuccia
MS-DOS:	Microsoft Inc.
Opus-CBCS:	Wynn Wagner III
OpusComm & ConfMail:	Bob Hartman, Spark Software
PCBoard:	Clark Software Development Corp.
QuickBBS:	Pegasus Software, Inc.
QM, QMail & AreaFix:	Greg Dawson & George Peace
RBBS:	Capital PC Users Group
TBBS:	eSoft, Inc.
Telix:	Colin Sampaleanu, Exis Inc.
TheDraw:	Ian Davis, TheSoft Services
TinyTerm, OECC:	George A. Stanislav
Turbo C:	Borland International
V-Series and Hayes:	Hayes Microcomputer Products
VT-100:	Digital Equipment Corporation
X00:	Ray Gwinn, RENEX Co.
Xmodem:	Ward Christiensen
ZIP:	Phil Katz, PKWare
Zmodem and MobyTurbo:	Chuck Forsberg

It's Canadian, eh?

Maximus is a Canadian product. As such, proper English spellings are used in most places. Carbon-based life forms living in the United States of America should make the following mental translations:

Proper spelling	American spelling
Colour	Color
Favoured	Favored
Licence	License
Neighbour	Neighbor

For your viewing pleasure, an alternate language file called AMERICAN.MAD is included in the Maximus distribution package. This language file is an "Americanized" version of the standard English file, including many of the above changes.

SYSTEM REQUIREMENTS

Although it is possible to run Maximus on a system with less than the following equipment, the following should be considered the realistic minimum with which you can get by:

*   An IBM (or compatible) personal computer running MS-DOS or PC-DOS, with at least 256K of available RAM, with at least 164K free.

*   MS-DOS or PC-DOS version 2.0 or greater, 3.2 or above preferred.

*   A Hayes-compatible modem. It is possible to use Maximus with a modem which is not Hayes-compatible. However, doing so will make your life unnecessarily complicated.

*   A hard disk. Capacity of 30 megabytes or greater is preferable.

*   A FOSSIL communications driver, revision level 5 or higher. (Common FOSSILs are X00, BNU and OpusComm.)

MAXIMUS OVERVIEW

This section of the documentation is designed to give you, the SysOp, a general overview of what Maximus can do. By any means, this is not complete, but it should give you a general idea of how the system operates.

Logging On

When Maximus starts up with a caller on-line, the Maximus name and version will be displayed, followed by the SysOp-created log-on screen (usually \MAX\MISC\LOGO.BBS). This screen should be kept under 2K in length, and it should not use ANSI graphics and high-bit IBM characters.

After the logo screen has been displayed, Maximus will then prompt the user to enter a name. Unlike other BBS programs, Maximus allows a user to use names with more than two words, so even a name such as "Jesse David Hollington" will be accepted. (On the other hand, Maximus will reject callers with a one-word name, unless you have the `Alias System' keyword uncommented in MAX.CTL.)

However, if the user does NOT exist in the user file, Maximus will display the NOTFOUND.BBS file. This serves to confirm that the user really wanted to enter the name which was entered at the prompt. In addition, Maximus will disable key stacking, just in case the user entered a name in the format of `Joe SysOp Y Password', and misspelled the name `Joe SysOp'. With other BBS software, this would result in a new user account being created under the wrong name.

If the name was not in the user file, Maximus will go through the new user routine. First, Maximus will display APPLIC.BBS; this normally gives the caller some information about your system, and it may also present an application form to be filled out. (If a caller hangs up before APPLIC.BBS has completed displaying, then the user's configuration will NOT be saved, and no entry will be made in USER.BBS.)

When APPLIC.BBS ends, Maximus will then prompt the user to enter his/her city, alias (if applicable), phone number, and so on. Finally, Maximus will display NEWUSER2.BBS, which can be yet another screen directed toward educating new users.

On the other hand, if the user's name was found in the user file, then Maximus will ask that user to enter his/her password. The user will get five tries to enter the correct password; if all five tries fail, or if the user pressed <enter> five times, Maximus will hang up and recycle. However, Max will display the file \MAX\MISC\BAD_PWD.BBS before doing so, which means that you can prompt the user to enter a message of explanation before hanging up.

If the user enters the correct password, Maximus will proceed with the log-on sequence, and display WELCOME.BBS to the user. Note! If a user's account has a BLANK password, then Maximus will treat that user as a `guest account'.  This means that Maximus will ask a user who is using this account for his/her configuration settings at every log-on, and Maximus will also skip the password prompt.  This allows the SysOp to create an account specifically for new users (while using a `Logon Level Preregistered' statement), such that users can look around the system, but the user file won't get cluttered with users who choose not to register. (User registration would presumably be done through an on-line questionnaire.)

Maximus also supports a concept known as a `custom welcome screen'.  A special welcome screen can be created to display to each individual user, BEFORE displaying the main WELCOME.BBS.  By placing a file called `#.BBS' in the Maximus system directory, where `#' is a user number, Maximus will display that .BBS file to the caller with that user number. (You can find out a user's user number by looking in the log file, or also by doing a find in the user editor.) For example, if you wanted to show a custom welcome to user #5, then you might create a file called `\MAX\5.BBS'.

As an added security feature, you can also disable remote sysop logons. If you place a "[isremote sequal hangup]" at the top of your WELCOME.MEC, Max will automatically hang up on remote callers with a priv level of SysOp. If you never call your system from remote, using this option can help prevent accidents.

If the user has ANSI or AVATAR graphics support, the user can use the cursor keys to edit any of the text on the command line. Editing can be performed through the <left>, <right>, <bs>, <del>, <ctrl-left>, and <ctrl-right> keys. To use this feature, the caller's terminal must support either "Doorway mode" or VT-100 keyboard codes.

The Main Menu

Although Max's menus are completely redefinable, this section attempts to explain the commands which would normally be found on the main menu. This is also where the commands appear in the default configuration.

Message Section

This enters the message section. From here all the message entering and reading features are available. See the section on the message areas for more detail.

File Section

This takes the user to the Files Section. From here a user can exchange files by uploading or downloading, or simply see what files are available. See the section on the file areas for more detail.

Change Setup

This takes the user to the Change Setup menu. From here, a user can modify his/her user profile. The user can set the screen length, change the graphics mode, password, toggle the full-screen editor, and more. See the section on the Change Setup menu for more details.

Goodbye

This option logs the user off the system and hangs up the phone. This is almost identical to what happens if a user simply drops carrier. Maximus will not `hang' if a user drops carrier, but will recycle as if they logged off using this command. This command simply asks the user to confirm that they want to disconnect, asks if he/she wants to leave a message to the sysop, and then hangs up on the user. By default, comments are placed in message area 1, although this is selectable in MAX.CTL.

Comments to the sysop are saved in the "Comment Area", as defined in MAX.CTL. (By default, comments are stored in area 1.) If message area 1 does not exist, or if you use a "Comment Area" statement with no area listed, users will not be asked if they want to leave you comments.

Statistics

This option displays the user's statistics, including the time the user has been online for the current call, the time online for the day, amount uploaded, amount downloaded, NetMail credit, and so on.

Yell

This command allows the user to attempt to contact the SysOp while he/she is on-line. You can set when you want users to be able to page you with this command, and you can toggle the local speaker noise with the local "!" command.

Userlist

This command simply displays a list of all the users who are currently in the user file. You can set the maximum and minimum privilege levels to display in the list through options in the control file. By default, will not display users with a privilege of SysOp, Hidden or Twit. These privilege levels are also selectable in the control file.

Version

This displays the version number and a few other statistics about the current revision of Maximus and the version of the operating system currently in use.

SysOp Menu

This command takes you to the Maximus SysOp menu. See below for more details.

Chat Menu

If enabled, this menu can be used to access all of Max's multi-node chat features, including paging, toggling chat availability, and the chat itself. This feature allows callers to talk with one or more other callers. Chatting with the SysOp is performed through the Yell command.

Who is On?

> If the multi-node chat system is enabled, this command can be used to display a list of callers who are currently logged onto other nodes of the same system. This command will display the user name, the node number, and that user's status. The status message indicates the current activities of the user, such as "Downloading a file", "Entering a message", and so on.

In addition, you can define other options in the main menu to call sub-menus, display text files, or run external programs. See the control-file reference section for more details on defining menus.

The Message Section

There are four basic types of message areas within Maximus. These area types are Local, NetMail, EchoMail and conference mail.

Local messages are messages entered by a user on your BBS. They can be either public or private, and they remain on your BBS. Other users can only read these messages by logging onto your BBS and selecting that particular message area.°

NetMail messages are messages that are sent from one BBS to another through a network that you are connected to. They are generally private messages.

Unless you are a host or hub, most NetMail messages you encounter will either be entered on your system, or will be sent to your system from another. Maximus is fully compatible with the FidoNet mail standard for these messages. A user entering a NetMail message will be prompted to enter additional information to tell your mail processor where to send the message. Maximus is capable of reading a FidoNet compatible Version 5 or Version 6 nodelist file in order to get its address and cost information. Generally your users will need to have credit in their user accounts in order to send NetMail. Under most circumstances you should only have one of these areas.

EchoMail messages are messages that are shared between several bulletin board systems over a wide area. An EchoMail message will be sent through your network to all other systems participating in that same conference. Generally, EchoMail areas only permit public message. To a Maximus user, an echo area appears identical to a local message area, except that messages entered in EchoMail areas will have special control information added to the end, such as origin and tear lines. In addition, when a user enters a message in an EchoMail area, Maximus will add that area's tag to your echo tosslog.

Maximus also supports a variety of message area toggles (or "attributes"), each of which can be set independently. Although a complete list can be found in the MSGAREA.CTL reference in the Maximus Technical Reference Manual, some of the more common attributes are:

Private Only

All messages entered in these areas will be marked private, and can only be read by the user who sent the message, the user it is addressed to, and the sysop.

Public Only

All messages entered in these areas will be public and can be read by any user. This flag is recommended for EchoMail areas.

Read-Only

Messages in these areas can be read by users, but only users with a privilege level of assistant sysop or higher can enter messages.

Anonymous OK

In an area has this attribute set, users can optionally enter messages under a pseudonym instead of that user's real name. Maximus will embed the user's real name within the message in such a way that only the SysOp can see it. If this causes some sort of security or anonymity problem, this embedding of the user's real name can be disabled. Please refer to the MSGAREA.CTL reference (in the Maximus Technical Reference Manual) for more details.

Maximus is also capable of assigning passwords to message and file areas, and re-assigning privileges within the area for certain passwords. In addition, you can assign user names to the barricade file, so only certain users can enter an area. See the section on extended barricades in MAX_REF.PRN for more details.

In addition, the private and public message-area attributes can be defined individually by privilege level, rather than globally for all callers. (In other words, you can allow the SysOp to enter anonymous messages, while still forcing normal users to use their real names.) See the control-file reference for more information on how to assign these attributes to areas.

The Message Menu

The following attempts to explain the options that would normally be used in a message area menu:

Area Change

This command allows the user to change to another message area. The user will be prompted to enter the message area they want to go to, or to enter a "?" for a list of the areas that are available to them. The user can also enter the "<" or ">" characters to go to the previous or next area in the list, respectively. If entering a barricaded area where a password is required, he/she will be prompted for the password before they are allowed to enter the area.

Next

This command will display the next message in the current area. To keep reading messages in this direction, the user can press the ENTER key at the next prompt. The ENTER key will repeat the last N)ext or P)revious command.

Previous

This command will display the previous message in the current area. To keep reading messages in this direction, the user can press the ENTER key at the next prompt. The ENTER key will repeat the last N)ext or P)revious command.

Enter a Message

This command will allow the user to enter a message. After the user selects this command, Maximus will prompt them for some information is needs to know to send the message, such as who the message is to, the subject of the message, and whether the message is public or private, and any other information allowed by the configuration of the current area.

If the area does not allow public messages, or does not allow private messages, the user will not be able to select whether they want the message to be public or private.

If the area allows anonymous messages, the user will be able to change who the message is from as well. If the message area is a NetMail area, the user will be prompted for a network address to send the message to. When entering the address, the user can either enter the address, or use the following keys to get listings:

#	This will display a list of all the nodes in the current net.

/ This will display a list of all the nets in the nodelist.

If you only enter a node number, Maximus will automatically default to your current net.

A special shortcut exists for addressing points off the current system. Suppose that you are currently entering a message on 1:106/114. If you wish to write a message to John Doe at 1:106/114.22, simply enter `.22' at the destination prompt. Maximus will automatically expand this address to the full `1:106/114.22'.

After entering this information, the user will be placed in the message editor to enter the message. For more details, refer to the section on message editors.

Finally, if your nodelist processor can create a FIDOUSER.LST, Maximus is capable of finding a network address based on a user's name. If you enable the "FidoUser D:\Path\Fidouser.Lst" statement in MAX.CTL, Max will search the nodelist for a particular SysOp name. If that name was found, Max will automatically enter that SysOp's netmail address for you.

Change a Message

The Change command allows users to modify an existing message. Before allowing a user to change a message, Max will check to make sure that the user actually WROTE that message. Max will also check to see if the message has been scanned out or read by the recipient. If either of these conditions is true, Max will display a message to that effect and abort the change. (However, this acts only as a warning to callers with a priv level of SysOp; Max will always let the SysOp modify a message, whether or not the message has been read or scanned out.)

Both the message header and the message body can be modified. When graphics are turned on, the message status bits in the NetMail area (such as crash, hold, and so forth) can also be modified.

Reply to Message

This command allows the user to send a response to the author of the current message. The reply command is similar to the enter command, except that some of the message fields will be filled in (the name of the author of the message to which you are replying will automatically be inserted in the To: field). Also, once in the editor, the user will be able to QUOTE the message they are replying to. See the section on the message editor for details. "RK" [R)eply/K)ill] will kill the original message after the reply is finished.

Read Non-Stop

This command will allow a user to read all of the messages in the current area, starting with the current message, without pausing between each message. This is useful if users want to capture the messages to a disk file for later perusal.

Read Original

This command will allow the user to display the original message to which the message they are reading is a reply. Messages that are replies to another will have a "*** This is a reply to #xx" tag at the bottom of the message.

Read Reply

This command will allow the user to display any messages that are replies to the message they are reading. Messages that have replies to them will have a "***See also #xx" tag at the bottom of them.

Read Current

This command will allow the user to redisplay the current message number. This command is useful when first entering an area (to re-read the message you read in your last session), and also when trying to redisplay a message which scrolled off the screen.

Browse

Browse is an extremely powerful command; it replaces the functionality of Max 1.02's Scan, Inquire, List, and mailchecker commands. Browse acts as a powerful database engine for the message bases. Users can select a set of messages and operations to perform, and have Maximus automatically identify and display the messages which were requested. Browse is broken down into 3-4 separate menus.

The first menu allows the user to select a set of areas to query. The user can select either the current area, the set of areas selected with the T)ag command, or all message area.

The second menu allows the user to specify a set of messages to select. The user can specify ALL MESSAGES (each message in every area specified), NEW MESSAGES (messages above the lastread pointer in each area), YOUR MESSAGES (messages above the lastread pointer and addressed to the current user), and FROM A MESSAGE NUMBER (such as from message #200 and up). Maximus also allows the user to perform a search based on keywords in the to, from and subject fields, in addition to the message body. Complex searches can be defined using the logical OR and the logical AND operators.

The third menu selects an operation to perform on the specified messages. Selected messages can be listed (one to a line, to/from/subject only), read (displayed in full, with the option to reply or kill), or packed into a QWK bundle for off-line reading.

Tag

The tag command allows each user to select a subset of message areas available on the system. Each individual user can select his or her own set of message areas to be stored across calls. This set of areas can be used with the Browse command, or also with the Download command on the off-line reader menu.

Edit User

This function invokes the user editor from the message menu; however, it also reads in the current message, checks the 'From:' field, and automatically positions the user editor on that user's user record. This may be useful for validating users, since you can pull up a user's record at the press of a key. (This command is for SysOps only.)

Goodbye

This is identical to the G)oodbye command at the main menu. It will log off the user.

Main Menu

This will return the user to the main menu.

Kill a Message

This command allows a user to delete a message in the current area. Unless a user has SysOp privileges, a user will only be able to kill messages which are addressed TO or FROM that individual.

Upload a Message

This command allows a user to directly upload a text file as a message to the current area. This is identical to the E)nter message command, except instead of invoking the editor, Maximus will start a file upload. The user may then upload a pre-typed ASCII text file which will be stored as a message. Any file transfer protocol may be used for uploading the message text. The maximum length of an uploaded message is 8K.

Forward

This command allows a user to make a copy of a message in the current area, and send it to someone else. The user enters the message number to forward, and the name of the person to forward it to. The user can also forward the message directly into another area by typing the area number when prompted. The F)orward command supports two special modifiers, as follows:

"FK" [F)oward K)ill] will delete the original message after forwarding it.

"FB" [F)orward B)ombrun] will allow a user to specify a filename containing a list of users to forward the message to. In order for a user to use this command,their privilege level must be equal to the privilege level required to create a message from a file as defined in the Maximus control file. (See the control file reference in MAX_REF.PRN for more information.) The format of the bombing run file is as follows:

```
<username> <dest_net/dest_node> [-x]
```

-x can be one of the following switches:

```
     -h   Message should be marked as HOLD FOR PICKUP
     -c   Message should be marked as CRASH
     -n   Message should be marked as NORMAL (default)
```

The bombing file can contain any number of lines.

The <dest_net/dest_node> and [-x] fields are only used for NetMail messages, and should be omitted for local bombing runs.

In the username field, spaces in a user's name must be represented by underscores.

For example:

```
SysOp                         225/337
Scott_Dudley                  249/106      -c
Hubert_Lai                    249/102      -h
Vince_Perriello               141/191      -n
Jesse_David_Hollington        225/1        -c
```

This would send carbon copies of a message to the five people names, sending the messages to SysOp and Vince Perriello as NORMAL messages, and the messages to 249/106 and 225/1 as CRASH messages.

Note! If you are performing a NetMail bombing run, make sure NOT to route your messages through anyone else. Routing bombing-run messages is against FidoNet policy.

Hurl

This command is used to move messages from one area to another. It will ask the user which message to hurl and which area to hurl it to.

In addition, you can define other options in the message menu to call auxiliary menus, display text files, or run external programs. See the control-file reference for more details on defining menus.

Xport

The Xport command, normally available only to SysOps, can be used to export a particular message to a specified filename.

This message will be formatted for 80 columns, and the file will also include the message header. TO PRINT A MESSAGE ON THE PRINTER, specify an Xport filename of "prn".

Message Entry

For ANSI and AVATAR callers, Maximus supports a sophisticated message entry screen. After selecting one of the options which begins the message entry process (such as E)nter, R)eply, or U)pload), Maximus will then display a "template" for the user to fill in. The template indicates whether or not the message is private or public, the name of the recipient, the recipient's matrix address (if in a netmail area), the subject of the message, and optionally, an "alias" or alternate name for the sender.

The user can move back and forth through the various items if they have an ANSI/VT-100 or IBM-compatible terminal emulator, and if not, users can also use the WordStar-like Control-E and Control-X keys to move up and down the fields (respectively). Control-Y will delete the contents of the current field, and pressing <escape> TWICE will abort the message. Assuming that all of the fields in the template have been filled, Maximus will invoke the message editor when the user presses <enter> on the last field in the header.

Maximus also supports a carbon copy feature. To utilize this, simply include your carbon copy list at the top of the message body in this form:

```
cc: name1, name2, name3, etc.
```

or this form:

```
cc: name1
cc: name2
cc: name3
cc: etc.
```

If you are in the Matrix area, you may optionally specify a matrix address after each name. However, Maximus will also look up the SysOp name in your FIDOUSER.LST and NAMES.MAX files; if a match can be found, the message will be automatically addressed to the right node.

Message Editors

Maximus supports two types of message editors:, MaxEd, the full screen editor, and BORED, the line-oriented editor. Maximus also supports an external editor for local use, but that is covered in the control file documentation.

MaxEd

MaxEd is the Maximus full screen editor. This can only be used by users who are capable of receiving ANSI or AVATAR graphics, and have a screen width of 80 columns and a length of at least 23 rows. The full screen editor has a number of advantages over the line editor, the most obvious being that it is far easier to use. MaxEd is more like a word processor than a BBS editor; you can use the cursor keys to move around, insert and delete text in the middle of paragraphs, and so on.

MaxEd uses a mixture of WordStar, generic VT-100 and IBM-PC commands, a list of which can be obtained by typing ^n (Control-N) from within the editor. (The help file is contained in \MAX\HLP\FSED.MEC.)

Also, if the message you are editing is a reply to another, then you can quote text from the original message, and place it inside your own, which greatly increases readability. You can look at four lines at a time through a "quote window", and optionally copy those lines into the message you are writing. You can also page through the original message in either direction, forward or backwards.

MaxEd also has a special menu, which is accessible via either ^kH (Control-K and then 'H'). or the <F10> key The options available on the MaxEd menu are similar to those found on the BORED menu, and includes the following:

Continue

> This will return to MaxEd and allow the user to continue entering the message.

To

This allows the user to change the addressee of the message.

Subject

This allows the user to change the subject of the message.

From

This will allow the user to change who the message is from. The privilege level of this command should be set rather high, as this command can be used from any area, whether it's anonymous or not.

Handling

This is another command for which the privilege level should be set high. It will allow the user to change the flags on the message. Flags such as Private or Public can be changed, in addition to NetMail-type flags such as Crash, Hold or File Attach.

Read File

This will allow the user to enter a path to a file on your hard disk and read it in as a message. The privilege level of this command should be set fairly high.

BORED

BORED is the Maximus line-editor. BORED can be used by anybody, regardless of whether they have graphics or not. Most users who have graphics will most likely prefer to use MaxEd.

BORED allows the user to enter a message, one line at a time. When the user is finished entering the message, they are presented with the editor menu. The commands available on the editor menu are as follows:

Save

This will save the message the user has just entered.

Abort

This aborts the message without saving it.

List

Lists the message, preceding each line of the message with it's line number.

Edit

This command edits a line in the message, to correct any mistakes. Firstly, the user must select the line number that they wish to edit, then enter the text that they wish to replace, followed by the text you wish to replace it with. To insert at the beginning of the line, just press <Enter> for the "Text to replace" prompt.

Insert

This command will insert a blank line in the message preceding a specified line number.

Delete

This command will delete a specified line of the message.

Continue

Allows the user to continue entering their message, by appending to the end.

Quote

This command allows the user to quote text from the message to which he/she is replying. The user must enter the starting and ending numbers of the lines that he/she wishes to quote.

To

This allows the user to change the addressee of the message.

Subject

This allows the user to change the subject of the message.

From

This will allow the user to change who the message is from. The privilege level of this command should be set rather high, as this command can be used in any message area, whether or not the area is declared as anonymous.

Handling

This is another command for which the privilege level should be set high. It will allow the user to change the flags on the message. Flags such as Private or Public can be changed, in addition to NetMail-type flags such as Crash, Hold or File Attach.

Read File From Disk

This will allow the user to enter a path to a file on your hard disk and read it in as a message. The privilege level of this command should be set fairly high.

The File Menu

The following attempts to describe the options that would normally be used in a file area menu, and how they are displayed in the default configuration.

Area Change

This command allows the user to change to another file area. The user will be prompted to enter the file area they want to go to, or to enter a "?" for a list of the areas that are available to them. The user can also enter the "<" or ">" characters to go to the previous or next area in the list, respectively. If entering a barricaded area where a password is required, they will be prompted for the password before they are allowed to enter the area.

Locate

This command allows a user to search all of the file areas accessible with his/her priv level. The text that the user enters will be matched anywhere in the filename or description, so wildcards are not required. There are a couple of modifiers to the L)ocate command.

"L*" [L)ocate New] will search all of the file areas for any files that have been uploaded since the user was last on the system.

"L?" [L)ocate ?)Help] will display help information on the L)ocate command.

File Titles

This command will display a list of files in the current area, along with their descriptions. New files will be identified by a flashing asterisk (*). An argument to this command can be specified in the same manner as for the L)ocate command, however F)ile Titles will only search the CURRENT file area.

View

This command will allow a user to display the contents of any ASCII text file. The file is checked to make sure that it is an ASCII file, and the user is informed if it is not.

NOTE! In prior versions of Maximus, this command was called T)ype.

Goodbye

This is identical to the G)oodbye command at the main menu. It will log the user off.

Main Menu

This will return the user to the main menu.

Download

The download command allows the user to download one or more files from the file areas. Max 2.0 has a new format for downloading; now, users can enter the files to download on as many lines as they like, pressing <enter> after each file. Wildcards will be automatically expanded. If FB (the Maximus file database builder) is used, files can be quickly downloaded from any file area on the system, regardless of the current area. (Of course, the user's priv level is checked before allowing such an operation.) If files were T)agged earlier, those files will be automatically added to the current download.

If no default protocol is selected, the user will also be prompted to select a file transfer protocol. Maximus supports Xmodem, Xmodem-1k, SEAlink, Telink and Zmodem internally; more protocols, including DSZ, BiModem and Mpt, can be directly added as external protocols.

The user can also specify one of several operations before beginning the transfer. Entering "/q" causes Max to abort the transfer without sending anything. Entering "/e" causes Max to revert to "edit mode"; this mode allows the user to delete and list files in the download queue. Entering "/g" causes Max to hang up after the transfer is completed.

To start the download, simply press <enter> on a blank line.

Tag

The Tag command allows the user to queue one or more files for later downloading. The Tag command is also accessible through the "t" command (on the More [Y,n,t,=] prompt) when performing a F)iles listing or a L)ocate.

Max will prompt the user to enter the filename to tag. Wildcards are allowed, and if FB is used, global downloading is also permitted. Max will NOT print a newline when tagging from a file listing to prevent the rest of the list from scrolling off the screen.

To download previously-tagged files, simply select the Download option and press <enter> at the "File(s) to download" prompt.

Upload

This is the reverse of the download command, and allows a user to send files to your system. Maximus will ask the user which protocol they are using to upload, and in some cases the name of the file they are uploading (if the user selects a batch protocol, such as Zmodem or SEAlink, the filename is transmitted automatically in the transfer, so Maximus won't bother prompting the user). The protocols are identical to those used for the Download command.

Statistics

This option displays the user's statistics, including the time the user has been online for the current call, the time online for the day, amount uploaded, amount downloaded, NetMail credit (if any), and so on.

Contents

The C)ontents command will allow a user to look into a compressed file and see what files are contained inside. The C)ontents can view any .ZIP, .ARC, .PAK, .ARJ or .LZH file. For other compression methods, you are on your own.

Raw Directory

This will display a listing of ALL the files in the current file directory, not just the files listed in the files listing. The privilege level of this command should be set fairly high.

Override Path

This will allow the user to supply a path to a different directory than the one specified in the FILEAREA.CTL file for the current file area. This command should, for obvious reasons, be accessible only to users with high privilege levels. All changes made with this command are temporary, and the area's path will revert back to normal once you leave the area.

Hurl

This command will allow a user to move a file from one area to another. It will ask the user the name of the file to move, and the number of the file area to hurl it to. This command should be set to a rather high privilege level.

Kill File

This command will allow the user to delete a file from a file area. They will be asked for the name of the file to kill, and will then be asked to confirm that they want to delete it. If they answer "n" to the "Delete?" prompt, they will be given the option of leaving the file but simply removing the entry from the file listing. For obvious reasons, this command should be set to a high privilege level.

In addition, you can define other options in the file menu to call auxiliary menus, display text files, or run external programs. See the SILT Documentation for more details on defining menus.

The Change Setup Menu

From this section, a user may change as many of their settings as you permit. Upon entering the change menu, the user's profile will be displayed. The menu options are as follows:

City

Allows the user to change his/her city.

Phone Number

Allows the user to change his/her phone number.

Real Name

Designed for an alias system. Allows user to change his/her real name.

Password

Allows the user to change their password. The user will be prompted to enter his/her old password, then enter his/her new password twice. If the user gets his/her old password wrong five times, he/she will be quietly disconnected. If the new passwords don't match, the password will not be changed. Users should be encouraged to change their passwords every so often to prevent other people from finding them out.

Help Level

Allows the user to change his/her help level. There are four different help levels available in Maximus:

NOVICE:	Full Menus.
REGULAR:	Abbreviated Menus.
EXPERT:	No Menus
HOTFLASH:	Full-screen, hotkey interface.

Nulls

Allows the user to change the delay after each transmitted line. Most users won't need to use this unless they are using a really ancient terminal.

Width

Allows the user to change the width of his/her screen. The screen width is used to determine where Maximus should word-wrap, and how wide the menus should be,among other things.

Length

Allows the user to change the length of his/her screen. The screen length is used to determine when the "More?" prompts appear.

Tabs

Allows the user to toggle the translation of tabs. Normally tabs are sent unaltered, which speeds up the display time marginally. If this option is off, tabs will be translated to spaces before being sent.

More

This allows the user to toggle the `More [Y,n,=]?' prompts on and off.

Video Mode

This allows the user to change Video modes. Currently, only TTY (plain), ANSI and AVATAR video modes are supported. More video modes will be added in future releases of Maximus-CBCS for compatibility with other systems.

Full-Screen Editor

This command allows users to toggle the use of the MaxEd full-screen editor. If MaxEd is turned off, BORED will be turned on by default.

Screen Clear

This allows the user to toggle the sending of screen clearing codes in case his/her terminal cannot handle the TTY clearscreen (ASCII character 12) or the ANSI`CLS' command.

IBM Graphics

This allows the user to toggle whether or not Maximus will send IBM extended ASCII characters. The IBM (and compatibles) have a special `extended' 8-bit character set, which allows things such as box-drawing and block graphics, while running in text mode. Most non-IBM systems do not support these extended ASCII characters. For those users that have this option turned off, Maximus will translate IBM extended ASCII characters into standard ASCII characters in the range of 32 (decimal) to 128.

Hotkeys

This option allows the user to turn the hotkeys setting on and off. Turning on hotkeys instructs Maximus to execute commands as soon as they are received, without requiring an <enter> after every second keystroke.

Language

The Language command allows the user to select an alternate system language. Max supports up to eight different language files, all of which can be available at the same time. Simply add the appropriate "Language" statements to LANGUAGE.CTL, and those languages will become available for the Language command. The user's language preference is automatically saved across calls; to prompt users for an alternate language every time they log on, place a "[menu_cmd chg_language]" token at the top of WELCOME.BBS.

Protocol

The Protocol command allows the user to select a default transfer protocol. If a default file transfer protocol is selected, the Download command will immediately shift to the "File(s) to download" prompt instead of asking the user to choose a protocol.

Archiver

The Archiver command allows the user to select a default archiving and compression program. Max will shell out to the archiver when performing certain functions, such as compressing QWK bundles and decompressing REP packets. Max will use the information defined in COMPRESS.CFG to display a list of supported archivers, and will then prompt the user to choose one of the selected options. (See the READER.CTL section in the Maximus Technical Reference Manual for more information.)

Show in Userlist

The Show in Userlist option allows each individual user to select whether or not his/her name should be displayed in the system userlog. By default, the user's name will be displayed. If you wish to stop users from using this command, simply change the priv level of the Chg_Userlist command to Hidden.

Full-screen Reader

The Full-Screen Reader option allows users to toggle between the old-style Max 1.0x message header (the default) and the Max 2.0 full-screen reader box. The full-screen reader (hereafter referred to as FSR) displays a pretty blue box at the top of the screen, including the to/from/subject fields, reply links, and net/node information. However, this box takes a bit longer to display on a remote system, so it is turned off by default.

Alias

The Alias command allows each user to select an alternate alias. (Note: by default, this command is disabled in the distribution MENUS.CTL.) This command will change the "alias" field in the user record to whatever the user desires, as long as the specified alias doesn't already exist on the system.

Quit

This will return the user to the main menu.

Due to the dynamic menu structure of Maximus, it is possible to configure the change menu to include any commands. The above are only the commands that are most often used in the change menu on most systems.

The SysOp Menu

User Editor

This invokes the Maximus Internal User Editor. This command should be made available only to SysOps, for obvious reasons. See the section on the user editor for more details.

OS Shell

The OS shell permits either a local or remote shell to the operating system. Note that <Alt-J> can always be used for a local shell to the OS. (Note: OS/2 users should use MaxPipe to run CMD.EXE, as opposed to redirecting COMMAND.COM.)

If you are using an alternate command processor under DOS (such as 4DOS), make sure to change the menu entry for this command to reflect the command processor (ie. 4DOS.COM) instead of COMMAND.COM.

The Chat Menu

Maximus supports a full-fledged, internal multi-node chat module. Users on different nodes can hold private discussions with one another; users can even engage in large group discussions on a "virtual CB channel".

CB Chat

The CB Chat function allows two or more users to participate in a real-time multi-node chat. Messages can be sent back and forth, one line at a time, to all users who are participating in that particular conference channel. (Maximus supports up to 255 virtual CB channels.)

Page User

The Page User command allows for private chatting between two nodes only. Selecting Page causes Maximus to send a "chat request" to the specified node number, and it will automatically place the paging user in chat mode, waiting for the other user to respond to the page.

Answer Page

The Answer Page command is used in conjunction with the Page User command; after a user receives a page message from another user, the paged user can select Answer Page to engage in a private chat with the first user.

Toggle Status

The Toggle Status function allows a user to toggle his/her chat availability. Users which are unavailable for chat will be unpagable through the Page User command.

The Off-Line Reader Menu

The Off-Line Reader menu is the key to Max's internal QWK mail packer. This menu can be used to select a default protocol and archiving program, select a set of message areas, download messages from those areas, and upload reply packets.

Tag Area

The Tag Area command (equivalent to the command of the same name on the message menu) allows the user to select a set of message areas for access with the Browse or Download commands. This area selection will be saved across calls, so it's possible to select a set of message areas that interest the user and keep referring to those tagged areas for future calls.

Download New Msgs

The Download command packs new messages from all of the currently-tagged areas, compresses them into an archive, and prepares the file for download. (This option is the equivalent of specifying the argument "t;n;p" for the Browse command.) If no default archiver or protocol is selected, Max will prompt the user to select a protocol before commencing the download.

Upload Replies

The Upload Replies option allows users to upload a .REP reply packet into the message areas. Max will automatically determine the type of archive (even if it's different from the one selected as the user's default archiver), decompress the reply packet, and toss the replies into the appropriate areas.

Protocol Default

The Protocol option allows the user to select a default file transfer protocol; this option is identical to the one found on the Change menu,

Archiver Default

The Archiver option allows the user to select a default archiving program; this option is identical to the one found on the Change menu.

INSTALLATION INSTRUCTIONS

This section of the manual describes how to install Maximus from scratch. There is no conversion procedure available for other BBS programs aside from CVTUSR, the Maximus user file conversion utility.

Users of Maximus/2 should also read OS2.DOC before beginning the installation.

Step 1: Where do I put these files?

This is the easiest part of installing Maximus. Max is distributed in four separate parts: MAX200-1.LZH, MAX200-2.LZH, MAX200-3.LZH and MAX200-4.LZH. Each file contains crucial system information, so you'll need ALL FOUR FILES to install Maximus.

The first step in the installation is to unarchive all four files into an empty subdirectory using LHarc. (Obviously, since you are reading this file, you already know how to operate LHarc!)

Most of the files in the distribution kit are packed using a proprietary FIZ compression method, so you'll see a few files with a .FIZ extension, some documentation files, and the install program itself. The only way to unpack the .FIZ files is through the installation program, so that's what you should run next.

When run, the installation program will display some copyright information, and then proceed with the installation. It will ask you where to place the contents of each .FIZ file; unless you know what you are doing, or if want to use a different directory structure, the defaults are recommended. To select the default path, simply press <enter> at each prompt. (The installation program will create directories which do not exist.)

You should make sure to specify that you are performing a NEW installation (the default), since you need to extract ALL of the Max 2.00 files.

After extracting each .FIZ file, the installation program will then proceed with the first-time configuration. Simply type in the appropriate text at each box, and use <tab> (or click with the mouse) to move between fields. To select a particular option from a radio button group, use the left/right keys to cursor over to the appropriate location and press <space> to select that option. Click or press <enter> on the OK button when you have specified the correct values.

Step 2: Configuring your Modem

The modem is your gateway to the rest of the world. It can also be your biggest headache. Since this documentation can't cover all aspects of installing your modem, you should refer to your modem's manual if you are experiencing difficulties.

To begin with, a Hayes-compatible modem is strongly recommended. Although you may be able to use Maximus with a non-Hayes-compatible modem, Maximus will be unable to use your modem to its full potential.

If you do have a Hayes-compatible modem, but you can't get it to work, chances are that are that your problems have to do with DCD, or Data Carrier Detect. DCD is a signal which your modem sends to your computer when it is connected to another modem. However, when most modems are shipped from the factory, they have DCD forced on all the time. Unfortunately, this default is exactly the opposite of what most bulletin board packages require, including Maximus. Without DCD set correctly, Maximus will not be able to tell when a user has hung up, and your mailer software may have problems handling incoming phone calls. There are two ways to change the way your modem handles DCD, depending on the type of modem you have.

If you own a 1200 bps modem, then chances are that your modem's DCD handling is controlled through DIP switches. DIP switches are those miniscule plastic bits on the front, rear or bottom of your modem. (You may have to remove one or more panels to access these switches.) On most 1200 bps modems, DIP switch #6 toggles whether or not DCD is forced, and it should usually be set in the UP position. (However, it is a good idea to check your modem manual, just in case the switch settings are different.) Set the switch so that DCD reflects the modem's actual state, instead of being forced on. Also make sure that your modem will send back verbal result codes (as opposed to numbers). The DIP switch to control these result codes varies by modem manufacturer, so you will need to consult your modem's manual to determine which one to use.

If you own a 2400 bps (or greater) modem, then you will be spared the trouble of fiddling with DIP switches. Since almost all of these modems use non-volatile RAM (or NVRAM) instead of DIP switches, you can modify the modem's settings directly using a terminal program. Once you load up a communications package, type in the command `AT' and press <enter>. If everything is well, your modem should emit a response of "OK". After you have established that your modem is working, type in the command

`AT&C1&S1&D2&W' and press <enter>. (This command may not work for all 2400 bps modems; consult your modem manual for details.) This command will set up your modem so that DCD will always reflect the modem's actual state, rather than being always forced high.

One other thing which may caused problems is your modem cable. If your cable does not have the right number of signals wired through, your modem may also behave as if DCD was set incorrectly, regardless of DIP switch or NVRAM settings. One way you can verify your cable configuration is by borrowing a modem cable from a fellow SysOp. If you determine that your cable is at fault, you can go to your local computer store or service centre, and ask them for a cable that has at least pins 2-8 and 20 wired straight through.

Step 3: Installing a FOSSIL

Note! Unlike DOS, OS/2 does not require a FOSSIL driver. OS/2 includes a device driver to handle serial output, so Maximus/2 users can skip this section.

Under DOS, Maximus uses an external serial port driver. Like most FidoNet-compatible packages, Maximus requires a FOSSIL. `FOSSIL' is an acronym which stands for `Fido/Opus/SEAdog Standard Interface Layer'. A FOSSIL is a program which handles all of Maximus' low-level serial communication needs, including sending and receiving characters.

Using a FOSSIL allows Maximus to be used on semi-compatible computer systems which don't have 100% compatible serial hardware. The Maximus distribution package is shipped with a copy of David Nugent's BNU FOSSIL, which will be more than enough to get your system running. However, if BNU is not suitable or will not run on your system, other FOSSILs are available. Some of the most common FOSSILs are X00 and OpusComm. (Later versions of BNU may also be available.) If you can't find any of these programs on a BBS near you, one can be usually be obtained from a local Software Distribution System ("SDS") node, or from the author's BBS. (See the program licence for details on the author's BBS.)

When you are searching for a FOSSIL driver, make sure that the FOSSIL supports at least "FOSSIL Revision 5" or above, since Maximus won't work with lower FOSSIL versions.

There are two principal techniques used to install a FOSSIL. Some FOSSILs, such as OpusComm and BNU, are loaded as TSR (Terminate and Stay Resident) programs in your AUTOEXEC.BAT file. Others, such as X00.SYS, are loaded via your CONFIG.SYS file. Although different FOSSILs have different set-up instructions, it's fairly easy to install a FOSSIL for a basic configuration.

Here are the basic installation instructions for the three most popular FOSSILs. These cover only a generic 2400 bps modem on COM1; you should consult your FOSSIL manual to use Max on a different COM port or at higher speeds.

OPUSCOMM Installation:

To install OpusComm on COM1, simply insert the following command at the beginning of your AUTOEXEC.BAT:

```
OPUSCOMM
```

Make sure that OPUSCOMM.EXE is on your current PATH, or else this will not work.

BNUCOM Installation:

To install BNUcom on COM1, simply insert the following command at the beginning of your AUTOEXEC.BAT:

```
BNU
```

Make sure that BNU.COM is on your current PATH, or else this will not work.

X00 Installation:

To install X00 on COM1:, simply insert the following command at the beginning of your CONFIG.SYS:

```
DEVICE=X00.SYS
```

Make sure that X00.SYS is in your C:\ root directory, or else this will not work.

If you have advanced requirements (such as if your modem is on COM2, or if you are running a 9600 bps modem), you should consult your FOSSIL's documentation for more information.

Step 4: Editing Configuration Files

In order to run Maximus on your system, you will need to make several changes to your system set-up, especially in your CONFIG.SYS and AUTOEXEC.BAT files. (DOS only.)

Note! The default OS/2 CONFIG.SYS does not need to be modified to use Maximus/2. OS/2 users should skip this section.

In the CONFIG.SYS file, you'll either need to EDIT the following items if they already exist, or ADD them to the end of CONFIG.SYS if they do not. A generic text editor can be used to edit your CONFIG.SYS file, but a word processor in `non-document' mode will also get the job done.

The first change you have to make is to the `BUFFERS=' statement. (Again, if you don't already have a BUFFERS= statement, add it to the end of CONFIG.SYS.)  If the system you are using is an XT or a PC, make sure that the line reads `BUFFERS=20'. However, if you are using an AT or an 80386, this line should read `BUFFERS=30' for improved performance. The BUFFERS statement controls the amount of space that DOS uses for buffering files that are being read from or written to disk. Setting the BUFFERS command to the wrong number may slow down your system, but Maximus will still function correctly.

A second change must be made to the `FILES=' statement. Unlike the BUFFERS= statement, Maximus will not be able to run properly unless you have this statement set to a number GREATER THAN OR EQUAL TO 20. If you are running under a multitasking environment such as DESQview or DoubleDOS, then this number should be at least double that. In other words, if you have no multitasker, use `FILES=20'.  However, if you are running a multitasker, you should use at least `FILES=40', or else Maximus will probably exhibit erratic behaviour. Again, if you do not already have a `FILES=' statement, simply add it to the end of your CONFIG.SYS.

The third change you must make is to add the line `DEVICE=ANSI.SYS' to the end of your CONFIG.SYS.  If you look on your MS-DOS distribution disk(s), you should find a file called `ANSI.SYS' on one of them. After making the aforementioned change to CONFIG.SYS, copy ANSI.SYS from your DOS disk to the root directory of your hard disk.

NOTE! If you are using IBM-compatible hardware and wish to set Maximus up to use direct screen writes (using "Video IBM" or "Video BIOS"), you do not need to install ANSI.SYS. Please refer to the MAX.CTL documentation for more details, since your system may or may not be able to handle this video mode.

Finally, if you are planning on using Squish areas in a multinode environment, you MUST install a copy of DOS's SHARE.EXE. This sharing module allows Max to share Squish message bases with other nodes or tasks; using Squish areas in a multinode environment without SHARE will certainly cause damage. To install share, simply make sure that SHARE.EXE is in your root directory, and then add the line "share" to the end of your AUTOEXEC.BAT.

Note to Novell users: Installing SHARE is not completely necessary. As long as you load INT2F.COM after Netware shell, you can get away without loading SHARE.

Once you have made all of these changes and saved the configuration files, you should press <Ctrl-Alt-Del> to reboot your computer. Unless you reboot, changes made to CONFIG.SYS and AUTOEXEC.BAT will not take effect.

Step 5: Setting Up Control Files

This is the most complicated step in setting up your BBS. Maximus has four basic configuration files which control the operation of your system: MAX.CTL, FILEAREA.CTL, MSGAREA.CTL, and MENUS.CTL. MAX.CTL is the main control file, and it controls almost everything about your system from the log-on screens displayed to the time `reward' given to users who upload. MSGAREA.CTL controls all of the message areas on your BBS; FILEAREA.CTL controls all of the file areas, and MENUS.CTL controls all of the menus and options. The control files are all straight ASCII text, so to edit these files, you must either use a text editor, or a word-processor in a "non- document" or ASCII mode.

There are a lot of fairly complicated commands in these control files, but they give you control over even the most minute aspects of your BBS. Along with this power to tailor your BBS to your particular needs, there also comes the potential for screwing things up. Therefore, if you are a new SysOp, it is advised that you refrain from playing too much with these files until you get your BBS up and running, and have become more familiar with the various options which are available.

The installation program has already customized most of the system control files, so you don't need to do anything just yet. However, make sure that you know how to edit your control files, and make sure that your editor can handle plain ASCII files.

Step 6: Compiling the Control Files

Once you have made all of the required changes to MAX.CTL, the next step is to `compile' your control files. If you make any changes to your control files and forget to compile them, Maximus will not realize that anything has changed, and will still run using your old configuration.

Note! The idiot-proof configuration program already compiled your control file for the first time. However, it's a good idea to compile it again here, just so that you can learn how to compile everything properly.

Compiling your control files is easy; the program that compiles all of your control files is called SILT. Just type in the command `SILT <ctlname>' and press <enter>, where <ctlname> is the name of your main control file. If you are working with the default configuration, then `SILT MAX' should get you running and compile all four of the control files.

The main MAX.CTL has `Include' statements near the end which tell SILT to read in MENUS.CTL, FILEAREA.CTL and MSGAREA.CTL automatically. You cannot compile the other control files individually, so you must always give SILT the name of the main control file.

The first time you run SILT, it will probably complain that a few directories didn't exist, and that it is creating them for you. You need not worry, since this is perfectly normal. If you have made any errors in the control file, such as misspelling a keyword, SILT will warn you about those too. If you have made any errors, restart your text editor, move to the specified line number, fix the error, and then recompile using SILT.

Step 7: Starting Maximus

Once you have compiled your control files, you are finally ready to start Maximus. Although your BBS is still fairly bare-bones, it will be partially usable.

To start up Maximus for the first time, CD to the \MAX directory and type in the command "max -c". The `-c' switch tells Maximus to create a new user file, so you should only do this the first time you run Max. (If you are converting from another BBS program, now would be a good time to run CVTUSR.)

After a bit of disk activity, Maximus should display a copyright banner, and print out a message about the lack of an existing user file. Maximus will then display the prompt:
"Your Name Here [Y,n]?", where "Your Name Here" is the name entered in the "SysOp" box of the installation program. Type the letter `Y' to continue your logon.

After doing this, Maximus will display a bit of text which describes your BBS. This text is contained in a file called \MAX\MISC\APPLIC.MEC. (Files with an extension of .MEC will be discussed in greater detail later in this document.)

Maximus will also prompt you for a few pieces of information, including your city, phone number, and password. Maximus will also prompt you for ANSI screen controls, the MaxEd editor, IBM-PC characters, and hotkeys. Answer `Y' to all four of these prompts.

After Maximus has finished the interrogation, it will display a welcome screen and a bulletin file, and will then finally place you at the main menu. All of these screens are completely redefinable. Such customization will be described later in this manual.

Now that you have Maximus working, you will probably want to look around for a while. Feel free to play around, and explore the different features of your new BBS. If you would like to send some test NetMail, try going into the user editor and giving yourself some matrix credit.

When you have finished looking around at your new BBS, type `G' from any menu to log off, and Maximus will exit back to the command prompt.

Step 8: Support for Remote Callers

To handle non-local callers, Maximus needs to be run from a batch file. Although Max can be run locally from the command-line, to properly support external callers, a batch file must be created to recycle your system after a caller hangs up. In the case of a power failure, your batch files should also be capable of restarting the BBS without intervention.

There are two mutually-exclusive methods of running Max:

1)   Max can be run using the internal Waiting For Caller (WFC) subsystem. WFC handles all of the modem manipulation inside of Maximus, so no external programs are required to answer the phone. Max will answer the phone, flush out any MNP garbage, and drop immediately to the main BBS.

2)   Max can also be run under a "front end". A front end is a program which answers the phone, optionally performs some additional processing, and then passes control to the BBS. Front ends are commonly used for FidoNet support, since the FTSC-0001 protocol layer is not built into most BBS software. If you wish to become a member of FidoNet, you must run a front end.

In addition, a combination of these two methods is available for multi-node systems. In most cases, only one node needs to have a FidoNet address, so one line of your system can run a front end, whereas the other lines can use the internal WFC module. (All of this is possible using the same set of control files!)

Using the Internal WFC Module

When run in this mode, Maximus will handle incoming callers on its own. The first principle of WFC mode is the "-w" command line switch. To start Max in the most basic WFC mode, the following command should be used:

```
     max -w
```

"-w" instructs Maximus to enter WFC mode. After this command is executed, Maximus will load up, display a few windows, initialize the modem, and wait for an incoming call. Max will autodetect the incoming caller's speed and switch baud rates automatically.

By default, Maximus is set up to run on any Hayes-compatible modem. If WFC doesn't seem to be answering the phone correctly, read the comments in the Equipment section of MAX.CTL. You may have to fiddle with the command strings, but most modems can be made to work with Maximus.

In addition to the standard "-w" switch, you can also use the "-b" (baud rate) and "-p" (com port) switches to specify an alternate port number and maximum baud rate for the current node, overriding the defaults in the control file. For example, to start WFC on port 2 with a baud rate of 38400, specify the following command:

```
     max -w -p2 -b38400
```

To run WFC with a locked baud rate, either see your FOSSIL documentation about locking your FOSSIL driver, or see the documentation on the "-s" command-line switch in the Maximus Technical Reference Manual.

No matter which options you use, the command line must always include a "-w" if you wish to handle remote callers. This command must be placed in your batch file where you wish Max to be run.

The WFC module can also handle timed events which are run at specified times of the day. Please see the section entitled "WAITING FOR CALLER SUBSYSTEM".

Using an External Front End

In this mode, Maximus will not answer the telephone by itself. Since you are probably using this mode for FidoNet compatibility, you'll need a "front end" or a "mailer" to handle incoming and outgoing calls. There are six or seven FidoNet-compatible mailers freeware/shareware market, and three or four in the commercial market, so you have plenty to choose from. (At the time of this writing, the two most common mailers are BinkleyTerm and FrontDoor.)

Although setting up your front-end mailer is beyond the scope of this document, you will find several sample batch files for different mailers in the appendices in the Maximus Technical Reference Manual.

Your mailer's documentation may have some specific details for hooking it up to a Maximus system; if so, you should follow those directions. If not, read on. When Maximus starts up with a caller already on-line, it expects to find a minimum of one parameter: `-b<speed>', where <speed> is the baud rate of the caller currently on-line. Generally, this is handled through errorlevels or through a batch file called SPAWNBBS.BAT or EXEBBS.BAT.

Maximus Errorlevels

Once you are able to run Max for a remote caller (either through the WFC or from your front end), you are halfway there. When Maximus terminates, it needs to tell your system what to do next. For example, if a user entered an EchoMail message, you may want to run a utility (such as Squish) to process that message, or you may wish to run some sort of logging utility. To accomplish this, Maximus sets a numeric variable in DOS called an `errorlevel'. As mentioned earlier, Maximus has several errorlevels to juggle for various events, such as a user entering an echomail message, a user entering a netmail message, logging off before the user enters a name, etc.

You'll note that in several places throughout the control file, you can tell Maximus which errorlevel to use in certain situations. Errorlevels are always numeric, and always have a value from 1 through 255. (However, Maximus reserves errorlevels 1 through 4 to indicate errors, so you should not use these in the control file.) The control file comes pre-set with certain errorlevels, so unless you have a special case, you should not need to modify these.

Once Maximus is set up to use errorlevels, the only other task is to make your batch file detect the errorlevel Maximus used, and react accordingly. Errorlevels can be detected in a batch file quite easily, through the use of the `If Errorlevel <e> <a>' statement. <e> is a number, and should indicate the actual errorlevel to check for, and should also correspond to the errorlevel specified in MAX.CTL. <a> specifies an action, which is simply a normal batch file command.

However, there is one item which you should pay special attention to: DOS process all errorlevels using a greater- than-or-equal-to operation. In other words,the statement `If ErrorLevel 10 echo Hi!' will display the word `Hi!' if the errorlevel was set to 10, but will ALSO display `Hi!' if the errorlevel was set to 11, 12, or above. For this reason, if you have more than one errorlevel to check for, you should always arrange the group of errorlevel statements in a DESCENDING order. For example, to check for errorlevels 1, 3, 9, 10, 11 and 12, this would be the proper way to place the statements in your batch file:

```
MAX -p%1 -b%2 (other switches here; use "-w" instead for WFC
mode)

If ErrorLevel 12  (Do op `A' here.)
If ErrorLevel 11  (Do op `B' here.)
If ErrorLevel 10  (Do op `C' here.)
If ErrorLevel  9  (Do op `D' here.)
If ErrorLevel  3  (Do op `E' here.)
If ErrorLevel  1  (Do op `F' here.)
```

The only other point to remember is that ALL programs modify the DOS errorlevel. In the example given above, if you were to run a program called (for example) ABCD.EXE for the 'If ErrorLevel 12' statement, ABCD would RESET the DOS errorlevel after ABCD.EXE terminated. Since the batch file is executed one line at a time, the errorlevel statements which follow would use the errorlevel set by the ABCD program, rather than the value set by Maximus. To get around this DOS limitation, you must instead use a GOTO statement for the `action' portion of the errorlevel statement.

The GOTO command allows your batch file to jump to a completely different location within the same batch file, and start executing commands from that point. This is accomplished by using a statement in the form of `GOTO <l>', where <l> is a LABEL. A label is a unique, alphanumeric, ONE-WORD name, which indicates where in the batch file you wish to jump to. Some valid labels could be called `DoScan', `Heres_The_Scoop', and `ScanBLD'. However, a GOTO statement alone is not enough to complete the GOTO operation. You must also place the same label somewhere else within the batch file, which lets DOS know where the GOTO statement should end up. You can do this simply by placing a colon (`:') at the beginning of a line, simply followed by the label name.

For example, the following sample batch file:

```
Line 1:   :BigLoop
Line 2:   echo This will be shown repeatedly
Line 3:   goto BigLoop
```

...would cause the line `This will be shown repeatedly' to continuously display on the screen, until the user hits control-C to abort. (Omit the `Line X:' tags when typing this in.) When DOS starts the batch file, it will process each line in sequence. When DOS reads line 1, it will recognize that `BigLoop' is simply a label definition, and will ignore it. Next, DOS will read line 2, and process the ECHO command, by displaying `This will be shown repeatedly' on the screen. Once DOS encounters line 3, it will realize that it contains a GOTO statement, and will parse the label `BigLoop' out of the command. Having done that, DOS will then scan for the prior `:BigLoop' label, and jump back to line 1, thus continuing the cycle.

However, the GOTO command does have practical applications. The above example could be re-written like this:

```
     MAX (command line switches here)

     If ErrorLevel 12  goto OpA
     If ErrorLevel 11  goto OpB
     If ErrorLevel 10  goto OpC
     If ErrorLevel  9  goto OpD
     If ErrorLevel  3  goto OpE
     If ErrorLevel  1  goto OpF

     :OpA
     REM * Do operation `A' here.
     goto End

     :OpB
     REM * Do operation `B' here.
     goto End

     :OpC
     REM * Do operation `C' here.
     goto End

     :OpD
     REM * Do operation `D' here.
     goto End

     :OpE
     REM * Do operation `E' here.
     goto End

     :OpF
     REM * Do operation `F' here.
     goto End

     :End
```

In this situation, DOS would first compare the errorlevel returned by Maximus to those listed in the `If ErrorLevel' portion of the batch file, and then jump down to the corresponding label. For example, if Maximus exited using errorlevel 10, DOS would jump down to `:OpC', and process the statements which followed. The REM statement is simply a remark, and is ignored by DOS. (Typically, there would be one or more programs run in that portion of the batch file, as opposed to just the REM command.) After processing the REM command, DOS then reads the next line of the batch file, and processes it. The `goto End' statement is necessary, to make sure that DOS doesn't keep going, and execute the commands for `OpD' as well. (Recall that DOS just ignores LABEL DEFINITIONS, such as `:OpD'. Without the extra `goto End', the batch file would just `fall through' to the statements under OpD, OpE, etc. The extra goto specifically instructs DOS to jump to the `:End' label, which is located at the end of the batch file. On the other hand, some times you may WANT the batch file to `fall through', since it allows one to have several similar commands to be processed, when using the same errorlevel. Fortunately, cases where intentional fall-throughs are needed are few and far between.)

Arranging the batch file like this allows for more than one command to be executed for a certain errorlevel, and in addition, gets around the above-mentioned problem of other programs changing the errorlevel.

Max's errorlevel numbers higher than 5 are completely configurable, but if you are using the default configuration, the following errorlevels will be used:

* Errorlevel 255: This means that Maximus terminated with an undefined error condition. Your batch file should return to your front-end mailer.

* Errorlevel 12: This indicates that a user entered EchoMail and perhaps also NetMail during this session. In response, your batch file should call whatever scanner/packer program you use, such as SQUISH OUT SQUASH. If you are using *.MSG areas, you should also call SCANBLD at this point. After calling all of these external programs, your batch file should then restart your mailer.

* Errorlevel 11: This indicates that a user entered NetMail but no EchoMail during this session. In response, your batch file should call your mail packer, such as SQUISH SQUASH or oMMM. After calling the packer, your batch file should then restart your mailer.

* Errorlevel 5: This indicates that a user logged off without entering either EchoMail or NetMail during his/her session. In response, your batch file can execute a program such as the SCANBLD mail database utility. Alternatively, you may have your batch file simply restart your mailer.

*   Errorlevels 4 and 3: These errorlevels are used to indicate an error condition. Detecting either errorlevel 4 or errorlevel 4 should cause your batch file to restart your mailer (or to reload Max in WFC mode).

*   Errorlevel 2: Errorlevel indicates that the caller hung up before entering a valid name at the log-on prompt.

*   Errorlevel 1: This errorlevel indicates that the SysOp pressed <Alt-X> at the WFC screen. When your batch file receives this errorlevel, it should exit back to the operating system.

A generic batch file for Maximus (in WFC mode) might look something like this:

```
echo off
rem * This is where you call Maximus itself.  Change
rem * the `%1' and `%2' as necessary, to make Maximus
rem * work with your mailer.  (See the appendix on Sample
rem * Batch Files for more information.)

:Loop
MAX -w (command line switches here)
if errorlevel 255 goto Error
if errorlevel 16  goto Error
if errorlevel 12  goto EchoMail
if errorlevel 11  goto NetMail
if errorlevel  5  goto Aftercall
if errorlevel  4  goto Error
if errorlevel  3  goto Error
if errorlevel  2  goto Loop
if errorlevel  1  goto Loop


:EchoMail
rem * Invoke scanner and packer here.  Next, go to the
rem * 'Aftercall' label to process any after-caller actions.
rem *
rem * For the Squish mail processor, use the following command:
rem
rem SQUISH OUT SQUASH -fc:\max\echotoss.log

goto Aftercall
```

```
:NetMail
rem * Invoke packer here, then go to the 'Aftercall' label.
rem
rem For the Squish mail processor, use the following command:
rem
rem SQUISH SQUASH

goto Aftercall



:Aftercall
rem * Invoke after-each-caller utilities here.
goto End



:Error
rem * Something funky happened, so let's say so.
echo There has been an error!
goto Down



:End
rem * This label should re-load your phone answering program.
rem * If you are using the Maximus WFC, you should use the
rem * following line to jump back up to the top of the loop:

goto Loop

rem * If you are using FrontDoor, BinkleyTerm or another front
rem * end, you should comment out the above 'goto' and
rem * uncomment the following line:
rem
rem runbbs
rem
rem * This assumes that you use RUNBBS.BAT to start your
rem * front end.  If not, simply insert the name of that
rem * batch file above.
rem *
rem * WARNING!  OS/2 users must always use the "Spawn"
rem * method of executing Maximus.  Please see OS2.DOC for
rem * more information on setting up your batch files to
rem * work with OS/2.


:Down
```

```
rem * The system arrives here if there was a problem
echo Error!  Maximus had a fatal error and could not continue!
exit
```

Finally, if you are using any *.MSG message areas, it would be a good idea to read the chapter on SCANBLD right now. (SCANBLD is discussed in the "Maximus Utility Documentation" section of this manual.) When using *.MSG areas, SCANBLD updates the index file for each area, so you'll have to use it after every caller. In the sample batch file above, the call to SCANBLD would go under the label marked ':Aftercall'.

Step 9: Events and Yelling

The distribution copy of Maximus comes with a pre-configured event file. This "event file" serves two purposes:

1) With the internal Waiting for Caller subsystem, the event file is used to schedule "external events". External events are used for running external programs at predetermined times. These events are useful for nightly maintenance, general cleanup, and anything else you may desire.

2) "Yell events" are also controlled through the events control file. Yell events are used to define when callers are allowed to page the SysOp, how long the page should last, and which tune to play while the SysOp is being paged. Yell events are configured just like external events, except that a yell number is specified instead of an errorlevel.

All events are kept in the file called EVENTSxx.BBS, where 'xx' is the current task number. (For a system with no task number, use '00' for 'xx'.) Each node on a multi-node system must have a separate events file. However, all of the event files use the same format, so you can simply copy a master events file to EVENTS01.BBS, EVENTS02.BBS, and so on.

Yell Events

The distribution version of Maximus comes with an event file called EVENTS00.BBS. This event file contains a single yell event which is active between 13:00 and 23:00 (local time). You should read the comments in the event file for details, but you can add alternate yell time slots by simply duplicating that line and inserting the appropriate starting and ending times.

One thing to note are the numbers in the "Y3/3/1" flag. The first "3" specifies that the bell or tune will be played three times. The second "3" specifies that the user can yell up to three times in a row before Max will automatically turn off the Y)ell command. The final "1" specifies that tune #1 will be played from TUNES.BBS.

Max has an external file called TUNES.BBS which can be used to play simple notes and melodies on the local speaker. The comments in TUNES.BBS describe the structure of the file in more detail, but it essentially contains any number of notes, each note consisting of a frequency and a duration.

The tunes file can contain more than one tune; Max will play the tune "Yell1" when a user yells, since the distribution events file includes a "1" in the default yell event. The distribution version of the tunes file also includes "Yell2", "Yell3", "Yell4" and "Yell5", so you can try changing the last number of the yell event to produce different sounds. If you desire, you can also create your own tunes file.

External Events

As mentioned above, the events file also supports external events. (These events are only active when using the internal WFC.) To add an external event, simply add a line to EVENTSxx.BBS with the appropriate starting time, and add an "E<erl>" flag to the end of the line. <erl> specifies the errorlevel that Max will exit with at the specified time. After creating an entry for the event in EVENTSxx.BBS, you should modify your batch files to "trap" the specified errorlevel and to run your external event when that errorlevel is found.

For more information, see either the comments in the distribution EVENTS00.BBS or the section entitled "EVENTS.BBS CONFIGURATION" in the Maximus Technical Reference Manual.

Step 10: About Priv Levels and Locks

Unlike other BBS programs, Maximus does not use numbers to represent a user's access level. Instead, Maximus uses a series of keywords, which are called `Priv Levels'. Listed in descending order, the following privilege levels are currently supported:

Hidden (special, see note below)
SysOp
AsstSysOp
Clerk
Extra
Favored
Privil
Worthy
Normal
Limited
Disgrace
Twit

All of these privilege levels, except for `Hidden', can be applied to an ordinary user or menu command.  The`Hidden' privilege level is a special case, and if applied to a user, it ensures that the user will NOT be able to log onto your system, since Maximus will disconnect as soon as the user enters his/her password. Similarly, setting a menu option or a message/file area priv of `Hidden' means that nobody can access that option - not even the SysOp. This is useful for hiding commands that you don't want even yourself to be able to access.

The only privilege level which confers special capabilities is that of `SysOp'.  For example, users with the privilege level of SysOp can read private messages in any area, regardless of who the actual addressee is.  It should be noted that simply having your name listed in the `SysOp' section of the control file does NOT automatically confer SysOp privileges upon you. Your actual user profile, contained in the USER.BBS data file, must have your privilege level set to `SysOp'.

The remaining privilege levels do not confer any special capabilities and can be assigned to any users you wish, regardless of what the name of the privilege level implies. The privilege levels that are required to access menu options and message/file areas are controlled in MENUS.CTL and MSGAREA.CTL/FILEAREA.CTL, respectively. These three files will be discussed later in this document.

When first setting up your BBS, try and define a set of rules for using privilege levels. For example: "First-time callers get a privilege level of DISGRACE, validated users get a privilege level of NORMAL, and users who have donated money to the SysOp receive a privilege level of PRIVIL." If you don't lay out a plan for assigning privilege levels when you first start out, you will find it very easy to lose track later in the game of who has access to what.

Privilege levels are not the only way to control user access to various areas or menu options, since Max has a "lock and key" system. Using Max's locks, you can give specific users access to certain areas or options independently of that user's priv level. Once an option or area is `locked' with a specific lock number, a user must have the same key number to access that particular option or area. Valid lock/keys are numbers from 1 to 8 and letters from A to X. To add a lock to a message/file area or a menu option, simply add a forward slash after the privilege level, followed by the lock characters you wish to use. To illustrate, an area with an access level of `Privil/167AQX' would be accessible to only those users whose privilege level was at least `Privil', and who had keys 1, 6, 7, A, Q, and X.

You can modify a user's keys in the user editor (through the keys command). User keys can also be modified on-line, using the keys 1 through 8 on your keyboard. Alphanumeric keys can be also toggled through the priv level window, accessible through the 'S' option on the SysOp screen.

Step 11: Customizing *.BBS Files

Now that Maximus is functional, you are probably interested in customizing the screens and menus. Your first step will probably be to modify the welcome screens and information files which came with the default Maximus package.

Almost all of the display files are stored in your \MAX\MISC subdirectory, so this is where you will be doing most of your customization work. Different files are used for different purposes, and each file has its own unique name. For example, the first screen displayed to a new user is stored in a file called APPLIC.BBS, the bulletins are stored in BULLETIN.BBS, etc. You can change these filenames in MAX.CTL if you want, but it will be simpler if you leave the names alone for now.

Each file which can be displayed to the user ends with the extension .BBS. However, you will not be working directly with these files; just like the *.CTL files, display files need to be compiled before they can be displayed. Although it is possible to directly enter text into the *.BBS file, it is usually much easier to edit the "source code" which contained in the .MEC file.

"MEC" is an acronym which stands for `Maximus Embedded Commands'. However, if you really have a burning urge to insert compiled MECCA codes directly into a .BBS file, a list of the MECCA codes and their translations can be found in the MECCA language reference, in the Maximus Technical Reference Manual.

Two of the advantages to using MECCA (as opposed to simply creating files with ANSI graphics) are:

1) You can imbed user- or system-specific information into any screen displayed to the user, which gives your BBS a personal touch.

2) Secondly, MECCA allows you to directly enter colour tokens and cursor controls. The advantage to this is that Maximus will STRIP OUT these colour and cursor controls for callers who don't support them, which makes the *.ANS and *.ASC kludges of other BBS programs unnecessary. Only one file is needed for any given screen, and Maximus will convert it on-the-fly for callers with or without ANSI graphics, with or without the ability to display IBM graphics characters, or any combination of the above. This greatly reduces the time required to maintain your BBS, and it saves disk space too.

A *.MEC file is composed of straight text, plus some optional "embedded commands". If you want to see an actual *.MEC file, start up your text editor, and load the file `\MAX\MISC\NEWUSER2.MEC'. As you can see, the file is mainly composed of straight ASCII text, but with a few special commands inserted in, mainly to control colour and to perform special functions such as displaying the user's name.

Generally speaking, anything in a *.MEC file which is NOT inside a pair of square brackets is treated as straight text, and is therefore displayed to the user without alteration. Anything which IS enclosed in square brackets is called a `token', and is interpreted specially by Maximus. Tokens have various functions, which can range from changing the colour of the text, to running an external *.EXE or *.COM program, to invoking the internal mail- checker

Although you can see many MECCA tokens in use in the distribution \MAX\MISC\*.MEC files, a complete list of these tokens is available in the MECCA documentation, in the Maximus Technical Reference manual. A complete walkthrough to creating a MECCA source file is given in that same documentation, so you should at least read the first few pages of that section.

Once you have finished creating or modifying a *.MEC file, you must then compile it using MECCA, the Maximus Embedded Command Compiler (Advanced). Compiling a file with MECCA is easy. Simply type in the command `MECCA <filename>', where <filename> is the name of the *.MEC file you wish to compile. For example, to compile the file called `APPLIC.MEC' into the file called `APPLIC.BBS', type in `MECCA APPLIC'. MECCA will then compile the specified file, and warn you if you made any errors.

It's also a good idea to test your creations before allowing your users to log on. Chances are that some of your compiled *.BBS files will have problems, so users might get stuck in an endless loop (or something equally embarrassing). The Oracle utility will allow you to view *.BBS files off-line without having to start up a local Maximus session. Please see the section on Oracle for more information.

Step 12: Customizing Msg/File Areas

The next step in customizing your bulletin board system is to set up your own message and file areas. Although the Maximus distribution kit came pre-configured with three message and three file areas, you will probably want to expand beyond this, particularly if you are a member of FidoNet and carry a number of EchoMail conferences.

The first thing you should know is that all message areas are defined in MSGAREA.CTL, and file areas are defined in FILEAREA.CTL. Both message and file areas have 9-character area "names", so you can have an unlimited number of message and file areas, in theory. However, it is usually a good idea to start with a small number of areas, and then create new ones only as the need arises.

Each area definition in both MSGAREA.CTL and FILEAREA.CTL begins with an `Area <area_no>' line and ends with an `End Area' line. Everything between those lines pertains to that specific area only. <area_no> specifies the area `number' that is being defined. Max supports alphanumeric area "numbers" up to nine characters long; area names such "CHATTER" and "HELP" are acceptable, as are numbers such as "1", "2" and "3". You can even mix the two, such as "SYSOP249".

Inside each area definition are a number of keywords which describe that area. There are many advanced options available, but only a small subset of those are required for a basic configuration.

Remember that message areas are defined in MSGAREA.CTL, and file areas are defined in FILEAREA.CTL. It's possible to mix and match the two, but your system will be much easier to handle if the message and file areas are separated into two different control files.

Options in MSGAREA.CTL

MsgAccess <priv>

This statement defines the access level that a user must possess to access this message area.

MsgInfo <desc>

This statement tells Maximus what to describe this message area as to the user.

Local <path>
EchoMail <path>
Matrix <path>

Depending on what type of message area this is, you will use ONE of the above three statements to tell Maximus the type of messages that are in this area and in what subdirectory the messages are located. The area type can be LOCAL (Messages entered in this area stay on your system alone.), ECHOMAIL (Messages entered in this type of area are broadcast to other systems who are participating in the EchoMail conference.), or MATRIX. (Messages entered in this area are sent directly to the destination node, sometimes referred to as `NetMail'.) <path> specifies the directory in which the messages are located. Note: Each message area must have its own separate subdirectory! Yes, that is correct. You must have a separate subdirectory for each message area, just as you must have a separate subdirectory for each file area. SILT will create this directory for you if it does not already exist.

Private Only
Public and Private
Public Only
Read-Only

Again, you can use only one of these four statements in a given message area. These commands tell Maximus which type of messages to allow users to enter in this area. The first three statements perform as expected. Users may enter only private messages, they may be given the option of entering either private messages or public messages, or they may enter only public messages. For most users, a read-only message area means that they cannot enter any messages. Instead, they can only read existing messages. This is useful for those sysops who want to set up a message area in which he/she can post bulletins for users to read. Obviously, there must be some way for messages to be entered, or users would have nothing to read. Users with a with a privilege level of at least AsstSysOp are permitted to enter messages in read-only areas. Users below AsstSysOp will receive the message, `This is a read-only area', and Maximus will not allow them to enter a message.

MsgName <tag>

THIS IS ONLY REQUIRED FOR ECHOMAIL MESSAGE AREAS:

This keyword tells Maximus what the `echo tag' of the current message area is. This tag should be the same as the one which you have defined for this area in the control file for your EchoMail utility, usually called AREAS.BBS. For example, `MUFFIN' is the echo tag of the Maximus SysOp support echo.

Options in FILEAREA.CTL

FileAccess <priv>

This statement defines the access level that a user must possess to access this file area.

FileInfo <desc>

This statement tells Maximus how to describe this file area to the user. You can add some descriptive comments if you like, so long as all of the text will fit on one line. For example, the <desc> could be: "General purpose utilities".

Download <path>

This statement defines the subdirectory in which the files for this file area are contained. In other words, this is where users will be able to download files from.

Upload <path>

This statement defines the subdirectory in which uploads for this file area will be placed. You have two options for defining an upload path:

* Set it to the same subdirectory as the download path. This means that users should be in the correct area when they upload files, since the file will be available for download from the specified area as soon as the user finishes uploading.

- Set ALL upload paths in ALL file areas to point to the DOWNLOAD path for area 0, which is normally accessible by only the sysop. This is the most secure option, since it allows the sysop to check files which are uploaded before they are put on-line. Only after the sysop has checked the file out and Hurled it to the appropriate area is the file available for download by users. This also means that users can upload a file of any type anywhere and not have to worry about getting it in the correct area, since there is only one area for uploads.

Note! By default, all log-off comments will be placed in message area 1. However, if you wish to change the comment area, you can edit the "Comment Area" string in SESSION section of MAX.CTL.

One last reminder. Don't forget to recompile your control files with SILT after you make any changes!

Step 13: Maintaining File Areas

Although message areas are easy to create (by simply adding the appropriate area definition to MSGAREA.CTL), file areas require a bit more maintenance. Not only do you still need to create the area definition, but you also need to create a listing of files which are available for download in each file area.

If you have no files to go into a particular area, you don't have to do anything. Maximus will create a file catalog as needed when a user uploads a file.

Creating File Listings

If you already have some files that you would like to place in a certain file area, setting up that area is a bit more involved. For starters, do a CD to the DOWNLOAD directory for that file area, as you specified it in FILEAREA.CTL. From the download directory, just copy in all of the files that you wish to have in that file area.

Once you have done that, to create the initial file catalog, simply type in the following command at the DOS or OS/2 command prompt:

```
     FOR %F IN (*.*) DO ECHO %F >> FILES.BBS
```

You should see a bit of disk and screen activity as the file catalog is created for you. Although this creates the file names, you are not done yet. Next, load up an ASCII text editor, and load in FILES.BBS. Inside this file, you should see a list of the files in the directory. If you wish to add a comment for a particular file, you can just add one or more spaces after the filename, and insert your comment there. (You can use up to forty-eight characters if you wish to keep the comment on one line. If the comment is any longer, then Maximus will automatically wordwrap it onto the next line. You can make the comment as long as you want, up to 255 characters in length.)

Here are the contents of a sample FILES.BBS for a hypothetical file area:

```
TEST.ZIP      This is a text file
ABCDEFGH.TXT  This is another interesting text file
ACKTHPPT.ZIP  A digitized Bloom County comic strip
```

If you want to add files to your catalog after performing the initial `FOR %F' command, you can simply use your text editor to place the filename on a new line of FILES.BBS, followed by the description. Similarly, to delete a file from the catalog, just delete the line containing the file entry you wish to remove. You will also need to delete the actual file from the subdirectory.

When using the default `File Date Automatic' setting in the control file, Maximus will automatically place the file size and date beside the filename, in addition to adding a bit of colour to the catalog.

In addition, you can have files selected for "free download bytes" (doesn't count against user's download limit), "free download time" (doesn't count against user's time limit), or both. Before a file's description, place a slash and a flag character. For a free time download, use "/t". For a free bytes download, use "/b". For both free time and bytes, use "/tb".

For example, to make the Max 2.00 files not count against the user's DL quota, use the following:

```
     MAX200-1.LZH /b This is Max 2.00!
```

If you want to count the download against the user's byte total (but not the user's time), this would do:

```
     MAX200-1.LZH /t This is Max 2.00!
```

Similarly, if you want both free time AND bytes to be given to the user, the following also works:

```
     MAX200-1.LZH /tb This is Max 2.00!
```

Finally, you may add comments to the FILES.BBS listing which are NOT specifically related to one file. If the FIRST character on a line is a dash or a space, Maximus will treat the line as a comment and display it to the user. All of the usual *.BBS file colours and tokens are acceptable. If the first character on the line is a dash, the colour will be set to WHITE before displaying the line. If the first character is a space, the colour will be left alone (usually cyan).

Global Downloading and Dupe Checking

If you want to enable global downloading, upload dupe checking, and the fast locate function, you must use a compiled file database (as created by FB.EXE).

Global downloading allows users to download a file from an area other than the current area. For example, if a file in area 10 is called QWERTY.LZH, a user could enter "d;qwerty.lzh" from area 1 and successfully download that file.

Upload dupe checking tells Max to check for duplicate file uploads. Whenever a user uploads a file, Max will check the compiled file database to see if that file already exists in one of your other file areas. If it does, Max will abort the upload and display a short message to the user.

The fast locate command simply makes the Locate and Newfiles commands much faster. The search times on a large system will be sped up by several magnitudes, especially if you have a CD-ROM or WORM available to users. The fast locate command will be automatically used if a compiled file database exists in the Max root directory.

To create a file area database (and to use all of the above features), you must run a program called FB (Filebase Build). FB scans the FILES.BBS directories in each file area, compiles it into a flat binary file, and creates a sorted index at the end. Max can use this index to process global downloads and dupe checking; the flat binary files are used for the Locate and Newfiles commands.

Creating a compiled file database is easy; after making any change to the file areas (whether it be adding, deleting, receiving a TICK file, or even modifying a file description), the following command should be run from the Max root directory:

```
     FB
```

This tells FB to scan all of the file areas listed in AREA.DAT. This command creates all of the binary and index files, and so as long as you remember to run FB, global downloading and upload dupe checking can be used with no special effort. (You may need to enable upload dupe checking through the "Upload Check Dupe" keyword in the SESSION section of MAX.CTL.)

If you have a large system, running FB in this manner can take a long time (especially if you make a lot of small changes). For information on compiling only one or two areas at a time, or for using an area data file other than AREA.DAT, please see the section on FB in this manual.

Step 14: Customizing Menus

Max's menus are completely redefinable and can become difficult to manipulate. If you are just starting out, leave the menus alone until you become more comfortable with Maximus. However, even novices can change the access levels of particular commands in MENUS.CTL. The access level is usually located in the second or third column of each menu option. The only other "safe" field is the "Command as it appears to user" option. This command will be shown on the user's screen for NOVICE-level menus. Remember, the FIRST character in the command will be used to activate that option, so make sure that no more than one of your commands uses a given letter as its first character. (For example, don't use both "F)ile List" and "F)ind File" on the same menu.)

The last thing which is safe to play around with is the `MenuFile' option. This option tells Maximus to display a customized *.BBS file, as opposed to generating a canned file. This will allow you to completely customize your BBS and make it look different from any other. Please see the section entitled `Using Custom Menus' for more information on this topic.

Step 15: Configuring the QWK Mail Packer

Maximus supports a built-in QWK offline mail packer. This feature allows callers to log on, pack up messages from one or more message areas, and download a compressed mail bundle for off-line reading and reply. This packer is fully integrated with the main BBS, so the packer will automatically adjust itself as areas are added to or deleted from your system.

All of the information for the off-line reader is stored in READER.CTL. Your next task is load this file into your text editor, since several keywords need to be modified.

1) First of all, you will need to modify the "Packet Name" keyword. This keyword specifies the filename to use when sending QWK packets to your users, minus the ".QWK" extension. This name should be eight or fewer characters, case insensitive, with no spaces. Try to make the name describe your BBS in some way; an abbreviation of your BBS name is normally used. For example, a BBS called "Fowl Weather Post" might use a packet name of "FOWLPOST", and a BBS called "123 Systems Inc." might use a packet name of "123SYS".

2) Second, you should modify the "Phone Number" keyword to reflect your actual phone number. Maximus doesn't use this information, but your system's phone number is normally placed in the .QWK download packets. The phone number format given in the control file is suggested, even if you live in another country, since some external readers depend on this number being in a certain format. Unfortunately, this is a problem with the readers, so there's nothing that Maximus can do about it.

3) Finally, you may want to customize the "Max Mesasges" limit. If you run a busy system and you want to restrict callers from downloading more than 200 messages at a time, you can set a maximum here. In the distribution control file, a maximum of 600 messages is set. To completely disable the maximum, comment out the "Max Messages" keyword.

That's all there is to it! Beyond the initial configuration, the QWK packer is completely self-maintaining. No extra maintenance is required. However, if you would like to add some features to your mail packer (such as bulletins and new file lists), please see the section entitled "QWK MAIL PACKER".

Step 16: Miscellaneous Information

You have now completed the installation of Maximus-CBCS. Although you are now officially finished, there are a few things that you should keep in mind:

For users of *.MSG areas: A renumbering utility is required. If you carry any EchoMail conferences, a renumbering utility will be especially crucial. Maximus comes bundled with a program caller MR; this will automatically delete, renumber and link messages based on information given in MSGAREA.CTL. MR doesn't use low-level disk calls, so it's safe to use on all systems. For more information on MR, please see the Maximus Utility Documentation.

For users of Squish areas: Although Squish normally handles renumbering on-the-fly, you may need to use the "SQPACK" utility on a semi-frequent basis. (Typical systems only need to run SQPACK once a week, but large systems may need to use SQPACK on a daily basis.) Squish areas gradually develop small holes over a period of time, and the SQPACK program can be used to remove these holes. In addition, if you are performing renumbering by date, you must use SQPACK, since renumbering by date cannot be performed on-the-fly. SQPACK is part of Max's companion mail processor, SquishMail. The SquishMail package includes a number of other useful utilities (such as diagnostic and repair aids for Squish-style areas), so getting a copy of SQSH_*.* is a good idea.

Finally, if you are having trouble installing Maximus, chances are that you have not followed these instructions to the letter. Try reading through the installation instructions once more to see if you forgot anything.

ACCEM: MECCA Decompiler

ACCEM does the reverse of what the MECCA utility does: it takes a compiled .BBS file, and converts it back to a human- readable .MEC file. This can be useful if you have lost the source for one of your .BBS display files, or if you are trying to change a compiled .BBS file which someone else has given you.

After running ACCEM on a .BBS file, you can freely edit the resulting .MEC file, and recompile it as you wish. ACCEM can convert a .BBS file back to the complete .MEC file. The .MEC file created with ACCEM should be identical to the original .MEC file, with one small exception: since label names are not stored in the .BBS file, MECCA will simply convert these into numeric labels in the form of `/L0', `/L1', `/L2', etc. However, the reverse-engineered .MEC file will still compile correctly, and after compiling, the output from the new .MEC file should be identical to the original .BBS file.

The format for ACCEM is:

```
     ACCEM <infile> [outfile] [-s]
```

<infile> is the name of the .BBS file to convert. If no extension is given, then ACCEM will automatically use an extension of .BBS.

[outfile] is the name of the .MEC file to write to. If no extension is given, or even if [outfile] is omitted, then ACCEM will default to the <infile> filename, but using an extension of .MEC.

[-s] tells ACCEM to split lines which are over 100 characters. Using this will make ACCEM place an empty brace at the end of each line, thereby limiting the length of lines in the .MEC file, but without affecting the .BBS output. This is useful if you are decompiling a .BBS file with some very long lines, and if your editor can only display a limited number of columns.

For example, to convert TEST.BBS to TEST.MEC, all of the commands following would work equally well:

```
     ACCEM TEST
          or
     ACCEM TEST.BBS
          or
     ACCEM TEST.BBS TEST
```

If one wanted to split lines over 100 characters in length, the following would work, too:

```
     ACCEM TEST -s
     ACCEM TEST.BBS -s
     ACCEM TEST.BBS TEST -s
```

ANSI2BBS/MEC: ANSI to MEC conversion

ANSI2BBS and ANSI2MEC are two programs which will process a file containing ANSI graphics, and convert it into a file displayable by Maximus. ANSI2BBS will convert a file containing ANSI graphics directly into a .BBS file, which can be immediately displayed by Maximus. On the other hand, ANSI2MEC will convert a file with ANSI graphics into a file containing MECCA commands, as opposed to the compiled AVATAR sequences which are generated when ANSI2BBS is run. ANSI2BBS is useful for a one-time translation, when you are sure that the output will look right. ANSI2MEC is useful if you wish to display a file containing ANSI graphics, but also want to add some special effects, such as customizing the screen with MECCA tokens, or adding menus. After running ANSI2MEC and making any changes,the .MEC file must then be compiled into a .BBS file through MECCA, the Maximus Embedded Command Compiler. Please refer to the MECCA command language reference manual for more details on the operation of MECCA.

The format for ANSI2BBS (and ANSI2MEC) is as follows:

```
     ANSI2BBS <infile> [outfile]
        or
     ANSI2MEC <infile> [outfile]
```

<infile> is the name of the input file which contains ANSI graphics. If no filename extension is specified, then ".ANS" will be used by default.

[outfile] is the name of the file to place ANSI2BBS/ANSI2MEC's output in. If no output filename is specified, then ANSI2BBS will add a `.BBS' extension to the input filename, and send the output to that file. ANSI2MEC will do similar, except it will use an extension of `.MEC' instead.

Although ANSI2BBS and ANSI2MEC will try do the best job they can when converting an ANSI file, due to some ambiguities in the ANSI cursor-movement syntax, it is not always possible to correctly convert all ANSI graphics files. ANSI2BBS and ANSI2MEC will particularly have problems with some `highly-animated' screens,including some of TheDraw's alternate scanning modes, such as `Diagonal', `Gate', `Squiggle', etc. Maximus can handle almost all straight-through ANSI files, so unless you are using one of those scanning modes, you should not have any problems.

However, once you have converted an ANSI screen, it is a good idea to put it in a place where Maximus can access it, and test it in local mode, or with the Oracle utility. If the file didn't convert correctly and has some formatting glitches, then you have two choices:

- If the file is animated, load the file using TheDraw, an excellent ANSI screen editor by Ian Davis, and turn off the animation by pressing Alt-J and then `N' to convert the drawing to normal mode. Then try ANSI2BBS again, and hope it works.

- Convert the file using ANSI2MEC, and try to edit the MECCA tokens to fix the problem, and then compile the .MEC file using MECCA.

- Leave the file as-is, and send straight ANSI codes to the caller. Although it won't be viewable by AVATAR or TTY callers, and it will look icky if you have the `Video IBM' statement enabled, it should work okay for remote ANSI callers, if you enclose the graphics inside `[colour]' and `[endcolour] tokens.

CVTUSR: User File Conversions

CVTUSR is a utility which will allow you to convert foreign user files into a format Maximus can handle, from several other popular BBS programs. In addition, CVTUSR is also capable of generating an Opus 1.10-style USER.DAT file (for use with external programs) FROM Maximus' own USER.BBS.

The command-line format for CVTUSR is

```
     CVTUSR [[-x]...]
```

where valid switches are:

-l and
-m These switches tell CVTUSR to reset the lastread pointers in a Maximus-style USER.BBS. This option is normally only used to fix cross-linked lastread pointers, so it should only be used when "Lastread ptr xlinked" error message appear in the log. This function may reset the lastread pointers for some or all users, but it will correct any crosslinked pointers. Both -l and -m function identically.

-o110 This switch tells CVTUSR that you are converting an Opus 1.10-style USER.DAT to a Maximus-style USER.BBS. This procedure will convert almost all of the Opus 1.10 fields, with the exception of the expiry dates, personal welcome screens, and any utility-specific fields which may be stored in the user file. Your old USER.DAT is NOT overwritten by this procedure, so you don't need to make a copy of it.

-q This switch tells CVTUSR to convert a QuickBBS or RA-style USERS.BBS to a Maximus USER.BBS. This conversion function is not as complete as some of the others; it will leave out things such as ANSI graphics and "More [Y,n]?" prompting. However, the next time the user logs on, Maximus will ask for all the information which couldn't be converted, so the loss is minimal.

When converting QBBS security levels, CVTUSR will use the following conversion system:

QBBS User Level	Max Priv Level
100-32000	SysOp
90-99	AsstSysOp
80-89	Clerk
70-79	Extra
60-69	Favoured
50-59	Privil
40-49	Worthy
20-39	Normal
1-19	Disgrace

-x110 This switch CVTUSR to convert the Maximus-style USER.BBS to an Opus 1.10-style USER.DAT. CVTUSR will translate all of Maximus' fields into the equivalent Opus 1.10 fields, and will also attempt to store some Maximus-specific information in some "spare" fields. This means that it MAY be possible to convert the Maximus USER.BBS to the Opus USER.DAT, run a program which modifies the Opus version of the user file, and convert it back. Although this theoretically should work without problems, it is not advised, and doing so may cause some fields to be lost in the Maximus portion of the user file.

-n This flag tells CVTUSR to convert an older Maximus 1.0x USER.BBS to a Maximus 2.00 USER.BBS file.

-s This flag tells CVTUSR to swap the `alias' and `name' fields in a Maximus 2.00 USER.BBS file. This function is used to accommodate the changes in the alias system available in Maximus 2.00 from older Maximus systems.

EDITCALL: Call Fudging Utility

EditCall is a small utility which was written to dummy up the `number of callers' count contained in BBSTATxx.BBS. This program is useful if you have recently changed from another BBS package, and want to set the caller count to reflect the actual number of callers to your system.

The command-line format for EditCall is:

```
     EDITCALL <task_num> [num_calls]
```

<task_num> should indicate the task number whose caller counter you wish to set. If you are running only one line, then use 0 for <task_num>.

[num_calls] should indicate the new number-of-calls variable you wish to set for the specified task. If you don't specify this parameter, then EditCall will simply display the number of callers for the specified task number, without changing anything.

FB: File Database Compiler

FB is the Maximus File Database compiler. FB compiles the ASCII listings in FILES.BBS into a format which can be used by the global downloading routines, upload dupe checker, and the fast Locate command. FB is not required, since Max can use FILES.BBS directly, but you'll be missing out on a lot of the new features if you don't use FB.

WARNING! FB can only be used for automatic file dating. If you are using "File Date Manual", you should make sure to remove all file dates from FILES.BBS before running FB. Using manual dating is much less of a necessity with FB anyway, since file sizes and dates are stored in the binary database.

The command line format of FB is as follows:

```
     FB [[area_dat] [area...] [/u]]
```

[area_dat] specifies the name of the area data file. If no command line parameters are specified, FB will default to AREA.DAT in the current directory.

[area...] is a list of zero or more area "numbers", as specified in the "Area <anum>" statement in FILEAREA.CTL. If no area numbers are specified, FB will rebuild the entire file database. Otherwise, FB will only process the file areas given.

[/u] is the optional `upload' switch. This causes Maximus to scan the UPLOAD path of the specified file areas rather than the download path. This parameter is used internally by Maximus in RUNFB.BAT (see below for more information).

When compiling a file area, FB will parse FILES.BBS and create the following files in each file area:

FILES.DAT	A compiled version of each file's name, size, timestamp, privilege level and flags.
FILES.DMP	A compiled version of each file description.

```
     FILES.IDX A sorted binary index of all files in the current
               area.
```

If you are using a FileList statement in FILEAREA.CTL, Max will simply chop off the file list's extension and add .DAT, .DMP and .IDX as appropriate. For example, if you specified the following in FILEAREA.CTL:

```
     FileList    D:\AREA1.LST
```

FB would then create files called D:\AREA1.DAT, D:\AREA1.DMP and D:\AREA1.IDX. This allows owners of CD-ROMs to store all of the file area information in an alternate location.

Max also supports a special feature for updating the file database after a file is uploaded. After processing all of the files uploaded in a single U)pload command, Max will try to find a file called RUNFB.BAT (or RUNFB.CMD for OS/2) in the Max root directory. After the upload, if this batch/command file is found, Max will execute it with the following parameters:

```
     RUNFB <area_dat> <areanum> -u
```

where <area_dat> is the name of the current AREA.DAT file, <areanum> is the area into which the files were uploaded. (-u specifies that FB should check the upload paths rather than the download paths.)

In the default distribution, RUNFB.BAT looks like this:

```
     fb %1 %2 %3
```

This causes RUNFB to automatically run FB, which causes the file database to be updated on the spot. However, if you don't have enough memory to run FB in the BBS window, or if you don't wish to waste time by compiling the file database while the user is on-line, you can use the following in RUNFB instead:

```
     echo fb %1 %2 %3 >>do_fb.bat
```

(OS/2 users should use 'do_fb.cmd' instead of 'do_fb.bat'.)

The above command creates a log of file areas to update. If you are using this deferred file database updating, you should insert the following command in your batch file after each caller:

(DOS 3.0-3.2 only)

```
     if exist do_fb.bat command /c do_fb.bat
     del do_fb.bat
```

(DOS 3.3 and above only)

```
     if exist do_fb.bat call do_fb.bat
     del do_fb.bat
```

(OS/2 only)

```
     if exist do_fb.cmd call do_fb.cmd
     del do_fb.cmd
```

These lines cause Max to perform all file database updating after the caller logs off, which saves on both memory and on-line time. Make sure to select the command which is appropriate for your system and DOS revision, and make sure that it's run after EVERY caller, regardless of whether or not that caller entered NetMail, EchoMail, or no mail.

MAID: Language File Compiler

MAID is Maximus Language File compiler. MAID takes a language file source, such as ENGLISH.MAD, and turns it into a form usable by Maximus. The language file support can be used to support non-foreign languages, or simply to change the prompts in the English version of Max.

MAID reads the source language from a file called <langname>.MAD. The distribution version of Maximus comes with one language file called ENGLISH.MAD. The different files associated with language file processing area:

<langname>.MAD: The Maximus International Definitions file. This file contains the language "source". This file can be edited with an ordinary text editor, and this file is used as the input file for MAID.

<langname>.LTF: The Language Translation File. This is the compiled version of the <langname>.MAD file, as produced by MAID. This is the only file that Max uses; if you don't want to change the language source, feel free to delete the <langname>.MAD file.

ENGLISH.LTH: The dynamic language include file for the C language. This is only required when recompiling the source code.

ENGLISH.H: The static language include file for the C language. This is only required when recompiling the source code.

If you are planning on changing the language files, you must keep ENGLISH.MAD and ENGLISH.LTF as a minimum. By default, the install program places language files in the \MAX\LANG directory.

The command line format for MAID is as follows:

```
     MAID <langname> [-s] [-d]
```

<langname> is the full path and name of the language file. Do NOT include an extension; MAID will add the ".MAD" automatically.

-s   Specifies that you want the static language include file to be generated. This is only required when recompiling the source code. The <langname>.LTH file is not created unless you use this switch.

-d Specifies that you want the dynamic language include file to be generated. This is only required when recompiling the source code. The <langname>.H file is not created unless you use this switch.

If you wish to change the source in ENGLISH.MAD, you must be wary of three points:

1) Make sure NOT to change the order of the statements within the file! Disastrous consequences may result if the strings get out-of-sync. You can add and delete blank lines or comments, but leave the order of the strings alone.

2) After changing the language file source, the file must be recompiled with MAID to create the .LTF version of the language.

3) After compiling the language file, you MUST recompile your .PRM file with SILT! Since Max overlays strings in memory, Max needs to know how long your language files are; the summary of this information is stored in MAX.PRM. SILT collects this information while compiling the .PRM file, so you can simply run SILT to create the language file. If Maximus detects that the language file has been changed at start-up, it will print out "Old language <langname>: recompile PRM file with SILT!" and exit. It is vitally important that the .PRM file be recompiled after changing any of the language files.

Finally, if you wish to create a modified language for other people to use, you can simply copy ENGLISH.* to MYLANG.* (or whatever you wish to call the language), and then add that language as a "Language MYLANG" statement in MAX.CTL.

NOTE! We are attempting to set up a "Maximus Language File Repository". If you have written a Maximus language file for another language (or if you have done a take-off of the English language, such as JIVE, VALLEY, etc.), please upload a copy of the .MAD file to the author's BBS. A special portion of the Maximus distribution area will be set up for alternate language files, acting as a central location for obtaining new language files. Language files are also welcome in SDSMAX, the Software Distribution System area dedicated to Maximus and related utilities.

MECCA: Display File Compiler

MECCA is a companion utility which will compile *.MEC input files into binary *.BBS files, which can then be displayed by Maximus.

The operation of MECCA itself is fairly simple. The command-line format is:

```
     MECCA <infile> [outfile] [-t]
```

<infile> is the name of the input file, and if no extension is specified, an extension of `.MEC' will be used by default. <infile> can include wildcards, so entering `MECCA *.MEC' is perfectly valid.

[outfile] is the name of the compiled output file. This parameter is optional, and if not specified, it defaults to <infile>, using an extension of `.BBS'.

In other words, typing `MECCA BULLETIN' would cause MECCA to try to compile the file `BULLETIN.MEC' into a file called `BULLETIN.BBS'.

[-t] tells Maximus to compare the date stamps of the input and output files, and to skip the current file if the output filename is newer than the input filename. This is useful for recompiling an entire directory of .MEC files, if you can't remember what has changed. Simply type `MECCA *.MEC -t', and MECCA will automatically scan all of the files in the current directory, and recompile those which have changed.

Documentation on the internal format of a *.MEC file itself, and the keywords used therein, is contained in the MECCA Command Language Reference, in the Maximus Technical Reference Manual.

MR: Maximus Renumbering Program

MR is the Maximus-specific renumbering program. MR is only used for *.MSG-style areas, since Squish areas renumber themselves on-the-fly. MR automatically reads the information given in AREA.DAT, renumbers, deletes and relinks messages in *.MSG-style areas.

The command line format for MR is:

```
     MR <area_dat> [area...]
```

[area_dat] specifies the name of your Maximus AREA.DAT file. This parameter must be given for MR to operate properly.

[area...] specifies one or more message areas to be renumbered. If no message areas are specified, MR will process all *.MSG areas on your system.

When renumbering, MR will check the "Renum Days" and "Renum Max" settings for each message area. (For more information on Renum Days and Renum Max, please see the MSGAREA.CTL section of the Maximus Technical Reference Manual.) If either of those two keywords are set, MR will also purge messages based on the specified criteria. Messages can be killed by message number, by age, or both.

MR automatically updates the Maximus lastread files and message area links. Essentially, just include a call to MR in your daily batch file, and all of your renumbering needs will be taken care off. If you want to delete messages, make sure to include a "Renum Max" or "Renum Days" statement in MSGAREA.CTL for the areas to be trimmed.

ORACLE: Display File Viewer

ORACLE is an off-line .BBS file viewer. Unlike other BBS programs with embedded command languages, Maximus allows you to view compiled .BBS files without logging on, while still having the screens displayed exactly as they would be through Maximus itself.

The command line format for ORACLE is:

```
     ORACLE <bbsfile> [[-x]...]
```

<bbsfile> is the name of the compiled .BBS file you wish to view. If no extension is supplied, then .BBS will be used by default.

[-x] can be any of the following command-line parameters:

-hX   Sets the current help level to `X', where `X' is the first letter of a valid help level. Valid options are `hN' (Normal), `-hR' (Regular), `-hE' (Expert), and `-hH' (Hotflash).

-i   Disables high-bit IBM characters. With this option enabled, ORACLE will automatically translate IBM Extended ASCII to the ASCII equivalent.

-kX   Sets the user's keys to X, where X is simply a listing of keys to assign to the user. Valid keys are from 1-8 and A-X. For example, using `-k1237AD' would give keys 1, 2, 3, 7, A and D to the user.

-mX   Sets the local video mode to X, where X is one of `D' (DOS), `F' (FAST), 'B' (BIOS) or `I' (IBM). By default, ORACLE will use the video mode defined in the control file. However, if you wish to use ORACLE from remote, it may be necessary to use the DOS video mode, since output from the IBM and FAST video modes normally cannot be redirected to a COM port.

-pX   This tells ORACLE to read the Maximus .PRM information from the file `X'. If no PRM file is specified, then ORACLE will default to using MAX.PRM, in the current directory. THIS PARAMETER IS REQUIRED!

-q   This option enables the `quick' hotkey mode.

-slX   This option sets the virtual screen length to `X' rows. This doesn't change your physical screen length; however, it does determine when the `More [Y,n,=]?' prompts are displayed. This option defaults to 24 lines.

-swX This option sets the virtual screen width to X columns. This doesn't change your physical screen width; however, it does change the screen width, and controls when virtual screen wraps will occur.

-t The -t parameter forces Oracle into TTY video mode. This will disable all ANSI and AVATAR graphics commands, and display the file just as it would be shown to a TTY caller.

-vX This sets the user's privilege level to `X', where `X' is the first letter of a valid priv level. For example, `-va' would set the user's priv level to AsstSysOp, while `-vl' would set the user's priv level to Limited.

In addition to specifying the above parameters on the command line, you can also permanently set these options through an environment variable. Instead of typing all of the parameters on the command line, you can simply place the same options into the ORACLE environment variable.

For example, issuing the following sequence of commands:

```
     SET ORACLE=-pD:\Max\Max2.Prm -vS -q
     ORACLE D:\Max\Misc\Bulletin
```

is identical to entering all of this at once:

```
     ORACLE D:\Max\Misc\Bulletin -pD:\Max\Max2.Prm -vS -q
```

Although the first example looks like more typing, you can easily place the SET command into your AUTOEXEC.BAT, and only type `ORACLE <filename>' for each future file you wish to display.

SCANBLD: Database Builder

SCANBLD is the Maximus *.MSG database update utility. SCANBLD is only required if you are using *.MSG areas; if you are using Squish-format areas (which have their own indexing), SCANBLD is not required and this section may be skipped.

SCANBLD's primary function is to speed up Max's internal mailchecker, plus a few of the other internal commands. Since the *.MSG storage system is very slow, an index file must be built for each area to provide reasonably fast mailchecking. SCANBLD is not REQUIRED for using the mailchecker, but running the checker on a non-SCANBLD-compiled *.MSG area will be extremely slow.

When SCANBLD runs, it creates a database of all of the messages in each area of your system. SCANBLD must be run after certain actions, including after running a message renumbering utility, after receiving EchoMail, and so on. This is somewhat inconvenient; however, unless you switch to the Squish message format, you'll have to use SCANBLD to maintain speed in your *.MSG areas.

The command line format for SCANBLD is as follows:

```
     SCANBLD <user_bbs> <area_dat> ...
          [ [All | Local | Matrix | Echo | Conf | @<afile> | ...
          <areaname> | !<areaname> | /x]...]
```

<user_bbs> specifies the name and location of your USER.BBS file. This parameter is required.

<area_dat> gives the name and location of your AREA.DAT file. This parameter is required.

After the two mandatory parameters, any of the following commands can appear in any order:

ALL — Specifies that SCANBLD is to scan ALL areas, regardless of what type of message area it is. This is the default, and all areas will be scanned if no parameters are specified. Note that SCANBLD will only scan *.MSG areas. Even if a Squish-format message area is specified for processing, that area will be automatically skipped.

LOCAL — Specifies that SCANBLD should scan LOCAL areas.

MATRIX
Specifies that SCANBLD should scan MATRIX areas.

ECHO
Specifies that SCANBLD should scan ECHOMAIL areas.

CONF
Specifies that SCANBLD should scan CONFERENCE areas.

@<afile>
Specifies that SCANBLD should read the named file which contains a list of area tags. SCANBLD will compare those tags to those specified for the `MsgName' keyword in each area, and process areas with matching tags. <afile> can be Max's own ECHOTOSS.LOG, or it can be the import data files produced by Squish, ConfMail or QM.

<areaname>
Specifies a specific area number/name for SCANBLD to process.

!<areaname>
Specifies that this area is to be excluded from a normal scan, and is not to be processed. This is useful if you have two separate area numbers pointing to the same physical message path, or if you want to exclude certain areas from one of the above EchoMail/Matrix/Local scans.

/c
Forces SCANBLD to do a full compile of each area processed. By default, SCANBLD will normally try to update the mail database in the areas processed without rebuilding the entire area. You should ALWAYS use this option after renumbering messages, or else the message database will become out of sync with the actual messages.

/nd
Informs SCANBLD that you do NOT want the @<afile> specification to be deleted after processing. This is useful if you have other utilities which need the specified file, even after SCANBLD has finished with it.

/q
This switch forces SCANBLD into quiet mode. Instead of displaying each area's statistics, SCANBLD will instead display a single hash sign (`#') for each area processed.

The options specified on SCANBLD's command-line are cumulative, so entering the following:

```
     SCANBLD user.bbs areas.dat echo matrix 45 !22 @et.log
```

would cause SCANBLD to process all EchoMail areas, in addition to all NetMail areas, plus area number 45, plus the areas listed in the ECHOTOSS.LOG-format ET.LOG, with the exception of area number 22.

It is suggested that you run SCANBLD as follows, since this procedure ensures that the mail databases always remain synchronized with the actual messages.

AFTER A USER ENTERS ECHOMAIL (usually errorlevel 12):

```
     SCANBLD user.bbs area.dat local matrix @echotoss.log
```

AFTER A USER ENTERS MATRIX MAIL (usually errorlevel 11):

```
     SCANBLD user.bbs area.dat local matrix
```

AFTER A USER ENTERS LOCAL MAIL (usually errorlevel 5):

```
     SCANBLD user.bbs area.dat local
```

AFTER IMPORTING ECHOMAIL:

```
     SCANBLD user.bbs area.dat local matrix @echotoss.log
```

AFTER RUNNING ANY MESSAGE-RENUMBERING UTILITY:

```
     SCANBLD user.bbs area.dat all /c
```

Finally, after using an external message editor, you must SCANBLD all of the areas which you entered messages in. If your editor can produce an ECHOTOSS.LOG-like file, then you should run SCANBLD after your editor, using the command shown for `If a user enters EchoMail'. On the other hand, if your external editor does not produce an ECHOTOSS.LOG (or similar) file, then you must scan all areas, using the following command:

```
     SCANBLD user.bbs area.dat ALL
```

IF THESE INSTRUCTIONS ARE NOT FOLLOWED TO THE LETTER, SCANBLD may miss messages which would otherwise be flagged as new mail.

SILT: Control File Compiler

SILT is the Maximus control file compiler. SILT takes the raw ASCII control files which you have created and turns them into something that Maximus can use directly. Optionally, SILT can be instructed to parse only part of your system control files.

Starting SILT is fairly easy; the command syntax of SILT is:

```
     SILT <ctl_file> [-a] [-m] [-o] [-p] [-s103] [-s110] ...
          [-u] [-x]
```

<ctl_file> specifies the name of the control file you want SILT to process, and is the only required argument. If only the name of the control file is given with no other arguments, then SILT will process everything EXCEPT the SYSTEM*.BBS files. Otherwise, SILT will only process the parts of the control file which are given on the command- line. When specifying the control file, do not include the .PRM extension.

-a   Tells SILT to compile only MSG/FILEAREA.CTL.

-m   Tells SILT to generate the *.MNU files from MENUS.CTL.

-p   Instructs SILT to generate MAX.PRM, and if requested, the Opus version 14 and 17 *.PRM files.

-s103   Tells SILT to create SYSTEM*.BBS files for Opus 1.03 compatibility, in addition to compiling only MSG/FILEAREA.CTL.

-s110   Tells SILT to create SYSTEM*.DAT files for Opus 1.10 compatibility, in addition to compiling only MSG/FILEAREA.CTL.

-u   Tells SILT to run in "unattended mode". Normally, SILT will prompt the user for input in certain situations, including when a specified directory doesn't exist. (In such a case, SILT would ask the user whether or not s/he want to create the directory.) Using `-U' will tell SILT not to stop to ask for directions, and to create any nonexistent directories.

-x   Causes SILT to compile everything, INCLUDING the SYSTEM*.BBS and SYSTEM*.DAT files.

-y   This switch tells SILT _not_ to sort message and file area numbers into alphanumeric order. By default, SILT will sort message and file area numbers before writing the final index. If you wish to override the default sort order, using the -y switch tells SILT to use areas in the order that they are processed (with no sorting). This option can be used to override the default sorting order.

RUNNING EXTERNAL PROGRAMS

Although Maximus itself offers a large selection of internal features, chances are that you'll want to execute programs OUTSIDE of Maximus while a user is on-line. Maximus can run almost all types of external programs, from customized file-transfer protocols, to `door' programs written for another BBS program.

Maximus can execute as many external programs as you wish, from either a menu option, or from a MECCA embedded command file. Since these two pieces comprise the whole of the Maximus software, it means that you can run any external program anywhere, at any time.

Execution Methods

In general, Maximus supports four different methods of running external programs. You can determine what type of exit you need by looking at the list below, and comparing the methods' advantages and disadvantages to the requirements of the program you wish to run.

DOS: This is the so-called `normal' exit type. Maximus will load a second copy of COMMAND.COM, and then run the external program.

- This is the only way to run a batch file as an external program.
- Since COMMAND.COM has to be loaded, your external program will have about 174K less space to work in. (164K/Maximus + 10K/COMMAND.COM = 174K)
- If the program is located on the DOS PATH, no explicit path needs to be given.

You can execute internal DOS commands using this method, such as DIR, TYPE, CHDIR, etc.

RUN: This exit is identical to `DOS', except that COMMAND.COM is NOT loaded. This means:

- Your program will have a bit more memory to run in, since a second COMMAND.COM is not in memory.
- You cannot run a batch file with this command.

- This method will run faster than the DOS method, because COMMAND.COM doesn't have to be loaded.

CHAIN: This command is just like `RUN', except the external program will be loaded ON TOP of Maximus. In other words, the external program will overlay Maximus in RAM. NOTE: this execution format may not work with all hardware and software combinations.

- Since the program is loaded on top of Maximus, it will be able to use all available system memory.

- This is a one-way command, and control will not be passed back to Maximus when the program terminates. It is the responsibility of the external program to reload Maximus with the appropriate parameters once the program has been executed. (See below about restarting Maximus after a CHAIN command.)

ERRORLVL: This command tells Maximus to exit completely from memory, and exit to the calling batch file or program.

- This command is slow, since the transient portion of COMMAND.COM must be reloaded.

- The only interface Maximus has with the external program is an errorlevel. (However, this is not totally true. See below about Errorlevel Batch Files.) Also, see below about restarting Maximus after an errorlevel exit.

ErrorLevel Batch Files

When exiting via an errorlevel exit, Maximus uses a concept similar to BinkleyTerm's `BBSBATCH' command, which allows Maximus to pass command-line parameters to an external program. To create an errorlevel batch file, instead of specifying only an errorlevel as the command to execute, add a single SPACE character (or an underscore, if you are running the external program through a menu option), and then the name of the command you wish to run.

ie. `Xtern_Erlvl 65' -> `Xtern_Erlvl 65_Myprg_Arg1_Arg2'.

When Maximus encounters an argument after the errorlevel, it will write a file called ERRORLVL.BAT in the Maximus startup directory, containing the argument specified after the errorlevel. (If you have a task number defined in MAX.CTL, then Max will write a file called `ERRORLxx.BAT' instead, where `xx' is the task number, in hexadecimal. However, aside from the filename change, the use of ERRORLVL.BAT is identical to that in a single-node environment.) In the case of the above example with MYPRG.EXE, the ERRORLVL.BAT file would contain:

```
     Myprg Arg1 Arg2
```

Once the errorlevel batch file has been written, then Maximus will exit with the specified errorlevel. You can then trap this errorlevel in your batch file, and use a `CALL ERRORLVL.BAT' command to execute the command. (If using DOS 3.2 or under, replace the `CALL' with `COMMAND /C'.) After executing the external program, you can then restart Maximus by the method described in the next section.

Restarting After Chain/Errorlevel

After executing an external program via the CHAIN or ERRORLEVEL exit methods, Maximus can restart itself exactly where it left off and appear as if Maximus had remained in memory for the entire time.

This feature is made possible through the `-r' command-line parameter. When Maximus is invoked using `-r', it will read a file called RESTAR*.BBS from the root directory. This file was written to disk just before Maximus executed the chain/errorlevel command. This file contains all of the information that Maximus needs to start up again, so Maximus will simply pick up right where it left off, whether Maximus was displaying a menu or in the middle of a *.BBS file.

Also make sure to specify the *.PRM file name on the command-line, if you are not using MAX.PRM. In addition, if you are using a NON-ZERO task number, then you MUST accompany the `-r' option with the `-nXX' (set task number) option.

WARNING! Never attempt to use an `[xtern_erlvl]' token before a new caller has reached the NEWUSER2 file. Maximus cannot perform a restart until it knows who the user is, and that means that the user must have entered their name, password, graphics selection, etc.

This is an example batch file which utilizes errorlevel batch files, and also the restart option:

```
REM * These first "%1 %2 %3" parameters will be passed to
REM * the batch file by your mailer.  However, they
REM * are not really important when dealing with errorlevel
REM * batch files, so we'll just assume that they are
REM * correct.  Also, make sure that the `:DoErlvl' label
REM * comes AFTER the main `Max -b%1 ...' command.

Max -b%1 -p%2 -t%3 -n2

:DoErlvl
if errorlevel 65 goto Outside
REM * [..more errorlevels here..]
if errorlevel  1 goto end
goto end

:Outside
REM * Replace the `Call' with a `Command /C', if using DOS
REM * 3.2 or below!  Also, make sure that the number after
REM * the `-n' parameter specifies the Maximus task number
REM * to use, if not the one specified in the control
REM * file.
REM *
REM * Finally, if you are using a non-zero task number, keep
REM * in mind that the filename Maximus writes will be
REM * `ERRORLxx.BAT', where `xx' is the hexadecimal task
REM * number.

call C:\Max\Errorlvl.Bat
Max -r -n2
goto DoErlvl

:End
```

After you have created a batch file such as this, using errorlevel exits becomes just as easy as any of the other exit types. In MECCA, instead of using something in this format:

```
     [xtern_run]D:\Path\Progname.Exe Arg1 Arg2
```

one could easily replace it with something like this:

```
     [xtern_erlvl]65 D:\Path\Progname.Exe Arg1 Arg2
```

As you can see, once you have added the errorlevel code to your batch files, adding new options requires only a minimal amount of work.

External Program Translation Characters

When passing a command-line to an external program (and also when parsing some special MECCA tokens), Maximus can include information about the user and SysOp by using special translation tokens. A format token consists of a percent sign and a single, case-sensitive letter or symbol. Maximus will interpret the character following the percent sign, and replace it with the variable which that character represents.

Maximus currently supports the following external program translation characters:

Char	Translation
%!	Embeds a newline in a string.
%A	The user's FIRST name, in upper-case.
%b	The user's baud rate. If the user is a local caller, then this will translate to `0'.
%B	The user's LAST name, in upper-case. (If the user has no last name, then this will translate into `NLN', `No Last Name'.)
%c	The user's city.
%C	The response to the last `[menu]' MECCA token.
%d	The area number of the current message area
%D	The area number of the current file area
%e	The user's password
%E	The user's screen length, in rows
%f	The user's first name, in mixed case.
%F	Path to the current file area.
%g	User's graphics mode -- `0' for TTY, `1' for ANSI, and `2' for AVATAR.
%G	User's Daily DL limit, in kilobytes
%h	The user's phone number.
%H	Number of kilobytes downloaded today
%i	Total downloads
%I	Total uploads
%j	Minutes on-line, this call
%k	The current node's task number. (`0' for no task number.)
%K	The current node's task number in hexadecimal format padded with leading zero to make it two characters. (`0' for no task number.)
%l	The user's last name, in mixed case. If the user has no last name, then this will translate into `NLN'.
%L	If the user is REMOTE, this will translate into the string `-pX -bY', where X is the port number (1=COM1,

2=COM2, etc) and `y' is the baud rate.  If the user is LOCAL, this will translate into a simple `-k'.

%m The name of the first file to transfer when invoking an external protocol.

%M Path to the current message area.

%n User's full name, in mixed case.

%N The name of your BBS, as defined in MAX.CTL.

%p The current port number (0=COM1, 1=COM2, etc).

%P The current port number (1=COM1, 2=COM2, etc).

%q Path to the current msg area (NO trailing backslash)

%Q Path to the current file area (NO trailing backslash)

%r The user's real name, if applicable.

%R All remaining stacked text, as entered at the last menu.

%s The SysOp's last name, in mixed case. If the SysOp has no last name, then this will translate into `NLN'.

%S The SysOp's first name, in mixed case.

%t The amount of time the user has left, in minutes.

%T The amount of time the user has left, in seconds.

%u The user's user number.

%U Simply translates to an underscore.

%v Path to the current upload area (with trailing backslash)

%V Path to the current upload area (NO trailing backslash)

%w The path to the current FILES.BBS-type file. This takes into account the alternate names which may be used by the `FileList' option.

%W The "steady baud rate", as passed via the -s command-line parameter.

%x Drive letter of current drive, in upper case.

%X The last read message number for the current message area. This only works while in a message area.

%Y The user's current language number, zero based (0 is first language, 1 is second, etc.)

%Z Translates to the user's full name, in caps.

In addition to the above translation characters, there is also another set of almost-identical characters, which can be used when giving Maximus the name of a file to display. However, the first character in sequence should be a "+", rather than the usual "%". The second character WILL be treated as shown above, and translated normally. For example, to display a file called D:\<#>.BBS, where <#> is the user's number, you would use the following command in MENUS.CTL:

```
     Display_File  D:\+u.BBS  Twit  "Display It!"
```

Please keep in mind that the usage of the "+" is only required when specifying a filename to display. The percent sign should be used in all other cases.

Finally, there is one additional shortcut for *.MNU menu names. If you wish to substitute the current task number in a filename, then substitute the "*" character where you wish the task number to appear, and Maximus will translate it automatically. For example, the following line...

```
     First Menu  MAIN*
```

would cause task 0 to display a menu called MAIN00.MNU when first executed, task 1 to display MAIN01.MNU, etc. (Keep in mind that the task number is in hexadecimal, and therefore the menu displayed for task 12 would be MAIN0C.MNU.)

Running Doors

A `door' is just a fancy name for an external program which can be run and can communicate with an on-line user. Most door programs contain modem routines, so they can keep track of a user's time limit, make sure that the user doesn't drop carrier while inside the door, etc.

However, running a door program presents a special problem. There are several conflicting standards for `door interfaces', which are what controls the way the BBS program passes information to the door. Most modern door interfaces can pass out the user's name, whether or not the user supports ANSI graphics, the name of the SysOp, etc.

Maximus includes built-in support for the Opus 1.03 LASTUSER.BBS standard, as well as the capability to DIRECTLY write ANY text-based door interface file. The distribution version of Maximus comes with MECCA scripts which allow you to create door interface files for the following formats: DORINFO1.DEF (QuickBBS and RBBS), CHAIN.TXT (WWIV), CALLINFO.BBS (WildCat!) and DOOR.SYS. In addition, you can write your own MECCA scripts, which allow you to generate a door interface file for almost any other system type.

Maximus can achieve this through the use of the `[write]' MECCA token. Although the `[open]' and `[post]' commands were originally used for on-line questionnaires, they serve a dual purpose under Maximus. The `[write]' token will simply write a line of text to the previously-opened file, while making translations to the string, as described in the `External Program Translation Characters' section, above.

For example, the only requirement to make Maximus write a QuickBBS or RBBS-compatible DORINFO1.DEF file is to copy the following MECCA script into a file called DORINFO.MEC, and compile it. (Note! If you are using the standard distribution package, then you can find this file, including the compiled .BBS version, in the \MAX\MISC subdirectory.)

When copying this into a file, be sure to line up all of the text against the left margin. Also make sure to change the [delete] and [open] tokens to reflect the path where you want the DORINFO1.DEF interface file to be placed.)

```
[delete]C:\Max\Dorinfo1.Def
[open]C:\Max\Dorinfo1.Def
[write]%N               [ comment Write the BBS name]
[write]%S               [ comment Write the SysOp's f.name]
[write]%s               [ comment Write the SysOp's l.name]
[islocal write]COM0     [ comment Write the COM port]
[isremote write]COM%P   [ comment (local is always COM0)]
[write]%b BAUD,N,8,1    [ comment Write the baud rate]
[write] 0               [ comment Say we are not networked]
[write]%A               [ comment Write the first name]
[write]%B               [ comment Write the last name]
[write]%c               [ comment Write the city]
[write]%g               [ comment Write the graphics]
[sequal write]100       [ comment Write the security level]
[sxclude write]50       [ comment Ditto]
[write]%t               [ comment Write the time remaining]
[write]1                [ comment Say we are using a FOSSIL]
[quit]                  [ comment And we are done!]
```

You can create similar files for other door interface types, by simply creating another MECCA file containing the appropriate commands. (A list of the external program translation characters has been provided in the prior section; however, you can use the above DORINFO.MEC file as a guide to designing your own door interface files.)

There are three ways to have DORINFO1.DEF (or any of the above-mentioned files) created when running an external program:

TO CREATE DORINFO1.DEF FROM A .MEC FILE:

To have the appropriate door file created, simply include the following line, whenever you wish to have DORINFO1.DEF written:

```
[link]C:\Max\Misc\Dorinfo
```

As mentioned above, the distribution version of Maximus also comes with MECCA scripts to generate several other types of door interfaces. The format for using these is similar to the interface described above:

```
[link]C:\Max\Misc\WWIV        - To create CHAIN.TXT
[link]C:\Max\Misc\CallInfo    - To create CALLINFO.BBS
[link]C:\Max\Misc\DoorSys     - To create DOOR.SYS
```

TO CREATE DORINFO1.DEF FROM A MENU OPTION:

Similarly, you can achieve the same results through a menu option, by simply linking the appropriate .BBS file to the menu option. (For more information, please see `Linking Menu Options' in the Maximus Technical Reference Manual.)

For example, to create a DORINFO1.DEF file for running a program called `C:\Max\Prg.Exe', you would use something similar to the following in MENUS.CTL:

```
NoDsp  Display_File   C:\Max\Misc\Dorinfo Twit "Run Prg.Exe"
       Xtern_Run      C:\Max\Prg.Exe      Twit "R"
```

Again, the same concept can also be applied to the other MECCA-created door scripts, by simply substituting the name of the script into the Display_File command.

TO HAVE DORINFO1.DEF CREATED AUTOMATICALLY:

If you wish to have DORINFO1.DEF written every time Maximus exits for an external program, for whatever reason, you can simply edit the `Uses Leaving' statement in MAX.CTL, such that it reads like this:

```
     Uses Leaving C:\Max\Misc\Dorinfo
```

This will instruct Maximus to create DORINFO1.DEF whenever Maximus runs an external program, without needing to be specifically instructed to.

What follows is a demonstration of how to install a non- Maximus door in a menu file, assuming that you have NOT implemented the above `Uses Leaving' statement in MAX.CTL.

In MENUS.CTL, you should add the following to the menu which you wish the door to appear on.

```
        Display_File  Misc\DorInfo   Disgrace   "Play BoreDoor"
NoDsp   Xtern_Dos     BD\Bore.Bat    Disgrace   "P"
```

The `Display_File' command tells Maximus to write the DORINFO1.DEF file, which will always be written to the C:\MAX directory (unless you have changed the .MEC file).

The `Xtern_Run' command tells Maximus to run the batch file called BD\BORE.BAT, which you'll need to create later. (The `NoDsp' in front tells Maximus not to show `P' on the menu a second time, since you only want the first `Play BoreDoor' to be visible. See the section on Linking Menu Options (in the Maximus Technical Reference Manual) for more details.)

When a user selects `P' from the menu, Maximus will execute the above options in order. That means that DORINFO1.DEF will first be written, followed by the execution of BORE.BAT.

Although the contents of the batch file are highly specific to the door program you'll be running, in general, you should use a format similar to this, in BORE.BAT:

```
echo off

REM * Change to the right directory
cd \Max\BD

REM * Copy the DORINFO1.DEF file from the
REM * main Maximus directory into the
REM * current directory, which is probably
REM * where the door program will look for
REM * it.
copy \Max\Dorinfo1.Def

REM * This is the door program itself.  The
REM * command-line parameters will be
REM * specific to the door you are running, so
REM * you should consult your door's installation
REM * instructions for more details.
BoreDoor

REM * Now change back to the Maximus
REM * directory.
cd \Max

REM * And exit back to Max!
exit
```

On-Line User Record Modification

Some door programs may be written specifically for Maximus, and may need to directly change part of a user's profile (such as the user's remaining time, ANSI/AVATAR preference, phone number, etc.), even while the user is on-line. Maximus supports this feature through a series of special keywords and characters, which cause it to re-read the LASTUSxx.BBS file after returning from an external program.

If you are running the external program through an option in MENUS.CTL, then the fastest way to enable on-line modification is to place the `ReRead' modifier in front of the usual `Xtern_xxx' option. In other words, instead of invoking the program like this:

```
          Xtern_Run   D:\Path\Prog.Exe   Disgrace  "Program"
```

you would place the following line in MENUS.CTL, which would enable on-line modification:

```
ReRead    Xtern_Run   D:\Path\Prog.Exe   Disgrace  "Program"
```

Similarly, you can perform the same operation when using the [xtern_xxx] MECCA tokens, by using an `@' as the first character in the program name. For example, instead of using this:

```
     [xtern_run]D:\Path\Prog.Exe
```

you would use this, instead:

```
     [xtern_run]@D:\Path\Prog.Exe
```

However, keep in mind that most programs don't need this feature. For security reasons, you should not use this feature, unless the external program's documentation states that on-line modification will be performed.

MULTI-LINE OPERATION GUIDE

In addition to general multi-line support, Maximus 2.00 supports an integrated paging and inter-node chat facility, which makes it ideal for multi-line systems. In addition, Maximus uses NetBIOS-compatible file opening calls (using the SH_DENYNONE attribute), which makes Maximus even more suited for network applications.

This section is merely a guide to running Maximus in a multi-line environment. Undoubtedly, there will be some problems which are not covered by this section, and there will be some questions left unanswered. However, this section will hopefully answer most of the basic questions, and at least give you a head start on installing a multi-node version of Maximus.

Installation

Installation of a network version of Maximus is fairly similar to a normal installation. Simply run the INSTALL program, and answer all of the questions it asks.

However, there are several important things to consider:

*   Normally, you'll need a SEPARATE batch file for EACH copy of Maximus you wish to run. You can reduce duplication by moving common parts of the batch file into a separate file and calling it with CALL or COMMAND/C, but you'll still need a separate batch file for each node you wish to run. However, all Max tasks can be run out of the same directory, so you can run everything out of \MAX.

  Fortunately, you only need one copy of the MAX.PRM file: you can use the \`-nXX' and \`-lX' command line parameters to adjust the task number and log filenames at runtime. However, you DO need to specify a separate log file for each task. Naming the log Line01.Log for task 1, Line02.Log for task 2, and so on is a reasonably way of handling log files. (If you don't want a log file for a certain node, then simply use the \`-l' command line parameter without specifying a filename.)

  Even if you use the same .PRM file for all tasks, you can still display node-specific files to the user through the use of the \`*' token.  When display a .BBS file, \`*' translates into the current two-digit task number, zero-padded and in hexadecimal. For example, if you specified \`D:\Max\Misc\Welcom*' as the welcome file in MAX.CTL, Maximus would display WELCOM01.BBS for task 1, WELCOME02.BBS for task 2, and so on.

Advanced users: Actually, it's possible to run multiple copies of Max with only one batch file. If you set an environment variable to equal the current node number, you can use that variable as a replaceable parameters in a single batch file. However, this requires a bit of knowledge about DOS, so it is not recommended for normal users.

- When setting up your batch files, you should make sure that ALL copies of Maximus are started from the same directory. This will allow you to share some files between nodes, in addition to providing a cleaner directory structure.

- If you are part of FidoNet, you may want to run a mailer on one line only. Fortunately, the internal WFC module can be used on a node-by-node basis with the same set of control files. For more information on using WFC, please see the section of the installation entitled "Support for Remote Callers".

- When looking for a compatible FOSSIL, it may take a bit of work to find one that runs correctly under your network software. If you are having mysterious communications problems, then try switching to a different FOSSIL. There are at least three different types for the IBM PC, so you should have no problem finding one which works with your hardware.

- In your AUTOEXEC.BAT, you may wish to include commands to delete ACTIVE*.BBS and UTASK*.* from the main Maximus directory, and IPC*.BBS from the inter-process communications directory. These are temporary files created by Maximus during execution, so they should not be left lying around. If you need to restart the network while Maximus is running, these files won't get deleted which may cause future problems. To fix this, you should include the above-mentioned delete commands in your AUTOEXEC.BAT to make sure that you start with a clean slate whenever you reboot. (In the case of a network, the delete commands should be placed in the server's AUTOEXEC.BAT. If you are running DESQview or some other multitasker on a single node, then you can also place those statements in your main AUTOEXEC.BAT.)

*   If you wish to use either of the multi-node chat or the paging features, your operating system must support file and record locking. Under DOS, this means that you must load the DOS "SHARE" program. Under OS/2, file locking is built into the operating system, so no special utilities are necessary. Although it is possible to use Maximus in a multi-node environment without loading SHARE (through the `No Share.Exe' option in MAX.CTL), this is strongly discouraged, and no guarantees are made if you don't load SHARE.

*   Make sure that all copies of Max have a unique and NON-ZERO task number. If the task number is set to zero, Maximus will assume that you are running in a single-node environment, and won't bother to check the inter-process communications area. In fact, none of the multi-node features will work if you are using a task number of zero.

Multi-Node Chat Operation

The main way in which Maximus takes advantage of multiple lines is through the integrated multi-node chat and paging facility. These features are much like those found in the commercial PCBoard and TBBS programs and are just as flexible. Users can toggle whether or not they can be paged by others, they can display a list of who is on-line, and they can actually enter into a real-time conversation with other callers.

The first step in configuring the multi-node chat is to enable the `Path IPC' statement in MAX.CTL.  (Make sure to follow the instructions in the `Path IPC' description about installing SHARE.EXE and creating a RAM disk!)

The second step is to edit MENUS.CTL and uncomment the Display_Menu option which calls the CHAT menu. Although you can use a custom MenuFile for the chat section, it is best to leave this for later, and use the built-in `MenuHeader Chat' for now. You can worry about tweaking the cosmetics once everything is running smoothly.

Having changed MENUS.CTL, the only remaining step is to recompile the control files. But before allowing users to call the system, you should first test it yourself, by logging onto two nodes locally. (You'll have to use two different user names, since Maximus will only let one user hog one node at a time.)

Before testing the chat mode itself, enter the Chat Section, and look at the menu display. The table should show the node number which you are logged on to (including your name, and the `(you)' designation), in addition to the same information about the second node. (If there is no display, check to make sure that you have implemented the `Path IPC' keyword, and that it points to a valid drive and directory. Another possibility is that you have forgotten to load SHARE.EXE.)

If the menu display seems to be in order, the next step it to try toggling your chat availability a few times. After your status has been toggled, the table should indicate whether or not you are available for chat, in the `Status' portion of the table. You can also check that the other node was informed of the change, by simply entering the Chat Section on the second node, and looking at the table on that system.

Finally, after you have confirmed that everything else is working, you can enter the multi-node chat itself. To initiate a chat, select the P)age option. Then enter the number of the other node you have logged onto, and wait for the chat request to register. (This should take no longer than about 15 seconds.)

After you have paged the user, you should see a `You are being paged by Joe SysOp (node XX)' message on the other node. This is the canned page message; to modify it, you can either edit the ENGLISH.MAD language file, or you can create a \MAX\MISC\CHATPAGE.MEC file. (For information on the latter, see the section on hardcoded filenames.)

To answer the chat request, simply select the A)nswer Page option and enter the node number of the user who sent the request. This should place you inside chat mode: the other user should see a `User Name joins the conversation' message, which indicates that the other user answered the chat request.

The user who answered the page won't see anything immediately; to find out who is participating in the conversation, you can simply type a `/w' command at the beginning of a line, and Maximus will display the list of callers on the same channel. To list all of the callers on the system, whether or not they re in chat, type '/s'.

Once in chat, users can send messages to each other by simply typing the text that they wish to send. Maximus will automatically word-wrap at the end of lines, and the text will be transmitted one line at a time. If possible, it's best to try typing a few times from each node to make sure that the chat function is working properly.

Once you are finished testing, you can use the `/q' command on each node to exit chat mode. (When a node exits chat, the other nodes participating in the same chat should see a `User Name leaves the conversation' message.)

In addition to the private chat facility (which is what you just tested), Maximus also supports a group chat, or a `virtual CB' chat. The CB chat is useful when you have three or more nodes, and want to have more than two callers in one conference. Maximus supports 255 concurrent `channels', which means that there can be up to 255 separate group conversations going on at the same time. However, the CB chat has no paging ability; it's up to the callers to look at the status screen in the Chat Section, and see which channel everyone else is using.

Aside from the differences in invoking the CB chat, once you get inside the chat mode itself, Maximus will behave just as it does inside the private chat, even using the same commands. For more information on using Max's multi-line chat, please see the chat help file, included in the Maximus distribution package. (Assuming a standard system, the help file is accessible using the `?' command from the chat menu, or through the `/?' command inside chat mode.)

USING CUSTOM MENUS

Maximus allows you to create custom menus with relative ease: simply insert a `MenuFile' command in the appropriate section of MENUS.CTL, and you are done. However, there are several tips and tricks you may find useful when designing custom menus, especially when using fancy ANSI or AVATAR graphics.

*   When using a menu which contains a `[cls]' MECCA token, you'll notice that output from some of the internal commands (such as Version or Statistics) disappears, since the [cls] command in the menu erases it, before it can be seen by the user. The solution for this is to link a `Press_Enter' menu option after the appropriate command, which will cause Maximus to wait until the user presses <enter>, before re-displaying the menu. (For more details, see the section entitled `Linking Menu Options' in the Maximus Technical Reference Manual.) For example, to make Maximus wait after displaying the user's statistics, you might use something like this:

```
          Statistics     Disgrace  "Statistics"
NoDsp     Press_Enter    Disgrace  "S"
```

*   If you are using a custom MenuFile statement in the message or file areas, you should never disable the MenuHeader statement. If all you want to do is to suppress the 'MESSAGE Section' banner and statistics information, use "SilentMenuHeader Message" or "SilentMenuHeader File" instead of the equivalent MenuHeaders. This will cause the appropriate menu processing to take place, but nothing will be displayed on-screen.

     However, if you REALLY need to disable even the SilentMenuHeaders, for whatever reason, you must modify your MenuFile to compensate for this. Due to the flexible way that Maximus handles menus, you need to inform the menu handler that a particular menu represents a message/file area so it can read certain pieces of information from AREA.DAT. Since the MenuHeader statement usually informs Maximus of this, disabling it will make Maximus think that the menu just represents a normal area. The solution for this is to place either the `[message]' and `[file]' MECCA tokens at the top of the custom MenuFile, depending on the type of area (message or file) you want the menu to represent. These tokens must be used before any of the message- specific tokens (such as `[msg_cname]') are used. The [message] or [file] token only needs to be used when a message area is first entered - this means that you can place the [message] or [file] token in the custom HeaderFile as well, although it will work equally well in the MenuFile.

- When designing a custom menu with an input prompt at the bottom, you may have some trouble getting the cursor to stop at the appropriate place. Most text editors automatically insert a carriage return after the last line of the file, and since Maximus reads the entire file, this will cause the cursor to skip down to the next line after the entire file is displayed. There are two solutions to this: the first is to use a text editor that DOESN'T insert a carriage return at the end of the file. The other solution, if you are using a .MEC file to create the menu, is to insert a `[quit]' token where you want the cursor to stop. As soon as Maximus encounters this token, it will stop displaying the file, without displaying an extra carriage return. On the other hand, if you are creating the MenuFile manually, you can insert the compiled equivalent directly into the text, which is `^oQ'.  (Control-O and then a capital letter `Q'.). This has the same affect as would the [quit] token, and this token will caused Maximus to behave in the desired fashion.

WAITING FOR CALLER SUBSYSTEM

Max 2.00 supports an internal Waiting for Caller (WFC) module. This module allows Max to initialize the modem, wait for a call, answer the phone, and pass control to the main BBS. WFC can be used on all nodes of a system, on selected nodes, or on no nodes. Nodes which do NOT use WFC will require an external program to answer the phone, such as BinkleyTerm or FrontDoor.

Starting WFC

The WFC module is activated with the `-w' command line switch. Optionally, the `-p' and `-b' switches can be used to override the starting com port and baud rate. If you specify just `-w', WFC will start up using the com port and baud rate specified in the control file.

Before using WFC, you must make sure that the modem strings in MAX.CTL are configured correctly. The distribution version of Max comes with a modem configuration which supports most Hayes-compatible modems. However, if the WFC module doesn't work out of the box, you may have to fiddle with certain strings (such as the "Answer" and "Init" strings) to make it perform as expected.

In particular, the distribution MAX.CTL defaults to using "manual answer". This means that, instead of telling the modem to automatically answer the phone when it detects a ring, Max will take care of ring checking itself. This means that the phone will only be answered when Max is ready to take a call, which is the preferred method of doing things.

However, this manual phone answering may not be compatible with all systems. If you wish to disable manual answering, change the last part of the Init string to read "S0=1" instead of "S0=0", and comment out the "Answer" string. This will instruct your modem to answer the phone automatically, which may work better on several brands of semi-compatible modems.

Screen Display and SysOp Keys

When WFC starts up, assuming that you are using Video IBM or Video BIOS, you will see four multicolored windows displayed on the screen.

The first window, "Status", gives you the time until the next event, the current modem status, the number of calls made to your system (both today and in total), and the name of the last caller on your system.

The second window, "Modem Responses", displays a scrolling list of past responses from the modem. In this window, Max will print out result codes send from the modem.

The third window, "Current Activity", is a scrolling window which displays log messages as they appear.

The fourth window, "SysOp Keys", contains descriptions for all of the keys which can be pressed while in WFC mode.

Pressing <Alt-K> will start a local copy of the BBS. Maximus will take the phone off-hook and then commence the normal log-on procedure.

Pressing <Alt-J> will start an OS shell. You can do file maintenance, make changes to your batch files, or perform other small changes while in the shell. Type `exit' to return to Maximus. Maximus will take the modem off-hook while you are in the shell.

Pressing <Alt-X> will take the system down. Max will put the phone off-hook, clear the screen, and exit to your batch file with errorlevel 1.

For more information on installing WFC, please see the section in the installation entitled "Supporting Remote Callers".

When using the internal WFC, Max can also handle "external events". External events are used to run a particular program at a given time, usually by exiting to your batch file with an errorlevel. Events are covered in detail in the EVENTS.BBS section of the Maximus Technical Reference Manual.

EXPIRATION/SUBSCRIPTION SYSTEM

Maximus supports a full-fledged user subscription and expiry system. Callers can be set to "expire" based on the current date or time used (in minutes). When a caller expires, Max can optionally demote that caller to a lower priv level, hang up, or delete that user's account.

To access the user subscription system, start up the Max user editor with "max -u". Until you get the hang of Max's subscription system, create a new user for testing purposes. (The "A" key will append a blank record to the end of the user file.)

To create a subscriber, first select the "," key; this allows you to set the "expire by" field. If you want the user's account to expire after a certain date, select "D". If you want the user's account to expire after a certain number of minutes, press "M". If you don't want the user to expire at all, press "N".

Next, press "." to set the expiry action field. If you want Max to hang up and delete that user's account, press "A". If you want Max to demote that user to a lower priv level, press "D" and enter the priv level. If you don't want any action to take place, press "N".

Finally, press "<" to set the expiry date/time. If you selected DATE for the "expire by" field, you can enter the expiry date of the user here. Otherwise, if you selected MINUTES, you can enter the number of on-line minutes to give the specified user.

After setting up the expiry controls for a user, the subscription system is completely self-maintaining. If a user expires, that user's account will be modified accordingly whenever that user logs on again.

In addition, when a user expires due to the current date, the file \MAX\MISC\XPDATE.BBS will be shown. When a user expires due to running out of minutes, the file \MAX\MISC\XPTIME.BBS will be shown.

MULTILINGUAL SUPPORT

Maximus 2.00 includes full-fledged multilingual support. Up to eight different languages can be defined in MAX.CTL, and users can switch to any of these languages at any time. The language files themselves encompass almost everything that Max displays to the user, including prompts, system messages and command keys. A separate language file can be created to use Oui and Non instead of Yes and No; even the keystrokes for various options can be changed.

Language files are divided into two distinct sections. Each language file has a set of strings to be displayed to the USER, and each also has a second set of strings to be displayed to the SYSOP. By default, the SysOp interface always uses the FIRST language file defined in LANGUAGE.CTL, regardless of the language currently in use by the user. This means that the user can be walking through the menus in German, but the SysOp will still be able to read the pop-up menus in English. Max also comes with a second language file, AMERICAN.MAD, which was modified to handle certain American spellings.

Max's multilingual support can be used to define different prompts, menus and custom display files for each individual language. Prompts are all handled by the language file itself, simply by editing the appropriate <langname>.MAD file. However, menus must be specially designed by the SysOp, since a separate set of menus should normally be used for each language. Likewise, most display files should be changed to accommodate each new language.

The principal method of supporting alternate menus and display files is through the "%Y" external program translation character. The "%Y" character translates to the user's current language number, with 0 being the FIRST language defined in MAX.CTL, 1 being the second, and so on. "%Y" can be used in many places, including the "First Menu" option in MAX.CTL, all Display_Menu options, and also as "+Y" in all Display_File commands. The careful placement of the "%Y" token can be used to handle most of Max's multilingual support.

For example, if you had the following language statements in LANGUAGE.CTL:

```
     Language  English
     Language  Sanskrit
```

using a "First Menu MAIN%Y" statement in MAX.CTL would cause "MAIN0" to be displayed for callers who selected the English language, and "MAIN1" would be shown to callers who selected the Sanskrit language.

This methodology can also be applied to display files; you can either use a "Display_File D:\Path\File%Y" to display different physical files, or the "[iflang]" MECCA token can be used within an individual display file to make decisions based on the current language.

By default, Max stores a user's language preference in the user file. However, if you want Max to prompt the user for a new language each time he/she logs on, you can do this by placing the token "[menu_cmd chg_language]" at the top of WELCOME.MEC.

QWK MAIL PACKER

Through the reader menu, Max allows users to download mail for off-line reading. Max supports the popular QWK format, so readers such as Deluxe2, Qmail, SLMR and OFFLINE can all be used to read the downloaded mail packets.

Configuration

The QWK mail packer is largely self-maintaining, since it uses most of the Max configuration files to a full extent. However, you do have to edit READER.CTL at least once to configure it for your system.

The first keyword to change is the "Packet Name" option. You should set this to a name describing your BBS, eight characters or less, with no spaces. Maximus will use this as the base filename when sending QWK mail packets.

You should also configure your phone number properly, and also make sure that you have all of the archivers defined in COMPRESS.CFG.

For more information on configuring the QWK mail packer, please see the section in the installation entitled "Configuring the QWK Mail Packer".

Bulletins, News Files and File Lists

In addition to mail, the QWK format also supports bulletins, news files and new file lists. Max supports these files in an extremely simple manner; anything which exists in the \MAX\OLR directory will be packed up with each mail packet.

The QWK format defines several standard files which will be displayed to the user. To use these features, simply place a file with the specified name(s) in the \MAX\OLR directory.

HELLO — Displayed when the reader first starts up. This is typically the equivalent of your WELCOME.BBS screen. This should be ANSI only; no AVATAR or MECCA codes allowed.

NEWS — Your BBS news file. This is usually available as an option from the QWK reader's main menu. This is normally a flat ASCII with no graphics.

GOODBYE — Displayed when the reader closes the packet from your BBS. This file can include ANSI graphics.

BLT-1.1 — Bulletin file 1. This is usually displayed as an option on the reader's main menu. In this case, the file extension is ".1", but you can use anything from ".1" to ".99" to provide up to 99 different bulletins.

NEWFILES.DAT — This file can contain a new files listing for your BBS. Max won't generate this for you, but it's easy to have your file list generator create a copy of a new files list in the off-line reader directory.

Again, all of these files are optional. However, since Max packs up everything in the \MAX\OLR directory when creating a packet for the remote system, simply placing one of the above files in that directory will cause it to be displayed on the remote side.

Remote Message Packing

The default Max configuration includes an "off-line reader" menu, but mail can also be packed from the regular message menu. All of the QWK mail packing functionality is built into the Browse command; in fact, the "Download" command on the reader menu is a simple macro which invokes the Browse command.

The default Download command passes the options "t/n/p" to Browse. This requests a scan of T)agged areas, N)ew messages, and P)ack in QWK format. Obviously, with the flexibility of the Browse command, many more operations can be performed. A selective download can be performed by using the search function (complete with AND/OR operators), and messages could be packed from only the current area, as opposed to all areas on the system. Since the QWK packer is seamlessly integrated with the rest of the Browse logic, an infinite number of combinations are possible.

When selecting the Pack option from the Browse menu, Max will gather all of the specified messages, print out how many it captured, and find out if the user wants to download them. If so, Max will compress the packet with the user's default archiving program, count to ten (giving the user a chance to abort), and then send the file using the default transfer protocol.

The reader menu also includes an upload option; this allows the user to upload a .REP file (created by one of the off-line readers). This .REP file will be decompressed based on the information given in COMPRESS.CFG, and messages contained within will be tossed to the appropriate message areas.

Local Mail Packing

In addition to remote use, the QWK routines can also be used in local mode. After compressing a packet, Max will simply leave the packed QWK file in the off-line reader directory. (By default, the file will be in the \MAX\OLR\NODExx\ directory, where 'xx' is the current task number). Max normally deletes the QWK file after it's sent, but Max will leave it there for a local session.

In local mode, if you want to "upload" a .REP packet, select the Upload option from the reader menu. If the caller is local, Max will prompt for the path and filename of the .REP packet. Enter the location of the packet (as created by your off-line reader), and Max will then decompress and toss that packet.

Unattended Mail Packing ("Vacation Saver")

In conjunction with the "-j" command-line parameter, local mail packing is an important feature. At predefined intervals, you can set up your batch files to call Max with the "-j" command line switch. This switch can be used to log on as a certain user, execute a download command, log off, and have your batch files copy the created QWK packet to a file area.

By doing this, you can pack mail "in advance" for certain users, or use it to save mail for yourself while you are on vacation. Since the packing process is completely controlled by the keystrokes you specify for the -j switch, almost any type of mail download is possible.

For example, if the keystrokes required to get from the main menu to the reader menu, download a packet and log off were "n;o;d;y;m;g", the following command-line would automatically start up Max, pack mail for the specified user, and then log off.

```
     max "-jJoe User;Y;Password;n;o;d;y;m;g"
```

Note! If your log-on sequence includes a "Press ENTER to continue" prompt, you should use the "|" character where you would normally press the <enter> key.

In addition, you can create multiple lines in your batch file for multiple users, as long as you remember to copy the packet from \MAX\OLR\NODExx into a safe place after each mail pack.

NetMail Messages

Since the QWK format was not designed with NetMail messages in mind, you must follow a special convention when reading and replying to NetMail messages in a QWK reader. When you download messages from your NetMail area, the first line of each message will look like this:

```
From: <addr>
```

where <addr> is the 4D network address of the sender. Since there's no place in the QWK header to store the origination address, Max places this information in the message body instead.

If you wish to create or reply to a NetMail message, Max expects to see a "To: <addr>" line as the FIRST line in the message body. For example, to send a NetMail message to 1:123/456, the first line of your message should look like this:

```
To: 1:123/456
```

Note that the "To:" text will be stripped before the message is written to the Max message base, so your QWK messages will look like normal messages to everyone else.

When replying to a message, there's an easy way to set the destination address; simply quote the original message, then change the "From:" line to a "To:" (after removing any quoting marks). This ensures that the destination address is correct, and Max will make sure that you reply is sent to its intended destination.

MISCELLANEOUS INFORMATION

This chapter is for miscellaneous information which didn't fit anywhere else in this documentation.

Filename Specifications

Wherever you specify a filename, make sure to specify a FULL path, including drive specifier and leading backslash. For speed reasons, Maximus changes the current directory as it executes. This mean that you can never assume anything about the current directory. SILT will try to compensate for this, but cannot do so in all circumstances.

Hard-Coded Filenames

Maximus uses several hard-coded filenames, whose names are not changeable:

<areaname>.DSC: For Squish areas only. This file will be displayed to a user when entering an area, only if that user's lastread pointer is set to zero. This is useful for giving a brief description of a message area, including the topics allowed and other appropriate information. For *.MSG areas, see DESCRIPT.BBS.

<areaname>.SQR: For Squish areas only. This file can be placed in the subdirectory containing the messages for that area. <areaname> is the same as the name for the Squish message database (defined by Matrix, Local, Echomail, or Conference keyword in MSGAREA.CTL). It will be displayed to callers each time they enter a message area. For *.MSG areas, see RULES.BBS.

<areaname>.SQX: For Squish areas only. This file will be displayed to a user who attempts to enter a message in a read-only area.

ACTIVExx.BBS: This file is created by Maximus whenever a user logs onto a given node. 'xx' is the hexadecimal task number of the node in question. This file will be deleted when the user logs off.

ATTRIB.BBS: This is the file displayed to users who press a "?" at the attribute prompt in the full-screen message entry header. This file explains the various attributes available, such as private, crash, hold, and so on. This file is located in the \MAX\MISC directory.

BADUSER.BBS: If a file named BADUSER.BBS resides in the main Maximus directory, Maximus can use it as a screen when a new user logs on, to keep out users with unwanted names. This file is a simple ASCII text file, containing a list of names not to be allowed on the BBS, one to a line. Each name listed in the file will be matched to either the first, last, or the entire name of the user. If Maximus finds a match, then it will try to display a file called BAD_USER.BBS in your miscellaneous directory, and then hang up. This file should be located in the \MAX directory.

BLT-1.1 - BLT-1.99: Bulletins to be displayed to the users of the QWK packer. Please see the section entitled "QWK Mail Packer" for more information. This file should be located in the \MAX\OLR directory.

BROWSE.BBS: The help file for the Browse command. This file should be located in the \MAX\MISC directory.

CHATHELP.BBS: This is the help file which will be displayed inside the multi-node chat. It is located in the Maximus MISC\ directory.

CHATPAGE.BBS: This file is displayed to the user when a chat request is received from another node. Maximus will display the "You are being paged ..." message first, followed by this file. For example, this file could be used to display information on how to access the Answer Page command. CHATPAGE.BBS should be located in the Misc directory.

CHG_SENT.BBS: This is the help file displayed when a user tries to edit a message which has already been sent, packed, or scanned as EchoMail. CHG_SENT.BBS should be located in the Misc directory.

CHG_NO.BBS: This is the help file displayed when a user tries to edit a message which was written by someone else. It is located in the \MAX\MISC directory.

DESCRIPT.BBS: If this file exists in a *.MSG directory, the contents will be displayed to users who enter this area, and whose lastread pointer is set to zero. For Squish areas, see the comments for <areaname>.DSC. To display a file to all users who enter an area (regardless of lastread pointer settings), see also RULES.BBS

EVENTSxx.BBS: The ASCII event control file used by Max's event controller, where 'xx' is the current node number. See the section in the installation entitled "Events and Yelling" for more information, and also see the "EVENT FILE CONFIGURATION" section in the Maximus Technical Reference Manual. These files should be located in the Max root directory.

EVENTSxx.DAT: The compiled version of the Maximus event file, where 'xx' is the current node number. The ASCII event file, EVENTSxx.BBS, is compiled by MAX.EXE at runtime. These files should be located in the Max root directory.

EXCBYTES.BBS: This file will be displayed to a user who attempts to download too many kilobytes in one session. (Maximus will display this after printing "That would exceed your daily download limit.") This file should be located in the \MAX\MISC directory.

EXCRATIO.BBS: This file will be displayed to a user who attempted to download a file that would exceed his/her file download ratio. (Maximus will display this file after printing "That would exceed your download ratio.") This file should be located in the \MAX\MISC directory.

EXCTIME.BBS: This file will be displayed to a user who attempted to download a file that would exceed his/her time limit. (Maximus will display this file after printing "That would exceed your time limit.") This file should be located in the \MAX\MISC directory.

FILES.BBS: This is the name of the ASCII file listing in each file directory. For more information on creating a FILES.BBS, please see the section in the installation entitled "Maintaining File Areas".

FILES.DAT: This is the name of the compiled file name listing in each file directory. The FB utility creates FILES.DAT and several other files from the master FILES.BBS.

FILES.DMP: This is the name of the compiled file description listing in each file directory. The FB utility creates FILES.DMP and several other files from the master FILES.BBS.

FILES.IDX: This is the name of the compiled file index in each file directory. The FB utility creates FILES.IDX and several other files from the master FILES.BBS.

FILE_BAD.BBS: This file is displayed when an uploaded file fails the upload virus check. Please see the MAX.CTL reference for more information on the "Upload Check Virus" keyword. This file should be located in the \MAX\MISC directory.

FILE_OK.BBS: This file is displayed when an uploaded file passes the upload virus check. Please see the MAX.CTL reference for more information on the "Upload Virus Check" keyword. This file should be located in the \MAX\MISC directory.

GOODBYE: This is the name of the file displayed (by the off-line reader) to a QWK user who closes your system's mail packet. This file should be located in the \MAX\OLR directory.

HELLO: This is the name of the file displayed (by the off-line reader) to a QWK user who opens your system's mail packet. This file should be located in the \MAX\OLR directory.

LASTUSxx.BBS: This file is created by Maximus as a record of the last caller on the system. In addition, this file is used by Maximus-specific door programs to obtain information about the user who is currently on-line. This file simply contains a copy of that user's user record, with the "time remaining" and "baud rate" fields filled out appropriately.

MAXFILES.IDX: This is the system-wide file index, as created by FB. This file is used for performing global downloading and upload dupe checking.

MTAG.BBS: This is a binary file used by Maximus to store message area T)agging information for each individual caller. This file is located in the \MAX directory.

NAMES.MAX: Max has a feature similar to FrontDoor's NAMES.FD. If you place a file called NAMES.MAX in your system directory, you can use it as an "alias" file for entering NetMail messages. NAMES.MAX has the following format, one alias definition to a line:

```
     <alias>,<name>,<address> [,<subject>]
```

You can have any number of aliases listed in NAMES.MAX. If Max spots a message addressed to "alias" (which can be done by entering the name directly at the prompt, when doing carbon copies, etc.), the message will be automatically readdressed to "name". <subject> is optional; it can be used to enter a default subject for the message. Example:

```
     jdh,Jesse David Hollington,1:225/1
     adf,Andrew Farmer,1:163/115
     sjd,Scott Dudley,1:249/106
     afx,Areafix,1:106/116,gronk
     jimbo,Jim Jones,1:106/114.5
```

Entering the initials in the left column instructs Max to readdress the message to the appropriate person, using the specified address.

If a "*" is placed at the beginning of an alias definition, that definition can only be used by callers with a priv of SysOp. This may be useful for protecting Areafix and Raid passwords. The NAMES.MAX file should be located in the \MAX directory.

NEWFILES.DAT: This file is displayed to a QWK user (by the off-line mail reader) when that user requests a new file listing from your BBS. Please see the section entitled "QWK Mail Packer" for more information. This file should be located in the \MAX\OLR directory.

NEWS: This is displayed to a QWK user (by the off-line mail reader) when that user requests a news file display from your BBS. Please see the section of the documentation entitled "QWK Mail Packer" for more information. This file should be located in the \MAX\OLR directory.

NOTIN.BBS: When a user yells and the sysop does not respond, Maximus will look for this file in the Maximus MISC\ directory. If it exists, then this file will be displayed. If it does not exist, Maximus will display the standard `Sorry, there's no answer'. (Compare to YELL.BBS.)

RAWDIR.BBS: If a file of this name exists in a file area directory, it will be displayed to the user when s/he attempts a R)aw directory command. It will be displayed AFTER the command is selected from the menu, but BEFORE the `Enter mask:' prompt.

READONLY.BBS: If this file exists in a read-only message area, and a user tries to enter a message, then this file will be displayed, as opposed to the built-in, "canned" message which Maximus normally displays.

RESTARxx.BBS: This file is created by Maximus (where 'xx' is the current node number) when performing an [xtern_erlvl] exit. This file is used to store state information about the current caller, and it was designed to put Maximus back exactly where it was before the xtern_erlvl was performed. This file is deleted by Max after restarting with the -r switch. Also, this file can be used to change some information which is normally not modifiable by external programs, such as the "time user logged on system" field, the current menu, and more. This file is located in the \MAX directory.

RULES.BBS: When placed in a *.MSG directory, Max will display this file to all callers who access this area. For Squish areas, see <areaname>.SQR. To display a file to callers only once, see DISPLAY.BBS and <areaname>.DSC.

TAG_FILE.BBS: This is the help file for the file area Tag command. This file is located in the \MAX\MISC directory.

TAG_MSG.BBS: This is the help file for the message area Tag command. This file is located in the \MAX\MISC directory.

TIMEUP.BBS: This file is displayed to callers when their time limits expire. This message will be displayed after Max prints "TIME LIMIT." This file is located in the \MAX\MISC directory.

TUNES.BBS: This is the file referred to by the 'Uses Tunes' command in MAX.CTL. This filename is not hardcoded, but several utilities look for it in the \MAX directory.

WHY_ANSI.BBS: This is the help file for the "Want ANSI [Y,n,?]" question at log-on. This file is located in the \MAX\MISC directory.

WHY_FB.BBS: This is the help file for the "Goodbye [Y,n,?]" question at log-off. This explains to callers why they should leave feedback to the system operator. This file is located in the \MAX\MISC directory.

WHY_FSED.BBS: This is the help file for the "Want MaxEd [Y,n,?]" question at log-on. This file is located in the \MAX\MISC directory.

WHY_HOT.BBS: This is the help file for the "Want Hotkeys [Y,n,?]" question at log-on. This file is located in the \MAX\MISC directory.

WHY_HU.BBS: This is the help file for the "Goodbye [Y,n,?]" question at log-off. This file is located in the \MAX\MISC directory.

WHY_PC.BBS: This is the help file for the "Want IBM Chars [Y,n,?]" question at log-on. This file is located in the \MAX\MISC directory.

WHY_PVT.BBS: This is the help file for the "Private [Y,n]?" question when entering a message in TTY mode. This file is located in the \MAX\MISC directory.

XPDATE.BBS: This file is displayed when a user's subscription expires due to the current date. This file should be in the \MAX\MISC directory.

XPTIME.BBS: This file is displayed when a user's subscription expires due to that user's number of on-line minutes. This file should be in the \MAX\MISC directory.

YELL.BBS: When yell is turned off and a user tries to yell for the sysop, Maximus will look for this file in the \MAX\MISC directory. If it exists, it will display the file to the user. If it does not exist, Maximus will display the standard `Yell is turned off' message.

INDEX