



# Synchronet System Operator Documentation

## Table of Contents

### [Features](#)

#### Installation

##### [Installation](#)

Windows

NT Services

Unix

Daemon

##### [Getting Started](#)

##### [How to Get Help](#)

##### [About this Manual](#)

#### Configuration

##### [System Configuration](#)

Configuration Wizard (Windows only)

SCFG

##### [Message Options](#)

##### [System Options](#)

##### [Toggle Options](#)

##### [New User Values](#)

##### [Advanced Options](#)

##### [Loadable Modules](#)

##### [System Security Level Values](#)

##### [Expired Account Values](#)

##### [Quick Validation Values](#)

##### [Adding Nodes](#)

##### [Node Configuration](#)

##### [Node Toggle Options](#)

##### [Node Advanced Options](#)

Configuration Files

Filter Files

##### [Creating User Accounts Locally](#)

##### [Creating a Sysop Account](#)

##### [Creating a Guest Account](#)

#### Sysop Commands (Remote / Local)

##### [Sysop Commands](#)

##### [Definition of a Sysop](#)

##### [Multiple Sysops](#)

##### [Local Only Sysop Commands](#)

##### [Remote/Local Sysop Commands](#)

#### User Management

##### [Editing Users Locally](#)

GUI User Editor (Windows)

uedit

##### [Online User Editor](#)

##### [Online User Editor Commands](#)

##### [User Security Fields Explained](#)

#### Security

##### [ARS Security](#)

##### [Introduction to ARS](#)

##### [ARS Keywords and Symbols](#)

##### [ARS General Usage Examples](#)

##### [Nesting ARS Logic](#)

##### [ARS Nested Logic Examples](#)

#### Message Bases

##### [Message Base](#)

##### [Electronic Mail](#)

##### [Adding Message Bases](#)

##### [Creating Message Groups](#)

##### [Main Options](#)

##### [Creating Message Sub-Boards](#)

##### [Main Options](#)

##### [Toggle Options](#)

##### [Network Options](#)

##### [Advanced Options](#)

##### [Importing Messages from Your Previous BBS Software](#)

##### [Posting](#)

[Remote OWK Functions](#)

## **Networking**

[Networking Message Bases](#)

[OWK Networking](#)

[Network Hubs](#)

[Networked Sub-Boards](#)

[Setting up a OWK Node](#)

[Setting up a OWK Hub](#)

[Transferring Files via OWK Network](#)

[FidoNet Networking](#)

[Step-by-Step Setup](#)

[Sending FidoNet NetMail](#)

[Internet Networking](#)

## **Internet Mail Server**

SMTP

POP3

## **File Section**

[File Transfer Section](#)

[Setting up the File Section Options](#)

[Main Options](#)

[Viewable File Types](#)

[Testable File Types](#)

[Download Events](#)

[Extractable File Types](#)

[Compressible File Types](#)

[Transfer Protocols](#)

[Adding File Libraries](#)

[Main Options](#)

[Adding File Directories](#)

[Main Options](#)

[Toggle Options](#)

[Advanced Options](#)

[Adding Files to File Sections](#)

[Creating an Offline Files Directory](#)

[Creating a Sysop Directory](#)

[Creating a User to User Directory](#)

[Creating a Default Upload Directory](#)

[Supporting Blind Batch Uploads](#)

[CD-ROM and other Slow Media Devices](#)

[Using Alternate File Paths](#)

## **FTP Server**

## **Chat Section**

[Chat Section Configuration](#)

[Artificial Chat Gurus](#)

[Multinode Chat Actions](#)

[Multinode Chat Channels](#)

[External Sysop Chat Pagers](#)

Internet Relay Chat

## **External Programs**

[External Programs](#)

[Fixed Events](#)

[Timed Events](#)

[Native \(32-bit\) Program List](#)

[External Editors](#)

[Adding Online Programs](#)

[Supported BBS Drop File Types](#)

[Callback Verifiers](#)

[Installing a New External Program](#)

[Configuring a New External Program](#)

[External Program Setup Examples](#)

[Legend of the Red Dragon \(LORD\)](#)

[Usurper](#)

[Trade Wars 2002](#)

[Barren Realms Elite](#)

[Internet Gateways \(Telnet, RLogin, etc.\)](#)

[Troubleshooting External Programs](#)

## **Text File Section**

[General Text Files](#)

[Main Options](#)

[Adding Text Files to a Section](#)

Telnet/Rlogin Server

Services

JavaScript services

Native Services

Defaults Services

Finger Service

NNTP Service

Gopher Service  
IRC Service

#### Synchronet Utility Reference

[CHKSMB](#) (Checks Message Base for Validity)  
[FIXSMB](#) (Fixes Message Base and Mail Indexes)  
[SMBUTIL](#) (Synchronet Message Base Utility)  
[SMBACTIV](#) (Checks for Active Message Bases)  
[ADDFILES](#) (Importing ASCII File Lists)  
[FILELIST](#) (Creating ASCII File Lists)  
[DUPEFIND](#) (Finds Duplicate Files in Synchronet)  
[DELFILES](#) (Deletes Files from File Base)  
[MLABELS](#) (Mailing Labels Creation Util)  
[QWKNODES](#) (Create QWKnet Users/Route/Node Lists)  
[ALLUSERS](#) (Command Line Bulk User Editor)  
[NODE](#) (Node Display/Control Utility)  
[SLOG](#) (Node/System Statistics Viewer)  
[DSTSEEDIT](#) (Node/System Statistics Editor)  
[TOTALS](#) (Credit Gain/Loss Util for Games)  
[ANS2ASC](#) (ANSI to Ctrl-A Conversion Utility)  
[ASC2ANS](#) (Ctrl-A to ANSI Conversion Utility)

#### Customizing your System

[Menus and Other Text Files](#)  
[Text Colors](#)  
[Node Action Text](#)  
[Trash Can Files](#)  
[Message Variables](#)  
[Message Color Codes](#)  
[SIF Questionnaire File Format](#)  
[GURU.DAT \(Guru Brain\) Modifications](#)

#### Appendix Information

[Appendix A](#) : Command Line Specifier Reference  
[Appendix B](#) : Synchronet (SBBS/SCFG) Command Line Options  
[Appendix C](#) : Synchronet Environment Variables  
[Appendix D](#) : Wait for Call Status Display (**v2 Only**)  
[Appendix E](#) : Node Status Display  
[Appendix F](#) : Log File Line Type Specifiers  
[Appendix G](#) : File Formats (Technical Specs for Developers)

#### Definitions of Commonly Used Terms

[Glossary](#)

#### [Back to Table of Contents](#)

---

#### [Copyright](#) © 2000 by [Rob Swindell](#)

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .

# Synchronet BBS

Multinode Bulletin Board System Software

---

[Back to Table of Contents](#)

## Synchronet Features

### General

- Designed from the ground up as a reliable and feature rich multi-node BBS
- Up to 250 simultaneous nodes with multi-node chat and external program (door) support
- Sysop inter-node control (remote or local):
  - Interrupting - hang up on any node
  - Locking - only sysops can logon a locked node until the lock is removed
  - Rerunning - re-executes BBS, reinitializing configuration, etc.
  - Downing - shutdown a node from another remote node or local console
- Real-time user database:
  - Amazingly complete user edit function
  - Remote changes to online-user data are immediately effective
  - Disk image is always current for enhanced system fault tolerance
  - Linked free-form data file for every user
  - International address and phone number support
- **RIPscrip** and **WIP** graphical user interfaces and mouse support!
- Online, sysop configurable, chatting artificial intelligence engine
  - Multiple personalities available for your users to experience
- Pay BBS features:
  - Time credits, user expiration dates, credits, and 900/976 billing

### Security

- **ARS** (*Access Requirement String*) security provides the most flexible, comprehensive, and user friendly security available in the BBS software industry
- Post/Call ratio, connect rates, time of day, and other user data fields can be elements of any **ARS** security field
- Impenetrable to remote hackers
- 100 security levels with 104 sysop configurable flags per user
- Sysop can allow or disallow users to choose their own passwords
- Sysop can force periodic password changes (uniqueness is also forced)

### Customization

- Programmable Command and Menu Structure:
  - Includes free Baja shell/module compiler
  - Users can select a shell of their choice including emulations of other BBS packages
  - Several shells included with Baja source code
- The most configurable BBS available today:
  - Powerful object oriented menu driven configuration program
  - All menus, text, and colors are sysop configurable without source code
  - New User Questionnaire is available and customizable
  - All standard new user questions can be disabled individually
  - Over 120 [message variables](#) (@-Codes) supported

### Messaging

- Hierarchical bi-level message areas with an unlimited number of messages groups and an unlimited number of sub-boards within each group
- QWK message off-line reader and networking support built-in
  - Off-line file requests
  - Off-line new message scan configuration and pointer adjustment
  - Send *FidoNet* or Internet NetMail from QWK REP packet
  - 32-bit CRC duplicate message checking
  - Multiple compression formats supported
  - File attachments optionally automatically included
- Inter-node message services:
  - Current node status - user online, action, connect rate, and more

User activity - credited downloads, mail activity, logons, and logoffs

Users can list active nodes and send private messages from any prompt  
(even between external programs and the BBS)

Multi-channel multi-node chat with optional channel password protection

Private real-time node to node character by character chat with remote split screen support

Users can send telegrams, single line messages, or enter private chat from any prompt

User information (gender, age, location, etc) can be included in node status display

- Extensive networking capabilities:
  - Internal QWK network hub and node support with file transfer ability
  - FidoNet* EchoMail and NetMail support
  - Send/Receive File Requests from within Synchronet
  - UTI drivers for *PostLink* Networks (RIME, ILink, etc)
  - Sub-boards can force real names
  - Allow/disallow/force private posts and anonymous posts per sub-board
  - Supports color codes for *WWIV*, *PCBoard*, *Wildcat*, and *Celerity*
  - Support for multiple networks of the same or different network technologies with different addresses and tag/origin lines
  - Duplicate message checking using 32-bit CRC for QWK and *Fido* Networks

## File Transfers

- Unlimited external transfer protocols with optional DSZLOG support
- Batch uploads, downloads, and bidirectional file transfers
- Transfer credit system is completely configurable
- File viewing, extraction, and partial downloading of archives
- Content rating, multi-disk numberings, and file upload dates can automatically be added to file descriptions
- **FILE\_ID.DIZ** and **DESC.SDI** files can be automatically imported into extended file descriptions
- Personal user to user file transfers
- Optional batch download file flagging for user convenience
- Support for offline directories
- Multiple sysop file removal/move/edit search criteria
- Directories can be sorted by filename or date, ascending or descending
- Complete CD-ROM (single and multi-disk changer) support
- Optional forced upload directory can be invisible to users
- Optional allowance of blind (unsolicited) batch uploads with automatic DIZ importation
- Download events available for adding files/comments just before download (i.e. CD-ROM)

## External Programs/Doors

- Unlimited number of external programs
- Multi-user or single user programs
- Programs that use DOS I/O (don't use COM port directly) supported
- 12 door file formats supported for an incredible level of compatibility including:  
**DOOR.SYS**, **PCBOARD.SYS**, **USERS.SYS**, **CHAIN.TXT**, **DORINFO#.DEF**, **EXITINFO.BBS**, **SFDOORS.DAT**,  
**TRIBBS.SYS**, **CALLINFO.BBS**, **DOORFILE.SR**, and **UTIDOOR.TXT**
- Supports native **WWIV** color code expansion
- Free Software Development Kit (SDK) available for program development
- Comes with multiplayer Synchronet Blackjack game (with C source code)
- No batch file editing/programming necessary!
- Configurable credit cost on a per program basis
- Separate access requirements to see the program and execute
- Multiple Timed and Fixed Events:
  - Logon and logoff events
  - New user events
  - Upload events (integrity testing, adding of comments, virus scan, etc.)
  - Node and system daily events
  - Timed events with support for forced exclusive execution

## Version 3 Features

### General

- All "nodes" run in a single multi-threaded process
  - Minimal resource requirements and improved performance
  - DOS Windows are automatically opened/closed for external DOS programs/doors
  - It's never been easier to run a 10+ node system (with DOS door support) on one computer
- Professional GUI *Synchronet Control Panel* (SBBSCtrl) to monitor all clients and servers simultaneously
  - Replaces old-style "*Waiting for call*" screen
  - Launch configuration and editor dialogs directly
  - New Sysop Availability toggle button (replaces v2.3 Scroll-Lock function)



- GUI User Editor
- Local and Remote Node Control/Spying
- Optionally plays sysop's choice of **.WAV** file on user connect/disconnect (*Windows only*)

## Internet

- Supports Telnet logins only (no more direct-dial user support)
  - No third-party FOSSIL/Telnet driver (SIO/MODEM, NetModem, COM/IP) required
  - Full FOSSIL/DOS/WWIV/XSDK program/door support (easy setup!)
  - Outbound Telnet module included for external MUDs/BBSs
  - Automatically detects and logs client IP and hostname
  - Optional auto-logon via IP address (requires V exemption)
- Integrated FTP server
  - Supports Anonymous (Guest) logins
  - Virtual directory structure mimics BBS filebase hierarchy
  - User security settings enforced
  - Supports long/mixed-case filenames
  - Dynamically generates index (description) files
  - Supports blind upload and sysop directories
  - QWK/QWKnet packet uploads and downloads
  - Upload description support
  - Transfers included in BBS and user upload/download statistics
  - Time/Credit system enforced
  - Custom login/hello/goodbye messages
  - Optional connect/disconnect sounds (.WAV files)
  - File aliases supported
    - Put links in your FTP root directory to often access files
    - Link to files on your local hard disk or your BBS filebase
    - Use general alias names to point to often-changing versioned filenames
  - Complete access to local file systems for remote sysops
- Integrated Mail (POP3/SMTP) server
  - Check BBS mailbox using standard Internet mail clients (Eudora, Outlook)
  - SPAM Filters supported: RBL, RSS, DUL
  - Configurable alias list
  - Optionally receive mail by user number (1@yourbbs.com)
  - Optional inbound/outbound sounds (.WAV files)
  - Optional DNS/MX record lookup (no SMTP Relay Server required)
- Outbound Telnet/RLogin gateway function to run external MUDs, BBSs, and TWGS
  - Includes any-time ^] menu to see who is on the BBS, send messages, etc.
- IP/hostname filter/trashcan files (ip.can/host.can) for all Internet servers/services

## Online Programs

- 16-bit DOS Door Support
  - Windows 95/98/ME FOSSIL/DOS Interrupt driver (sbbsexec.vxd)
  - Windows NT/2000 FOSSIL/DOS Interrupt driver (sbbsexec.dll)
- 32-bit/Socket Door Support
  - Duplicate socket handle passed in **DOOR32.SYS** drop file (new standard)
  - Duplicate socket handle passed on command line via %H specifier
  - Support for external programs written with 32-bit XSDK (v3+)
- Most popular DOS doors fully supported
  - Users can be ;INTERrupted while running ANY external program
  - Non-exclusive events do not take any nodes off-line
  - No more hassling with baud rates and initialization strings
  - DCDWATCH no longer necessary to support WWIV/DOS externals

## [Back to Top](#)

---

**Copyright © 2006 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Document](#) .



[Back to Table of Contents](#)

## [1.0] - Synchronet Installation Instructions

### Finding the Latest version of Synchronet

You can find the latest released public distribution of Synchronet at <http://www.synchro.net>.

### Versions and Supported Platforms

Synchronet is currently available in three separate major versions:

- **Version 2.3 for DOS and OS/2**  
Supports serial/modem and local logins  
Native 16-bit and 32-bit console mode executables  
Single node per instance (up to 250 total)  
Version-specific features are noted with: **(v2 Only)**
- **Version 3 for Windows**  
Supports TCP/IP logins only  
Native 32-bit DLLs and GUI front-end application  
Up to 250 nodes per instance  
Version-specific features are noted with: **(v3+ Only)**
- **Version 3 for Linux and Unix-like OSes**  
See [sbbsunix.txt](#) for details

All configuration and data files are compatible between version 2.3 and 3.x so you can combine instances of different versions to create a single "hybrid" BBS.

### Upgrading to a Newer Version of Synchronet

Because of the many variances when changing to a different version or revision of Synchronet, this topic is not covered here. When upgrading an existing Synchronet installation to a newer version or revision, you should **ALWAYS** consult the upgrade documentation included with the distribution (usually in the form of an [UPGRADE.TXT](#) file). Upgrade-only distributions are usually made available in a more compact archive form (e.g. **sbup300c.zip**).

### Windows Installation

After downloading the Synchronet distribution file (e.g. **sbbs300c.zip**), you will need to extract the archive file that you downloaded into a temporary directory on your hard drive. You will need [WinZip](#) or an equivalent archive program to uncompress and extract the installation files.

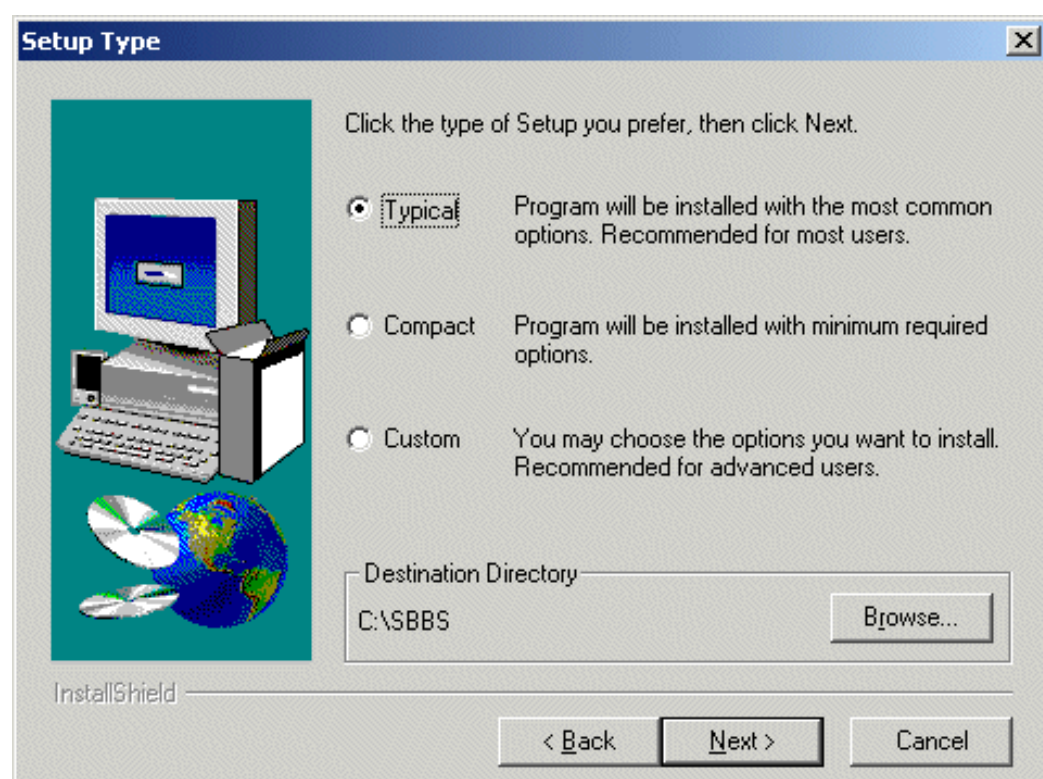
Once you've extracted the installation files, go to the directory you extracted the compressed files to and run **SETUP** to start the installation.

The default installation directory is recommended for most users. If you choose to change the installation directory, it is **critical** that you **DO NOT** use a path with long filenames or spaces as Synchronet still relies on some 16-bit DOS utilities which require DOS-compatible 8.3 file and directory names to function. It is recommended you use a directory named **SBBS** off of the root directory on one of your hard drives (e.g. **C:\SBBS**).

You should now be at the Setup Type Installation menu. Choosing a *Custom* setup from this menu, will allow you to select the items that you want (or don't want) installed. If you are upgrading an existing installation of Synchronet (v2.3 or v3.0), it is advised that you choose *Custom* Setup and disable the following installation components:

- Configuration Files
- Text and Menu Files
- External Programs



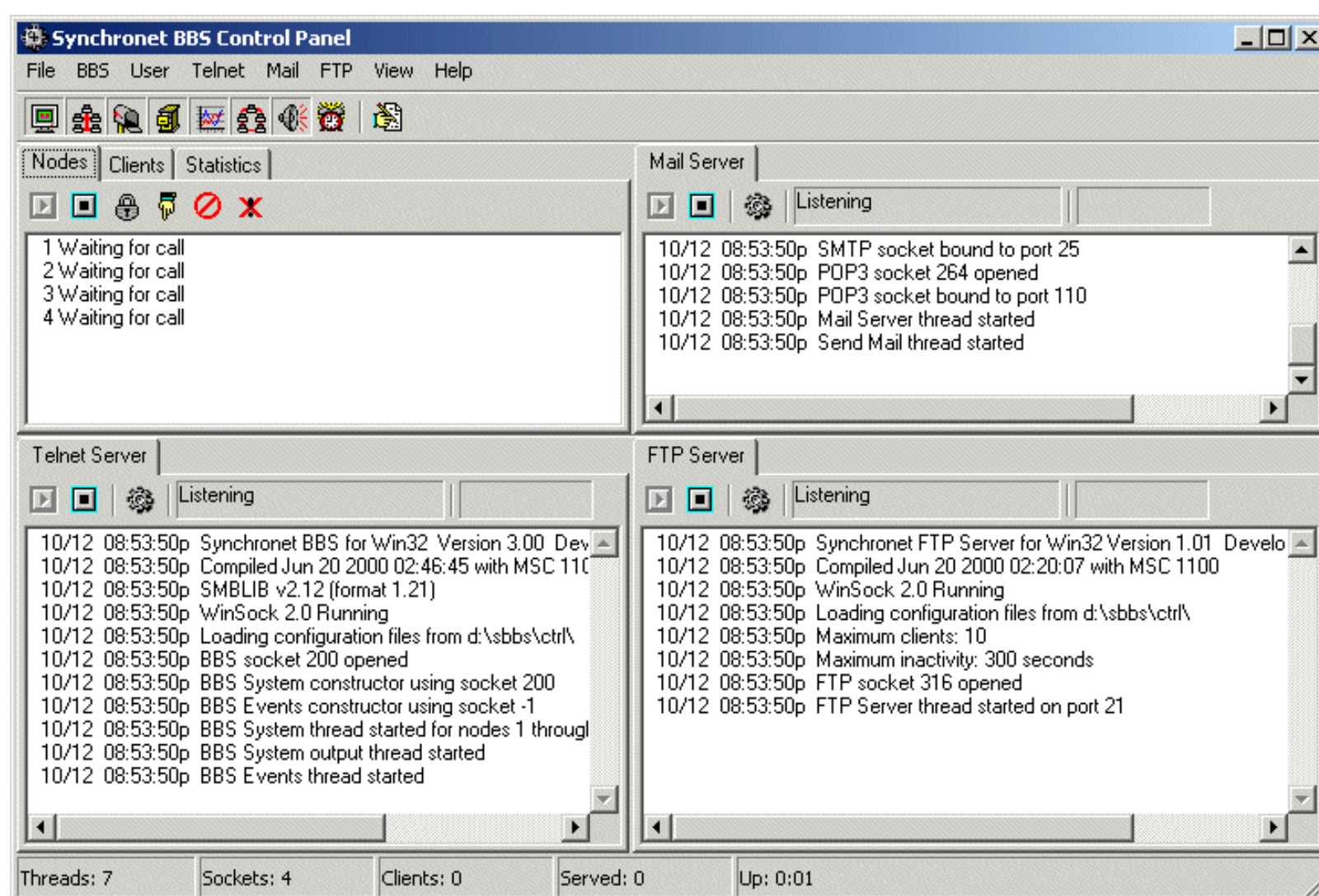


After completing your selection and the installation is completed you will find a new Icon on your Desktop and in your StartUp Folder in your Start Menu. If you do not want Synchronet to start automatically during boot, you may safely delete the Synchronet shortcut from your StartUp Folder.

## [1.1] - Getting Started

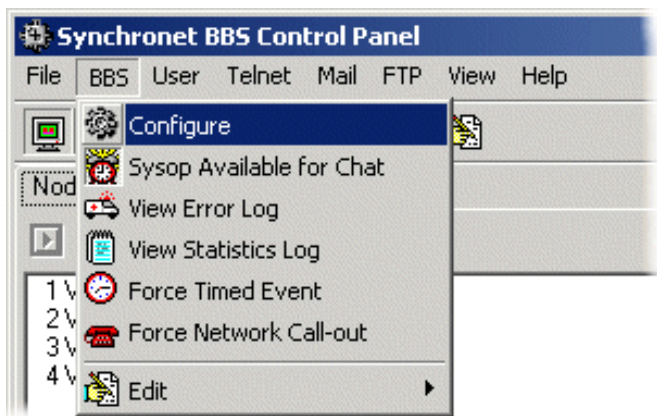
### Getting Started with Synchronet v3 for Windows

Upon Executing Synchronet the screen will look something like the one below.



After starting Synchronet for the first time there will be several pop-up windows that are running programs and events required before the system can be used.

The last window that opens will start the Synchronet Configuration Program ([SCFG](#)). To run SCFG again later you can load it from the *Synchronet Control Panel* (SBBSCtrl) by selecting *BBS* and then *Configure* from the menu bar.



Before putting the system online you should [configure your BBS](#) with [SCFG](#) to set your system options and settings according to your preferences. It is recommended you spend some time reading and becoming familiar with your [configuration](#) and the many options available.

After completing [configuration](#) you should then follow the proper steps to [add a sysop account](#). Once you [create the account](#) you should become familiar with the [Synchronet User Editor](#) and set the [security](#) settings for the account(s) you have created.

Once you have completed the initial setup of your system and sysop account, you have a basic functioning Synchronet BBS System. Please refer to the rest of the [documentation](#) to finish the setup of your new BBS System.

[Back to Top](#)

## [1.2] - How to Get Help

There are a number of avenues that can be used to find help on Synchronet.

This list is but a small one of the available resources:

**Synchronet Homepage** : <http://www.synchro.net>

**Support BBS (Vertrauen)** : <telnet://vert.synchro.net>

**Echomail support:**

FidoNet (zone 1) : **SYNC\_SYSOPS** - Synchronet BBS Software Discussion

For detailed instructions on how and where to get Synchronet technical support, see [support.html](#).

## [1.3] - About this manual

This manual is an ongoing work in progress. As [Synchronet](#) is constantly evolving, so is this manual. This manual aims to give you comprehensive information about installing, configuring, updating, and maintaining your [Synchronet](#) system. There are additional helpful documents that you may find in your Synchronet *docs* directory.

[Back to Top](#)

---

**Copyright © 2006 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Document](#) .

# Synchronet BBS

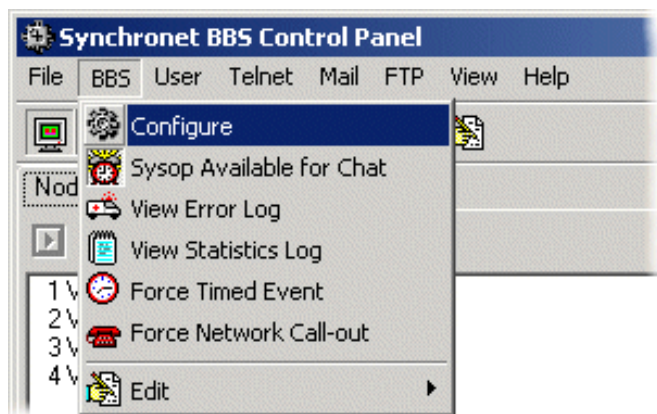
Multinode Bulletin Board System Software

[Back to Table of Contents](#)

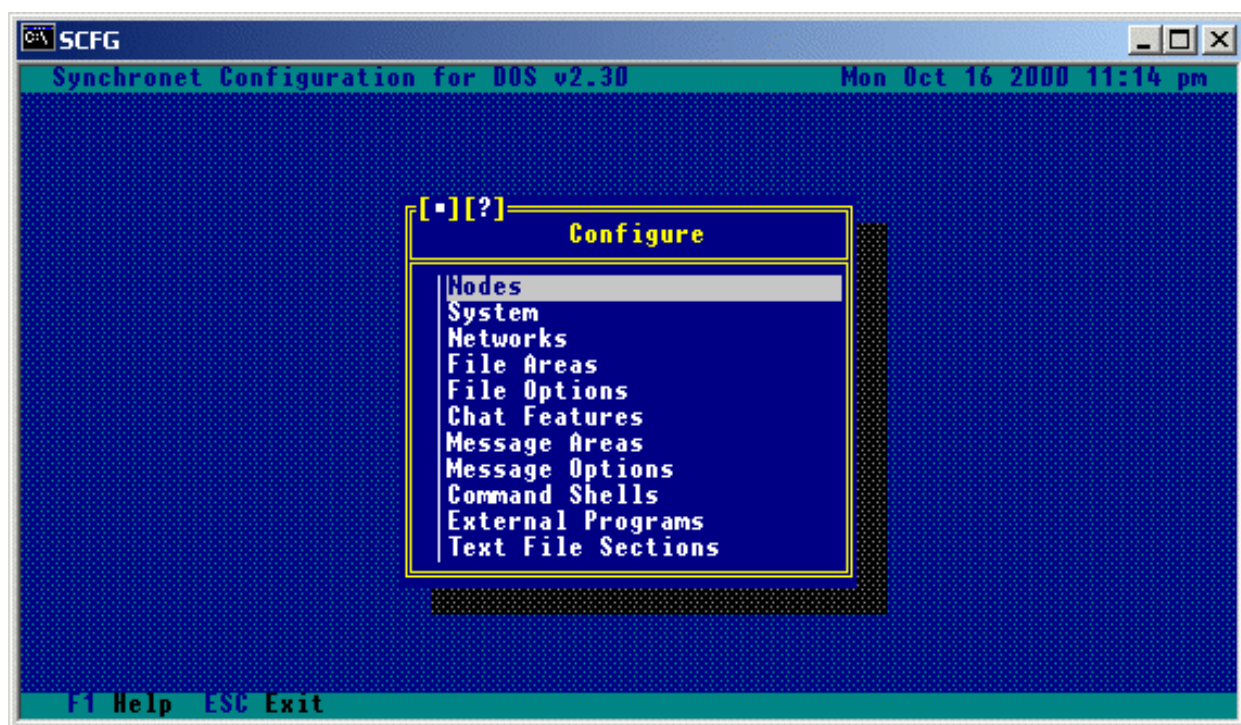
## [2.0] - System Configuration

SCFG is the Synchronet Configuration Utility. It is a stand-alone console mode application (e.g. SCFG.EXE for Windows).

To run SCFG from the Synchronet Control Panel (Windows), select *Configure* from the *BBS* sub-menu on the menu bar.



Once SCFG has initialized, you will see a main menu titled "Configure" which looks something like this:



To manipulate the light-bar, use the *UP* and *DOWN ARROW* keys or type a letter or number in the option you wish to highlight. To highlight the first option of the menu, hit the *HOME* key. To highlight the last option of the menu, hit the *END* key. You'll notice that if you hit *UP ARROW* key when the light-bar is at the first option of the menu, the last option will become highlighted. This wrap effect also occurs when hitting the *DOWN ARROW* key when the light-bar is at the last option of the menu. To select the current highlighted option, hit *ENTER*. To exit SCFG, hit the *ESC* key.

Under "System", set your system's name (BBS Name), the sysop's name (You), the system location (City, State), and system password. The system password you enter here will be required for any remote sysop operations and is prompted for with an "**SY:**" prompt. Under "[Message Options](#)", set your system's QWK ID (Up to 8 character BBS name abbreviation). Under "Toggle Options" set "Allow Aliases" to "Yes" or "No" depending on if you want users to be known by their real names on your BBS or a personal alias.

Remember that at any time within SCFG, you can hit the *F1* key to get online help about the current configuration window. Following is a more specific list of the commands available in the SCFG under System options.

## [2.1] - Message Options





#### BBS ID for QWK Packets:

This is the ID that will be used in QWK packets. It is important that you set this to an abbreviation of your BBS name before users start using the QWK functions of your BBS. Only valid DOS filename characters can be used and the ID must begin with an alphabetic character. This ID will also serve as your system's QWK Network address, should you choose to join a QWK message network (e.g. [DOVE-Net](#)).

#### Local Time Zone:

This should be set to the time zone where your BBS is located.

#### Maximum Retry Time:

This is the maximum number of seconds the Synchronet message base library will wait for a locked message base to become unlocked. Under normal conditions, message bases are only left locked for an extremely short period of time.

#### Maximum QWK Messages:

This is the maximum number of messages which will be packed when a QWK packet is created. If the number of new messages to be packed exceeds this, the packing will be stopped after packing this number of messages. Setting this value to 0 sets the number of messages per packet to unlimited. Private e-mail is not included in the total number of messages and QWK network nodes are automatically exempt from this maximum.

#### Pre-pack QWK Requirements:

If this option is used (not blank), Synchronet will pre-pack any new messages into a QWK packet in the DATA\FILE directory for each user that meets this requirement. When that user calls to download their packet, it will automatically extract the pre-packed QWK (if it exists) and append to it. It is mainly intended for QWKnet nodes that feed from your BBS, but can also be used for long distance users that wish to save connect time and always read their messages via QWK.

#### Purge E-mail by Age:

This will allow electronic mail for your users (stored in DATA\MAIL.\*) to be purged after a specific number of days (read or unread).

#### Purge Deleted E-mail:

This option can be set to "Daily" or "Immediately". If set to "Immediately", e-mail marked for deletion will be physically removed from the e-mail database immediately after the user exits the reading mail menu. If set to "Daily" (the suggested setting), e-mail is marked for deletion isn't physically removed from the database until the automatic daily event executes (sometime after midnight).

#### Duplicate E-mail Checking:

This option allows the setting of a number of message CRCs to be kept to check e-mail to insure that it is not a duplicate of another piece of e-mail. This option should be set to 0 (disabled) unless you specifically want duplicate e-mail disallowed, in which case a value of 2000 should be sufficient.

#### Allow Anonymous E-mail:

Setting this option to Yes will allow users with the 'A' exemption to send anonymous e-mail.

#### Allow Quoting in E-mail:

To allow users to quote from e-mail, this option should be set to Yes.

#### Allow Uploads in E-mail:

To allow users to attach files to e-mail, this option must be set to

Yes.

**Allow Forwarding to NetMail:**

If you allow users to send NetMail on your BBS and want to allow your users to set their account to forwarding their e-mail to a NetMail address, set this option to Yes.

**Kill Read E-mail:**

If you would like to have any e-mail that has been read by the recipient automatically deleted by the message base maintenance program (SMBUTIL) set this option to yes.

**Users Can View Deleted Messages:**

This option allows all users or sysops/sub-ops the ability to view messages (posts/e-mail) marked for deletion (and optionally undelete them) before they are permanently removed from the message base.

**Extra Attribute Codes...:**

This option will give you a sub-menu of toggle options. These options tell Synchronet whether or not it should interpret color codes which are supported by other BBS packages.



**[2.2] - System Options**



**BBS Name:**

This is the name of the BBS. Try to be original. :-)

**Location:**

This is the city, state and/or other pertinent location information.

**Operator:**

This is the name of the system operator. Doesn't have to be the same as user #1, but usually is.

**Password:**

This is the super-secret system password that only sysops should know.

**Users Can Change Password:**

If you want enhanced system security, it is suggested that you force



users to keep the original randomly generated password they were assigned by setting this option to No. If you do allow users to change their passwords, you can make them change their password periodically with this selection.

**Days to Preserve Deleted Users:**

If a user is deleted, his slot will be preserved for this many days since his last logon date. Preserved slots will not be written over by new users. Preserved slots may be undeleted by the sysop.

**Maximum Days of Inactivity:**

If you wish that users that haven't logged on in a certain number of days be automatically deleted, set this value to the maximum number of days a user can be inactive before he is deleted. Users can be exempted from the automatic deletion with the 'P' exemption. Setting this value to 0 disables this feature (Unlimited inactivity).

**New User Password:**

If this field has a value, new users will have to enter this password correctly before being able apply for access.

## [2.3] - System Toggle Options

Selecting this option will bring you to the following sub-menu:



**Allow Aliases:**

If you wish the users of the BBS to be allowed to use aliases publicly, set this option to Yes.

**Allow Time Banking:**

Set this option to Yes to allow users to store their time in a time bank, or to convert their credits to time via the BBS.

**Allow Credit Conversions:**

Setting this option to Yes will allow users to be able to convert credits to time.

**Allow Sysop Logins:**

To be able to perform sysop actions remotely, this option should be set to Yes.

**Echo Passwords Locally:**

If this option is set to No, all BBS passwords will not be displayed locally (characters will be replaced by an X). This includes passwords entered during logon, passwords in the user editor, as well as passwords normally displayed on the status line.

**Short Sysop Page:**

If set to Yes, this sysop page will be a short series of beeps, otherwise a continuous sysop page will be used.

**Sound Alarm on Error:**

If the above option (beep locally) is toggled off, but you still wish to have errors produce an audible alarm, you should toggle this option to Yes.

**Include Sysop in Statistics:**

It is suggested that you set this option to No, so that the sysop's activity on the BBS is not included in the usage statistics.

**Closed to New Users:**

If you wish to disallow access to any new users, set this option to

Yes.

**Use Location In User Lists:**

If you wish the location (City, State) of the user to be displayed in user listings instead of the user note, set this option to Yes. Setting this option to No will display the user's note (sysop created) if one has been created. The option should be set to Yes if using Caller-ID since the user's note may contain their phone number.

**Military (24 hour) Time Format:**

Use this option to toggle your BBS between 12 and 24 hour time formats.

**European Date Format (DD/MM/YY) :**

This option should ONLY be toggle to Yes if you are in a European country which uses this date format. All countries using the MM/DD/YY format should leave this option set to No. If your BBS has been running for any length of time with this option set to No, you should NOT change it to Yes.

**User Expires When Out-of-time:**

Useful for BBSs which charge users based on time, this option, when set to Yes, will set a user to the expired account values once the user's time runs out.

**Display Sys Info During Logon:**

Display "hard-coded" system information during BBS logon process.

**Display Node List During Logon:**

Display list of active BBS nodes during BBS logon process.

**[2.4] - New User Values**



This option allows you to modify the security values assigned to a new user account. You can also set the number of credits and minutes new users start off with. If you have configured an alternate command shell or external editors, you may select one of these as the default for new users. See the [User Editor](#) chapter for more information about user account values.



The 'Default Toggles' are used to set the account defaults that users will have when they log on to the system as a new user. These account defaults can be changed by the user at logon, or while on the system from the user defaults

menu.



The 'Question Toggles' are used to enable/disable the different questions which new users will be asked when logging on to the BBS. The 'Force Unique...' question toggles are used to force users to pick a string that is not being used by any of the other users currently in the system user database.

## [2.5] - Advanced Options



### New User Magic Word:

If this field has a value, it is assumed the sysop would have put some reference to this "magic word" in NEWUSER.MSG and the user will be prompted for this after he enters his own assigned password. If he doesn't enter it correctly, it is assumed he didn't read the text displayed to him and he is disconnected.

### Data Directory:

This is the path to the directory where all the data files for Synchronet will be stored. This value should not be changed unless necessary.

### Logs Directory:

This is the path to the directory where all the logs files for Synchronet will be stored. This value should not be changed unless necessary.

### Exec Directory:

This is the path to the directory where all the executable files for Synchronet are stored. This value should not be changed unless necessary.

### Mods Directory:

This is the path to the directory where sysop-modified modules (.bin and .js files) for Synchronet are stored. Normally, this directory will be empty. If a sysop wishes to modify a stock/standard module and protect the modified file from being over-written during subsequent Synchronet upgrades, the sysop should copy the file from the **Exec** directory to the **Mods** directory where it will take precedence over the file by the same name in the **Exec** directory.

**Input SIF Questionnaire:**

This is the name of a SIF file that resides the text directory that all users will be prompted to answer upon logging on the first time. See SIF for more information.

**Output SIF Questionnaire:**

This is the name of the SIF file that is used by the sysop to view the users' answers to the input SIF questionnaire. If this value is left blank, the input SIF questionnaire is used. This output SIF questionnaire should be identical to the input SIF with the exception of the text content. See SIF for more information.

**Credits Per Dollar:**

This is the monetary value of a credit (How many credits per dollar). This value should be a power of 2 (1, 2, 4, 8, 16, 32, 64, 128, etc.) since credits are usually converted by 100 kilobyte (102400) blocks. To make a dollar worth two megabytes of credits, set this value to 2,097,152 (a megabyte is 1024\*1024 or 1048576).

**Minutes Per 100k Credits:**

This is the value of a minute of time online. Credits can be converted to minutes by the user if allowed by the command shell. Credits are only converted in 100k (102400) blocks. This field is the number of minutes to give the user in exchange for the 100k credit block.

**Maximum Number of Minutes:**

This value is the maximum total number of minutes a user can have. If the user has this number of minutes or more, he will not be allowed to convert credits into minutes. A sysop can add minutes to a user's account regardless of this maximum. If this value is set to 0, the user will have no limit on the total number of minutes he can have.

**Warning Days Till Expire:**

When a users account is about to expire, the BBS will begin sending expiration warning messages to the users notifying them this many days in advance.

**Default Status Line (v2 Only):**

This is the number of the status line that will be displayed by default at the bottom of the screen while the user is online.

**Last Displayable Node:**

This is the number of the last node that will be viewable by the users. Any nodes which exist above this number will be "invisible" nodes and cannot be seen by anyone.

**Phone Number Format:**

This is the format used for phone numbers in your local calling area. Use N for number positions, A for alphabetic, or ! for any character. All other characters will be static in the phone number format. An example for North American phone numbers is *NNN-NNN-NNNN*.

**Sysop Chat Override:**

Any user meeting the criteria set here will be able to page the sysop regardless of the status of the sysop availability.

**User Database Backups:**

Setting this option to anything but 0 will enable automatic daily backups of the user database. This number determines how many backups to keep on disk (i.e. data/user/user.#.dat and data/user/name.#.dat).

**Mail Database Backups:**

Setting this option to anything but 0 will enable automatic daily backups of the mail database. This number determines how many backups to keep on disk (i.e. data/mail.#.\*).

**Control Key Pass-through:**

This value is a 32-bit hexadecimal number. Each set bit represents a control key combination that will **not** be handled internally by Synchronet or by a Global Hot Key Event.

To disable internal handling of the *Ctrl-C* key combination (for example) set this value to **8**. The value is determined by taking 2 to the power of the ASCII value of the control character (Ctrl-A is 1, Ctrl-B is 2, etc.). In the case of Ctrl-C, taking 2 to the power of 3 equals 8.

To pass-through multiple control key combinations, multiple bits must be set (or'd together) to create the necessary value, which may require the use of a hexadecimal calculator.

If unsure, leave this value set to **0**, the default.

## [2.6] - Loadable Modules





The Loadable Modules options are used for loading JavaScript (.js) or Baja (.bin) modules during various stages of the BBS operation. Selecting one of the functions will prompt you for the base filename of the module (excluding the file extension - .js or .bin) to be used when that event occurs. JavaScript (.js) modules take precedence over Baja (.bin) modules.

Following is a brief description of when each of the Loadable Modules occur during the BBS operation:

**Login:**

This module is **REQUIRED** for remote and local logins. Occurs when a user is connected, immediately after the Synchronet copyright notice is displayed.

**Logon Event:**

Occurs immediately after Login (above).

**Sync Event:**

This is a Synchronization Event and occurs each time the BBS performs a node synchronization (e.g. when node messages are received, node status is read, etc...basically continuously while a user is online).

**Logoff Event:**

Occurs **ONLY** when a user does a slow logoff. Does **NOT** occur when users hang up on the BBS or do a fast logoff.

**Logout Event:**

This is an offline event that occurs after a user has disconnected from the BBS, either by logging off or hanging up.

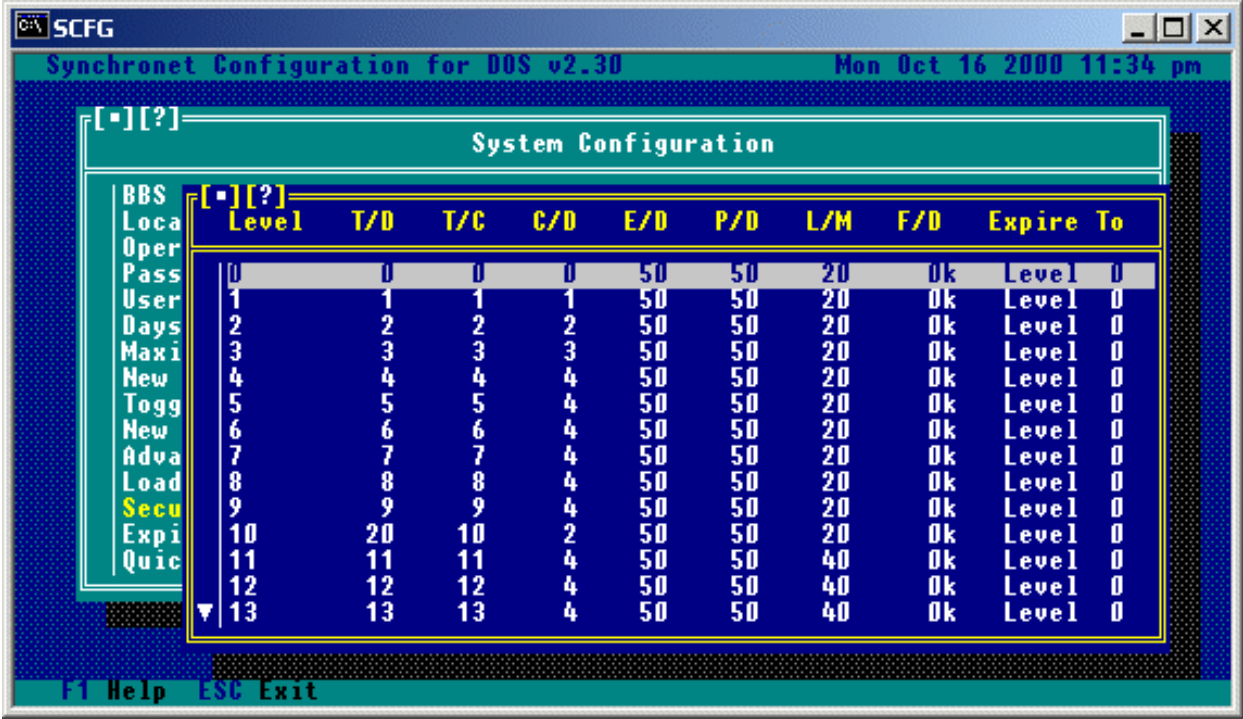
**New User Event:**

Occurs at the end of a new user procedure (e.g. after the user has logged on, left new user feedback, and anything else that is required of a new user).

**Expired User:**

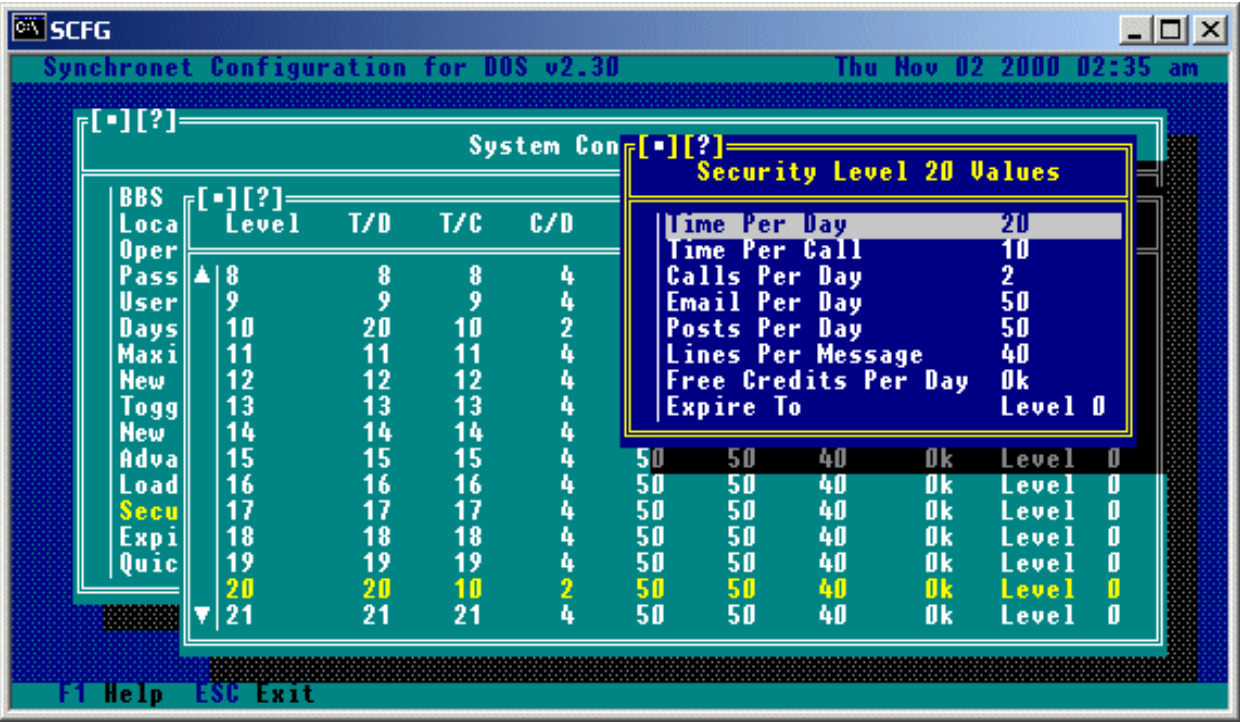
This is an offline event that occurs during Synchronet's internal daily event (after the first caller at the beginning of a new day).

## [2.7] - Security Level Values



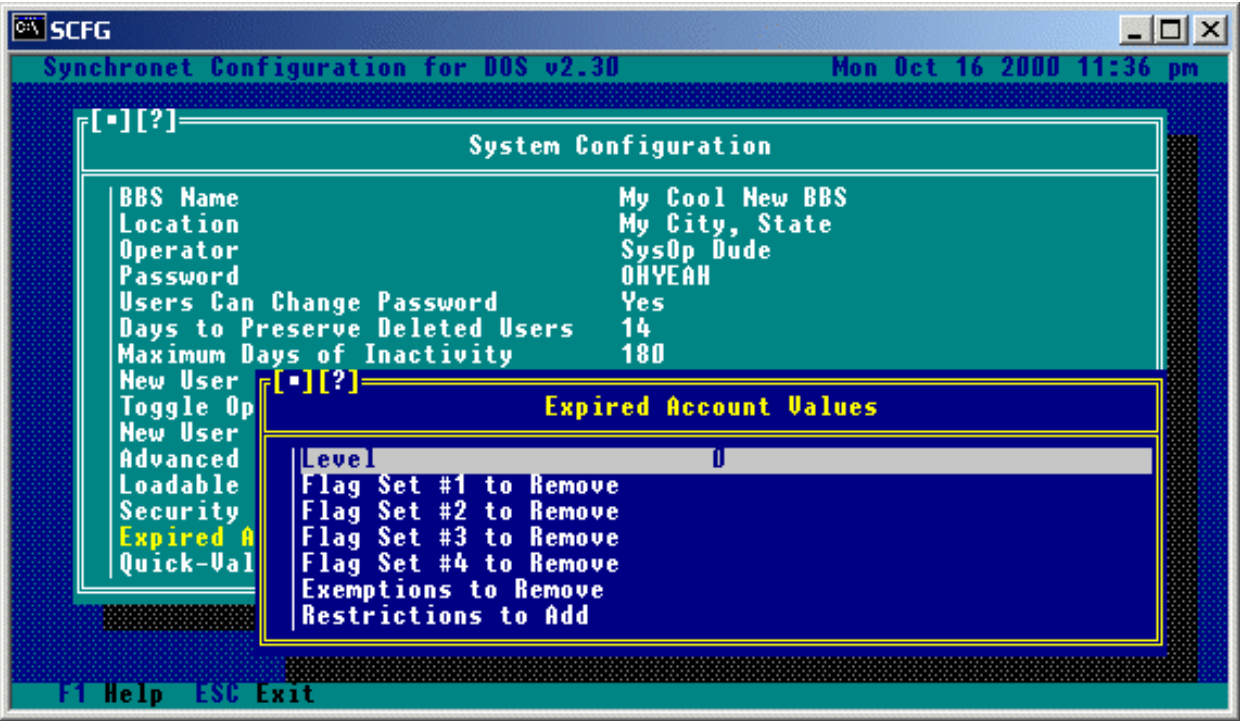


This option allows you to define the operating parameters of the Security Levels (0-99). Selecting the desired security level will bring you to another menu similar to the following:



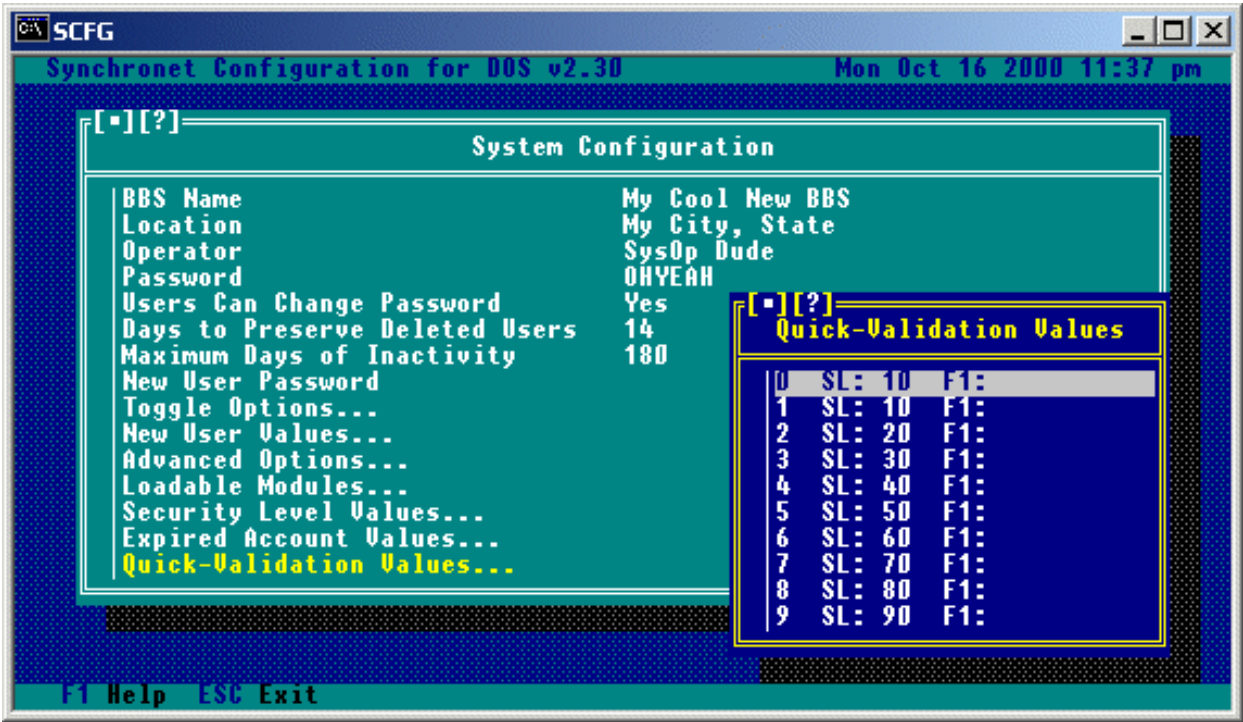
Here you can modify the corresponding values for that particular security level. These values will be given to ALL users who have the selected security level. Many of these values may be overridden by giving certain users the necessary exemptions (if required). For example, the 'T' exemption will exempt a user from any *time* limitations.

## [2.8] - Expired Account Values

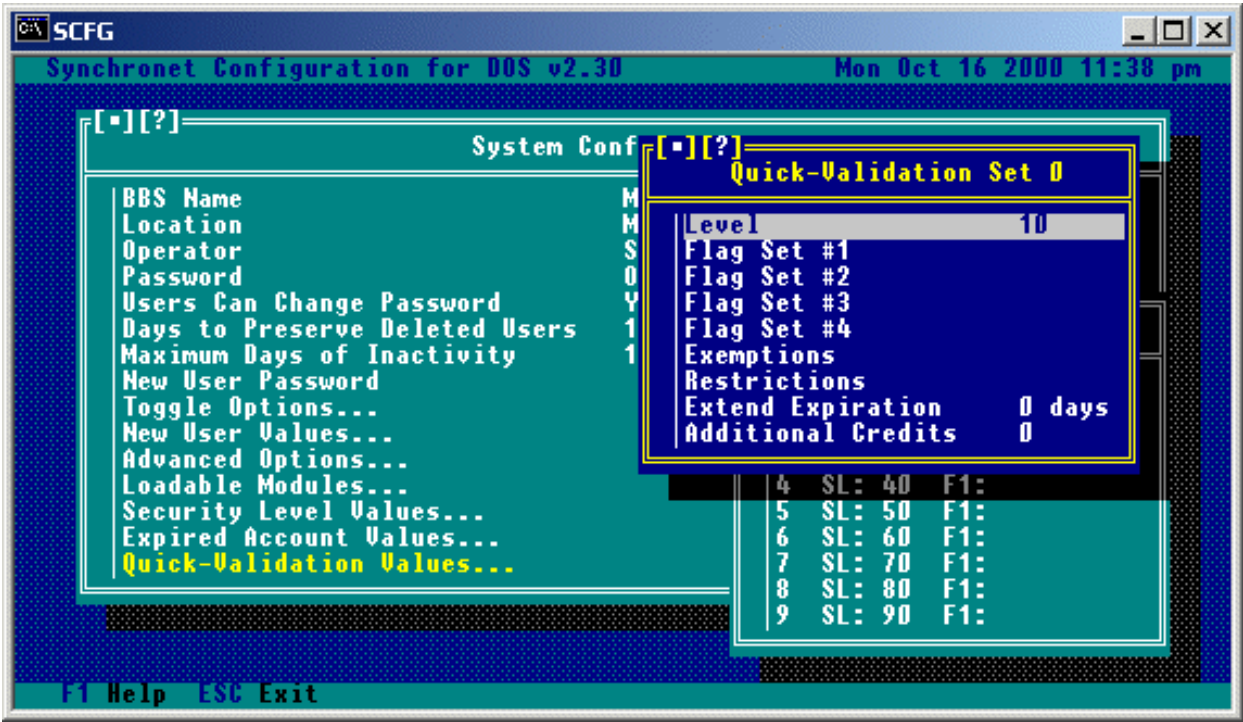


This sub-menu contains the account values for expired user accounts. When a user account expires, the information contained here will be applied to that user account.

## [2.9] - Quick Validation Values



This is where you define the values of the quick-validation sets (0-9). Selecting a level from this list will bring you to another menu similar to the following:



These are used to quickly set a user's Level, Flags, Restrictions and Exemptions, as well as extend their expiration date and add credits to their account. A user can be quick-validated with the 'V' command from the Telnet/RLogin-based User Editor.

## [2.10] - Creating User Accounts

Users may be created using one of the available user editors or by logging on as "new" via Telnet or RLogin.

### [2.10.1] - Creating a Sysop Account

The first user account created (via Telnet or RLogin) will automatically be given Sysop-level access (e.g. Security level 90, all flags and all exemptions). For this reason, the System Password is prompted for (with the **SY:** prompt) during the first new user creation process.

The sysop account is traditionally the first user record in the database (i.e. user #1). This user account **should not** have the alias of Sysop - use your real name or a handle instead. Synchronet automatically treats the alias of Sysop special and forwards any mail received for "Sysop" to user #1.

### [2.10.2] - Creating a Guest Account

During the default Telnet/RLogin logon process, the sysop user account will be prompted whether to create a "Guest" account or not (via the exec/makeguest.js module). It is recommended that you answer "Yes" to create a Guest account. This will allow anonymous FTP logins as well as "read only" access to your other servers and services which encourage users to experience your BBS without requiring them to create a user account first.

The sysop account's Security Flag 4G is used by the default logon module (exec/logon.js) to determine whether or not to prompt the sysop to create the Guest account. If you answered "No" to the create Guest and "Ask again later?" prompts, you can either run the makeguest.js module manually, or restore the 4G

security flag to the sysop account to be prompted during the next logon.

[Back to Top](#)

---

**[Copyright](#) © 2006 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Document](#) .

# Synchronet BBS

Multinode Bulletin Board System Software

---

[Back to Table of Contents](#)

## [5.1] - Sysop Definition

There is usually one sysop (System Operator) per BBS - the owner and operator of the hardware which Synchronet is running on and accepting remote users. Quite often systems require more than one sysop and Synchronet is flexible in this respect. Within Synchronet, a sysop is defined as any user with a main level of 90 or above. Any user that qualifies as a sysop, is prompted with an "SY:" to enter the system password at logon. Any time a sysop attempts to perform a sysop function that may breach the system security, he is prompted with an "SY:" to enter the system password before he can proceed. Users with sysop access can cause serious damage to the system. Give out this ability with caution.

## [5.2] - Multiple Sysops

As stated above, any user with a security level of 90 or greater can perform sysop functions, "So why have levels 91-99?" you might ask yourself. Well, you can still restrict a user with sysop access from accessing Message Groups or Sub-boards or File Libraries or Directories by setting the minimum required level above that user's level. For example: If you have a remote sysop with a security level of 90, you can still have a Message Group or Sub-board with a minimum required level of 91. Now this remote sysop will not be able to access that Group or Sub-board. While it is true that sysops can edit their own and subordinate user's accounts, they can not edit a sysop of higher level's account (when calling remotely). A remote sysop can't raise any user's level above his own, or add flags or exemptions that he himself doesn't have.

## [5.3] - Local Sysop Commands *(v2 Only)*

Most sysop functions will be performed locally, while some sysop functions can only be performed locally. The following special key combination are only available while online locally (not from the Waiting for Call screen).

### Macros:

All the function keys (F1-F12) and combinations with Shift, Ctrl, and Alt (48 macros total) are available for the sysop. A macro is a short way to enter many keystrokes. Each macro is stored in a separate file in the TEXT directory with a MAC extension. The name reflects which key combination will initiate the macro. For regular F1 through F12, the name is F1.MAC through F12.MAC respectively. For Alt-F1 through Alt-F12 the name is ALT-F1.MAC through ALT-F12, Ctrl-F1 through Ctrl-F12 are named CTRL-F1.MAC through CTRL-F12, and Shift-F1 through Shift-F12 are named SHFT-F1.MAC through SHFT-F12. Each file is a basic ASCII text file with one exception; an extended key combination (such as Alt-D) can be entered by placing a Ctrl-@ (ASCII 0, NULL) in the file followed by the scan code of the key you wish to use.

### Alt-key combinations:

#### **Alt-U (User Edit)**

At any place within Synchronet, pressing Alt-U at the local keyboard will pop the local console into User Edit while the remote side (if there is a remote user) will see [WAIT]. Alt-U is available while using the terminal mode as well. See User Edit for more information. See UEDIT command below.

#### **Alt-# (Quick Validation)**

Pressing Alt and then any of the numbers 0-9 will change the current users level/flags/exemptions/restrictions/credits and extend their expiration date to the settings for that quick-validation slot set in the system configuration. SCFG->System->Toggle Options->Quick Validation Hot Keys must be set to 'Yes' for this feature to work.

#### **Alt-S (Crazy Cursor Toggle)**

Pressing Alt-S will toggle the state of the crazy (spinning) cursor, but doesn't change the current user's default settings regarding it.

#### **Alt-E (Local I/O Only)**

If a user is on remotely, this allows the local console to use the BBS while the remote user just sees [WAIT]. Pressing Alt-E again re-enables the remote console.

**Alt-L (Capture)**

This starts and stops the local capture of text to a file. If capture is active the status line will reflect this state with a blinking 'C' as the far left character. The file the BBS opens to capture to will be appended if the file already exists. The filename to capture to is prompted for when capture is started. The filename defaults to CAPTURE.TXT in the current node's directory. To direct output to the printer, change the filename to PRN.  
See ANSCAP command below.

**Alt-Q (Quiet Remote User)**

If a user is connected remotely, hitting this key will disable him from being able to enter any keystrokes into the BBS. The user will see all the BBS's output, but any characters sent from his side will not be accepted as input. Hitting Alt-Q again enables remote input again.

**Alt-P (Sysop Page Off)**

If someone is currently paging the local sysop (annoying sounds at the local console), hitting Alt-P will turn them off. Scroll-lock toggles local sysop availability.

**Alt-H (Disconnect)**

Hangs up on the current user immediately.

**Alt-I (Interrupt)**

Hangs up on the current user after displaying a message the node has been closed for maintenance.  
See INTR command below.

**Alt-A (Sysop Alert)**

When the current user logs off the BBS, the local sysop will be alerted by a message and a continuous alarm.

**Alt-R (Rerun Node)**

When the current user logs off the BBS, this node will automatically rerun itself.  
See RERUN command below.

**Alt-X (Down Node After User Logs Off)**

Hitting this key combination while the user is in the BBS will toggle the down status of this node. When a node is to be downed, it will be automatically shutdown when the current user logs off.  
See DOWN command below.

**Alt-- (Subtracts 5 Minutes)**

Subtracts 5 minutes from the user's time allowed online this logon.

**Alt++ (Add 5 Minutes)**

Add 5 minutes to the user's time allowed online this logon.

**Alt-N (Lock Node)**

Pressing this key will disallow any non-sysops or users without the 'N' exemption from logging on after the current user logs off.  
See LOCK command below.

**Alt-T (Temp Sysop)**

Gives the current user sysop status temporarily for this logon. Hitting Alt-T again removes the temp sysop status. This doesn't actually raise the security level of the user to 90, but gives him access to all sysop functions. The status line reflects Temp Sysop state with a blinking asterisk to the far left of the status line.

**Alt-G (Guru Chat)**

Pops the user in and out of apparent local chat with the system's guru.

**Alt-C (Local Chat)**

Pops the user into chat with the local console.

**Alt-D (DOS Shell)**

Pops the local console into DOS. If a user is on remotely, he sees [WAIT] till the local console exits the shell. The sysop can use the 'D' command to shell to DOS from the wait for call screen.  
See DOS command below.

**Alt-Z (Local Key Menu)**

Displays the local key menu.

**Status Line:**

The status line displays one line of user information at any given time. To change the current line, use CTRL and the UP and DOWN arrow keys to scroll through different information lines, or CTRL-HOME to go to the default (1st) or CTRL-END to go to the last.



The available status lines are:

```
0: UserName SecLevel Password ModemType Birthday Age Sex Phone
1: UserName SecLevel RealName/CompanyName "Alt-Z for help"
2: UserName SecLevel RealName/CompanyName Age Sex Phone
3: UserName SecLevel Location Phone
4: UserName SecLevel Note/Caller-ID Phone
5: UserName SecLevel Flags1 Age Sex Phone
6: UserName SecLevel Flags1 Expiration
7: UserName SecLevel DateFirstOn DateLastOn Expiration
8: UserName SecLevel Credits Minutes Expiration
9: SecLevel Flags1 Flags2
10: Exemptions Restrictions
11: ComputerType ModemType ChatHandle
12: Address Location ZipCode
13: UploadBytes UploadFiles DownloadBytes DownloadFiles LeechCount
14: Posts EmailSent FeedbackSent EmailWaiting TotalLogons TotalTimeOn
15: NetMailForwardingAddress
16: SysopComment
```

You can set the default status line number in SCFG->System->Advanced options. This will be the default status line that appears until manually scrolled.

There are status flags on the status line that indicate specific status items. They appear blinking on the left or right edges of the status line. They are defined as:

```
C      Local text/ANSI capture is on
*      Temporary sysop status
A      Alert sysop when user logs off
R      Rerun this node when user logs off
D      Down this node when user logs off
E      Run node daily event when user logs off
L      This node is locked for sysop logons only
```

## [5.4] - Extra Online Sysop Commands

Most of the additional commands available to sysops online (local or remote) are initiated from the main or transfer sections. A menu of the available commands can be viewed with the '!' command from the main or transfer sections with the Synchronet Classic command shell. Additional sysop commands are available in many other sections of the BBS and are displayed after the normal menu with the '?' command. All sysop commands from the main or transfer sections of the Synchronet Classic command shell begin with a semicolon (;) character, end with CR (ENTER), and some have optional parameters (shown in square brackets). Users with certain exemptions can use some of these commands and have access to the menus via the '!' command using the Synchronet Classic command shell.

### **UEDIT [x] (User Edit)**

This command initiates the User Edit function. If 'x' is specified and is a valid user number, that user will be the current user when User Edit starts, otherwise the user online will be the current user. This command is also available from the transfer menu. Locally, Alt-U is the preferred method of initiating User Edit. See the chapter on User Edit for more information.

### **SPY [n] (Spy on Another User) (v3.00c+ Only)**

This command allows the sysop to remotely monitor or control another node running in the same instance as the node the sysop is logged into.

### **CHUSER (Change into Another User)**

This command allows the sysop to temporarily change to another user account. If changing to an account with a higher level, the password of that user must be given. Changing to an account of non-sysop status automatically enables Temp Sysop mode so that the sysop can change back to his/her own account before logging off (a suggested action).

### **ANSCAP (Toggle ANSI Capture Mode) (v2 Only)**

Normal ANSI escape sequences are not written to the capture file. The sysop can use this command to enable all characters (including ANSI escape sequences) to be written.

### **LOCK [x] (Lock Node)**

The sysop can use this command to prevent users logging onto a node until the lock is removed. If 'x' is specified, that node's lock status will be toggled, otherwise a list of all nodes is given and the sysop is prompted for the node to lock or unlock. Sysops and users with the 'L' exemption can use this command and log onto locked nodes. A locked node will have an 'L' in parenthesis after the node information in the node listings.

**INTR [x] (Interrupt Node)**

This command allows a sysop (or user with the 'I' exemption) to hang up on a user on another node. If the user is currently executing an external program (editors and transfer protocols included), he will not be disconnected until control returns to Synchronet. A message will be displayed telling the user that the node has been temporarily closed for maintenance before carrier is dropped. If 'x' is specified that node's interrupt status will be toggled, otherwise a list of all nodes is displayed and the sysop is prompted for the node who's interrupt state he wishes to toggle. The current interrupt status of a node is reflected by an 'I' in square brackets following the node information line in the node listings.

**DOWN [x] (Down Node)**

A sysop can toggle the down status of a node with this command. When a node has the down status (noted by the [D] flag on the node status line) it will be shutdown immediately after the user logs off. If a user is not online, it will shutdown immediately.

**ANON (Anonymous)**

This command makes the current node anonymous. Sysops and users with the 'Q' exemption have access to this command. To a sysop, an anonymous node appears normal with the addition of an 'A' in square brackets following the node information line.

**QUIET (Quiet Mode)**

Using this command, a sysop (or user with the 'Q' exemption) can make his node appear to be "Waiting for call" to other users of the system. Users with this ability can also make this state their default state upon logon in the user defaults section. Local users can logon in this state with the WFC command SPACE, then 'Z' or user number 1 can logon fast and quiet with the WFC command SPACE, then 'Q'. To a sysop, a node in this state appears normal with the addition of a 'Q' in square brackets following the node information line.

**RERUN [x] (Rerun Node)**

This command allows a sysop to rerun any node on the system by specifying the node number on the command line. If there is a user on the node that is to be rerun, the node will not rerun until that user logs off. A node that is going to be rerun is noted with an 'R' in square brackets following the node information in node listings.

**DOS (Shell to DOS)**

Using this command, a sysop can go straight to the DOS command line locally or remotely. If the sysop is on locally, ALT-D is the preferred method of shelling to DOS. If a local sysop wishes to shell to DOS from the wait for call screen, he can use the 'D' command.

**EDIT [s] (Edit Text/MSG File)**

A sysop can edit any ASCII text (and MSG format) files on the system with this command. If 's' is given, it will be used as the filename to be opened or created if it doesn't exist. If 's' is not specified, the filename will be prompted for.

**LOG (Today's Detail Log)**

A sysop can view today's detailed log file with this command. This command views the same file as the 'L' wait for call screen command.

**YLOG (Yesterday's Detail Log)**

A sysop can view yesterday's detailed log file with this command. This command views the same file as the 'Y' wait for call screen command.

**NS [x] (Node Statistics)**

This command will give today's statistics for node 'x' if specified, or the current node.

**SS (System Statistics)**

This command will give today's statistics for all nodes combined.

**NLOG [x] (Node Statistics Log)**

A sysop can view the history of statistics information for any node on the system. If 'x' is specified, that will be the node who's statistics are viewed, otherwise the current node. Same as 'N' from the wait for call screen.

**SLOG (System Statistics Log)**

Using this command, a sysop can view the history of statistics information for the system. Same as 'S' from the wait for call screen.

**MEM (Available Memory) (v2 for DOS Only)**

This command displays the amount of memory available to Synchronet and any external programs it executes (without swapping).

**LIST [s] (View Text/MSG File)**

A sysop can use this command to view any ASCII text (or MSG format)

file on the system. If the filename 's' is not specified, it is prompted for.

#### **GURU (View Guru Log)**

A sysop can use this command to view the GURU.LOG file which contains discussions that users have had with the system guru (GURU.DAT). See GURU.DAT for more information.

#### **MAIL (Read All Mail)**

Using this command (or 'M' from the WFC screen) a sysop can read all the e-mail on the system.

#### **BULKMAIL (Send Mass Mailing)**

Use this command to send a single e-mail message to multiple users.

#### **CALL [Hub-ID] (Force QWKnet Call-out)**

Forces a QWKnet call-out to the hub "Hub-ID" the next time the call-out node is at the WFC screen. Example: CALL VERT

#### **EXEC [Command] (Execute a DOS program or Baja module)**

Execute internal DOS commands (DIR, TYPE, etc.) by prepending "command /c" to your command line. Execute Baja modules by prepending "\*" to the command line. Baja modules must be located in your EXEC directory. Command line specifiers (see appendix A) may be used.

#### **CHAT**

Users with the 'C' exemption can use this command to page the sysop.

#### **OLD (Search for Files not Downloaded Since New-Scan Date)**

A sysop can use this command to remove, edit, or move files that were uploaded before the new-scan date (set with the '&P' transfer section command) and have not been downloaded since.

#### **OLDUL (Search for Files Uploaded Before New-Scan Date)**

A sysop can use this command to remove, edit, or move files that were uploaded before the new-scan date (set with the '&P' transfer section command).

#### **CLOSE (Search for Files Currently Open)**

If there are no users online and all nodes are running, there should be no file records open. A sysop can check for open file records with this command. Any file records that are left open erroneously, can be closed with this command. This circumstance should not happen.

#### **ALTUL [x] (Alternate File Path Uploads)**

If there are alternate file paths configured, a sysop can use this command to set the current upload path to one of the alternate file paths. All subsequent upload commands will use the alternate upload path as the storage directory regardless of the storage path specified for the directory being uploaded to. This is a useful command for those who wish to have one directory with files listed from multiple CD-ROMs. See CD-ROM for more information on creating alternate file paths.

#### **UPLOAD (Bulk Local Upload)**

This command searches for files that are on disk, but not in the database of the directory. If any are found, a description is prompted for and the file is then added to the database.

#### **RESORT (Re-Sort and/or Compress)**

A sysop can use this command to re-sort a directory if he has changed the sort order in the directory configuration or can use this command to remove deleted file records from the database to conserve memory and disk space. If there are many deleted files in the database of a directory, performance will also be affected. Deleted file records are used by new uploaded files, but if a substantial amount of files have been removed from a directory, a sysop may wish to use this command.

#### **OFFLINE (Search for Offline Files)**

This command will search the disk for files that are in the database of a directory but not actually on the disk. The sysop can then remove, edit, or move these files to another directory. Offline directories are ignored in this search.

#### **DIR [s] (Directory of Files)**

This command displays a DOS directory of the path specified. If no path is specified, a directory of the current file transfer directory is displayed.

#### **GET [s] (Download File From Anywhere)**

A sysop can download a file from any drive or directory on the system by using this command. The file does not have to be in the file database.

#### **PUT [s] (Upload File to Anywhere)**

A sysop can use this command to upload a file to a local disk without having to add it to the file database.

Note: The Baja source code file for these commands is: **exec/str\_cmds.src**

[Back to Top](#)

---

**[Copyright](#) © 2000 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .

# Synchronet BBS

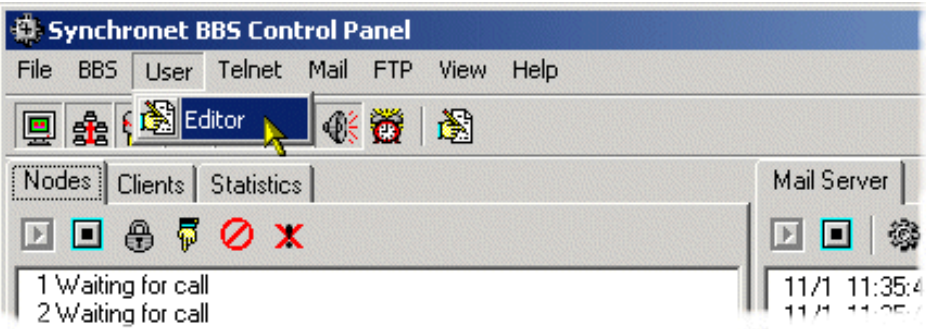
Multinode Bulletin Board System Software

[Back to Table of Contents](#)

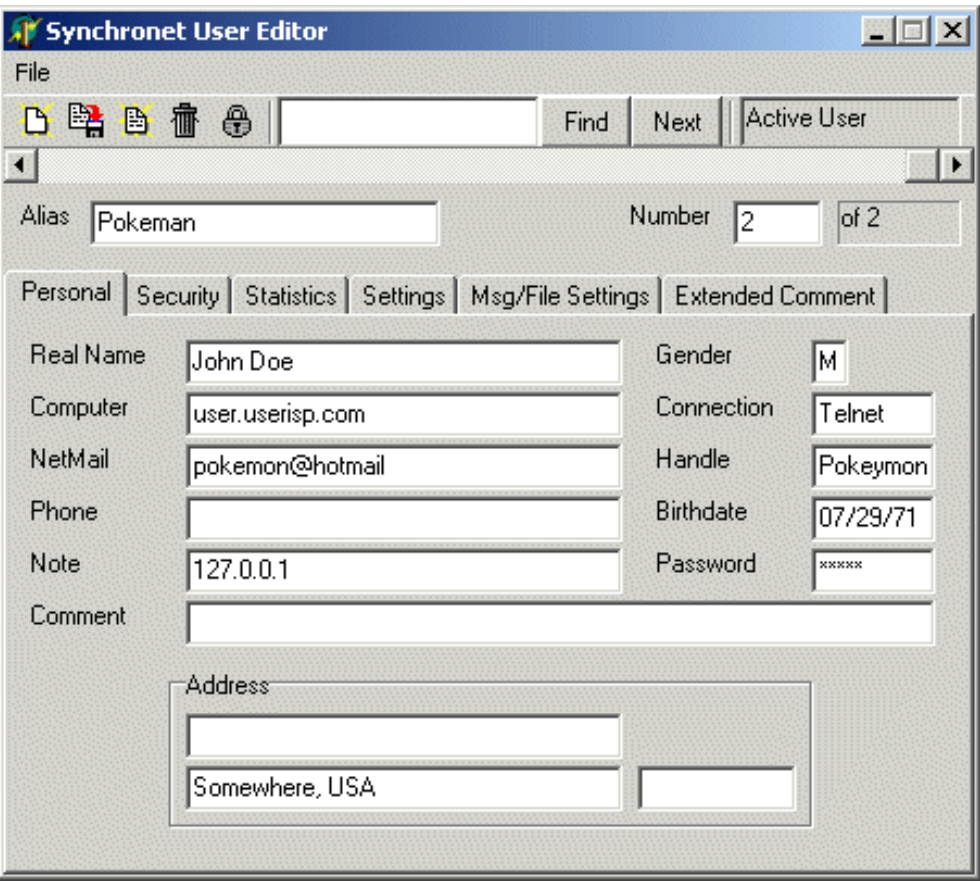
## [6.0] - Editing Users Locally

### Graphical Synchronet User Editor *(Windows Only)*

The Windows graphical user editor application can be executed from the menu bar of the Synchronet Control Panel as shown below.



Below is a screenshot of the GUI User Editor with a sample user and data shown. How this screen looks will vary depending upon your configuration and what data you decide to collect from new users. There are many options and settings available in the User Editor and you should spend some time getting familiar with this feature.



## [6.1] - Online User Editor *(Telnet/RLogin)*

U  
At the Reading Mail or Reading All Mail prompts, hitting 'U' will pop the sysop into User Edit with the author of the previous message as the current user. Hitting 'U' from the Reading Messages (O)perator menu will start User Edit too.

### UEDIT Sysop Command

Typing ';UEDIT' at either the Main or File prompts (when using the Synchronet Classic command shell) will also pop the sysop into User Edit and if a user number or name is specified (e.g. ;UEDIT 20 or ;UEDIT BOB), then that user will be the current user account in the editor.

Other command shells may have different methods of accessing sysop commands from the top-most level menus.

Using the online user editor, a sysop cannot raise any user's security level above his own and cannot give a user any Flags or Exemptions that he/she doesn't have. A remote sysop can not edit any information of any users with a Level higher than his own and their passwords, real names, and phone numbers aren't displayed to him/her.

### Online User Editor Screen Example:



```
Alias      : The Albatross           Password   : MYPASS    10/12/91
Real Name  : Joe Schmoe              Phone number : 714-555-1212
Address    : 404 N.S. West Ln.       Age/Sex/BD  : 31 M 03/08/61
Location   : Irvine, Ca              Zip Code    : 90001
Note       :                         Handle       : J Schmoe
Computer   : 386DX-25 ISA SVGA 4 60 ST506 Modem type : 14.4/V32

First on   : 07/19/91  Expire      : 12/18/95  Last on    : 09/12/92  20:13
Time on    : 2831      Today       : 0    105   Last call : 9    90    Extra : 0
Logons     : 156       Today       : 0    6     Posts    : 39   24    Today : 0
E-mails    : 22        To sysop    : 3        Waiting   : 0        Today : 0

Uploads    :           3,085 bytes in 1 files
Downloads  :       90,389,401 bytes in 282 files
Credits    :       4,007,324 free: 0 (0 per day)
Minutes    :           0

Sec Level  :
Flags #1   :   CD           L N           Flags #3   : AB   E
Flags #2   :           G           Flags #4   :           F H
Exemption  :                   Restricts   :
```

**User Edit Screen Explanation**

The 1st line contains either the alias (if aliases are allowed) or the name of the user, the user's password, and the date the password was last modified.

The 2nd line contains the user's real name and the user's phone number.

The 3rd line contains address of the user, the user's age, sex, and birthday in the form MM/DD/YY or DD/MM/YY if European Date format is enabled.

The 4th line contains the user's location (normally City, State) and Zip (or postal) code.

The 5th line contains a public note about the user and the user's chat handle.

The 6th line contains the user's computer description and modem type.

The 7th line contains the user's private comment - only readable or editable by a sysop. If there is a '\$' after the "Comment" label, then this user has a message file currently attached and is view/editable with the '\$' command. If the comment is blank and there isn't an extended comment, this line is blank.

The 8th line contains the date the user was first on, the user's expiration date, and the date and time the user was last on. Dates are in the form MM/DD/YY or DD/MM/YY if European date format is enabled.

The 9th line contains the total time (in minutes) the user has been online, how much time today, the maximum time he is allowed per day, amount of time on his most recent call, the maximum time he is allowed per call, and any extra time the user may have accumulated through file uploads.

The 10th line contains the total number of logons for this user, the number of logons today, the maximum number of logons this user is allowed per day, total number of messages posted by this user, the user's current post/call ratio, and total posts today.

The 11th line contains the total number of e-mails sent by this user, total number of e-mails sent to user #1, number of e-mails currently in user's mailbox, and the number of e-mails this user has sent today.

The 12th line contains the user's NetMail forwarding address. If the user doesn't have a NetMail forwarding address specified, this line is blank.

The 13th line contains the total number of bytes uploaded by this user and the total number of files.

The 14th line contains the total number of bytes downloaded by this user and the total number of files. If the user has had any suspected leech downloads, the number of leech downloads is displayed here too.

The 15th line contains the number of credits this user currently has. The free credits are given on a daily basis as determined by the sysop.

The 16th line contains the amount of time credits this user has (in minutes).

The 18th line contains the user's Security Level.

The 19th line contains the user's Flag Sets 1 and 2 (if any).

The 20th line contains the user's Flag Sets 3 and 4 (if any).

The 21st line contains the user's Exemption and Restriction Flags that are set (if any).

The 23rd line contains the User Edit command prompt which contains the number of the current user and the total number of user slots on the system.

If the current user is deleted or inactive, then the 1st line will contain a string that specifies this status and all other lines will be shifted down one.

## [6.2] - Online User Editor Commands

As with most Synchronet command prompts, hitting '?' will display a menu of commands.

A	Alias and Chat Handle	S	Sex (Gender)
B	Birthday	T	Set Text Search String
C	Computer	U	Upload/Download Stats
D	Delete/Deactivate/Restore	V	Validate User
E	Exemptions	W	Password
F	Flags	X	View Extended Comment
G	Go to a User	Y	Copy User to Another Slot
H	Edit/Create Extended Comment	Z	Restrictions
I	Inspect/Edit User's Defaults	]	Go Up One User
J	Edit Minutes	[	Go Down One User
K	Edit Dates	}	Search Up
L	Location and Address	{	Search Down
M	Security Level	/	Set ARS Search String
N	Note	~	Set Leech Download Counter
O	Comment	+	Adjust Credits
P	Phone Number	*	Adjust Minutes
Q	Quit to Main Menu	\$	Edit Credits
R	Real Name	#	View User Questionnaire

The available commands are listed below with explanations.

**A - Alias and Chat Handle:**

This command will let you edit the user's alias - a 25 character field that must start with a non numeric character. You can also edit the user's chat handle (an 8 character field) with this command.

**B - Birthday**

You can edit the user's birthday with this command. The form of the date is forced as either MM/DD/YY or DD/MM/YY (depending on the setting of the European date option).

**C - Computer Description:**

You can edit the user's 30 character computer description field with this command.

**D - Delete/Deactivate/Restore/Activate:**

If the current user is at normal status (not deleted or inactive), hitting this command will prompt you if you wish to delete or inactivate this user. If you delete a user, his account will be saved for number of days since his last logon. This number of days is specified by the sysop in the SCFG. After this number of days, a new user may obtain this slot. Inactive slots are the same as deleted ones, except that they are reserved and can't be used by new users.

If the current user is deleted or inactive (as specified by the 1st line of the screen), this command will allow you to restore the user to normal status.

**E - Exemption Flags:**

Exemptions are used by the sysop to give user's extended privilege to the system. For example, you might want to give a trustworthy user the right to remove, move, and edit descriptions of all files in the Transfer Section. Normally users can only remove or edit the description of a file they've uploaded, but a user with the 'R' exemption can move any file, remove any file, or edit any file description in the Transfer Section. This ability is normally only given to sysops.

Using this command you can toggle the state of an Exemption Flag by hitting the corresponding letter. To view a menu of the available exemptions, hit '?' or [ENTER] to stop editing the Exemption Flags.

The following is a list of the available exemptions and their descriptions (\$ indicates that sysops have this capability):

**A - Anonymous**

Allows user to post and send e-mail anonymously.

**C - Chat \$**

Allows user to page the local sysop with the ;CHAT command regardless of the sysop availability state (Scroll-Lock).

**D - Download**

Allows user to download files even when the user doesn't have enough credits.

**E - Expire by Time**

Prevents user account from expiring when the user runs out of time and the SCFG->System->Toggle Options->User Expires When Out of Time is set to YES.

**F - FidoNet Crash/File Request/Return Receipt NetMail**

Allows user to send NetMail using title specifiers:

"CR:" to over-ride default crash status to ON

"FR:" to send file requests (FREQ)

"FA:" to send file attachments

"RR:" to request a return receipt

See Sending FidoNet NetMail for more information.

**G - Multiple Nodes \$**

Allows user to be logged on to more than one node at a time.

This exemption is useful for Guest accounts.

**H - No inactivity**

Allows user to be logged for extended inactive periods without being automatically disconnected.

**I - Interrupt Nodes \$**

Allows user to interrupt (hang up on) other nodes with the ;INTR command.

**J - Chat Channel Cost**

User will not be charged credits to change chat channels.

**L - Logons per day**

Allows user to logon the system an indefinite number of times per day.

**M - Modem Rate \$**

Allows user to logon any node at any modem speed regardless of what the minimum modem speed for that node is.

**N - Node Locking \$**

Allows user to toggle the lock status of any node with the ;LOCK command and log on a locked node.

**P - Permanent**

This user account will not be automatically deleted because of inactivity.

**Q - Quiet/Anonymous Node \$**

Allows user to toggle the quiet or anonymous state of his node with the ;QUIET or ;ANON commands respectively. Nodes that are in quiet mode appear to be "Waiting for call" to the other nodes. Anonymous modes appear to have "UNKNOWN USER" online.

**R - Remove/Move/Edit Descriptions of Files \$**

Allows user to remove, move, or edit the descriptions of any file in the Transfer Section that the user has access to.

**S - Send NetMail Cost**

User will not be charge credits for sending NetMail.

**T - Time Online**

Allows the user to remain online indefinitely - no time limit. The inactivity timer remains active.

**X - External Programs**

User will not be charged credits to run external.

**F - Flags:**

This command lets you toggle the state of all Flags. Hitting a number will allow you to edit that particular flag set. Hitting '?' will display a menu that is used to remind the sysop of the meaning of each Flag. These menus are located in the TEXT\MENU directory and names are FLAGS1.ASC, FLAGS2.ASC, FLAGS3.ASC, and FLAGS4.ASC. Hitting [ENTER] will end the editing of the flags.

The Flags in conjunction with the Level can be used by the sysop to limit access to Message Groups, Sub-boards, External Programs, General Text File Sections, Transfer Libraries, and Directories.

**G - Go to a User:**

This command will let you change the current user into another user by entering the user's name, partial name, or number. It should be noted that the fastest way to change the current user is to type the number of the user directly at the User Edit prompt.

**H - Edit/Create Extended Comment:**

This command will let the sysop edit or view an existing extended comment or create a new one. An extended comment is a free-form message file that a sysop creates that contains information pertaining to this user. A '+' after the "Comment:" string on the 5th line of the User Edit screen specifies that an extended comment already exists.

**I - Inspect or Edit User's Default Settings:**

This command will bring up the current user's default settings for ANSI terminal emulation, screen length, screen pausing, spinning cursor, auto new file scan, clear screen after messages, auto menu display, and default QUIET mode (if the user has the 'Q' exemption).

**J - Edit Minutes:**

This command allows the sysop to edit the current user's minute field.

**K - Edit Date Fields:**

The command will let you edit the date fields for this user's last logon, first logon, expiration date, and password modification date. All dates are forced to the form MM/DD/YY or DD/MM/YY if European date format is enabled.

**L - Location:**

This command will allow the sysop to edit the user's address, location, and zip/postal code fields.

**M - Security Level:**

The sysop can edit the user's Security Level with this command. The user's Level is a decimal number in the range 0 to 99. Higher levels indicate higher access. Users with a level of 90 or higher are considered to be sysops. The sysop can use the Level in conjunction with the Flags to limit access to Message Groups, Sub-boards, General Text File Sections, External Programs, Transfer Libraries, and Directories. The Level also determines the user's maximum time online per day, per call, maximum logons per day, and maximum number of lines per message (post or e-mail).

**N - Note/Name:**

If aliases are allowed on the system, this command will allow the sysop to edit the user's public note. This field is displayed next to the user's alias in user listings and the logon list.

If aliases are not allowed, this command allows the sysop to edit the user's name. The user's name is a 30 character field that must begin with a printable non-numeric character. The first letter of each word in the user's name is forced uppercase.

**O - Sysop Comment:**

This command allows the sysop to edit the private 60 character comment field regarding this user.

**P - Phone Number:**

The sysop can edit the user's phone number.

**Q - Quit User Edit:**

Exits the User Edit function and returns the sysop to where he was prior to initiating User Edit.

**R - Real Name (if aliases are allowed):**

This command allows the sysop to edit the user's real name - a 25 character field. The first letter of each word in this field is forced uppercase.

**S - Sex:**

A single uppercase character is used to describe the sex of the user (normally 'M' or 'F'), but can be changed to any uppercase character with this command.

**T - Set Text Search String:**

Use this command to set the text string to search for (using the { and } commands).

**U - Upload/Download Statistics:**

The sysop can edit the user's upload and download statistics with this command. The statistics are the number of bytes and files uploaded and the number of bytes and files downloaded. These are numeric fields.

**V - Quick-Validate User:**

This command will display the ten quick-validation sets (0 to 9) for quickly setting the user's Security Level, Flags, Exemptions, Restrictions, Expiration Date, and/or Credits. The Quick-Validation sets are set by the sysop in SCFG->System->Quick-Validation Values.

**W - Password:**

Use this command to change the user's password. The password modification date doesn't change when using this command (use 'K' to change the password modification date).

**X - View Extended Comment:**

This command allows the sysop to the user's extended comment if it exists.

**Y - Copy User to Another Slot:**

This command will copy the user data of the current slot to another valid slot. This will overwrite any user information in the destination slot.

**Z - Restriction Flags:**

A sysop can give a user restrictions from certain sections or features



of the BBS by setting one or more valid Restriction Flags. Typing the letter of a restriction will toggle its status, '?' will display a menu of the valid restrictions and [ENTER] will end the flag editing procedure. The valid restrictions and their descriptions follow:

**A - ANSI and Ctrl-A Codes**

User can't write messages containing either ANSI escape sequences or Ctrl-A attribute codes.

**B - Beep**

User can't put beeps (Ctrl-G, ASCII 7) into messages (posts, mail, or chat).

**C - Chat**

User can't use any chat functions.

**D - Download**

User can't download files.

**E - E-mail**

User can't e-mail other users (includes M) (can still e-mail sysop).

**F - Forward Mail**

User can't forward mail.

**G - Edit Defaults**

User can't edit their default settings.

**J - Quoting**

User can't use internal message quoting.

**K - Read Sent Mail**

User can't read/kill sent mail.

**L - Logon Once a Day**

User will only be allowed to logon a maximum of once a day regardless of what his Level allows normally.

**M - Networked Mail**

Can't send NetMail.

**N - Networked Subs**

User can't post on networked sub-boards.

**P - Post**

User can't post messages on Sub-Boards.

**Q - QWK Network Node**

This user account is used for a QWK Network Node BBS system. This restriction automatically puts the user at the QWK prompt upon logon and no other sections of the BBS are accessible. This restriction also allows the user to upload REP packets containing messages that are from users other than the current user.

**R - Remove/Edit Descriptions of Files**

User can't remove or edit file descriptions of files he has uploaded.

**S - E-mail Sysop**

User can't send e-mail (feedback) to user #1.

**T - Transfer Section**

User can't access the Transfer Section at all.

**U - Upload**

User cannot upload files to the Transfer Section.

**W - Auto-message**

User cannot write to auto-message.

**X - External Programs**

User can't access the external program menu ('X' from Main Menu).

**Y - Are you sure (y/N) ?**

The user will be forced to answer Yes prior to performing any function from the Main or Transfer Sections.

**] - Move Up One User Slot:**

Increment the current user number by one. If the current user is the last user, the current user will become user number 1 (wrap).

**[ - Move Down One User Slot:**

Decrement the current user number by one. If the current user is the first user, the current user will become the last user (wrap).

**(Number) - Change to User Number:**

Entering a user number directly at the User Edit command prompt will make that user the current user.

**/ - Set Search String:**

Use this command to set the AR string to search for (using the { and } commands). See the ARS Security chapter for explanation of AR string.

**} - Search Forward:**

Searches from the current user to the last user for the string set by the '/' (ARS) or 'T' (text) command.

**{ - Search Backward:**

Searches from the current user to the first user for the string set by the '/' (ARS) or 'T' (text) command.

**~ - Set Leech Download Counter:**

Every time a user downloads a file and the BBS detects the possible

use of a 'leech' protocol, a counter is incremented and the value is displayed on the Downloaded Bytes and Files line if it is not zero. To change the value of this counter, use the '~' command.

**+ - Add or Subtract Credits:**

This command will let you add or subtract credits to/from a user's account. Putting an 'M' after the number will multiply the amount by one megabyte (1,048,576) or a 'K' will multiply the amount by one kilobyte (1024). Placing a '\$' will multiply with the number by the number of credits per dollar in the current system configuration. Placing a minus sign before the number will perform a subtraction of credits.

**\* - Add or Subtract Minutes:**

This command will let you add or subtract time from the user's minute field. Putting an 'H' after the number will multiply the amount by 60 minutes (an hour). Placing a minus sign before the number will perform a subtraction of minutes.

**\$ - Edit Credits:**

This command allows you to edit the amount of credits of the current user.

**# - View SIF Questionnaire:**

If the sysop has specified a SIF questionnaire in the SCFG and the current user has answered the questionnaire, this command will allow the sysop to view the answered questionnaire. An output SIF (usually an abbreviation of the input SIF) will be used if specified in the SCFG, if not specified the output SIF will be used for viewing.

**[Back to Top](#)**

---

**[Copyright](#) © 2006 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Document](#) .

[Back to Table of Contents](#)

## [7.1] - Introduction to ARS Security

ARS stands for Access Requirement Strings. Access requirement strings are used to specify the requirements of a user to have access to features/sections of a Synchronet BBS. The string can consist entirely of English key words and numbers or use short-hand symbols to fit a large number of security requirements into the 40 character space allowed for access requirement strings. Short-hand symbols and key words may be used interchangeably and combined in the same string. For clarity, it is suggested you use the key words whenever possible. The string syntax is as follows:

**usage:** [not] [parm] [not] [equal] <value> [or] [and] [...]

*where:* **not** is the word "NOT" or the symbol '!' to specify reverse logic  
**parm** is one of any keywords (or short-hand symbols) that specifies a specific required parameter (default is LEVEL)  
**equal** is the word "EQUAL", "EQUALS", the words "EQUAL TO", or the symbol '=' to specify exact equality required  
**value** the required value (for either minimum requirement or equality)  
**or** the word "OR" or the symbol '|' used to specify that ANY of two or more parameter values MAY be met to match the requirement  
**and** the word "AND" or the symbol '&' may be used for clarity in reading when specifying that two or more parameter values MUST be met to match the requirement

## [7.2] - Key words and Symbols

Key word	Symbol Description
AND	& More than one requirement (optional)
NOT	! Logical negation (e.g. NOT EQUAL)
EQUAL	= Equality required
OR	Either of two or more parameters is required
AGE	\$A User's age (years since birthdate, 0-255)
ANSI	\$[ User has ANSI terminal (no value argument)
BPS	\$B User's current connect rate (bps)
CREDIT	\$C User's number of credits in Kilobytes (0-65535)
DAY	\$W Day of the week (Sun, Mon, Tue, etc. or 0-6)
DIR	\$J Current file directory (Internal code or 1-65535)
DOS	Current node is running SBBS4DOS (no value argument)
EXEMPT	\$X Exemption flag (A-Z)
EXPERT	User is in expert menu mode (no value argument)
EXPIRE	\$E Days till user account expires (0-65535)
FILE_CMDS	Number of executed file menu commands (0-65535)
FLAG	\$F User's flag (1-4) and (A-Z)
GROUP	\$M Current message group (1-65535)
LASTON	\$Y Days since last logon (0-65535)
LEVEL	\$L User's level (0-99)
LIB	\$I Current file library (1-65535)
LOCAL	\$G Logged on locally (no value argument)
MAIN_CMDS	Number of executed main menu commands (0-65535)
NODE	\$N Current node (1-250)
OS2	Current node is running SBBS4OS2 (no value argument)
PCR	\$P User's post/call ratio (0-100)
QUIET	User is in quiet mode
RANDOM	\$Q Random number between 0 and value argument (0-65535)
REST	\$Z Restriction flag (A-Z)
RIP	\$* User has RIP terminal (no value argument)
SEX	\$S User's sex/gender (M or F)
SUB	\$H Current message sub-board (Internal code or 1-65535)
SYSOP	User has level 90 or temp sysop (no value argument)
TIME	\$T Time of day (HH:MM, 0-23:59)
TLEFT	\$R User's time left online (minutes, 0-255)
TUSED	\$O User's time online this call (minutes, 0-255)
UDR	\$K User's upload/download ratio (0-100)
UDFR	\$D User's upload/download file ratio (0-100)
USER	\$U User's number (1-xxxx)
WIP	User has WIP compatible terminal (no value argument)
UNIX	BBS is running on a UNIX clone

## [7.3] - ARS General Usage Examples

### General Example #1

LEVEL 60

This string specifies that only users with a level of 60 OR HIGHER meet this requirement. The space between the level and the required value is optional:

LEVEL60

is functionally equivalent, but doesn't read as well. You may also use short hand symbols in place of key words to save space:

\$L 60

is also functionally equivalent. Once again, the space between the short hand symbol and the required value is optional. If a parameter is not specified, the requirement is assumed to be LEVEL. So the following string:

60

is also valid and evaluates the same as the above strings.

### General Example #2

NOT LEVEL 60

This string specifies that only users with a level BELOW 60 will meet the requirement. The "NOT" key word can be include before or after the parameter:

LEVEL NOT 60

evaluates identically to the first string in this example.

Short-hand symbols may be used to abbreviate the string:

LEVEL !60

\$L!60

!60

have exactly the same meaning as the first string in this example.

### General Example #3

LEVEL EQUAL 60

This string specifies that only users with a level EQUAL TO 60 will meet this requirement. The key word "EQUALS" or the combination "EQUAL TO" may be used in place of "EQUAL" if preferred:

LEVEL EQUALS 60

and

LEVEL EQUAL TO 60

are both functionally equivalent to the first string in this example. Again, short-hand symbols may be used to make the string shorter:

LEVEL = 60

evaluates identically to the above example strings and the space on either side of the '=' symbol are also optional:

LEVEL=60

is valid and may be preferred by some sysops. Since level is the default parameter type, the following string is also valid:

=60

### General Example #4

LEVEL 60 AND FLAG 1A

This string indicates that the user must have a level of 60 or higher AND have the flag 'A' from flag set #1. The user must meet both requirements in order to be considered as meeting all the requirements of the string. The "AND" portion of the string is only included for clarity in reading and is optional:

LEVEL 60 FLAG 1A

has the same meaning, but may be confusing when read. When specifying a flag



from flag set #1, the set number may be eliminated:

LEVEL 60 AND FLAG A

is functionally the same and is the preferred method of referring to flags from flag set #1. When specifying the flag set, spaces can be on either side of the number, or eliminated entirely:

LEVEL 60 AND FLAG1 A

and

LEVEL 60 AND FLAG 1 A

and

LEVEL 60 AND FLAG1A

have the same meaning as the first string in this example, but are probably not as easy to read as the second string in this example.

The following strings all have the same meaning as the first string in this example but have been shortened using symbols and other short-cuts:

LEVEL 60 & \$F A

\$L60 AND \$FA

\$L60\$FA

60\$FA

but for reasons of clarity, the first string is preferred if space allows.

#### **General Example #5**

SEX F OR LEVEL 90

This string specifies that the user must be of FEMALE gender OR have a level of 90 or higher. The following strings are functionally equivalent to the above:

SEX F | LEVEL 90

SEXF|LEVEL90

\$SF | \$L90

\$SF|\$L90

#### **General Example #6**

USER NOT EQUAL TO 20

This string specifies that user #20 will not meet the requirement.

Shortened using symbols:

\$U!=20

#### **General Example #7**

BPS 9600 OR NOT TIME 19:00

This string specifies that the user must be connected at 9600bps or higher, OR the time of day must be before 7pm. The value for the TIME parameter is in the format 24 hour format of HH:MM. If the specified time is on an even hour, then ":00" portion may be omitted:

BPS 9600 OR NOT TIME 19

has the same meaning.

You may also omit the "00" from the BPS value to shorten the string:

BPS 96 OR NOT TIME 19

Shortened versions:

\$B 9600 OR NOT \$T19

BPS9600|!TIME19

\$B96|!\$T19

### General Example #8

BPS 9600 OR TIME NOT 18:00 OR TIME 21:30

This string specifies that the user must be connected at 9600bps or higher, OR the time of day must be before 6pm, OR the time of day must be after 9:30pm.

When specifying multiple parameters of the same type consecutively (in this example, two of the required parameters are "TIME"), the parameter does not need to be restated. For example, the string:

BPS 9600 OR TIME NOT 18:00 OR 21:30

has the same meaning as the first string in this example.

Shortened versions:

\$B 9600 OR NOT \$T 18 OR 21:30

\$B96|\$T!18|21:30

### General Example #9

FLAG A OR FLAG B OR FLAG C OR LEVEL 90

The above string specifies that a user must have flag A, flag B, flag C, (all from flag set #1) or a level of 90 or higher to meet the requirement. Using the "sticky" parameter type feature of AR Strings, we can shorten the line:

FLAG A OR B OR C OR LEVEL 90

Utilitizing symbols, we can shorten it even further:

FLAG A|B|C OR LEVEL 90

Taking it to extremes:

\$FA|B|C|\$L90

### General Example #10

USER EQUALS 145 OR LEVEL 90

This string indicates that user number 145 will meet the requirement and all users with level 90 and higher (sysops).

Best shortened as:

USER=145 OR LEVEL 90

For ultimate compression:

\$U=145|\$L90

### General Example #11

LEVEL 60 AND FLAG X AND FLAG Y AND FLAG Z

Compressed examples:

LEVEL 60 AND FLAG X AND Y AND Z

LEVEL 60 AND FLAG X Y Z

LEVEL 60 FLAG XYZ

LEVEL60 FLAGXYZ

\$L60 \$FXYZ

60\$FXYZ

### General Example #12

FLAG 2A OR FLAG 2B OR FLAG 4Z

This string specifies that the user must have either flag 'A or 'B' from flag set #2, or flag 'Z' from flag set #4. Using the "sticky" parameter type feature, this could be shortened to:

FLAG 2A OR B OR FLAG 4Z

If the flag set is not #1, you MUST specify the flag set number when using the FLAG keyword.

Example:

FLAG 2A OR FLAG B OR FLAG 4Z

is NOT the same as the first two strings in this example. The second flag (B) would be considered as from flag set #1, since a flag set number was not specified.

Correct and shortened version:

FLAG2A|B OR FLAG4Z

\$F2A|B|\$F4Z

Note: Multiple flag sets were developed for advanced sysops who found that the 26 flag limitation (A-Z) was insufficient and requested additional flag capabilities. Most sysops will probably not find it necessary to use more than 26 flags, and therefore have no need for flag sets 2 through 4.

**General Example #13**

NOT FLAG 2G

This string specifies that the user must not have flag 'G' from flag set #2. The NOT keyword can appear in several places in this string while keeping the exact same meaning:

FLAG NOT 2G

FLAG 2 NOT G

Condensed versions:

!\$F2G

\$F!2G

\$F2!G

## [7.4] - Nesting Expressions

All of the above string examples have been single expressions that evaluate to either true or false. Sometimes, you may wish to require access to an area using "nested" logic.

Example: If the user is level 90+, OR the user has the A flag, AND is 21+ years of age or older.

The above example could be read two ways.

1: The user must have level 90+, and either the A FLAG or be 21+ years old.

2: The user must be 21+ years old and have either the A FLAG or level 90+.

To clarify which of the above methods were intended, we use parentheses to separate the two true/false expressions.

1: LEVEL 90 OR (FLAG A AND AGE 21)

2: (LEVEL 90 OR FLAG A) AND AGE 21

Strings 1 and 2 are interpreted differently and correlate to the above two English logic statements.

Whenever using a combination of AND and OR logic, nested expressions are required. The following string:

LEVEL 90 OR FLAG A AND AGE 21

is INVALID and would most likely NOT produce the desired results.

Expressions may be nested, using parenthesis, as deep as is required.

Example:

((LEVEL 80 OR FLAG S) AND AGE 18) OR LEVEL 90

is a VALID string that would evaluate with the following:

User with level of 80 and age 17 would not qualify  
User with level of 80, flag S and age 17 would not qualify  
User with level of 80 and age 18 would qualify

User with flag S and age 18 would qualify  
Any user with level 90+ would qualify regardless of age or flags

## [7.5] - ARS Nested Logic Examples

### Nested Example #1

BPS 9600 OR (BPS 2400 AND TIME NOT 15:00)

User must be connected at 9600bps or higher, OR be connected at 2400bps or higher and the time of day being before 3pm. 9600bps and higher callers would always meet this requirement. 2400-7200bps callers will only meet this requirement between 12:00 midnight and 3pm in the afternoon. 1200bps and slower callers would never meet this requirement.

Note, that when nesting expressions, the parameter type DOES NOT stick from one expression to another. Example:

BPS 9600 OR (2400 AND TIME NOT 15:00)

Is an INVALID string. The 2400 would be interpreted as a LEVEL requirement since it is as the beginning of a new expression and LEVEL is the default parameter type.

The first (correct) string in this example, can of course also be shortened using symbols (and eliminating the AND key word):

\$B9600|(\$B2400\$T!15)

### Nested Example #2

(SEX M AND AGE 21) OR (SEX F AND AGE 18)

This string specifies that only male users of 21 years and older and female users of 18 and older will meet the requirement.

Shortened Example:

(\$SM\$A21)|(\$SF\$A18)

The logic of the above string is more verbose than necessary and could be reduced to:

AGE 21 OR (SEX F AND AGE 18)

producing the same results.

### Nested Example #3

(BPS 2400 AND PCR 20) OR LEVEL 90

This string would indicate that users of any level will have access if they are connected at 2400bps and have a Post/Call ratio of 20% or higher - OR the users has as least level 90 (sysop).

Utilizing the default parameter type of level, the LEVEL key word can be omitted:

(BPS 2400 AND PCR 20) OR 90

to create a functionally equivalent, though more confusing, AR string.

A better way to shorten the string would be to eliminate the AND keyword and use symbols:

(\$B 2400 \$P 20) | \$L 90

Eliminating the spaces creates an even more compact string:

(\$B2400\$P20)|\$L90

### Nested Example #4

NOT (USER=1 OR USER=20)

All users will meet this requirement accept user #1 and user #20.

The above string could also be stated in a non-nested format with the same results:

NOT USER=1 AND NOT USER=20

though the meaning is not quite as clear, they are functionally the same.



Note: The second method requires 2 bytes less in raw logic storage because it is not nested and will be interpreted slightly faster (by a matter of microseconds). Just in case you were wondering...

#### **Nested Example #5**

```
LEVEL 90 OR (TIME 12:00 AND TIME NOT 18:00)
```

Users with level 90 or higher will always meet this requirement. Users with levels below 90 will only meet this requirement between 12 noon and 6pm.

Re-arranged:

```
(TIME 12:00 AND TIME NOT 18:00) OR LEVEL 90
```

the string has exactly the same meaning. Eliminating the unnecessary occurrence of the second "TIME" parameter shortens the string while keeping it easily readable:

```
(TIME 12:00 AND NOT 18:00) OR LEVEL 90
```

Maximum compression:

```
($T12!18)|90
```

#### **[Back to Top](#)**

---

#### **[Copyright](#) © 2000 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .



[Back to Table of Contents](#)

## [8.0] - Message Base

The Synchronet message base is used for storing messages from a user or sysop to other users or a specific user. The messages are divided into sub-boards where each sub-board has a unique message topic. Sub-boards are grouped together into message groups.

Sysops can control which users have what level of access to each group and sub-board. If a user doesn't have access to a specific group or sub-board, the user won't even know the specific group or sub-board exists.

The following is an example group/sub-board configuration:

Group Name	Sub-boards
Main	Sysop Notices General Discussion Debate Entertainment
Advertisements	
Adult	General Discussion Entertainment Singles Only
Technical	Hardware Help Software Help Programming

The Synchronet configuration program ([SCFG](#)) is used for adding, deleting, and configuring message groups and sub-boards.

## [8.1] - Electronic Mail (E-mail)

Synchronet has a separate private personal mailbox for each user on the system. Users may receive local E-mail, NetMail, and Attached Files in their mailbox. Upon logon, each user is asked if they wish to read any mail in their box. Mail will stay in the user's box until it is deleted. Mail in a user's box may only be viewed or deleted by the receiving user, the sending user, or a sysop.

E-mail is preferred over Private Posts (on a sub-board) for sending private messages to users on the same BBS. Private posts are only useful for sending private messages on message networks that do not support or allow NetMail.

An E-mail file attachment may be uploaded by a user if SCFG->System->Message Options->Allow Uploads in E-mail is set to Yes. When reading E-mail with a file attached, the receiving user will be prompted if he or she wishes to download the attached file at that time. If the user downloads the file, the file is deleted. If the message is deleted without the file being downloaded, the file is deleted as well.

Users with the 'A' exemption may send Anonymous E-mail if SCFG->System->Message Options->Allow Anonymous E-mail is set to Yes.

When replying to another message in E-mail, the original message may be quoted if the SCFG->System->Message Options->Allow Quoting in E-mail is set to Yes.

### [8.2.1] - Internet (SMTP/POP3) Mail Server *(v3+ Only)*

Set-up:

1. Set SCFG->Networks->Internet->Address to your system's Internet address (domain name or static IP).
2. Set SCFG->Networks->Internet->Allow Sending of Netmail to "Yes".
3. Set SCFG->Networks->Internet->Allow File Attachments to "No" (File attachments from the BBS are not supported at this time).
4. Set SCFG->Networks->Internet->Send NetMail using Alias however you wish (Users can receive e-mail to their Alias OR real-name regardless of this setting).

5. Set SBBCTRL->Mail Server->Configure->*DNS Server* to the address of your ISP's Primary DNS server (run **"ipconfig /all"** from the Windows command line to display your ISP's Primary DNS server) OR set *Relay Server* to the address of your ISP's SMTP server (check your e-mail program for your configured SMTP server hostname).

**Notes:**

Users will be able to send Internet e-mail from your BBS now (if you have a customized **TEXT.DAT** make sure you have updated line 62, "*InternetMailing*").

Your users will be able to receive e-mail at your BBS using any of the following ways:

```
First.Last@your.bbs.domain
First_Last@your.bbs.domain
RealFirst.Last@your.bbs.domain
UserNumber@your.bbs.domain  (if RX By User Number is toggled on)
"First Last"@your.bbs.domain
"RealFirst Last"@you.bbs.domain
```

E-mail user names are not case sensitive (e.g. Digital.Man and digital.man are both valid user names). In order for users to receive mail via user number, that option must be checked in SBBCTRL->Mail->Configure.

E-mail sent to **"SYSOP"**, **"POSTMASTER"**, your BBS's QWKID, or the sysop name (as configured in SCFG) will automatically go to user #1 on the BBS.

Additional aliases can be configured by creating the text file **ctrl/alias.cfg** with the following syntax:

```
-----[ Begin ]-----

; This is a comment
;
; Alias      User Name or number

root      sysop
sysadmin  sysop
webmaster sysop
great.drummer  Rob Swindell

-----[ End ]-----
```

If a user has their account configured to forward their e-mail to an Internet address and they receive an Internet e-mail, it will automatically be forwarded to the configured address.

The sysop of QWKnet nodes can receive e-mail at their QWK-ID @ their hub's Internet domain/ip address. Example: If YOURBBS (QWK-ID) polls VERT for QWKnet, you (the sysop) can receive e-mail at yourbbs@vert.synchro.net.

You (or any of the user's on YOURBBS) can send Internet e-mail from YOURBBS through VERT by sending a QWKnet NetMail message to NETMAIL@VERT with the actual destination Internet address as the first line of the message.

**Relaying:**

As of v1.10, the Synchronet Mail Server can be used to relay e-mail to external servers. Due to the security (SPAM) risks of opening the SMTP server up to just anyone, you must specify which hostnames or IP addresses you wish to allow to relay mail. You do this by creating the file **text/relay.can** and list the IP addresses or host names of the systems you wish to give relay access to. If you wish to allow relay access to all hosts (not recommended), simply create a **relay.can** with one line containing ".~".

**[8.3] - Setting Up the Message Base**

**[8.3.1] - Creating Message Groups:**

Run SCFG and select "*Message Base*" from the SCFG main menu and add any additional message groups you want. A message group is a group of message sub-boards that have a similar subject matter or other common element. It is common to put all of your local sub-boards in one group and your networked sub-boards (where messages are shared with other BBSs) into other groups. An example configuration that separates local and networked sub-boards into groups:

Group	Sub-boards
~~~~~	
Local	General
Local	Sports
Local	Debate
Local	Literature
Local	Technology

```
Local      Advertisements
DOVE-Net   General
DOVE-Net   Advertisements
DOVE-Net   Entertainment
DOVE-Net   Debate
DOVE-Net   Hardware/Software
DOVE-Net   Programming
DOVE-Net   Synchronet
DOVE-Net   Synchronet Sysops
FidoNet    Interuser Discussion
FidoNet    Politics
FidoNet    Unix
FidoNet    OS/2
FidoNet    DESQview
```

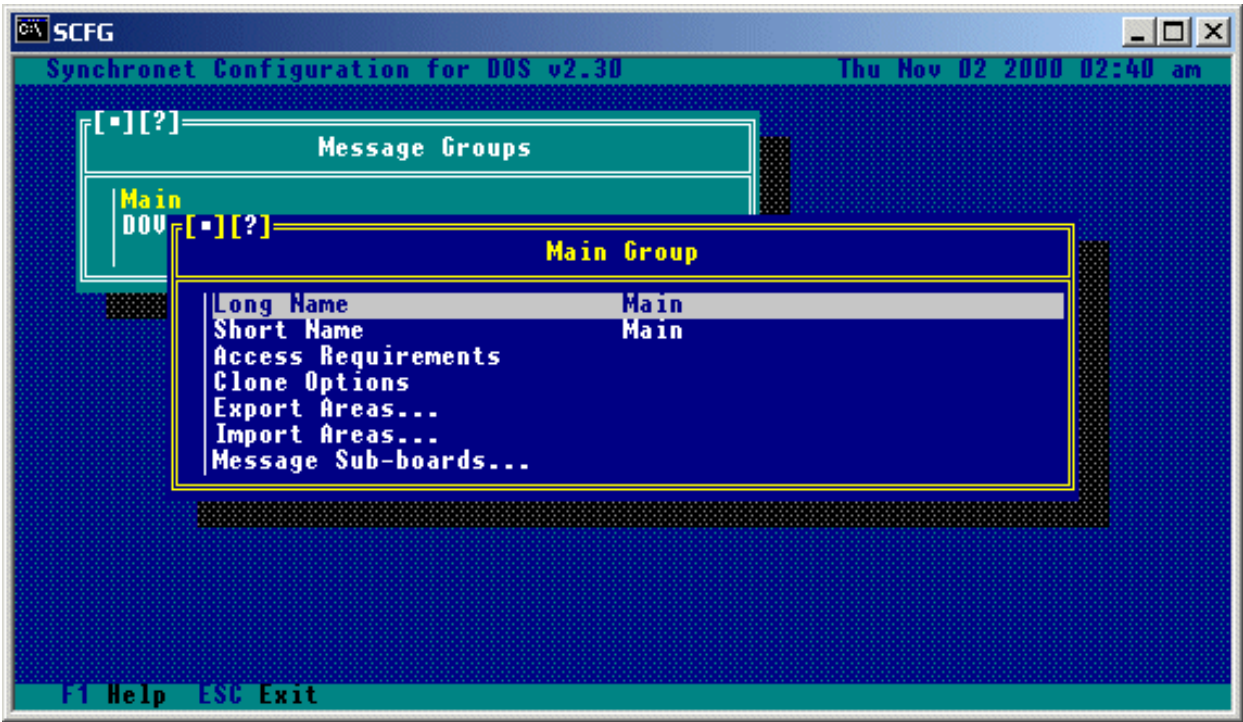
Another use of message groups is to group sub-boards of similar topic together:

Group	Sub-board
~~~~~	
Main	Sysop Messages
Main	General
Main	Debate
Main	Entertainment
Main	BBS Ads and ANSI
Technical	Software Help
Technical	Hardware Help
Technical	Programming
Adult	General
Adult	Singles Only
Adult	Dear Abby
Adult	Jokes
Adult	Financial

And you certainly can mix the two examples of message group utilization given above or invent your own. Or, you can just have one message group for all of your sub-boards - forgoing the organizational advantages of using multiple groups.

To add message groups, select the position you want the new message group to be in with the arrow keys. If you want the new message group to be the first group, highlight the top line. If you want the message group to be added to the end of the group list, highlight the bottom line. Hit *INS* (insert key) to add a new message group, or *DEL* (delete key) to remove an existing one. Following is a list of options available when creating or modifying a message group.

### [8.3.2] - Message Group Options



**Long Name:**  
This is a description of the message group which is displayed in message group listings.

**Short Name:**  
This is a description of the message group which is used for prompts.

**Access Requirements:**  
Use this option to set specific user requirements for access this message group.

**Clone Options:**  
Use this option to "copy" the options from the first message sub-board in this group to all other existing sub-boards in this group.



**Export Areas:**

If you wish to export the area information of all the sub-boards in the current message group to a text file, use this option.

**Import Areas:**

If you wish to import message area information from a text file, use this option.

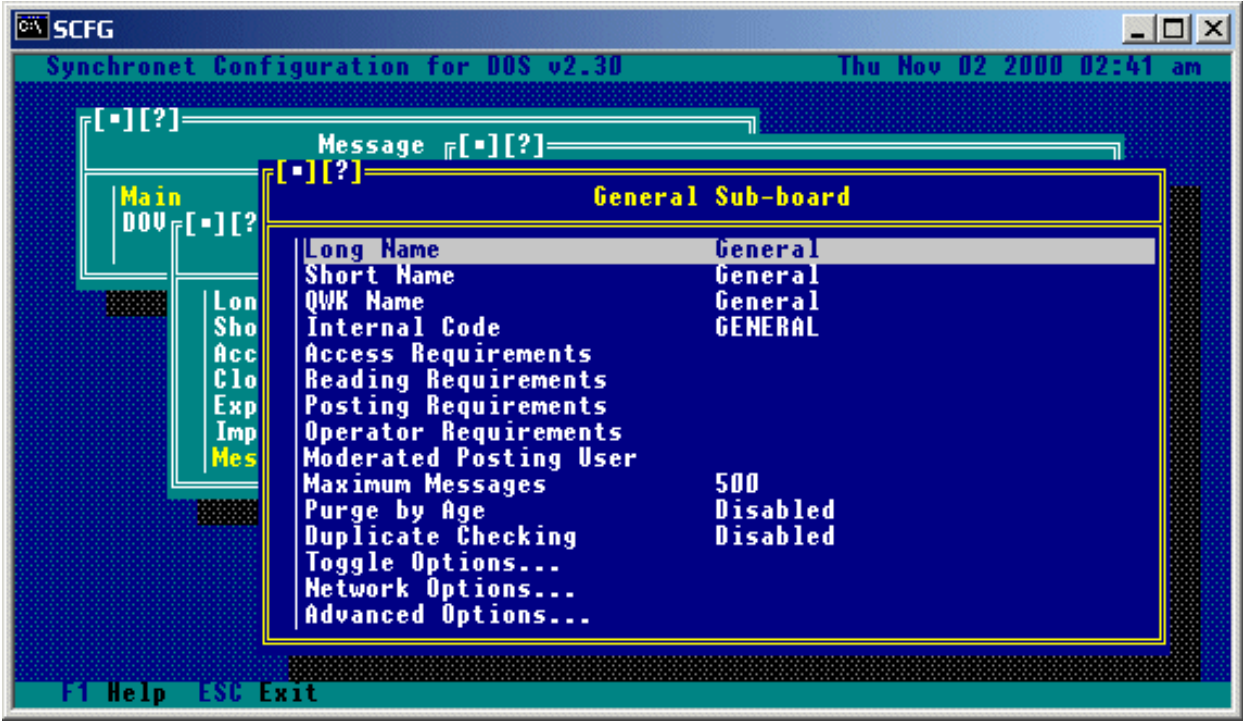
**Message Sub-boards....:**

Select this option to get a list of, and be able to edit, the sub-boards within this group.

**[8.3.3] - Creating Message Sub-boards:**

To create sub-boards, hit enter on the message group you want to add the sub-boards to, then select "Message Sub-boards". Adding sub-boards is very similar to adding groups except you will be prompted for an internal code. The internal code should be a short description (name abbreviation) for the sub-board. To configure the particulars for the sub-board, select it by hitting enter on it from the sub-board list. Following is a list of options available when creating or modifying a message sub-board.

**[8.3.4] - Sub-board Main Options**



**Long Name:**

This is a description of the sub-board which is displayed in sub-board listings.

**Short Name:**

This is a description of the sub-board which is used for prompts.

**QWK Name:**

This is the name of the sub-board used for QWK off-line readers.

**Internal Code:**

This is an internal code for SBBS to distinguish this sub-board from the others on the system. This should be a unique name of up to 8 valid DOS filename characters.

**Access Requirements:**

These are the requirements that a user must meet to be able to access this sub-board.

**Reading Requirements:**

Use this option to set the security requirements to read messages on this sub-board.

**Posting Requirements:**

Use this option to set the security requirements to post on this sub-board.

**Operator Requirements:**

Use this option to set the security requirement to be a sub-op on this sub-board (note: sub-op is a sysop of a sub-board).

**Moderated Posting User:**

Users meeting the criteria set here will have their posts "moderated" when posting on this sub-board. Moderated messages cannot be read by users or exported until validated by a sub-op or sysop.

**Maximum Number of Messages:**

This is the maximum number of messages that will be allowed on the sub-board. This number is used by SMBUTIL to maintain the message base and remove the oldest messages when this limit is exceeded.

**Purge by Age:**

This is the maximum length of time (in days) that old messages will be kept in a sub-board.

**Duplicate Checking:**

Use this option for duplicate message checking (comparing a new message against those previously imported to detect a duplicate message created by program or user error). Enabling duplicate message checking requires you specify a maximum number of CRCs. This is the number of previously imported messages to keep track of (using CRC calculation). This number, if used, should be set to about one weeks worth of messages. Large numbers in this field will slow down message importing.

**[8.3.5] - Toggle Options**



**Allow Private Posts:**

If you want to allow users to post private messages to another user on the sub-board, set this option to Yes. Since Synchronet has an internal e-mail system, this option is usually only set to Yes on networked sub-boards that allow private posts. You can also force private posts on this sub-board by setting this option to Only.

**Allow Anonymous Posts:**

If you want to allow users that have the 'A' exemption to be able to post anonymously on this sub-board, set this option to Yes. You may also force anonymous posts by any one (with or without the 'A' exemption) by setting this option to Only.

**Post Using Real Names:**

If you wish posts on this sub-board to be posted using the real name of a user instead of his/her alias, set this option to Yes. Many networks require BBSs that allow aliases to force real names on networked sub-boards.

**Users Can Delete Posts:**

You may allow users to delete their posted messages by setting this value to Yes. You may allows users to delete their messages only if it is the last message on the sub-board by setting this value to Last.

**Default On for New Scan:**

If this option is set to Yes, this sub-board will default to 'On' when a user does a new message scan on this sub-board for the first time.

**Forced New Scan:**

If this option is set to Yes, the sub-board will not be skipped in new message scans even if the user has configured his or her new scan to not contain this sub-board.

**Default On for Your Scan:**

If this option is set to Yes, this sub-board will default to 'On' when a user does a new message scan for personal messages on this sub-board for the first time.

**Public 'To' User:**

If you wish all posts to be prompted for a 'To' user, set this option

to Yes. Many networks do not allow private messages on networked sub-boards, so the only way to send a message 'To' someone is to send it publicly with a 'To' field. Normally, 'To' users are only prompted for on private posts.

**Allow Message Quoting:**

If you wish for users to be able to quote from other messages on this sub-board, set this option to Yes.

**Permanent Operator Msgs:**

Setting this option to Yes will cause messages posted by the Sysop (or sub-op) to be marked as permanent (will not be deleted by normal maintenance).

**Kill Read Messages:**

If you wish for messages to be deleted after the recipient has read the message, set this option to Yes. To delete only those messages marked as private after they've been read, set this option to Private.

**Compress Messages (LZH):**

When this option is set to 'Yes', the BBS will utilize the LZH compression method when storing messages. This will cause message importing to be slower, but can conserve disk space (30% to 50%) which can mean tens or even hundreds of megabytes of extra space on BBSs with large message bases.

**[8.3.6] - Network Options**



**Append Tag/Origin Line:**

If you want a tagline (QWK network) or origin line (FidoNet) to be appended to messages sent out on the network from this sub-board, set this option to Yes.

**Export ASCII Only:**

Set this option to Yes if this sub-board should only be exporting ASCII characters to message networks.

**Gate Between Net Types:**

If this sub-board is simultaneously networked with multiple networks using different network technologies, and you want messages imported from one network to be exported to the other network(s), you must set this option to 'Yes'. Gating messages between networks usually requires specific authorization from the moderator of each network. Do not set this option to 'Yes' unless you are absolutely sure you understand the consequences.

**QWK Networked:**

Setting this option to Yes will limit posted message titles to the QWK packet limitation of 25 characters and allow QWK network accounts to upload and download messages to/from this sub-board.

**QWK Tagline:**

This is an optional tagline that will be used for this sub-board. If this field is blank, the default tagline will be used. If "Append Tag/Origin Line" is set to No, then no tagline will be used.

**Internet:**

If this sub-board is networked via the internet, this toggle option should be set to 'Yes'.

**PostLink or PCRelay:**

If this sub-board is networked via PostLink or PCRelay, set this option

to Yes.

**FidoNet EchoMail:**

If this sub-board is networked via FidoNet, set this option to Yes.

**FidoNet Address:**

This is the FidoNet address used for this sub-board.

**EchoMail Origin Line:**

This is an optional origin line that will be used for this sub-board.

Do not include the FidoNet address in this line. Synchronet automatically adds the " \* Origin: " and "(Zone:Net/Node.Point)".

If this field is blank, the default origin line will be used. If "Append Tag/Origin Line" is set to No, then no origin line will be used.

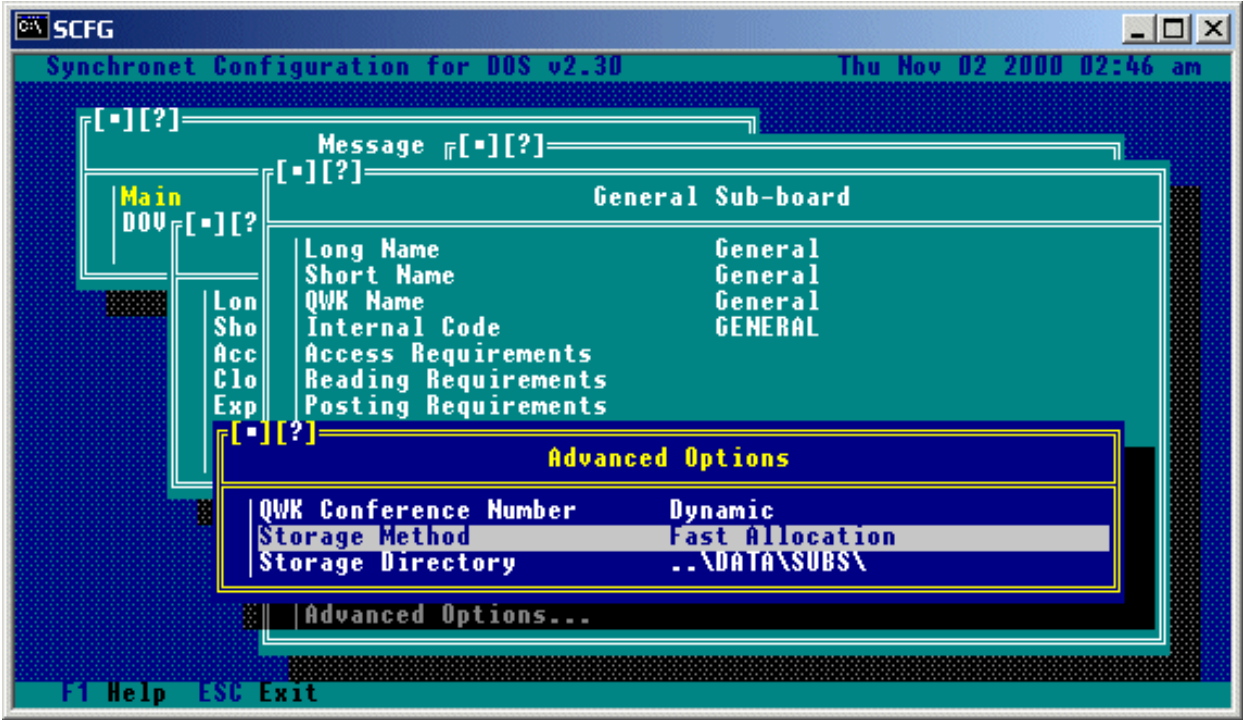
**EchoMail Directory:**

This is the storage directory where FidoNet style (FTS-0001 stored message format) messages are imported from and exported to for this sub-board (using SBBSFIDO a defunct method of importing/exporting FidoNet echomail). If this option is blank, then the EchoMail base directory is used with the internal code of this sub-board as the sub-directory where the FidoNet messages are stored.

**EchoMail Semaphore File:**

This is the path and filename of a file that should be created to trigger your front-end mailer to export FidoNet EchoMail.

**[8.3.7] - Advanced Options**



**QWK Conference Number:**

This option allows you to use a Dynamic (automatic) or Static (hard-coded) numbering for the QWK conference number of each sub-board. Dynamic numbering gives "as is" conference numbers, where sub-board 1 in group 1 would be 1001, and sub-board 1 in group 2 would be 2001, etc. Static numbering allows you to give each conference whatever number you prefer, which is extremely useful because it allows you to add and remove conferences without altering the numbers of other sub-boards.

**Storage Method:**

This is the method which will be used for storing messages when a new message is posted on this sub-board. There are three message storage methods. Their advantages and limitations are as follows:

**Self-packing:**

Self-packing is the most efficient message storage method because it looks for and uses deleted message blocks for new in-coming messages. Because of this behavior, it is the slowest storage method, but eliminates the need to run "SMBUTIL P" on the message base regularly. It is suggested however, that you do run "SMBUTIL P" on Self-packing message bases that have accumulated a large number of deleted message blocks. This can happen if you lower the maximum number of a messages for a message base and then run "SMBUTIL M" or import a large number of networked messages into this message base, far exceeding the maximum, and then running "SMBUTIL M". Under these circumstances, you will be wasting disk space on deleted message blocks that will not likely be used again unless you reach that number of messages again.

An example:

A Self-packing sub-board is set to maximum of 500 messages. The sub-board has 500 messages in it taking up approximately 2MB of disk space. A large network message packet is imported containing 200 messages for this sub-board. The sub-board now has 700 message in it taking up approximately 3MB of disk space. "SMBUTIL M" is run and the oldest 200 messages are deleted, bringing the total messages down to 500 again, but still taking up 3MB of disk space. The sub-board will not increase in size (disk consumption) again unless the total number of messages exceeds 700. If this is not likely to happen, then running "SMBUTIL P" on this sub-board will save you approximately 1MB of disk space.

#### **Fast Allocation:**

Fast Allocation storage method is faster than Self-packing because it does not search for deleted message blocks to use for new messages, it always adds to the end of the message base. Because of this behavior, the message base will continually grow in size (consuming disk space) until "SMBUTIL P" is run. Since the idea of using Fast Allocation is speed, it is also a good idea to speed up your "SMBUTIL P" event by specifying a minimum number of packable kilobytes before actually packing a message base. This will cause SMBUTIL to analyze the message base before actually packing it. This is be done by specifying the number of kilobytes on the SMBUTIL command line. If for example, you only want it to pack the message base if it can save 500k or more of disk space, then use "SMBUTIL P500" for a command line. This will keep SMBUTIL from packing the message base every time it is run (which can be time consuming) even if only a small amount of disk space will be saved from the operation.

You can switch between fast Fast Allocation and Self-packing storage methods at any time. You can even have one program importing into a message base using Self-packing and another using Fast Allocation and no harm will be done to the message base.

#### **Hyper Allocation:**

Hyper Allocation is much like Fast Allocation, except it is much faster because it does not update the message base allocation files necessary to be downward compatible with the Self-packing storage method. For this reason, you cannot change a message base from Hyper Allocation to Self-packing or Fast Allocation without first deleting all the messages in the message base (SCFG will do this for you). You must also be sure that all programs that write to this message base support the Hyper Allocation storage method (as defined in v1.20 of the SMB specification). If you are using any third party message import programs, do NOT use this storage method unless you are positive the program supports it. Ask the developers if you are not sure.

Like Fast Allocation, you must run "SMBUTIL P" on Hyper Allocated message bases regularly. The minimum packable kilobytes analysis feature of SMBUTIL (i.e. "SMBUTIL P500") takes longer with a Hyper Allocated message base but the actual packing operation is much faster.

To get the fastest importing/writing speed, disable Duplicate Message checking and LZH compression for the message base.

Retrieving messages (exporting/reading them from the message base) is the same speed with all storage methods. LZH compressed message bases will export slower than non-compressed message bases.

#### **Storage Directory:**

Where Synchronet should place the data files for this sub-board. Normally this option should be left BLANK unless it is required that the data be placed on a different drive letter.

## **[8.4] - Importing QWK Packet from Previous BBS Software**

If you are upgrading to Synchronet from another BBS package and have your messages in QWK packet format, you can import them into Synchronet as follows:

1. Create message groups and sub-boards under Synchronet to accept the messages from your old system.
2. Write down the conference numbers for each of your message areas from the old system.
3. Rename your exported QWK packet to TEMP.QWK and copy it into your DATA directory.
4. Add a QWK network hub in SCFG using TEMP for the QWK ID.
5. Add each of your sub-boards to the list of Networked Sub-boards and enter the conference number from the old system. Select "Strip-out" for Ctrl-A handling. Set Node 1 for the call-out node.
6. Run SBBS from your NODE1 directory and your messages should be imported immediately.
7. Run SCFG and delete the TEMP QWK network hub.



## [8.5] - Posting a Message

When posting a message on a sub-board, users or sysops can specify that the message is NOT to be distributed across a message network (kept local only). This is done by starting the message title with "NE:" which stands for "No Echo" (only useful on networked sub-boards).

## [8.6] - Remote QWK Functions

Following are examples on how to perform various functions remotely via QWK. Note that some of these functions are intended only for users who are using QWK mail doors to retrieve their messages from the BBS, and not for BBSs which are QWK netted together, these commands are noted as being for users only. Although intended for users, some of the user commands will work properly on a QWK netted BBS (e.g. FILES would place the resulting file list into the BBSs inbound directory), but some will not (e.g. the YOURS function).

In order to invoke the various remote QWK functions via a QWK network, you must post a message to SBBS on one of the conferences which you are receiving from your QWK hub. Note that the ADD and RESET functions are conference specific, meaning that your message MUST be posted on the conference where the function is to take effect.

**NOTE:** In the following examples, 'ptr' is a pointer indicating the message number to start at (e.g. specifying 100 would set the message pointer to message number 100), '-msgs' sets the message pointer to a number of messages from the end (e.g. -100 to set the pointer to 100 messages from the last message), and 'mm/dd/yy' sets the message pointer to the date specified (e.g. 01/01/90 sets the message pointer to January 1, 1990).  
The '|' symbol shown in the options means OR. Function parameters in square brackets '[']' are optional, parameters in less than/greater than symbols '< >' are required. Specifying an ON/OFF function without the ON or OFF parameter will toggle the function OFF.

*Examples:*

To: **SBBS** (All messages must be sent to SBBS)

Subj: **DROP [conf#]**

Note: Drop current conference (or specified conference #) from future packets.

Subj: **ADD [YOURS] [ptr | -msgs | mm/dd/yy]**

Note: Add current conference to future packets and optionally set the message pointer.

If "YOURS" is specified, only mail addressed to you will be packed for this conference.

The YOURS option is for users only.

Subj: **YOURS [ptr | -msgs | mm/dd/yy]**

Note: Same as "ADD YOURS".

For users only

Subj: **RESET [ptr | -msgs | mm/dd/yy]**

Note: Set message pointer for current conference.

Subj: **SUBPTR [ptr | -msgs | mm/dd/yy]**

Note: Same as "RESET".

Subj: **RESETALL [ptr | -msgs | mm/dd/yy]**

Note: Set message pointers for all conferences.

Subj: **ALLPTR [ptr | -msgs | mm/dd/yy]**

Note: Same as "RESETALL".

Subj: **FREQ <filename>**

Note: File Request from file transfer database (not attachments). QWK netted BBSs will receive the requested file into their hubs IN sub-directory, normally located in the \SBBS\DATA\QNET directory

Subj: **FILES [ON | OFF | mm/dd/yy]**

Note: Include files list in packet and/or specify new-scan date.

Specifying files with the date only will turn this option ON.

For users only

Subj: **ATTACH [ON | OFF]**

Note: Include file attachments in packet automatically (e-mail only).

For users only

Subj: **OWN [ON | OFF]**

Note: Include messages from you (affects all conferences).

For users only

Subj: **MAIL [ALL | ON | OFF]**

Note: Include private mail-box (ALL includes previously read mail).  
For users only

Subj: **DEMAIL [ON | OFF]**

Note: Automatically delete mail-box after successful packet download.  
For users only

Subj: **CTRL-A [KEEP | EXPAND | STRIP]**

Note: Ctrl-A color/attribute codes - leave-in, expand to ANSI, or remove.

Subj: **NDX [ON | OFF]**

Note: Include index (.NDX) files (not necessary for Synchronet QWKnet)

Subj: **CONTROL [ON | OFF]**

Note: Include control files (DOOR.ID, CONTROL.DAT, NETFLAGS.DAT, etc)

Subj: **VIA [ON | OFF]**

Note: Include message path (@VIA) line in messages.

Subj: **TZ [ON | OFF]**

Note: Include time zone (@TZ) line in messages.

### Downloading Files in QWK

If there are any files in the batch download queue when a QWK packet is created, they will be added to the QWK packet automatically. This is what makes the **FREQ** control command so useful. A user (or QWK network node) just posts a message (on any sub-board) to SBBS with the title "**FREQ FILENAME.EXT**" (where FILENAME.EXT is the filename and extension of the file to download). When the REP packet is extracted, the requested file (if found) is added to the batch download queue. Then when a QWK packet is created, the file is automatically included. Multiple file requests can be made and all files will be included in the QWK packet.

### Sending Netmail via QWK

If a user wishes to send netmail via a QWK packet generated from his offline mail reader, he must post a message in the E-mail conference (number 0) with the name and address of the recipient in the TO field of the message. (e.g. John Doe @1:103/715 would be a valid Fidonet netmail address, or jdoe@easyst.com would be a valid Internet netmail address). This will not work if the user does not have the ability to send netmail.

If the entire netmail address is too long to fit into the TO field (up to 25 characters), the user can alternately place the word "NETMAIL" in the TO field, and place the netmail address (name@addr) of the recipient on the FIRST LINE of the message body. Note that the '@' symbol MUST be present in a netmail address if used in the 'to' field.

### [Back to Top](#)

---

### [Copyright](#) © 2000 by [Rob Swindell](#)

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .

# Synchronet BBS

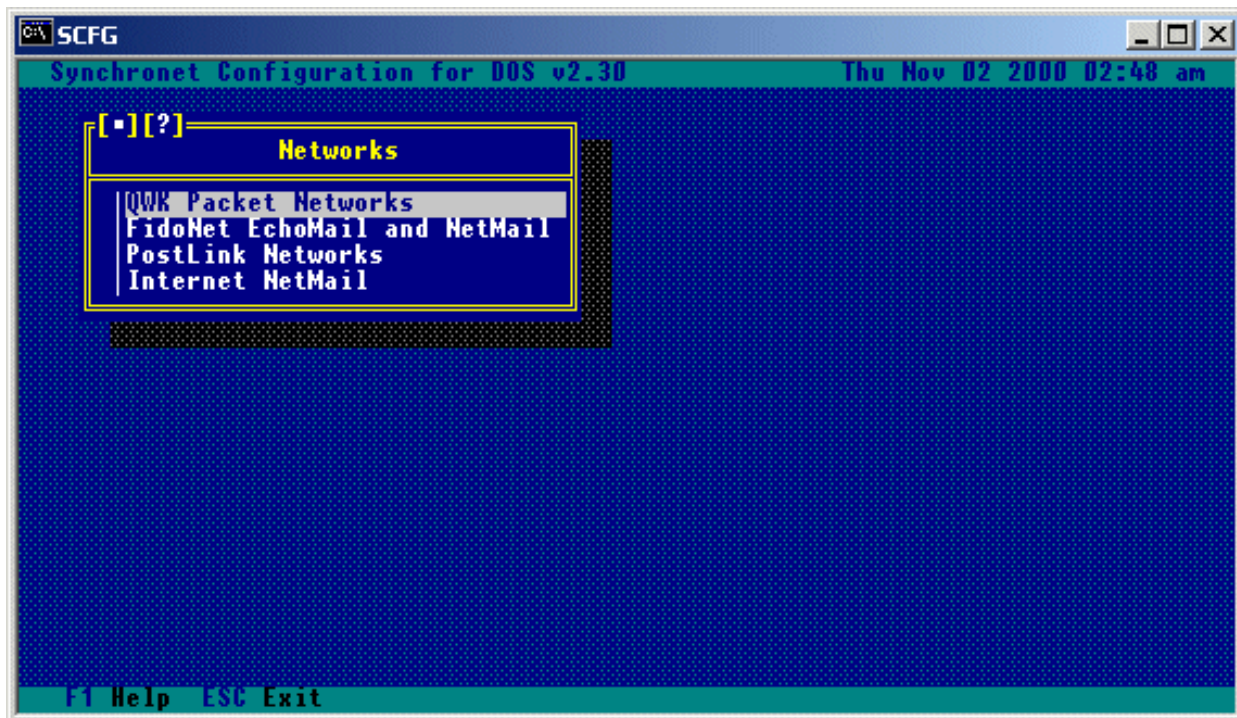
Multinode Bulletin Board System Software

---

[Back to Table of Contents](#)

## [9.0] - Networking

Selecting the Networks option from the SCFG will produce a list of available network technologies.



Select the network technology of your interest and refer to the following section that pertains to your selected network technology.

### [9.1] - QWK Packet Networking

Let's start with some basic definitions. A QWK netted BBS is either a node, a hub, or both. A node is a BBS that calls another QWK net hub to transfer packets (receiving QWK and sending REP). A hub is a system that receives calls from other nodes and transfers packets (sending QWK and receiving REP).

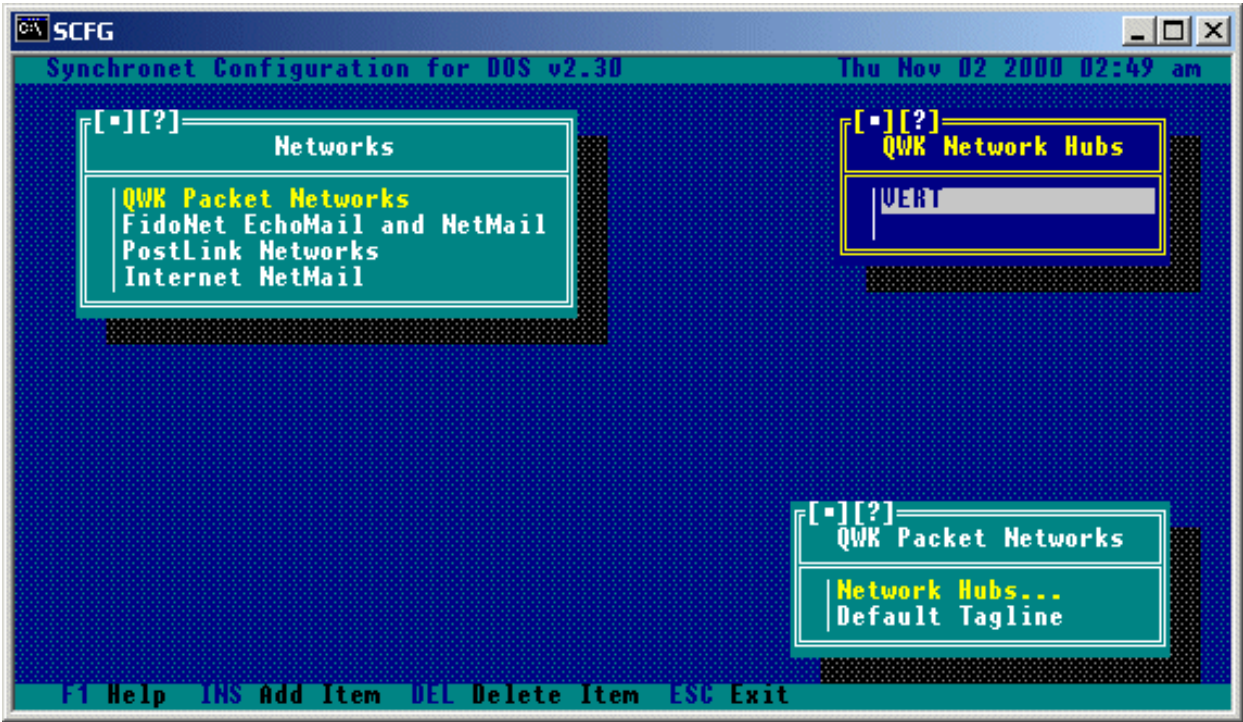
There are two options from the QWK Packet Networks menu, Network Hubs and Default Tagline. Selecting the Default Tagline will allow you to create or edit the tagline that will be used for messages that are sent out on the network from your QWK netted sub-boards. Individual sub-boards can have a different tagline to override this one, or use no tagline at all. Ctrl-A codes can be used in taglines so that other Synchronet systems will see the tagline in your preferred color scheme. The beginning of the tagline is not configurable. It consists of a tear line and the Synchronet product name:

```
---
p Synchronet p
```

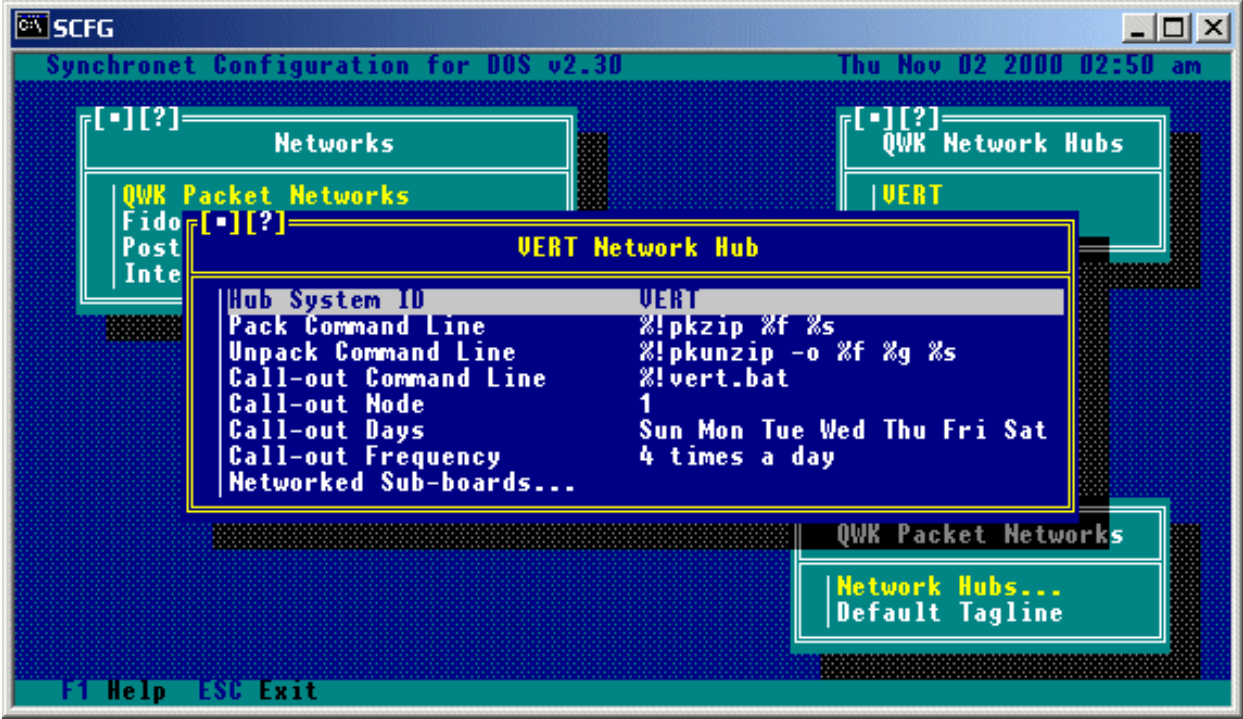
The configured tagline text will be attached to the end of the above tagline.

#### [9.1.1] - Network Hubs

Selecting this option produces a list of configured hubs that your system calls to upload REP packets to and download QWK packets from. If a system calls your system for messages, it is a node and your system is the hub and you should not have that system listed as a hub. If you do not call any systems for networked messages, then your system is a hub and the only configuration you need to do is set your taglines and the network options for each of your netted sub-boards. Use INS to add hubs and DEL to delete hubs from this hub list.



Selecting a hub from the list of available hubs will produce a sub-menu:



- Hub System ID:**  
This is the QWK system ID of the hub BBS.
- Pack Command Line:**  
This is the command line to execute to pack messages.
- Unpack Command Line:**  
This is the command line to execute to unpack messages.
- Call-out Command Line:**  
This is the command line to execute to perform the call-out.
- Call-out Node:**  
This is the number of the node which should perform the call-out.
- Call-out Days:**  
These are the days to perform the call-out.
- Call-out Time/Frequency:**  
This is either the specific time to call-out, or the number of times per day to call-out.

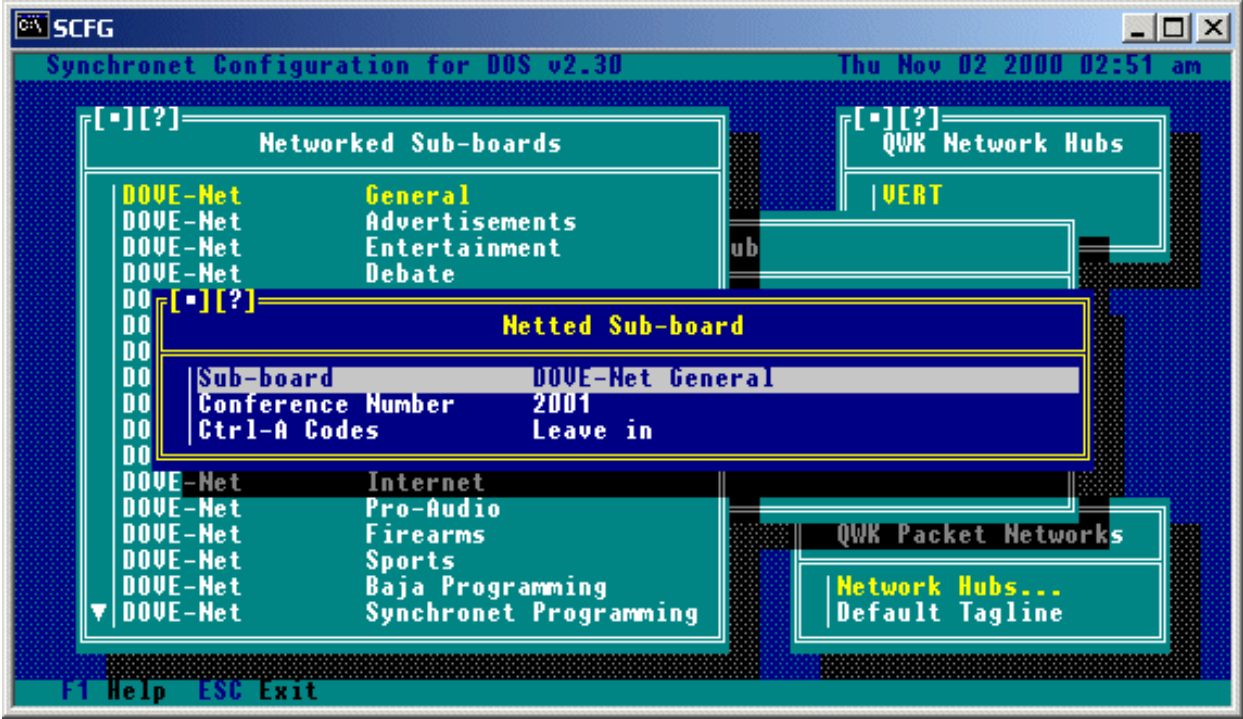
### [9.1.2] - Networked Sub-boards

Selecting this option will give a sub-menu containing a list of networked sub-boards that will look something like this:





For each sub-board the following options are available:



**Sub-board:**

Which sub-board is to be networked (Group and Sub-board name).

**Conference Number:**

This is the conference number of the sub-board on the HUB system.

If the hub is a Synchronet version 2 BBS and the sub-board is the second sub-board in the first group, the conference number is 1002; if the sub-board is the first sub-board in the third group, the conference number is 3001, etc. The conference numbering scheme for BBS programs other than Synchronet is usually just 1, 2, 3, etc.

**Ctrl-A Codes:**

This option allows you to determine how to handle Ctrl-A codes in messages. If the hub is a Synchronet BBS, you will want to set this to "Leave in". If the hub is not a Synchronet BBS, you will almost always want to set this to "Strip out". If the hub is not a Synchronet BBS, but allows ANSI escape sequences in messages, set this option to "Expand to ANSI".

**[9.1.3] - Configuring Your BBS as a QWK Node**

In the SCFG program, you'll need to specify your system's tagline (usually the system name and phone number), the hubs to call, when to call, what node will do the calling, and which sub-boards to carry for each hub (a sub-board can send and receive messages from more than one hub!).

The hub's System ID is important and must match the System ID of the hub BBS. For each hub, you can specify a list of sub-boards to network. For each sub-board, you must also set the conference number for the sub-board on the hub BBS. In Synchronet, conference numbers are based on the Group/Sub-board relationship. Group 1 / Sub 1 is conference number 1001, Group 2 / Sub 3 is conference number 2003, etc. (Note: this is NOT the conference number for the sub-board on your BBS. It is the conference number on the hub BBS.) Also, for each sub-board you must set the method of Ctrl-A code handling. If the hub system is a Synchronet BBS, you'll want to leave Ctrl-A codes in. If it is not, you'll probably want to strip them.

You'll need to decide which node will do the calling-out and configure it so.



You'll also need to decide if you want your BBS to call-out multiple times per day at any given time based on a total number of calls per day (usually, if all your hubs are local) or if you wish the BBS to call-out at a specific time every day (usually if one or more hubs are long distance).

You also need to set the command line to execute to perform the call-out. If your hub is a Synchronet BBS, use the included QNET module (\*QNET), otherwise you probably use a batch file to execute a scripted terminal program. Example scripts for Telix, Robocomm, and Qmodem are included in the EXEC directory (SBBSQNET.\*).

### **[9.1.4] - Configuring Your BBS as a QWK Network Hub**

Being a QWK network hub, just means that other QWK network compatible systems are going to logon to your system and upload and download messages. All you have to do is create a user account for each QWK network node that is going to call your system. The user name/alias should be the QWK system ID of the node BBS. The account needs to have the 'Q' restriction.

The 'Q' restriction is used only for QWK network node accounts. An account with this restriction will receive the QWK: prompt immediately upon logon and can't access any other facility of the BBS. This restriction also allows that account to send messages that are from other users as well as receive private posts that are to other users.

In addition to the 'Q' restriction, the 'L' (logons per day) and 'T' (time per day) exemptions may be helpful, depending on how many times the node will call your system a day and the time per call/day and logons per day allowed by the level you've given the account.

Be sure to give all node accounts sufficient access to read and write messages on the sub-boards networked between the node and your BBS.

## **[9.2] - Transferring Files Through QWK Network**

Synchronet allows the transferring of files between a QWK network node and hub with great simplicity. If you wish to send a file to a QWK network hub (your system calls directly) or QWK network node (their system calls yours directly), create a DOS subdirectory of DATA\QNET\QWKID.OUT, where QWKID is the QWK ID of the system to which you are sending the file. Copy any files you wish to send to this system into this directory. The next time the systems network with each other, the files will be sent.

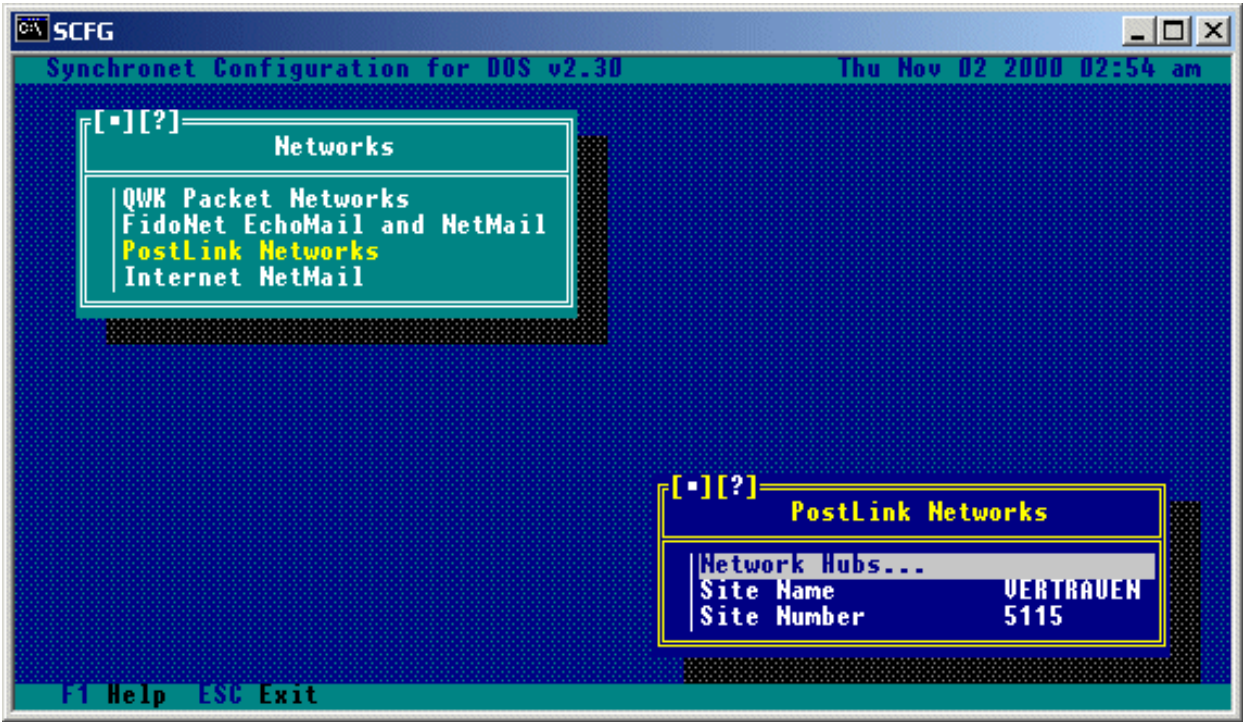
If your system is a QWK network node, you may request a file from your hub by posting a private message to SBBS on any of the networked sub-boards you get from your hub. The title of the message must contain "FREQ FILENAME.EXT" where FILENAME.EXT is the filename and extension of the file to download.

When files are received through a QWK network, the sysop is notified of the received file upon next logon. The file will be automatically placed in the subdirectory DATA\QNET\QWKID.IN, where QWKID is the QWK ID of the system that sent the file.

The transferring of files can be made automated for other programs to share data over the network. One popular use for this is Online Games that support inter-BBS play. One such game is Barren Realms Elite (BRE), see the external programs section for more information on this.

## **[9.3] - PostLink Networking**

PostLink is a message networking program written by Kip Compton and available for purchase from Bonnie Anthony (see Contacts Appendix). It is a derivation from PCRelay software. RelayNet (AKA RIME), headed by Bonnie Anthony, ILink and a few other networks use this technology. Selecting this option from the networks menu will bring up the following sub-menu:



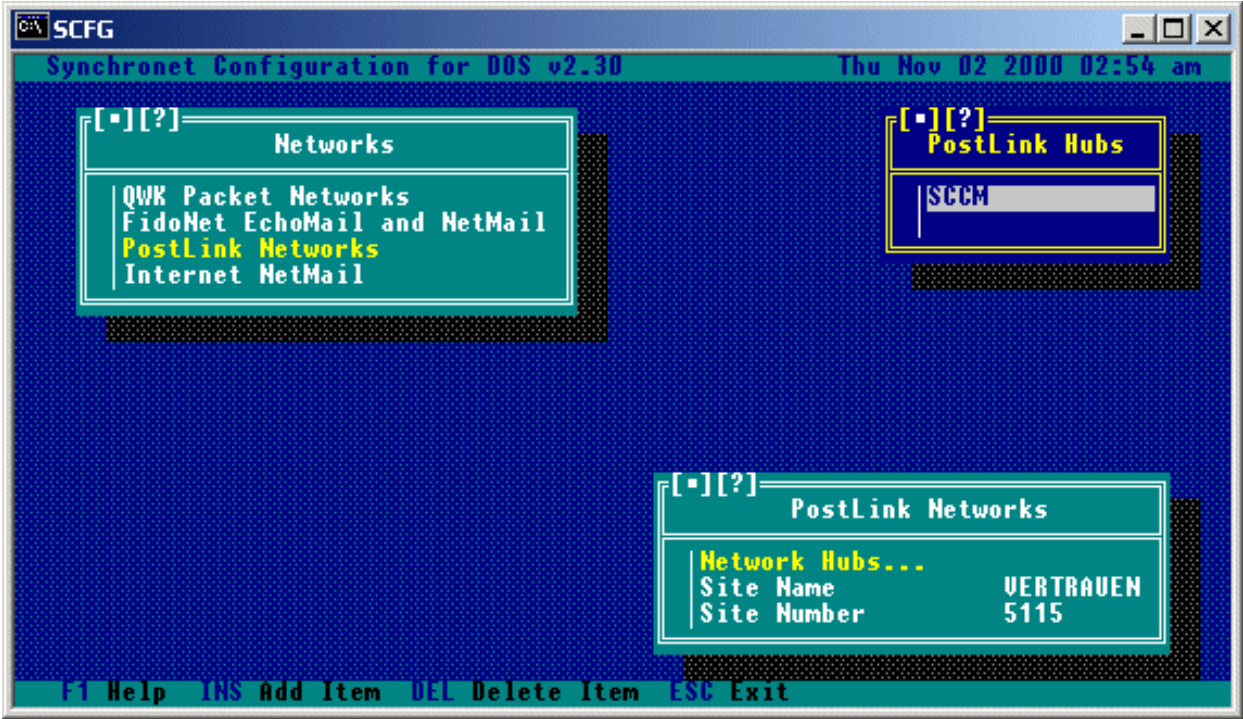
**Site Name:**

This is the site name to be used for your BBS.

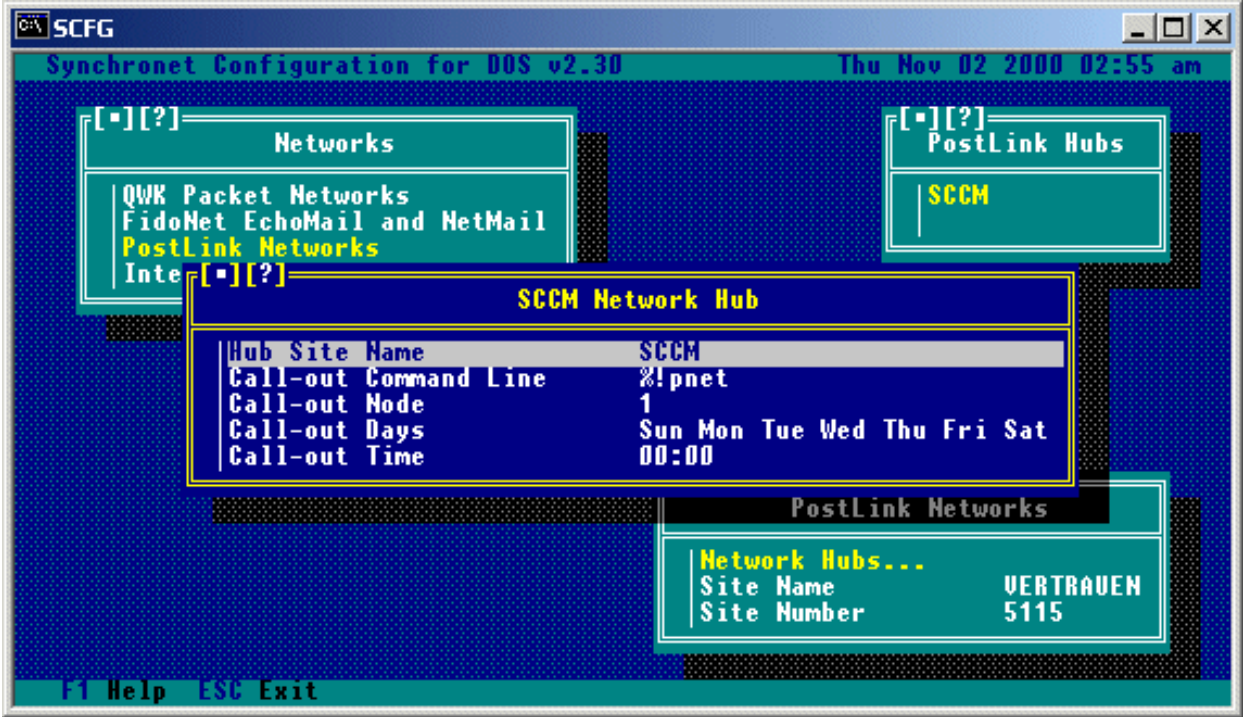
**Site Number:**

This is the site number which is assigned to you by the network coordinator.

Selecting Network Hubs from this menu will display a list of available hubs. Hubs can be added and/or removed from this menu.



Selecting one of the available network hubs from this list will allow you to configure the call-out information for that hub.



**Hub Site Name:**

This is the name of your network hub.

**Call-out Command Line:**

This is the command line which will be executed to perform the call-out for this hub.

**Call-out Node:**

This is the number of the node which will perform the call-out for this hub.

**Call-out Days:**

This allows you to select the days on which call-outs to this hub will occur.

**Call-out Time:**

This allows you to select the time when call-outs to this hub will occur.

Synchronet supports PostLink by letting the sysop configure the call-out schedule and which node will perform the call-out. Multiple PostLink networks are supported simultaneously by specifying multiple hubs. The Site ID entered in SCFG is only used for documentary purposes. Synchronet includes a UTI (Universal Text Interface) driver for PostLink. It is a set of 6 programs (UTI\*.EXE) included in your EXEC directory. See the Utility Reference for more information on the Synchronet UTI Driver.

Be sure to set the "PostLink Networked" Network Option to 'Yes' for each of your PostLink networked sub-boards.

## [9.4] - FidoNet Networking

FidoNet is both a network technology (referred to as FTN) and an actual message network. The technology was developed specifically for FidoNet, but is used by many other networks today as well. These non-FidoNet networks are referred to as FTNs (FidoNet Technology Networks) because they share the same technology, but don't necessarily have anything more in common with FidoNet.

Setting up a FidoNet is likely the most complex task a sysop could attempt. There are many steps associated with joining a FidoNet and most of the steps involve new terminology to even the most experienced sysop.

The first step is to find a FidoNet Front-end mailer program. This program will need to be run to make and accept FidoNet mail calls. It is usually run 24 hours a day, but is usually only required during a certain period (called Mail Hour and defined by the individual networks). The most popular of these programs are FrontDoor, D'bridge, InterMail, and Binkley. Both FrontDoor and Binkley will require the use of a FOSSIL (serial communications) driver as well. The most popular FOSSIL drivers are BNU and X00 (many DOORs require FOSSIL drivers as well).

When running Synchronet from the Front-end, you will need to pass the current connect rate, tell Synchronet to quit after the call, and possibly pass the minutes till the next event.

Example: SBBS Q C14400 E60

The above command line tells Synchronet that someone is currently connected (at 14400bps), to quit back to DOS after the caller logs off, and that the next scheduled event is in 60 minutes. See Appendix B for all the available switches for running Synchronet.

SBBSecho is an included Synchronet utility (registered separately) to import/export NetMail and EchoMail bundles/packets.

NetMail will be exported to and imported from the directory specified in the Networks configuration in SCFG. EchoMail will be imported to and exported from the EchoMail path specified for that sub-board, or off the base EchoMail directory specified in SCFG Network options in a subdirectory named after the internal code of the sub-board.

### [9.4.1] - Step-by-Step instructions for setting up FidoNet using SBBSecho and InterMail *(v2 Only)*

One of the most difficult tasks that the novice sysop will encounter is the setting up of FidoNet. These instructions will take you step by step through the process of setting up FidoNet on your BBS. Be warned that these steps are very basic, just enough to get FidoNet up and running on your system, but it may still be necessary for you to refer to the documentation which accompanies the programs referred to in these steps. For any advanced operations you will definitely need to refer to the documentation of programs referred to in these steps. If you know of anyone who is already on FidoNet that is willing to help you out, by all means ask them for their help!

NOTE that while it doesn't cost anything to have a node number (for sending and receiving netmail) or to receive local conferences, there is usually a nominal fee (called "cost recovery") to receive conferences from the FidoNet backbone or files from the FidoNet filebone.

- 1) First of all you will need to obtain a copy of FidoNet's Policy 4 document

- 2) Read the Policy 4 document, it will give you information about FidoNet as well as some specifics on how to obtain a node number for your system.
- 3) If you do not plan to run InterMail (or FrontDoor which is very similar) you must set up the front-end mailer you plan to use by following the directions which accompany the program and then skip to step number 9. The documentation will also tell you how to "compile" the nodelist file you've downloaded so that your front-end mailer can use it. Use the temporary address specified in Policy 4 when specifying your node address in the front-end mailer.

- 4) Install InterMail per IM.DOC (included with InterMail).
  - a) If using FrontDoor, follow the FD.DOC file and substitute the letters FD where ever you see an IM in this section.
- 5) Copy the MAILER.BAT included with SBBSecho and the EXEBBS.BAT from the Synchronet DOCS directory into the InterMail directory (C:\IM).
- 6) Change into the InterMail directory, run IMSETUP, and perform the following:
  - a) Under Global->Address->Main, enter your FidoNet address (or the temporary address specified in Policy 4 until such time as you have received your own address).
  - b) Under Global->FileNames, verify that the paths are set similar to the following:

[illegible]

- c) Under Modem->Advanced Setup->Connect Strings set ALL "BBSexit" levels to 100. (For FrontDoor under Mailer->Errorlevels set all baud rates to 100, Received Mail to 50, and Create .BAT file to Yes).
- d) Under Manager->Events, make sure all your events are set to exit when mail is received (under Event Behavior). This allows for the immediate importation of new messages into the BBS.
- e) Under Manager->Events, add the following event:

```

◦ Tag X
◦ Days -----A
◦ Modifier Ignore eventbase (*)
◦ Start time 04:30
◦ Length 00:01
◦ Errorlevel 75
◦ Not used
◦ Behavior
◦ Use alias 1:103/705
◦ Retry delay 0
◦ Comment Synchronet exclusive event
◦ ~~~~~
◦ Event tag A..YZ - eXternal event, errorlevel is required

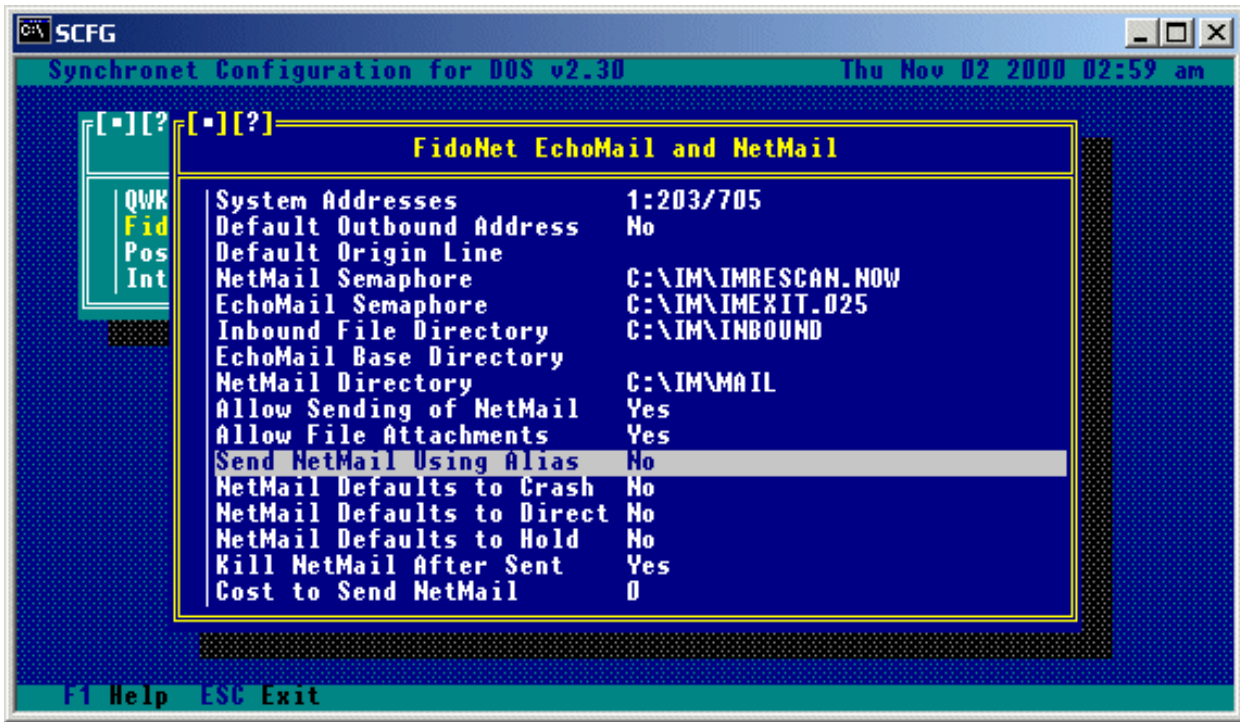
```

[illegible]

- f) Under Mailer->Misc, set "Trigger exit upon receipt of netmail" and "Upon receipt of ANY file" to Yes. This will allow for the immediate importation of NetMail and any File Distribution Echoes you may carry.
  - g) Under Modem->Hardware, set "Lower DTR when busy" to "No".
  - h) Under Modem->Command strings, set the "Down" string to "ATM0H1|".
  - i) Exit IMSETUP.
- 7) Unarc the nodelist file (NODLIST.???) that you downloaded previously into your nodelist directory.
  - 8) Now run IMNC to compile the nodelist.

## Setting up FidoNet in SCFG

- 9) Run SCFG from one of your node directories, or hit 'C' from the waiting for caller screen. Then select Networks->FidoNet EchoMail and NetMail. You'll then be brought to a menu that should be set up similar to this:



- a) For FrontDoor the NetMail Semaphore file should be set to C:\FD\FDRESCAN.NOW and the EchoMail Semaphore file should be set to C:\FD\FDEXIT.025.
  - b) For FrontDoor version 2.11, the EchoMail Semaphore file should be set to C:\FD\FDXIT.025.
  - c) For FrontDoor version 2.20 (commercial, multinode), the EchoMail Semaphore file should be set to C:\FD\FDXIT#.025, where # is the node number (0 based).
- 10) Set your address to your system's FidoNet address (or the temporary address specified in Policy 4 until such time as you have received your own address). This is the address used for sending and receiving NetMail. It will also be used as the default address for message sub-boards.
  - 11) Create an origin line for your system that most likely contains your BBS name and phone number.
  - 12) Following is an overview of the remaining options available on this menu:
    - a) **System Addresses:**

This option will bring you to a sub-menu to allow you to configure different addresses (AKAs) for your system. Following is an example for a BBS configured for two fido type networks.





**b) Default Outbound Address:**

When a user sends netmail, this is the 'send to' address that is used as a default (useful for FidoNet<->Internet gateways).

**c) NetMail Semaphore:**

This is the name of the trigger file which will cause your front-end mailer to exit and scan for outgoing netmail.

**d) EchoMail Semaphore:**

This is the name of the trigger file which will cause your front-end mailer to exit and scan for outgoing echomail.

**e) Inbound File Directory:**

This is the name of the directory where your front-end mailer stores incoming files.

**f) EchoMail Base Directory:**

This is the base directory for your echomail subdirectories

**g) NetMail Directory:**

This is the name of the directory where your front-end mailer looks for and places netmail.

**h) Allow Sending of NetMail:**

When set to 'Yes' users can send netmail from your system.

**i) Allow File Attachments:**

When set to 'Yes' users can send netmail file attachments from your system.

**j) Send NetMail Using Alias:**

When set to 'Yes' the user alias will be used (rather than their real name) on outgoing netmail.

**k) NetMail Defaults to Crash:**

If set to 'Yes' netmail sent from your system will default to crash status (send immediately and directly).

**l) NetMail Defaults to Direct:**

If set to 'Yes' netmail sent from your system will default to direct status (send directly).

**m) NetMail Defaults to Hold:**

If set to 'Yes', netmail sent from your system will default to hold status.

**n) Kill NetMail After Sent:**

Setting this to 'Yes' will delete outgoing netmail after it has been sent.

**o) Cost to Send NetMail:**

This is the cost (in credits) that a user will be charged when sending netmail.

**13) Setup SBBSecho by running the ECHOCFG program and reading the documentation for information on the available options.**

a) Under Paths...->Outbound Directory be sure to enter the path where outgoing mail packets and bundles should be placed (i.e.: C:\IM\OUTBOUND).

b) Select the appropriate Mailer Type.

c) Enter your registration number (if you are a registered owner of the

SBBSecho program).

- 14) Once you have done this you must now run the file MAILER.BAT (if using an Attach-Style mailer) to use Synchronet with your front-end mailer. MAILER.BAT must be run from your front-end mailer's directory (e.g. C:\IM). You will no longer run the file SBBS.BAT to start this node (if you run a multi-node system, you will still use SBBS.BAT to start those nodes which are not using a front-end mailer).
  - a) If you are NOT using InterMail as your front-end mailer, be sure to modify your MAILER.BAT file to reflect the front-end mailer that you are using.
- 15) Follow the instructions given in the Policy 4 document to determine who you should send netmail to in order to obtain your own node number.
- 16) Now, from your front end mailer, send a netmail to this person following the instructions contained in the Policy 4 document. Once you have done this and the netmail has been sent, you should patiently await a netmail reply containing your node number and the node number of your hub (where you will send/receive your echomail and netmail to/from).
- 17) Once you have received your node number, you'll want to replace the temporary node number you used in your front-end mailer as well as in SCFG->Networks->FidoNet Echomail and Netmail->System Addresses to it.
- 18) Send your hub a message with the Area Manager (for the backbone <echomail>) and Areafix (for the filebone <files>) passwords you wish to use on their systems. You will need to remember these passwords as you will be using them to do remote maintenance (such as adding and removing conferences and file areas that you wish to receive).
- 19) Your hub should respond with Area Manager instructions and a list of areas available for you to connect to. If not, contact your hub for more information.
- 20) Look through the list of areas available to you and determine which conferences you'd like to carry. In SCFG->Message Areas create a message group(s) for the conferences you'd like to carry. When setting up a sub-board for a conference, be sure to use the "Area Tag" name given in the area list as the sub-board Short Name.
  - a) Under Network Options... for each sub-board toggle the FidoNet Echomail option toggled to YES.
  - b) If the conference requires a different address, configure the address under Network Options...
  - c) If you allow aliases on your system, under Toggle Options... set Use Real Names to YES.
  - d) If you are unfamiliar with adding message groups and/or sub-boards, consult that section of the Synchronet System Operator's Manual.
  - e) For a more complete list of conferences available on the FidoNet backbone you may wish to download the file FIDONET.NA from either your hub or a local FidoNet BBS. This file contains a list of ALL conferences carried on the backbone.
- 21) Once you have set up all of the conferences you'd like to carry, you'll need to go to SCFG->Message Areas-><group>->Export Areas...->AREAS.BBS (SBBSecho) to create an AREAS.BBS file for SBBSecho to use.
- 22) Now following the Area Manager instructions, send a message to your hub's Area Manager and turn on all of the conferences you wish to carry. Note that there are some administrative conferences for FidoNet that your users shouldn't be able to read or post on, and some that even you shouldn't post on, there is also usually a "test" conference which you may want to set up to post messages on and insure that echomail is being sent and received properly.
- 23) That's about all there is to it. You are now set up on FidoNet and should be able to send and receive FidoNet echomail and netmail.

#### **Examples for Receiving File Echoes via TICK (third party file echo program)**

Example TIC.CFG:

```
-----[ Begin ]-----  
IN c:\fd\file  
ZONE 1 c:\fd\mailout  
NET 2  
NODE 3  
HOLD c:\fd\tickhold
```

```
QDIR c:\fd\tickqdir
FDLog
```

```
AREA j:\fido\backbone BACKBONE
1:2/1 xxxxx *
```

```
AREA j:\fido\SOFTDIST SOFTDIST
1:2/1 xxxxx *
```

-----[ End ]-----

In MAILER.BAT, at the end of the ":inmail" block, before "goto top":

-----[ Begin ]-----

```
set tz=PST8EDT
tick >> fd.log
set tz=
%sbbs%\exec\addfiles * /diz
```

-----[ End ]-----

## Setting up ALLFIX (third party file echo program) with Synchronet BBS Software

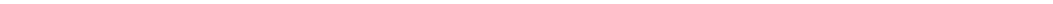
In ASETUP->System data->Global Options:

```
Oiiiiiiiiiiiiiiiiiiiiiiiii Global options .
```

```

3
3 Keep original file date Yes      Replace extension      No      3
3 Mailer rescan      FrontDoor      Max age of requests      0      3
3 Log style      FrontDoor      Keep exported messages      No      3
3 Mailer mode      FrontDoor      Days to keep on hold      30      3
3 Download counters      Size of stat file      0      3
3 Update DESCRIPT.ION      No      Max TIC archive size      0      3
3 Date format      mm-dd-yy      Max size to pack      0      3
3 Min HD space to import      1024      UTC offset      -8      3
3 Process local requests      No      Max Msg size      0      3
3 MSG compatibility      Fido      3
3 Dupe checking      Filename      3

```

3  3

```

3 LongDesc character      One line LongDesc      No      3
3 Filter LongDesc      No      Max len of LongDesc      0      3
3 Spaces to indent      1      Add tag to desc      No      3
3
3

```

[illegible]

Edit your MAILER.BAT:

```
Search for "set sbbs="
Add "set allfix=c:\allfix"
```

```
Search for ":inmail"
Add "%allfix%\allfix file mgr"
Add "%sbbs%\exec\addfiles * /diz"
```

```
Search for ":outmail"
Add "%allfix%\allfix file"
```

## [9.5.2] - Sending FidoNet NetMail

Once you have a FidoNet address and have your Front-end mailer functioning properly, you may send NetMail messages and allow, if you wish, your users to send NetMail. You can set a NetMail cost in credits for users in SCFG->Networks->FidoNet EchoMail and NetMail->NetMail Cost. Sysops and users with the 'S' exemption will not be charged credits when sending NetMail. Users with the 'M' restriction are not allowed to send NetMail. Users will not be allowed to send NetMail at all, if the sysop has set SCFG->Networks->FidoNet EchoMail and NetMail->Allow Sending of NetMail to No.

Users can optionally set a NetMail forwarding address for their account using the "Account Defaults" menu. If the user has a NetMail address specified, any E-mail set to that account will be automatically forwarded to the NetMail address if the sending user wishes. This is useful for users who can receive FidoNet NetMail on a "home" BBS system and prefer to receive their E-mail there instead of on this particular system.

Sysops and users with the 'F' exemption can also perform other special functions with NetMail. If NetMail Defaults to Crash status is OFF, sysops and 'F' exempted users can override this by starting their NetMail title with "CR:" setting the Crash status ON for that NetMail message.

Sysops and 'F' exempted users may also send File Requests by starting the title with "FR:" and then the filename(s) being requested follow on the title. If multiple filenames are requested, they must be separated by a space.

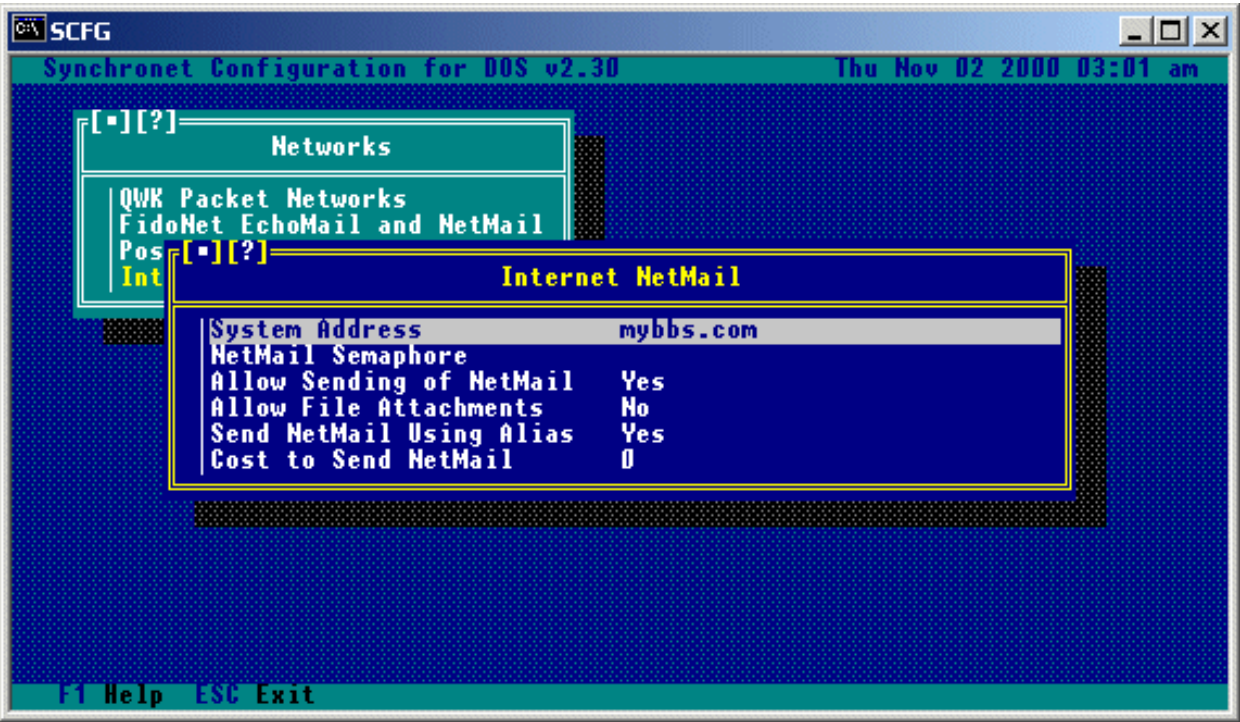
Example: "FR: FILE1.EXT FILE2.EXT".

Sysops and 'F' exempted users may also send File Attachments from anywhere on the system (potentially dangerous) by starting the title with "FA:" and then the filename(s) to attach to the netmail message. This allows the sysop to send a file attachment to multiple users without creating multiple copies of the attachment or requiring the attachment to be copied into the Synchronet DATA\FILE\xxxx.OUT directory of the sending user. Since the sending user (sysop or 'F' exempt) can attach ANY file on the system, the 'F' exemption and sysop security level (90+) should given out with extreme caution.

Sysops and 'F' exempted users can also request a return receipt by starting the title with "RR:". "CR:", "FR:", "RR:", and "FA:" may be used in combination with one another, but must be specified in that order.  
Example: "CR: RR: This is my title".

The "xx:" specifiers and an optional trailing space are eliminated from the final title for the NetMail message.  
Example: "CR:Hello" or "CR: Hello"  
Becomes: "Hello" before it is sent out by Synchronet.

## [9.5] - Internet Networking



**System Address:**  
This is the Internet address for your BBS.

**NetMail Semaphore:**  
This is the name of the trigger file which will cause your Internet mail program to exit and scan for outgoing netmail.

**Allow Sending of NetMail:**  
When set to 'Yes' users can send Internet netmail from your system.

**Allow File Attachments:**  
When set to 'Yes' users can attach files to the Internet netmail that they are sending.

**Send NetMail Using Alias:**  
When set to 'Yes' the user alias will be used (rather than their real name) on outgoing Internet netmail.

**Cost to Send NetMail:**  
This is the cost (in credits) that a user will be charged when sending Internet netmail.

**(v2 Only)**  
In order for your BBS to support Internet networking (including such things as sending/receiving Internet netmail and newsgroups, FTP, Telnet, etc.) it is required that you obtain a third party program such as NetXpress or BBSNet. It is also possible (but not recommended) to use a Fido->Internet or a QWK->Internet gateway (usually available as shareware software).

[Back to Top](#)

---

**Copyright © 2000 by Rob Swindell**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)

PO Box 501

Yorba Linda, CA 92885

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .



# Synchronet BBS

Multinode Bulletin Board System Software

[Back to Table of Contents](#)

## [10.0] - File Transfer Section

The Synchronet file transfer section is used for transferring data and program files between the BBS and the remote user. Files are stored in a logical hierarchy of libraries and directories. A library is a group of directories that contain files of a similar subject matter.

An example file library/directory configuration:

Library Name	Directory
-----	-----
Main	Text Utilities Business Graphics Games Communications Miscellaneous
Adult	Text Animation GIF files TGA files
DOS CD-ROM	Utilities Device Drivers Business Games Communications Programming
Windows CD-ROM	Utilities BMP files WAV files Device Drivers Games Desktop Publishing Fonts

## [10.1] - Setting Up the File Transfer Section

Run the SCFG utility, and select File Options from the menu. Following are example screens and the options available from the File Options sub-menu.

### [10.1.1] - File Options



#### Min Bytes Free Disk Space:

This is the minimum about of disk space (in kilobytes) that is necessary for a user to be allowed an upload.

**Max Files in Batch UL Queue:**

This is the maximum number of files that can be stored in the batch upload queue. The definite maximum is 500 files.

**Max Files in Batch DL Queue:**

This is the maximum number of files that can be stored in the batch download queue. The definite maximum is 500 files.

**Max Users in User Transfers:**

This is the maximum number of destination users in user to user transfers. The definite maximum is 500 users.

**Default Credit on Upload:**

This is the default amount used for newly created directories.

**Default Credit on Download:**

This is the default amount used for newly created directories.

**Leech Protocol Detection Percentage:**

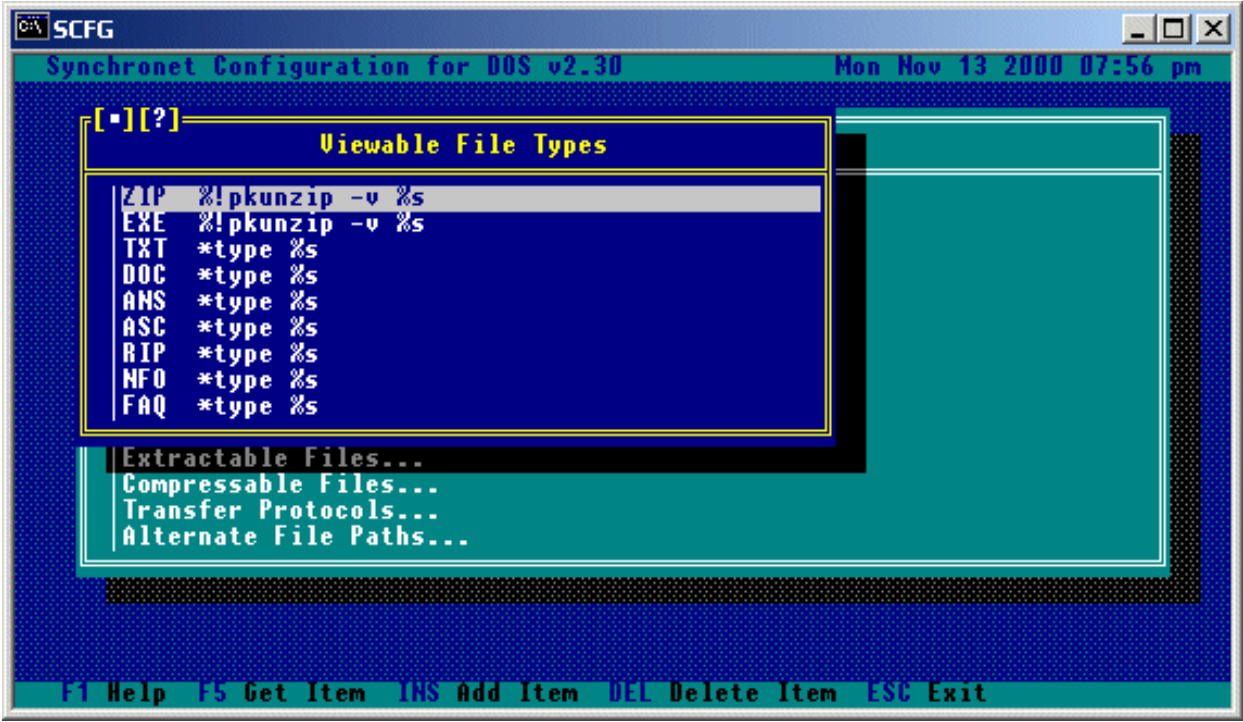
This value is the sensitivity of the leech protocol detection feature of Synchronet. If the transfer is apparently unsuccessful, but the transfer time was at least this percentage of the estimated transfer time (based on the estimated CPS of the connection result code), then a leech protocol error is issued and the user's leech download counter is incremented. Setting this value to 0 disables leech protocol detection. This option also allows you to set the minimum amount of elapsed transfer time to be considered for a possible leech download.

Leech protocol programs are file transfer programs (usually using Zmodem technology) that attempt to "fool" the BBS into thinking the file was not successfully transferred, when in reality it was. This is accomplished by the transfer program requesting a reposition (ZRPOS) after the last successful block and then aborting (ZCAN). A file transferred in this manner will not be considered a successful transfer by Synchronet, but will be caught as a possible leech download and notify the sysop (if this option is used).

This feature is also useful for detecting the partial download of image (GIF) files. If you charge your users credits for downloads, this can be a very useful feature in detecting dishonest users. If the user accumulates a large number of leeches (as displayed in User Edit) and the user never successfully downloads a file previously logged as a possible leech download, the user is probably trying to get something for nothing, though this is hard to prove without actually watching the file transfer in progress.

[10.2] - Viewable Files

This is a list of file types that have content information that can be viewed through the execution of an external program or Baja module. Here are a couple of command line examples for a few file types.

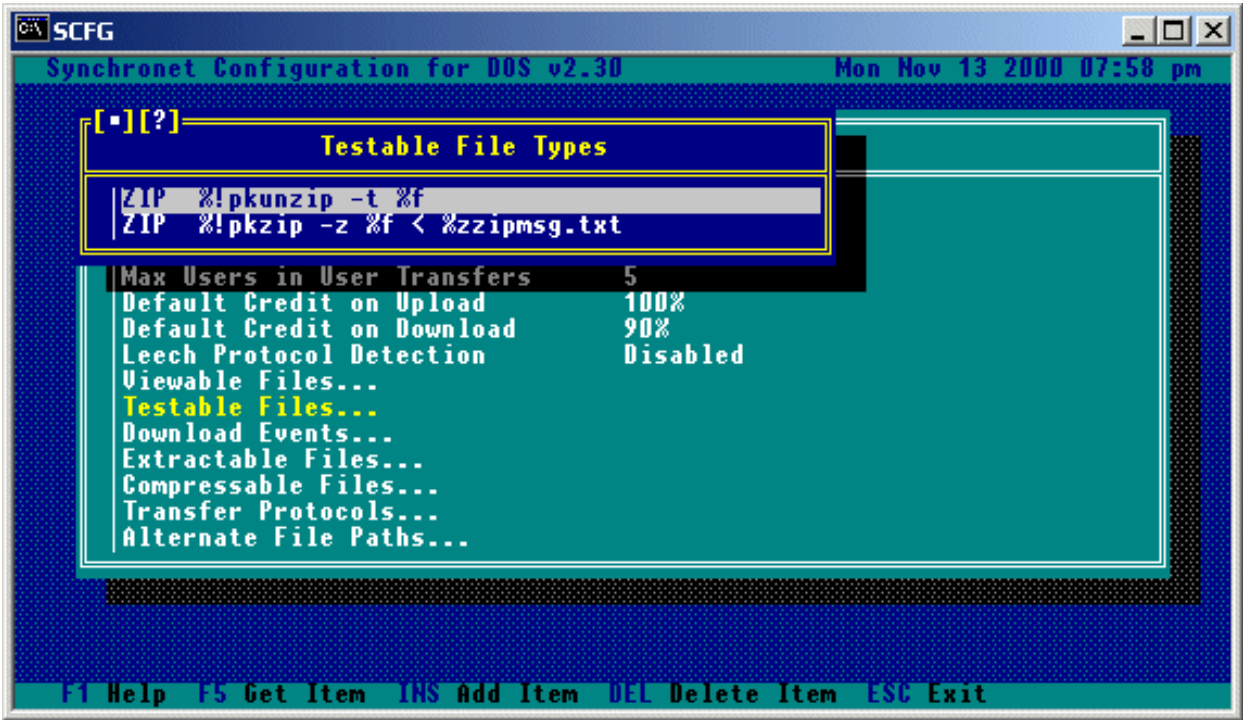


**NOTE:** When you select a file type from this list, you will be given one additional option (Access Requirements) which is not shown here. Users not meeting any access requirements which are set will not be able to perform, or be affected by, that function.

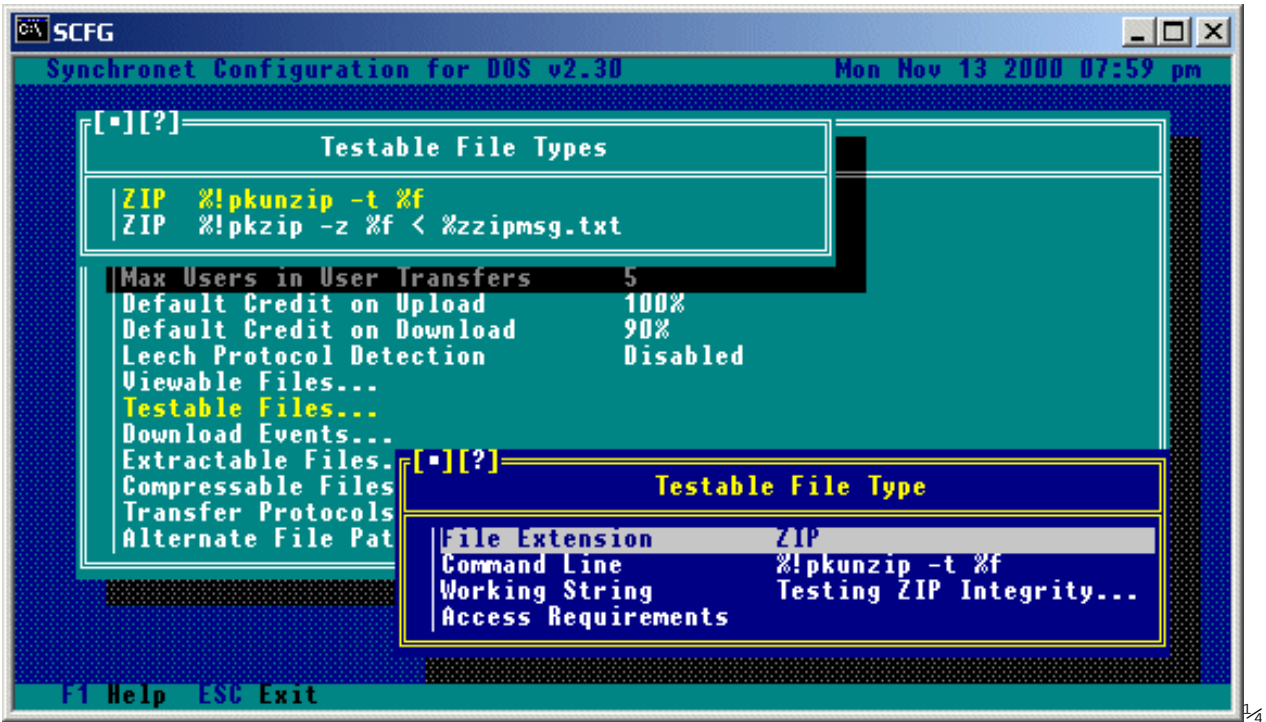
[10.3] - Testable Files

This is a list of file types that will have a command line executed to test the file integrity upon their upload. The file types are specified by extension

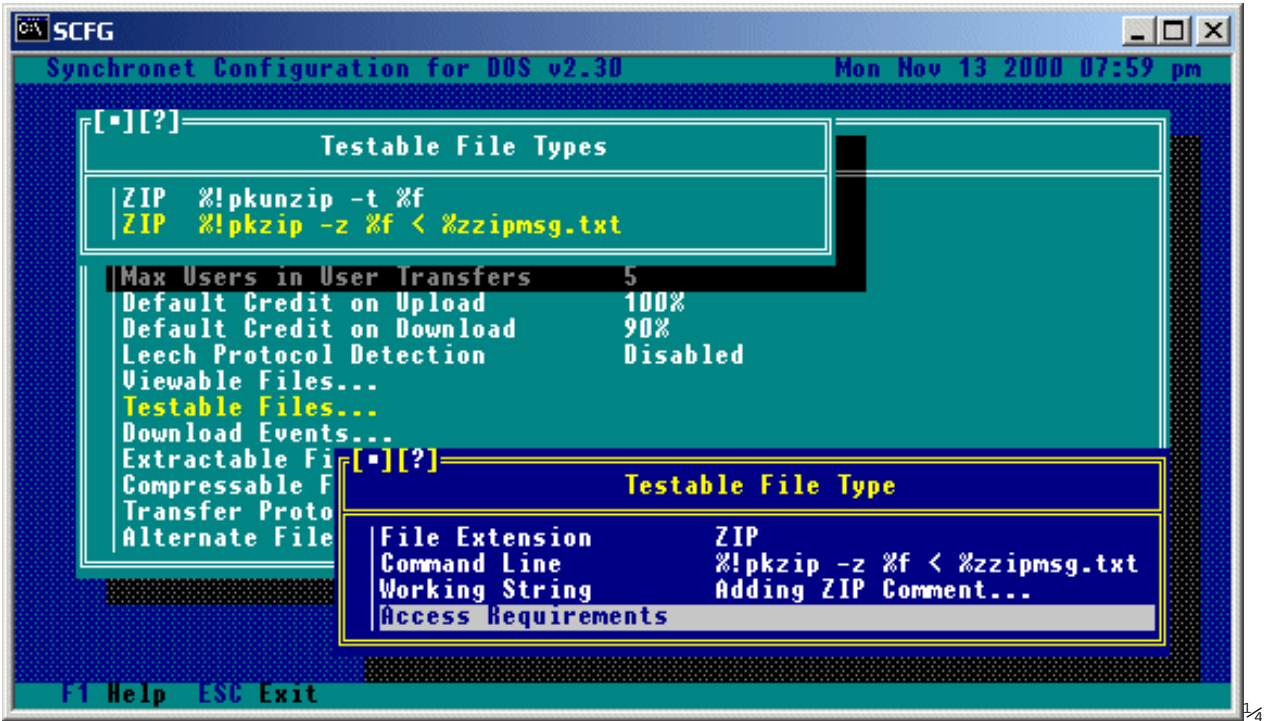
and if one file extension is listed more than once, each command line will be executed. The command lines must return a DOS error code of 0 (no error) in order for the file to pass the test. This method of file testing upon upload is also known as an upload event. This test or event, can do more than just test the file, it can perform any function that the sysop wishes. Such as adding comments to an archived file, or extracting an archive and performing a virus scan. While the external program is executing, a text string is displayed to the user. This "working" string can be set for each file type and command line listed. Shown is an example list of file types, and their respective sub-menus and configurations:



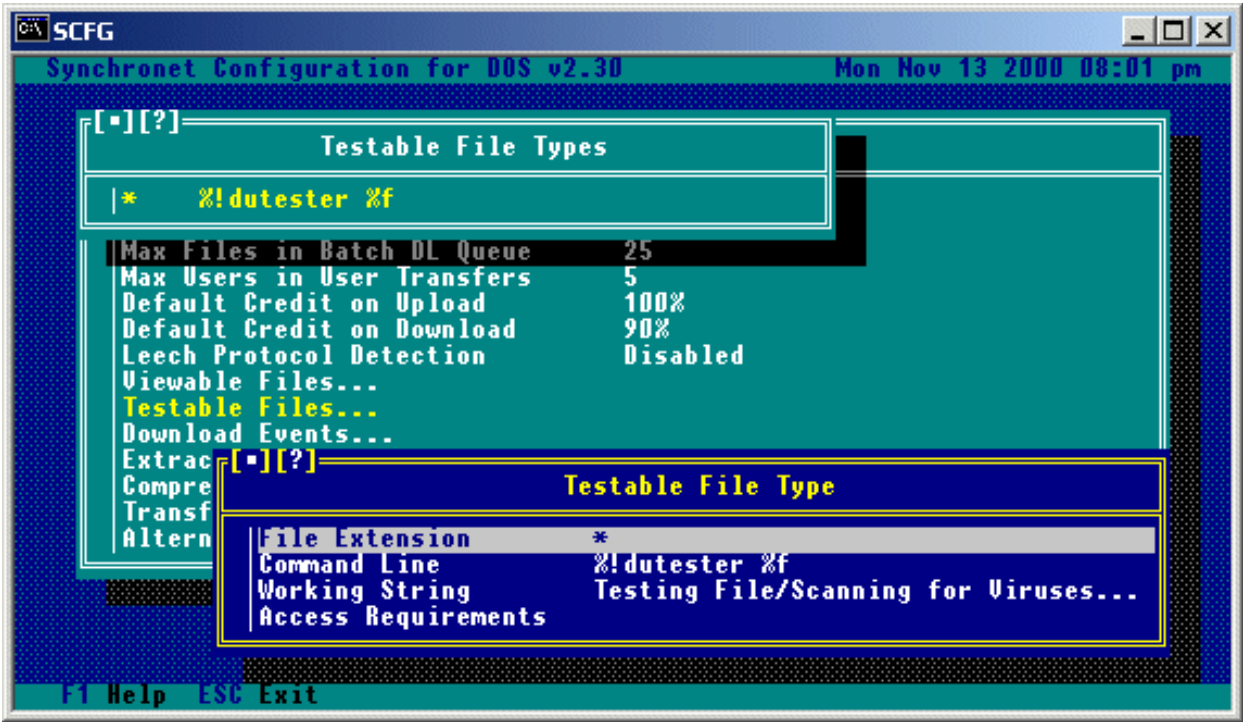
ZIP type files:



To add a ZIP comment to a ZIP type file:



If an extension of '\*' is specified, all files uploaded will be tested with this command line. An example would be when using Domain Upload Tester from Domain Entertainment:

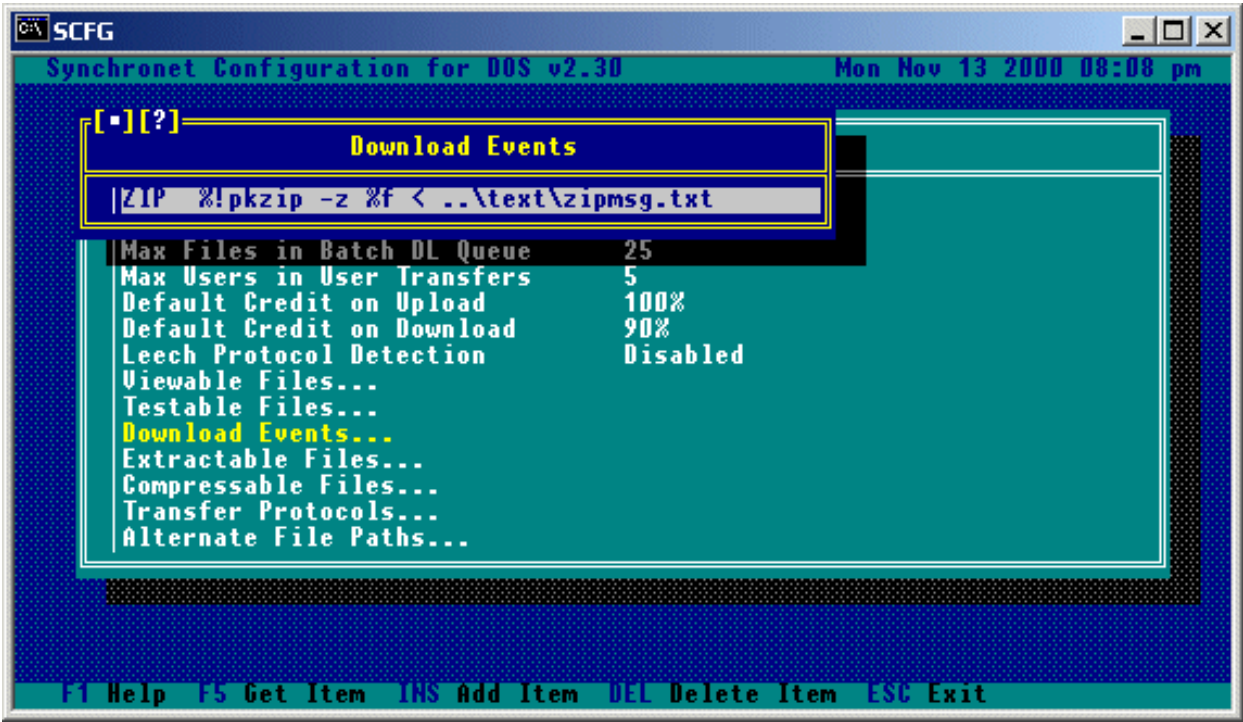


**NOTE:** Users not meeting any access requirements which are set will not be able to perform, or be affected by, that function.

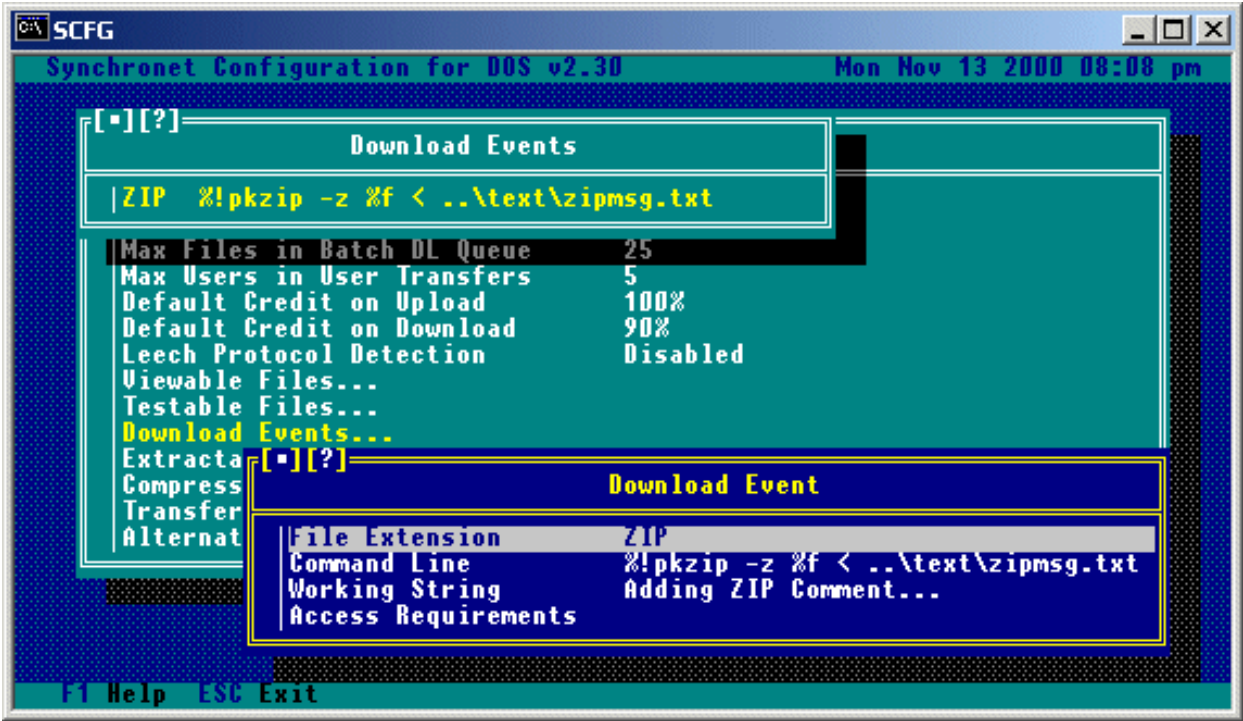
## [10.4] - Download Events

Download events are an event that you wish to have executed before a user downloads a file from your BBS. This can be useful adding a comment to a file from a CD-ROM before it gets sent to the user (as the example shows), or to perform a virus scan, or any other function you can think of.

Here is an example showing how to change the comment in ZIP type files when a user attempts to download a file.



Selecting the available event will allow you to edit the options available.

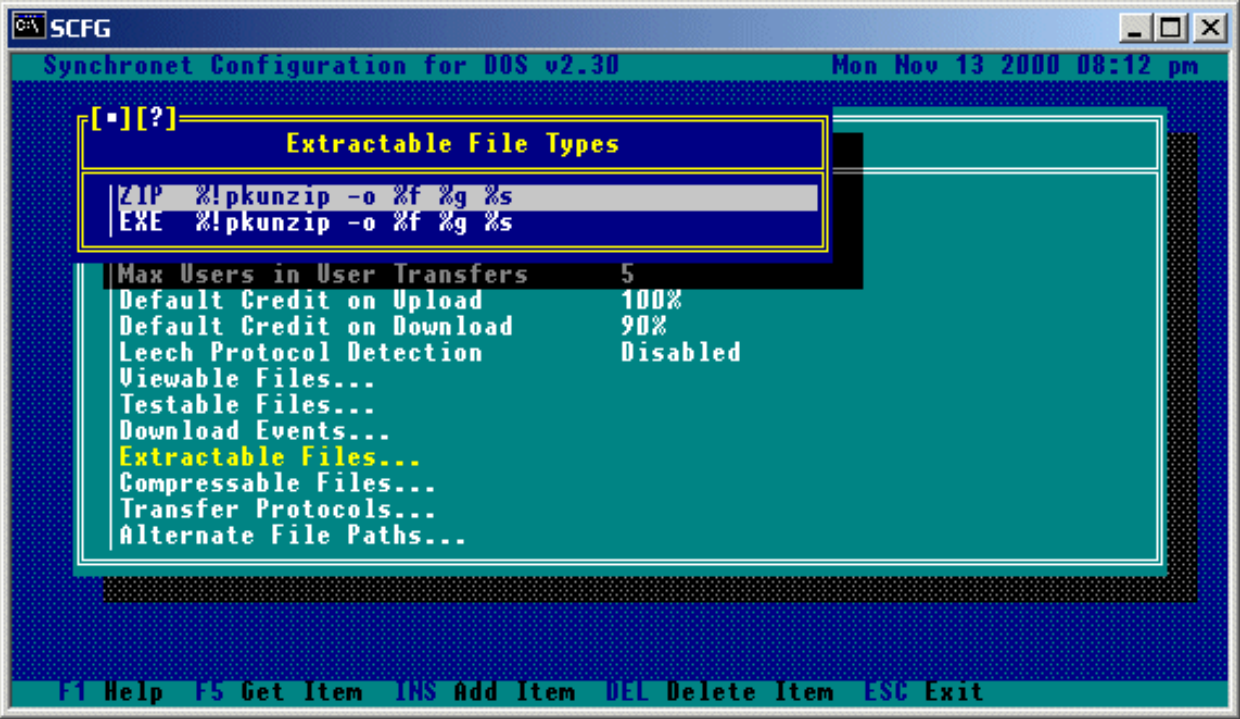


## [10.5] - Extractable Files

This is a list of archive file types that can be extracted to the temp



directory by an external program. The file types are specified by their extension. For each file type you must specify the command line used to extract the file(s). Here are some example command lines for extracting various files:



**NOTE:** When you select a file type from this list, you will be given one additional option (Access Requirements) which is not shown here. Users not meeting any access requirements which are set will not be able to perform, or be affected by, that function.

**Notes on command lines syntax:**

The use of conversion specifiers are supported in all command lines within SCFG. Conversion specifiers are proceeded by a percent (%) symbol and are not case sensitive. The above command line examples use the following specifiers:

**%! EXEC directory:**  
If the program you are executing with this command line is not located in your EXEC directory, you can specify the directory where the program is located or, if the program is in your search path, start the command line with just the program name (no prefix). We suggest specifying the location of the program for security reasons.

**%f Filename:**  
This is the filename of the archive that is to be extracted.

**%g Temp Directory:**  
This is the path to the temp directory. In the above command lines, it is the destination of the extracted files.

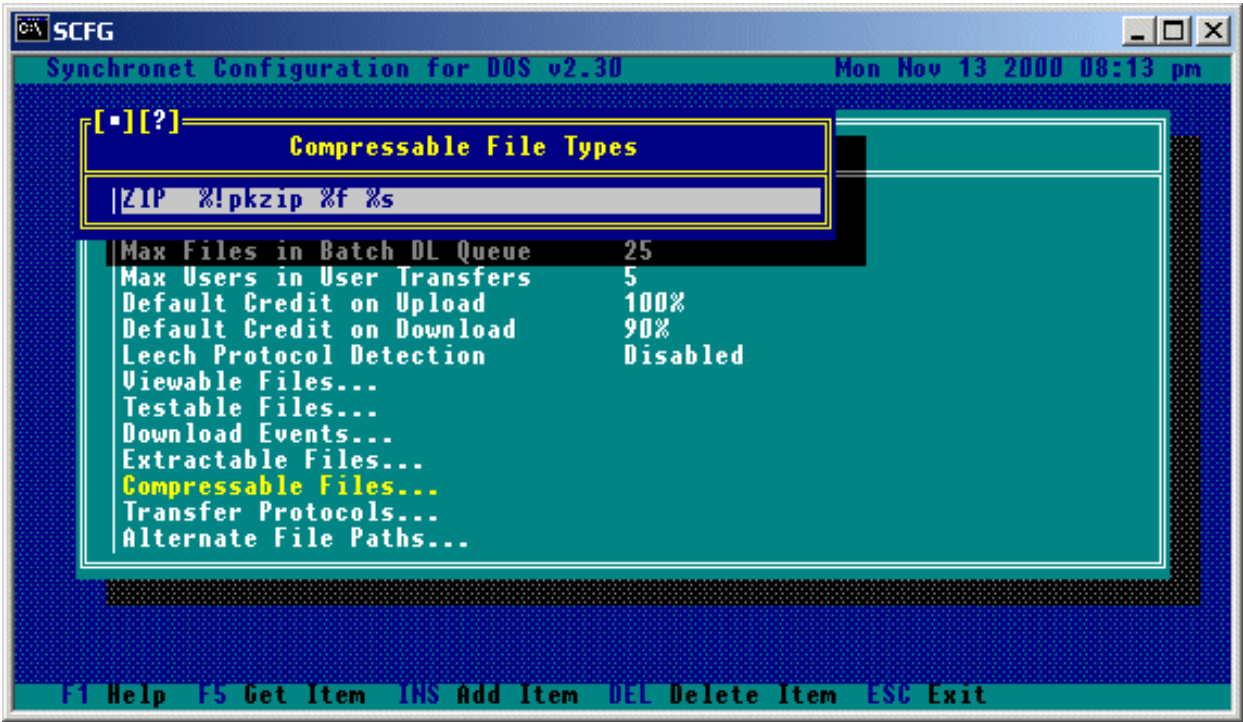
**%s File Specification:**  
This is the file specification to extract from the archive.

See Appendix A for a complete list of the available specifiers.

## [10.6] - Compressable Files

This is a list of archive file types that can be created for QWK packets and temporary archives for download. The file types are specified by their extension. For each file type you must specify the command line used to create the archive. Here are some example command lines for various compression types:





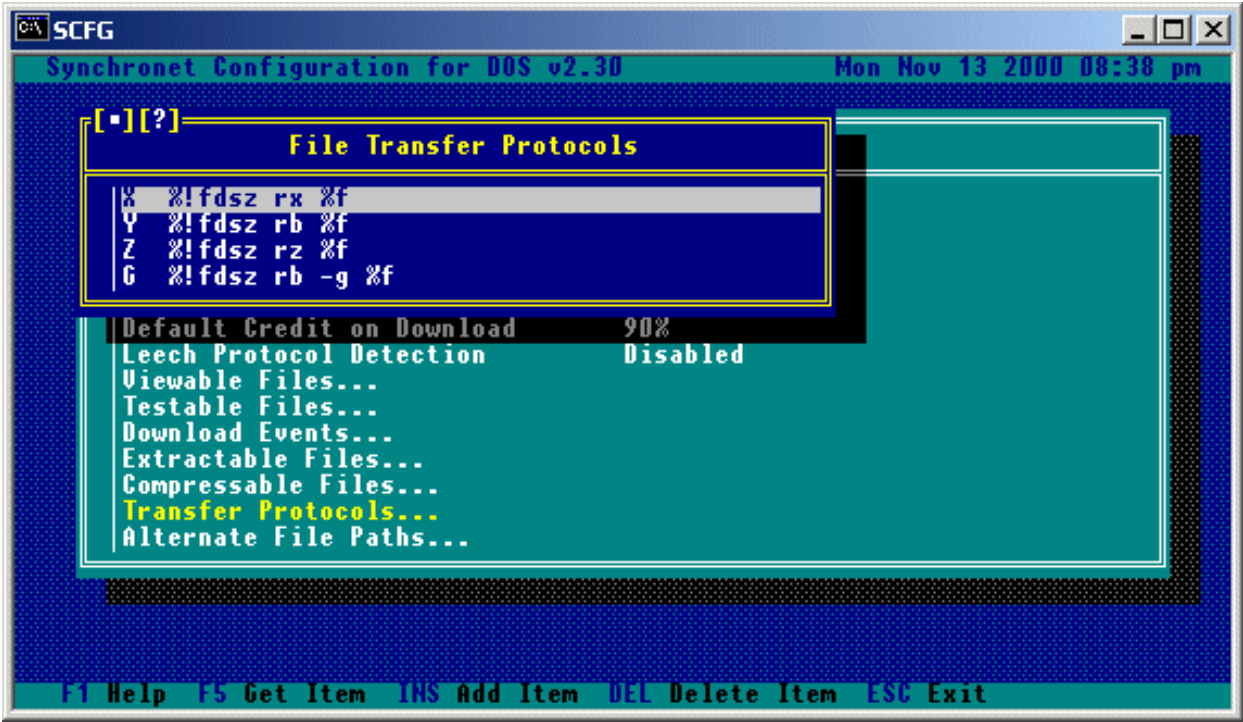
For these compression command lines to work, you must have the appropriate executable files (PKZIP, ARJ, and LHA) in your EXEC directory.

## [10.7] - Transfer Protocols

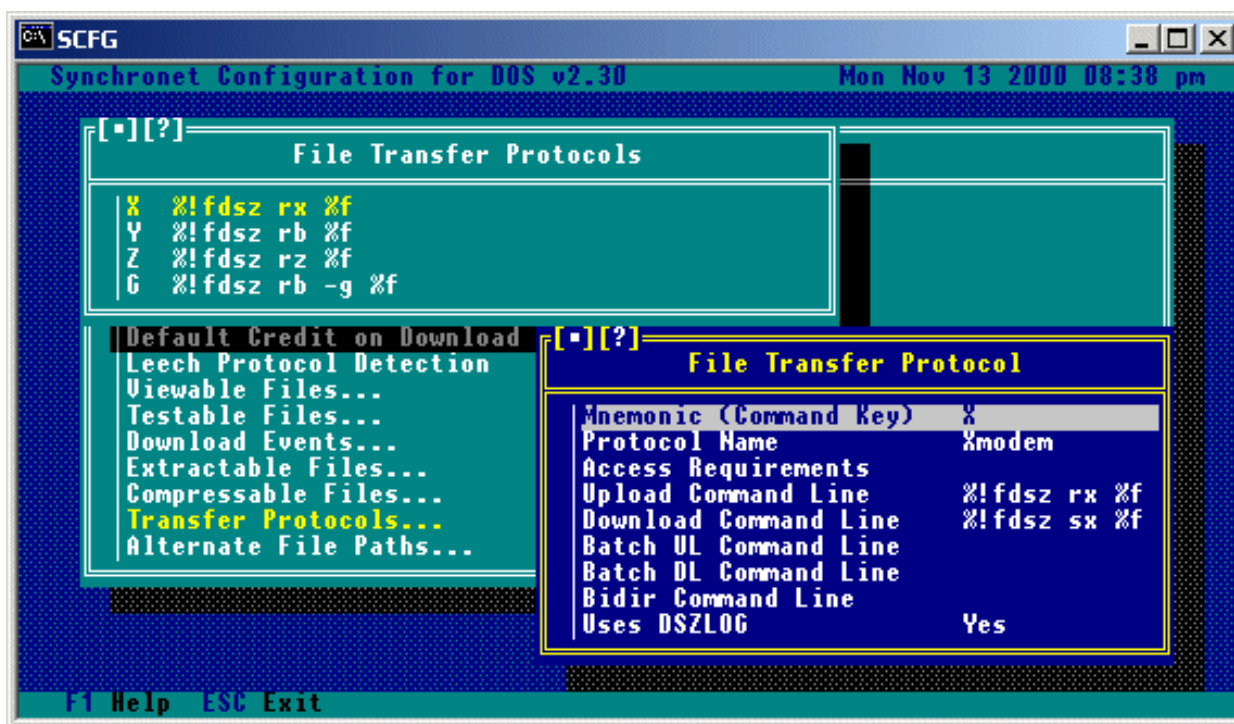
This is a list of file transfer protocols that can be used to transfer files either to or from a remote user. For each protocol, you can specify the mnemonic (hot-key) to use to specify that protocol, the command line to use for uploads, downloads, batch uploads, batch downloads, bi-directional file transfers, and the support of DSZLOG. If the protocol doesn't support a certain method of transfer, or you don't wish it to be available for a certain method of transfer, leave the command line for that method blank. Be advised, that if you add or remove any transfer protocols, you will need to edit the protocol menus (ULPROT, DLPROT, BATUPROT, BATDPROT, and BIPROT) in the TEXT\MENU directory accordingly. The '%f' command line specifier is used to represent the filename or batch file list. The following is an example list of protocols and their respective sub-menus and options:

### [10.7.1] - Version 3.0 (Win32 / Telnet) using FDSZ :

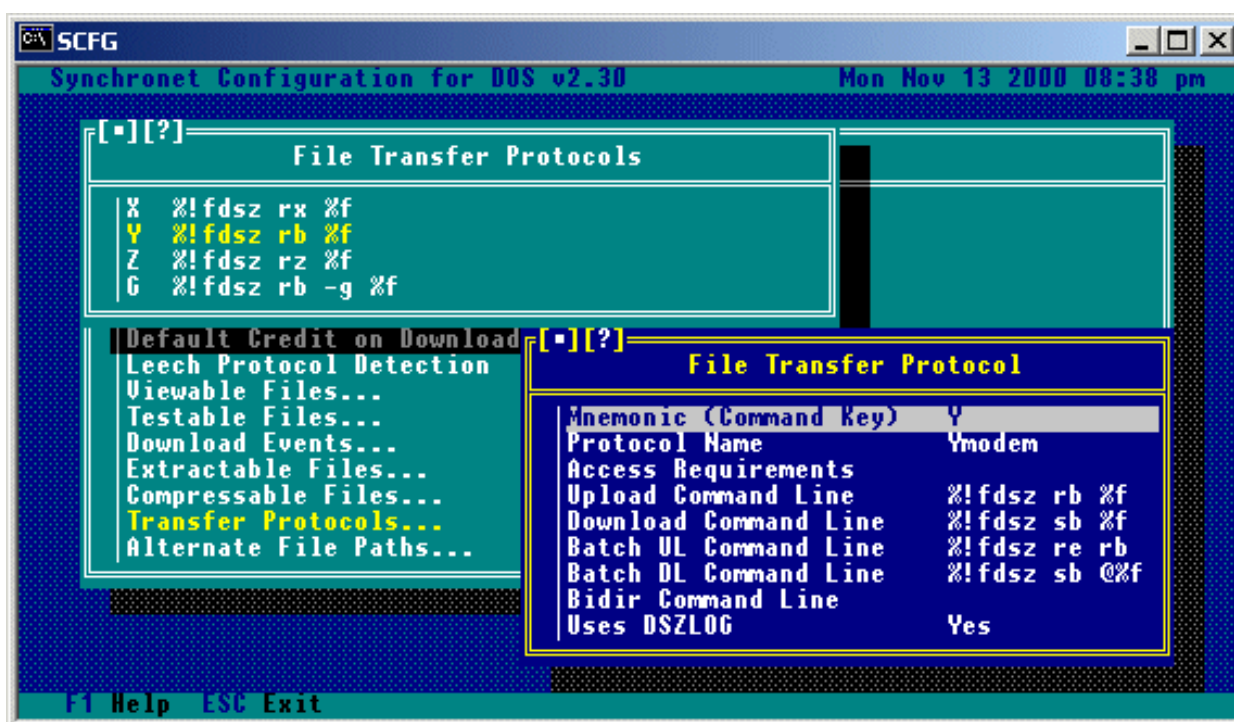
**IMPORTANT:** If you are running Synchronet as a Telnet Server then you must use a Protocol that supports FOSSIL Communications such as FDSZ or CEXYZ. Synchronet v3.0 comes pre-configured for use with FDSZ.



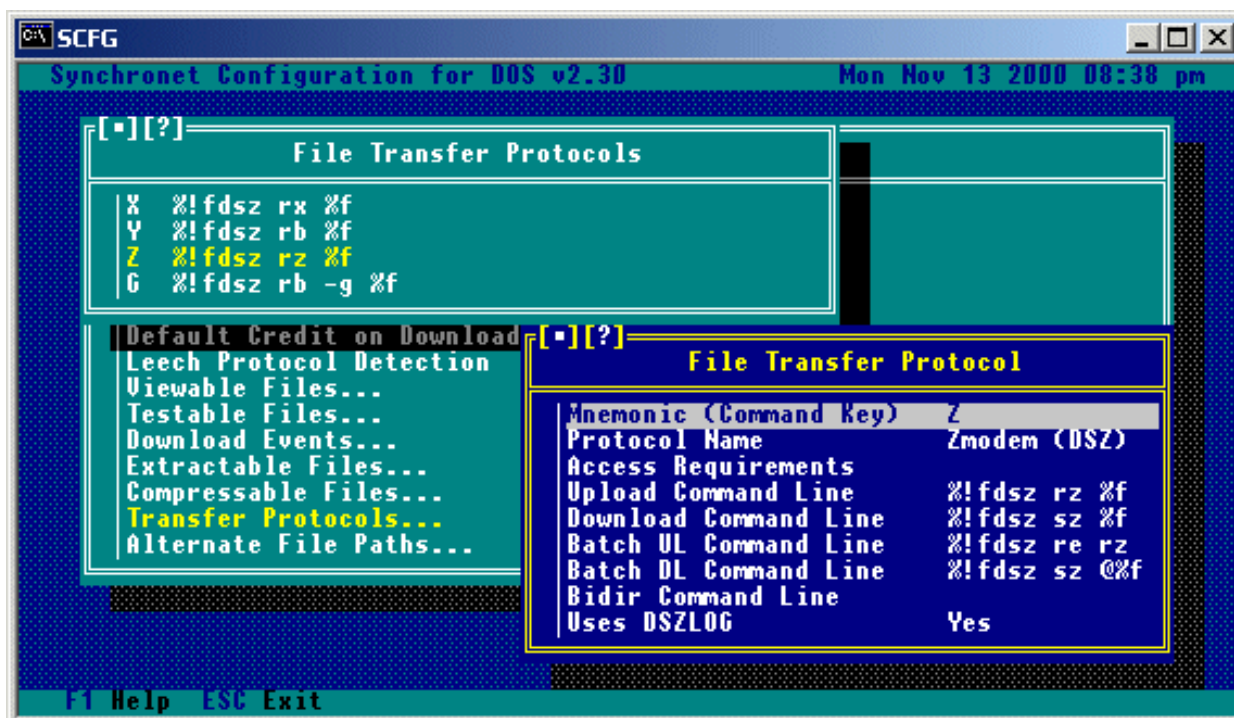
FDSZ Xmodem



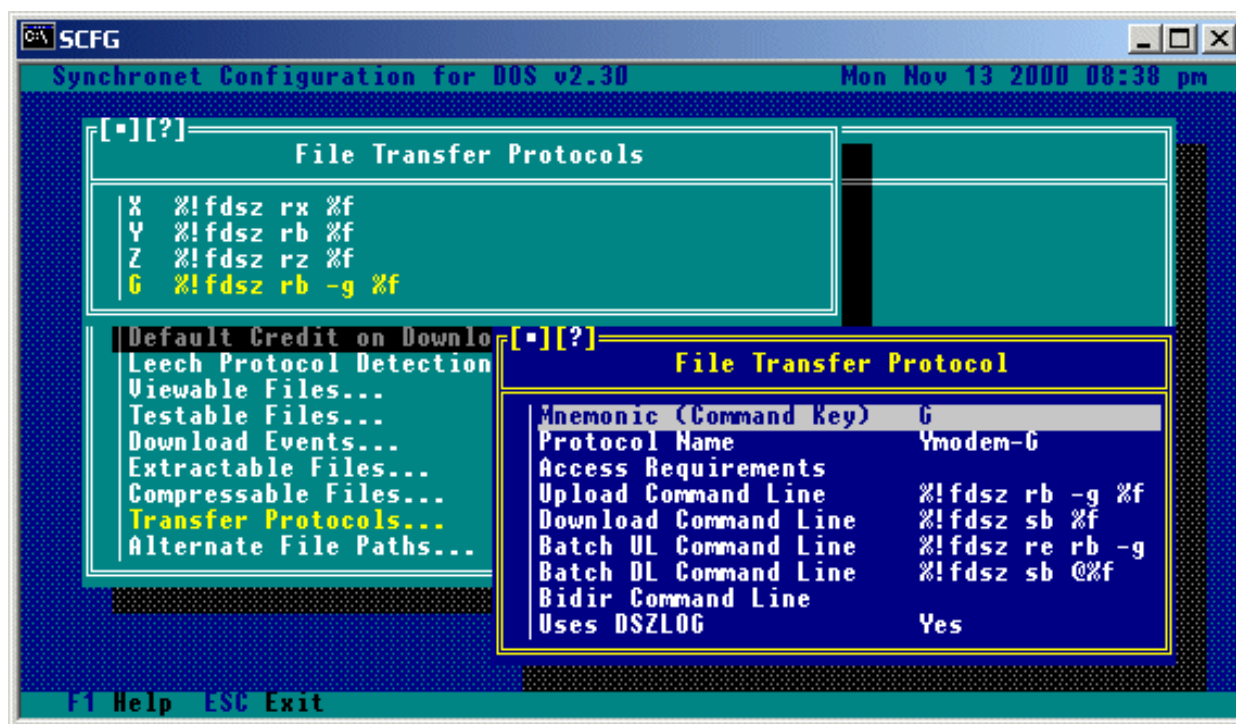
FDSZ Ymodem



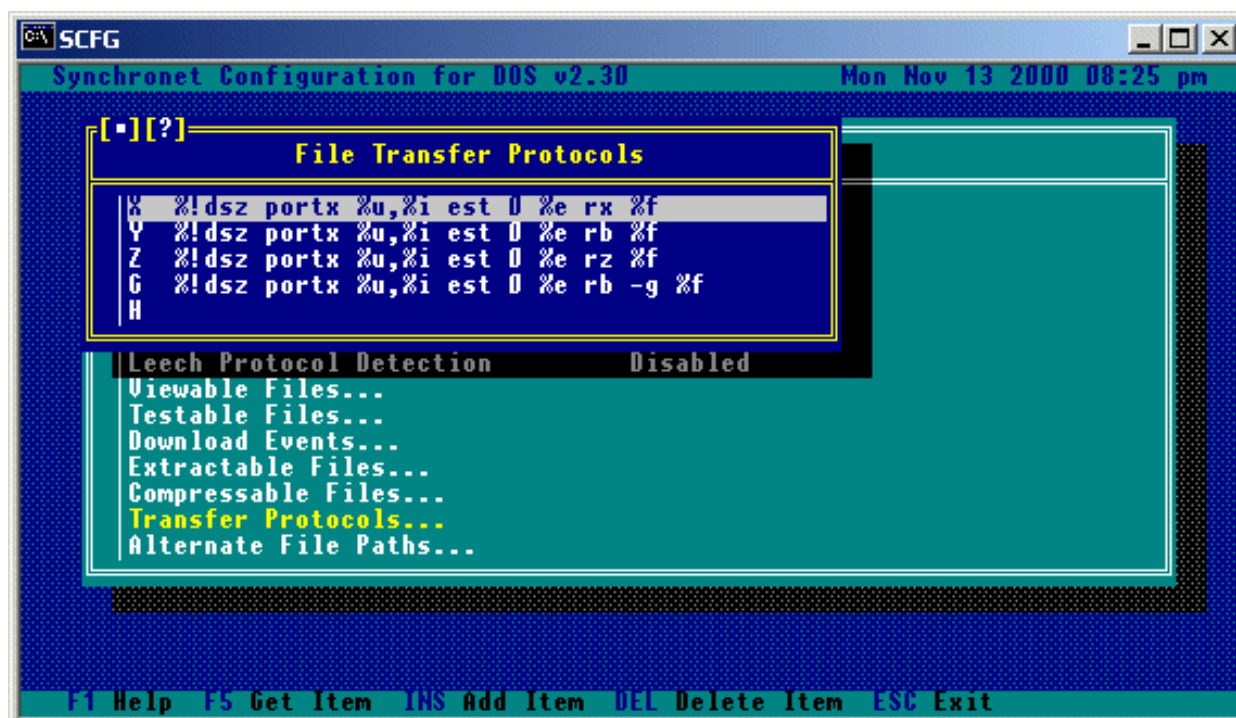
FDSZ Zmodem



FDSZ Ymodem-G



## [10.7.2] - Version 2.30 (DOS/OS2) using DSZ :

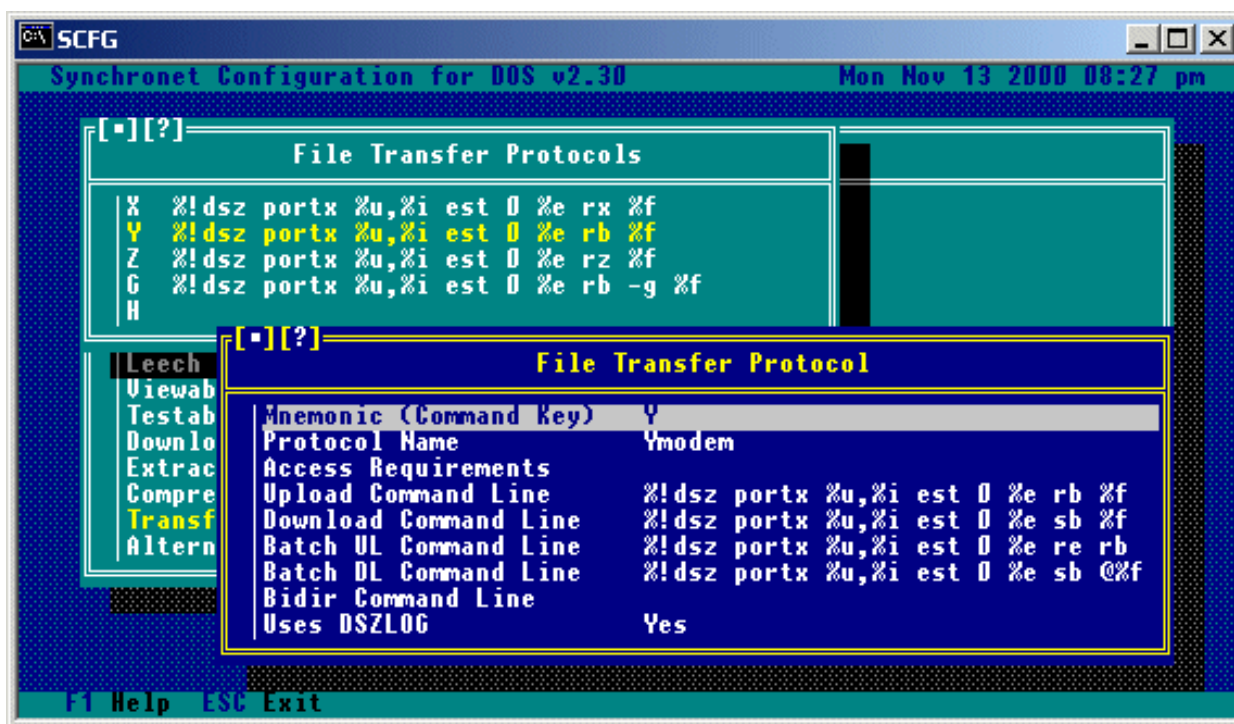


DSZ Xmodem:

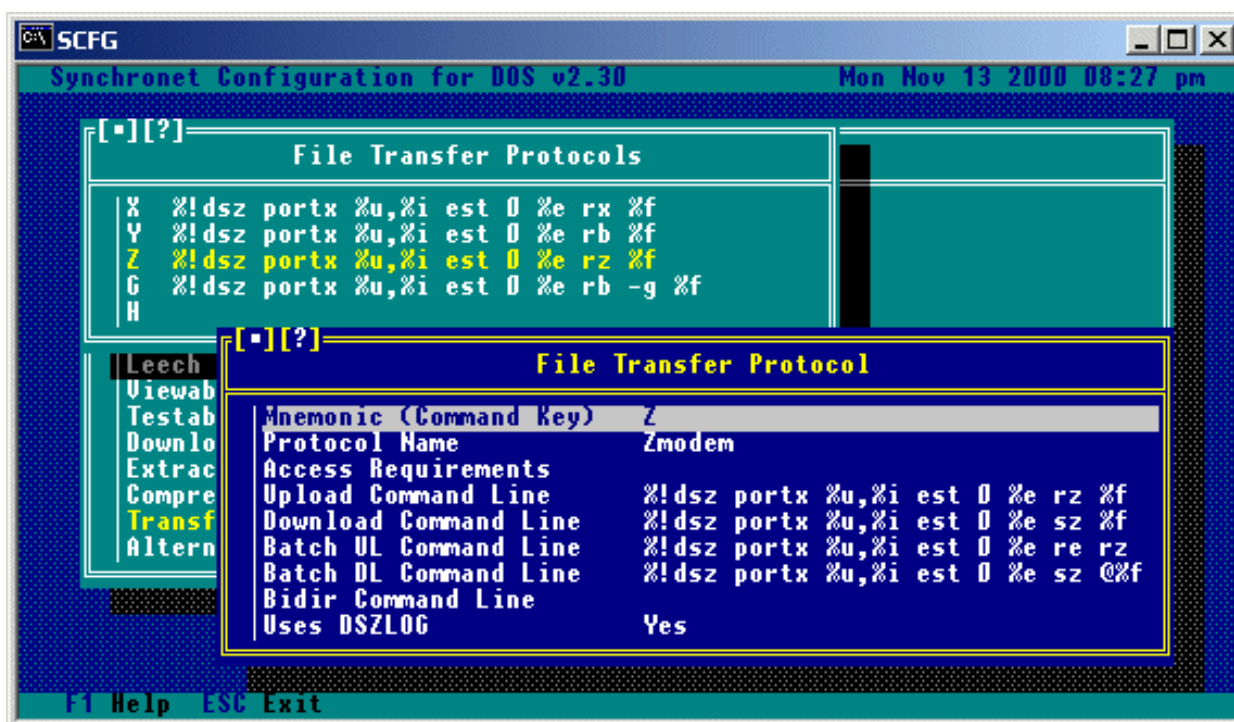


DSZ Ymodem:

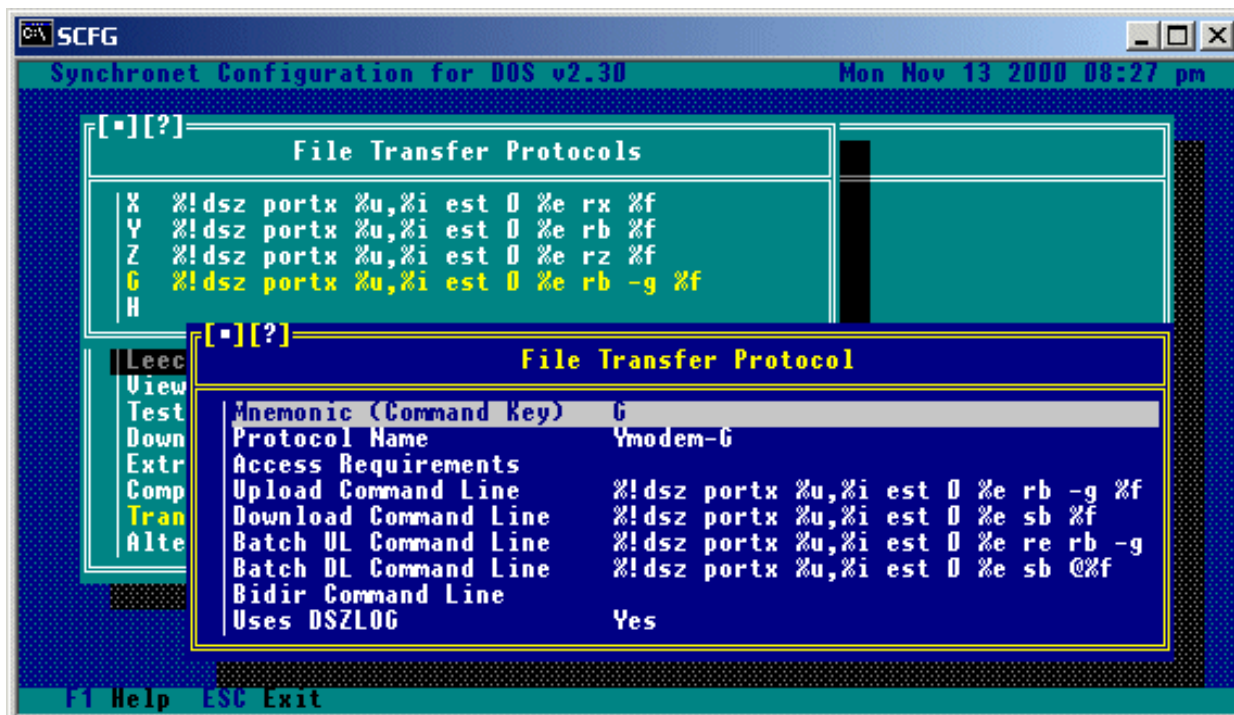




DSZ Zmodem:

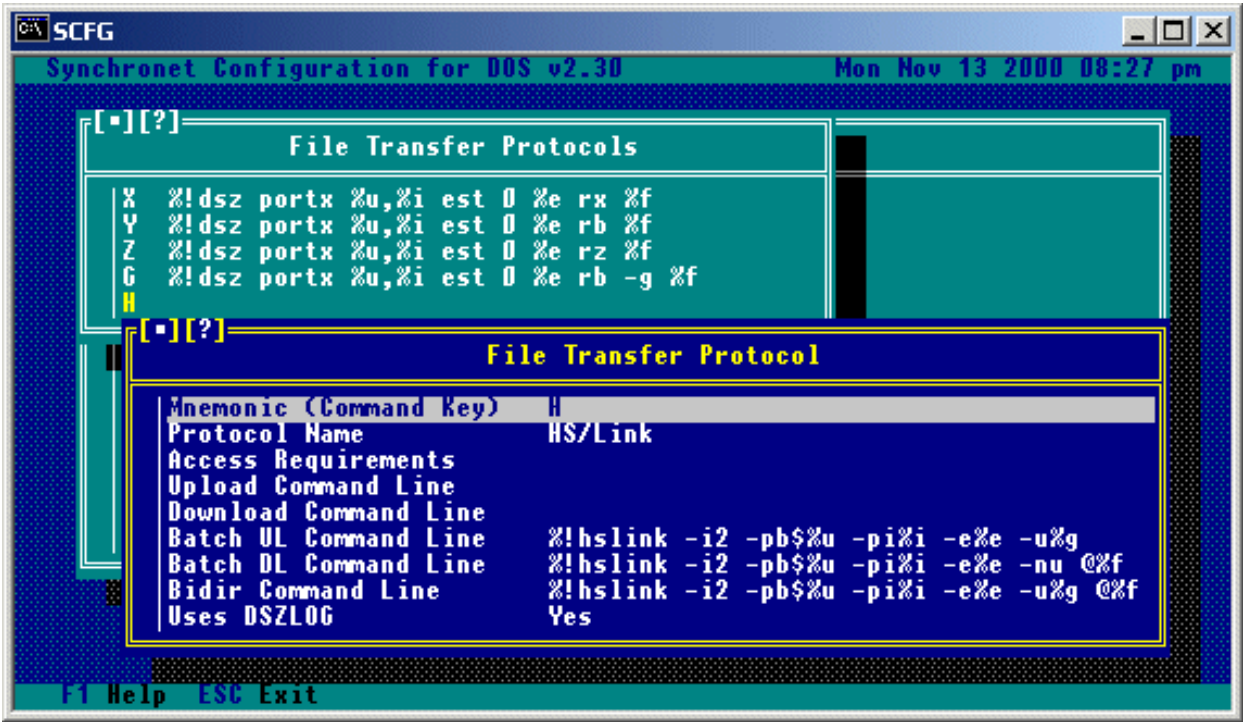


DSZ Ymodem-G:



### [10.7.3] - Version 2.30 (DOS/OS2) using HS/Link :

HS/Link Bi-Directional Transfers:



**NOTE:** Users not meeting any access requirements which are set will not be able to use that protocol.

**IMPORTANT:** If you have Synchronet set to use anything other than UART you will not be able to use the DSZ transfer protocols and will need to locate a set of protocols that will work with your setup. If you are using a UART serial board, there is no reason why Synchronet should not be set to use UART. Just because a front-end mailer or door game requires a FOSSIL driver, it does NOT mean that you must also set Synchronet to use a FOSSIL.

## [10.8] - Creating File Libraries

Select "File Transfers" from the SCFG main menu. Then select "File Libraries". Hit INS (insert key) to create a file library. File libraries are groups of file directories that have a similar subject matter or other common element. File libraries are often used to separate Program files and Data files or files stored on a Hard disk and files stored on CD-ROM. An example configuration that separates programs from data:

Library	Directory
-----	
Programs	Games
Programs	Utilities
Programs	Business
Programs	Communications
Programs	Graphics Programs
Programs	Programming
Data Files	Text (Documentation, Stories, Cheats)
Data Files	Program Source Code (C, ASM, PAS, BAS)
Data Files	Still Pics (GIF, PCX, TIF)
Data Files	Animation (FLI, GL, DL)
Data Files	Sound (MOD, WAV, MID)
Data Files	Other

Example configuration that separates hard disk directories from CD-ROM directories:

Library	Directory
~~~~~	
Hard Disk	Text
Hard Disk	Games
Hard Disk	Utilities
Hard Disk	Business
Hard Disk	Graphics
Hard Disk	Communications
Hard Disk	Graphics Programs
Hard Disk	Programming
Hard Disk	Graphics, Animation, and Sound
GIF CD-ROM	G Rated
GIF CD-ROM	G Rated 640x480
GIF CD-ROM	G Rated 800x600+
GIF CD-ROM	R Rated
GIF CD-ROM	R Rated 640x480
GIF CD-ROM	R Rated 800x600+
GIF CD-ROM	X Rated
GIF CD-ROM	X Rated 640x480
GIF CD-ROM	X Rated 800x600+
PD/ShareWare CD-ROM	Games - Mono, CGA, EGA
PD/ShareWare CD-ROM	Games - VGA
PD/ShareWare CD-ROM	Utilities
PD/ShareWare CD-ROM	Communications

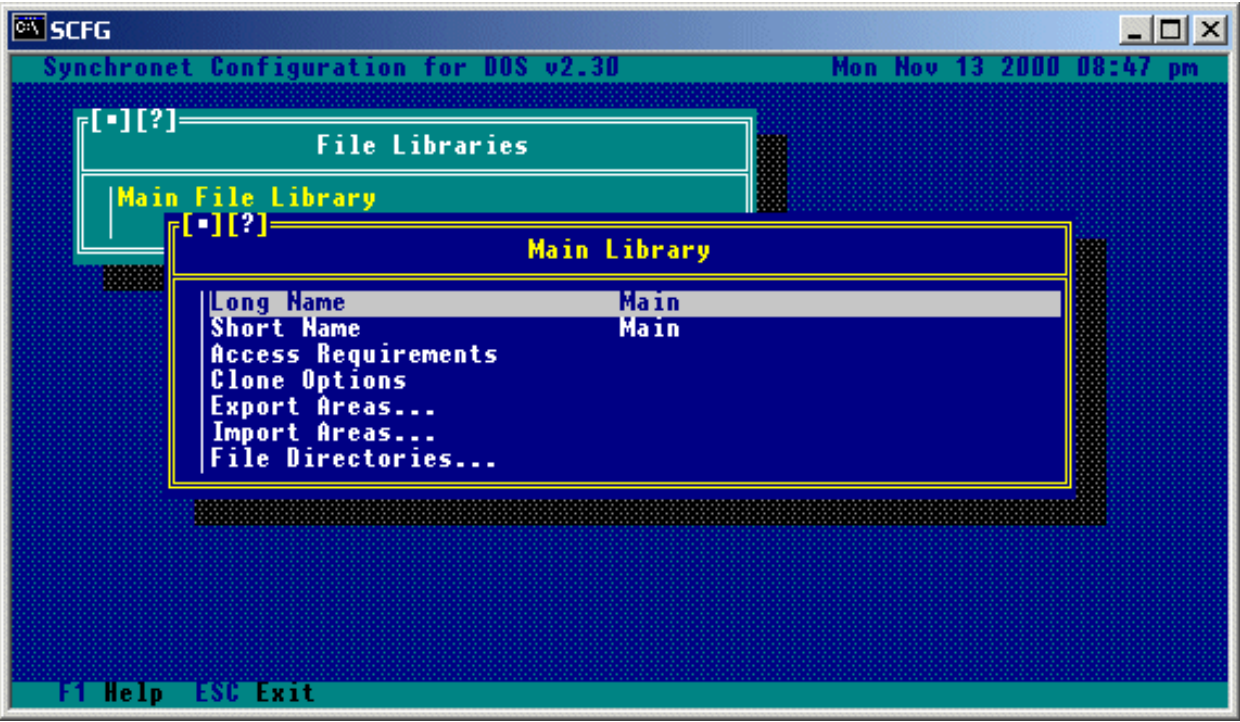


PD/ShareWare CD-ROM Graphics and Desktop Publishing  
PD/ShareWare CD-ROM BBS Software and Utilities  
PD/ShareWare CD-ROM Programming  
PD/ShareWare CD-ROM Windows Programs, Fonts, Icons, etc.

You must have at least one file library and one file directory for anyone to be able to use the file transfer section. Following is a list of options available when creating or modifying a file library.

## [10.8.1] - Library Options

Selecting a library name from the list of available libraries will give you a sub-menu of options for that library.



**Long Name:**  
This is a description of the library displayed in library listings.

**Short Name:**  
This is a description of the library displayed at prompts.

**Access Requirements:**  
These are the requirements which must be met by a user account in order to access this library.

**Clone Options:**  
Use this command to copy the options of the first directory in this library to all the other directories in the library.

**Export Areas:**  
If you wish to export the area information of all the directories in the current file library to a text file, use this option.

**Import Areas:**  
If you wish to import a text file of area information into the current file library, use this option.

## [10.9] - Creating File Directories

After you have created at least one file library, you need to create directories for the library. Creating a file directory is much like creating a file library except that you will be prompted for an internal code to use. The internal code should be an abbreviation of the name of the directory or something else that specifies the contents of the directory.

For each directory, you need to specify the file storage path. This is where the files are actually located (Drive and directory) on your system. If you do not specify a storage path, a sub-directory will be created off of your DATA\DIRS directory with the directory's internal code as the name of the sub-directory. If you wish to use the default storage path (DATA\DIRS\<code>), you'll need to move any files that you wish to be added to the BBS file database to that sub-directory.

If you are adding CD-ROM directories, be sure to set the "Check for file existence" and "Search for New Files" toggle options to "No", and "Slow Media Device" to "Yes" for each of your CD-ROM file directories.

Following is a list of options available in the SCFG when creating or modifying a file directory.

### [10.9.1] - Main Directory Options



**Long Name:**

This is a description of the file directory that is displayed in directory listings.

**Short Name:**

This is a description of the file directory that is displayed at prompts. Some short names have a predetermined meaning:

**Temp:**

This short name is reserved and cannot be used.

**Offline:**

This short name specifies that this directory is to hold offline files for the selected library and is treated a bit differently than other directories. It will not be included in new-scans and will be the default location of files that are removed or moved. A directory with this short name should have the upload level set to 90 and the access level and flags set the same as the library. There can only be one directory per library with this short name.

**Sysop:**

This short name specifies that this directory will be the destination for files uploaded by users with the 'Z' command from the transfer section. There should only be one directory with this short name and it should belong to the most accessible library. A directory with this short name should have the access level set to 90 and the upload level and flags set the same as the library.

**User:**

This short name specifies that this directory will be the storage point for user to user transfers. There should only be one directory with this short name and it should belong to the most accessible library. Users can upload a file to another user or group of users with the '/U' transfer section command. Users download files sent to them with the '/D' transfer section command. A directory with this short name should have the access level set to 90 and the upload level and flags set the same as the library or at whatever level the sysop wishes to allow users to use the user to user transfer facilities.

**Uploads:**

This short name specifies that this directory will be used for uploads if the user attempts to upload to a directory where he or she does not have sufficient upload access. If you wish all uploads to automatically go into one specific directory, set the required upload level for all other directories to 90 or higher (Sysop). If you don't want users to be able to see the files in this directory or download them until they are moved by the sysop, set the required access

for your "Uploads" directory to level 90 or higher. Creating an "Uploads" directory also allows "Blind" batch uploads from the users (files need not be added to the upload queue first).

**Internal Code:**

This is an internal code for SBBS to distinguish this directory from the others on the system. This must be a unique name of up to 8 valid DOS filename characters. The code TEMP is reserved and cannot be used.

**Access Requirements:**

This is a list of security requirements to access this directory.

**Upload Requirements:**  
This is a list of security requirements to upload to this directory.

**Download Requirements:**  
This is a list of security requirements to download from this directory.

**Operator Requirements:**  
Any users meeting these criteria will be able to perform Sysop type activities for this directory.

**Exemptions Requirements:**  
Any users meeting these criteria will be considered exempt from credit deductions when transferring files from this directory.

**Transfer File Path:**  
This is the actual path for the storage of the files that belong to this directory. If no path is specified, the directory DATA\DIRS\CODE, where CODE is the internal code for this directory, will be used to store the files.

**Maximum Number of Files:**  
This is the maximum number of files that will be allowed in this directory. Once this number of files is reach, no uploads will be allowed.

**Purge by Age:**  
This is maximum age of files (in days) to keep in this directory. The DELFILES utility (included with Synchronet) must be run as a timed event to automatically purge old files from your file areas.

**Credit on Upload:**  
If Credit Uploads is set to Yes in the toggle options for this directory, users will be credited with this percentage of the file size (in bytes) in credits when uploading a file to this directory.

**Credit on Download:**  
If Credit Downloads is set to Yes in the toggle options for this directory, users will be credited this percentage of credits for subsequent downloads of any file that they upload to this directory.

[10.9.2] - Toggle Options



**Check for File Existence:**  
If this option is set to 'Yes', when files are listed in the transfer section, each file is checked to see if it actually exists in the DOS directory. If the file doesn't exist, the credit value is displayed in high intensity blinking and is followed by a minus symbol. Setting this option to 'No' speeds up the file listings, but files not actually on disk will appear just as the others. Directories with a short name of Offline should have this value set to 'No' as they should solely contain files that aren't actually in the DOS directory. Directories that are stored on CD-ROM or other slow random access device should have this option set to 'No'.

**Slow Media Device:**  
If this option is set to 'Yes', files will be copied from the normal storage directory into the temp directory and downloaded from there. This option is helpful in reducing the overhead associated with multiple simultaneous user access to a slow media device such as

CD-ROM.

**Force Content Ratings:**

If this option is set to 'Yes', when a user uploads a file he is prompted to rate the content of the file with a single character (usually G, R, or X), and the file description will begin with the rating letter contained in brackets.

**Upload Date in Listings:**

Setting this option to 'Yes' will cause Synchronet to place the date that a file was uploaded onto the first line of the file description, similar to the Multiple File Numberings option.

**Multiple File Numberings:**

If this option is set to 'Yes', when a user uploads a file he is asked if the file he is uploading is part of a set of files. If the user answers 'Yes', he is then prompted for the total number of files, the number of the file he is uploading, and the file description will end with the numbering in the format "[n/t]", where n is the file number and t is the total number of files.

**Search for Duplicates:**

If this option is set to 'Yes', this directory will be searched for duplicate filenames when a user attempts to upload a file.

**Search for New Files:**

When this option is set to 'Yes' this directory will be included in new file scans (for those users than have access). Set this option to "No" for CD-ROM directories or directories that will not be receiving new files on a regular basis.

**Search for Auto-ADDFILES:**

When this option is set to 'Yes' this directory will be searched for FILES.BBS by the ADDFILES program when using the auto-add feature.

**Import FILE\_ID.DIZ:**

When set to 'Yes', Synchronet will attempt to import the FILE\_ID.DIZ or DESC.SDI of any files uploaded to this directory into the file description.

**Free Downloads:**

If you want all files downloaded from this directory to be free for the downloader (not cost any credits regardless of the credit value), set this option to 'Yes'.

**Free Download Time:**

If you do not want time spent during downloading to be subtracted from the users time online, this option should be set to 'Yes'.

**Deduct Upload Time:**

If you want the time spent uploading a file to be subtracted from the users available online time, this option should be set to 'Yes'.

**Credit Uploads:**

If you want users who upload files to this directory to get credit for their upload based on the "Credit on Upload" percentage, set this value to 'Yes'.

**Credit Downloads:**

If you want users who upload files to this directory to get credit for subsequent downloads of the file based on the "Credit on Download" percentage, set this value to 'Yes'.

**Credit with Minutes:**

Setting this option to 'Yes' will cause the uploading user to receive time (in minutes) for their upload rather than credits. The minutes a user receives will be based upon the percentage awarded multiplied by the time it took for the recipient of the file to download it.

**Anonymous Uploads:**

If you want users with the 'A' exemption to be able to upload file anonymously to this directory, set this option to 'Yes'. If you want all uploads to this directory to be automatically forced anonymous, set this option to Only.

**Purge by Last Download:**

This option is for use in conjunction with the Synchronet DELFILES program. When deleting files from the filebase, if this option is set to 'Yes', it will compare the maximum age specified by the "Purge by Age" field to the date the file was last downloaded. If set to 'No', the maximum age will be compared to the date the file was uploaded.

**Mark Moved Files as New:**

If this option is set to 'Yes', then when a file is moved from this

directory to another, it's upload date will be reset to the current date/time automatically.

### [10.9.3] - Advanced Options



**Extensions Allowed:**  
This is a list of file extensions (separated by commas) that are allowed to be uploaded to this directory.

**Data Directory:**  
This is the path where the data for this directory will be located.

**Upload Semaphore File:**  
This is the path and name of a semaphore file used to trigger your front-end mailer when a file is uploaded.

**Sort Value and Direction:**  
Directories can be sorted either by filename or upload date in an ascending or descending order. This option selects the value to sort on and in which direction.

### [10.10] - IMPORTANT - Adding files to the BBS database

Now that you've created at least one file library and one file directory inside that file library, any files located in the storage path for that directory are not automatically available for users to download. The file has to be added to the BBS file database first.

#### Manual Upload

Logon to the BBS as yourself (most likely, user #1), go to the file transfer section. Select the file area you wish to add the file to and hit 'U' for upload. Now enter the filename. If the file is already in the storage directory that you specified in SCFG you will be prompted for file descriptions and other questions about the file. If the file is not already in the storage directory, you will be prompted for the path from where to copy the file.

#### ADDFILES

If you have an ASCII list of your files with descriptions (often called FILES.BBS or DIR##), you'll want to use the Synchronet ADDFILES program to import this file and description information into the BBS file database. This is usually only the case for CD-ROMs and when converting a file database from another BBS. See the Utility Reference Chapter for more information on ADDFILES.

#### Bulk Upload

To add the files that are located in the storage paths of your configured directories, exit the configuration program, logon to the BBS as yourself (User #1), go to the file transfer section ('T' from the main menu), and type ";UPLOAD ALL" then hit ENTER. If there are any files that are in your storage paths, but not already in the BBS file database you will be prompted to enter a description for each file.

### [10.10.1] - Creating Offline File Directories

Each library can have an offline file directory. This is where you can place files that have been removed from the system, but the file information (name, description, uploader, etc.) remains in the database. Offline directories are treated special. They are not scanned for new files and are automatically



used as an optional destination for the file information when a file is removed by a sysop. Offline file directories are specified by having the short name of "Offline" (not case sensitive). Only one offline directory can be specified per library. Offline file directories should have the "Check for file existence" toggle option set to "No". The minimum security level to upload should be set to 90 so that users will not be allowed to upload to this directory.

### [10.10.2] - Creating a Sysop Directory

A sysop file directory is a directory that users can upload to, but they can not see nor download from. You can only have one sysop directory on the system and it can belong to any one of the file libraries. A sysop file directory is specified by the short name of "Sysop" (not case sensitive) and will be used as the destination directory for uploads with the 'Z' (upload to sysop) transfer section command. The minimum security level to access should be set to 90 or higher so that only sysops can view the directory and download from it and the upload access level should be set to 0 (or higher, if you want to restrict the number of users who can upload to the sysop dir).

### [10.10.3] - Creating a User-to-User Transfer Directory

A user-to-user transfer directory is used for users to send a file directly to another user (or group of users) with only that user (or group of users) being able to see the file and download it. It's like a private e-mail file directory. Users send files to other users with the "/U" command and download files from other users with the "/D" command. This feature is only enabled if the sysop creates a user-to-user transfer directory. This directory can belong to any library on the system, but must have a short name of "User" (not case sensitive). The minimum security level to access should be set to 90 or higher so that only sysops can view the directory and download any file they wish from it and the upload access level should be set to 0 (or higher, if you want to restrict the number of users who can upload user-to-user files). This directory should have the "Allow extended descriptions" toggle option set to "Yes" so that the uploader of the file can attach a more detailed description if he or she wishes.

### [10.10.4] - Creating a Default Upload Directory

A default upload directory is used if you want all uploads to go to a single directory. If you wish to use this feature, you must create a directory with the short name up "Uploads" (not case sensitive). Set the minimum level to access to 90 or higher if you do not wish to allow users to see this directory. Set the upload access level to 0 (or higher, if you want to restrict the number of users who can upload to this directory). You should set the required upload access level of all other directories in the library to 90 or higher so that all upload attempts to those directories will force the file uploaded into this directory.

### [10.10.5] - Supporting Blind Batch Uploads

Some sysops would like for their users to be able to do a batch upload of files to a directory without having to enter the files into an upload queue before hand. These are called Blind Batch Uploads. In order for this feature to work, you MUST have created a default upload directory (see the section above describing this procedure). Once you have created a default upload directory, the user need only go to the Batch Transfer menu and begin his upload.

## [10.11] - CD-ROM and Other Slow Media Devices

It is assumed that you have already installed your CD-ROM drive and any necessary device drivers so that it can be accessed as a DOS device (like a hard disk or floppy disk).

Explore your CD-ROMs directories to find out what directories contain what kind of files. Make notes of the directory paths and a description you would like to give that directory. Example:

Path	Description
-----	
D:\G-GIFS	G Rated GIFs
D:\GAMES	Games
D:\UTILS	Utilities

Now find the ASCII description lists on the CD-ROM for the individual directory contents and note which description files are for which directories. If multiple list formats are supplied, use the format that most closely matches this (FILES.BBS format):

FILENAME.EXT This is a file description

or the DIR## format:

```
FILENAME.EXT      530114  07-16-91  This is a file description that will be too
                                | long, so it wraps to the next line.
```

Example for file list notes:

File List	Description
-----	
D:\DESC\GIFS.LST	GIFs
D:\DESC\GAMES.LST	Games
D:\DESC\UTILS.LST	Utilities

Or preferably:

File List	Description
-----	
D:\GIFS\FILES.BBS	GIFs
D:\GAMES\FILES.BBS	Games
D:\UTILS\FILES.BBS	Utilities

Now run SCFG from your node directory. Go to "File Transfers", then "File Libraries". Hit the INS key to add a new file library. Name it "CD-ROM" or a description of what the CD-ROM contains. Now hit ENTER on the new library and select "File Directories". Hit INS to add each directory specifying the name and file storage path for each. Example:

Long and Short Name	Internal Code	File Path
-----		
GIFs    CD-GIFS    D:\GIFS		
Games	CD-GAMES	D:\GAMES
Utilities	CD-UTILS	D:\UTILS

Now hit HOME and ENTER to edit the first directory. Select "Toggle Options" and hit enter. Set "Check for File Existence" and "Search for New Files" to "No" and set "Slow Media Device" to "Yes". Hit ESC three times to go back to the "CD-ROM File Library" menu. Select "Clone Options" and hit ENTER. Answer "Yes" to clone the options of the first directory into the rest of the library.

Now exit SCFG and save changes.

From the DOS prompt, type:

```
SET SBBSNODE=C:\SBBS\NODE1
```

But use the correct path for your NODE directory. If you installed Synchronet into a directory named "BBS" on drive D:, then you would type

```
SET SBBSNODE=D:\BBS\NODE1
```

Now run ADDFILES (from your EXEC directory) for each of your CD-ROM file description lists (or only once if a FILES.BBS exists in each directory). See the Utility Reference chapter for specifics on how to use ADDFILES. Example (assuming description lists are in DIR## format):

```
ADDFILES CD-G-GIFS +D:\DESC\G-GIFS.LST 33 13
ADDFILES CD-R-GIFS +D:\DESC\R-GIFS.LST 33 13
ADDFILES CD-X-GIFS +D:\DESC\X-GIFS.LST 33 13
ADDFILES CD-GAMES +D:\DESC\GAMES.LST 33 13
ADDFILES CD-UTILS +D:\DESC\UTILS.LST 33 13
```

The following command will search ALL configured directories for FILES.BBS lists and import the files from them:

```
ADDFILES *
```

If you have a FILES.BBS format list which has a different filename, this will search ALL configured directories for that file and import the files from them:

```
ADDFILES *FILES.LST
```

These will add the file descriptions to your Synchronet file database. Now you should be able to run SBBS and have users download files from your CD-ROM.

### [10.11.1] - Alternate File Paths

Since a CD-ROM disk is "read only", you cannot alter the physical organization of files and directories on the CD-ROM. Normally, you would have one logical Synchronet file directory for every physical DOS directory. But for CD-ROMs that have their files spread across a large number of directories, you may not want your logical directory structure to mimic the physical structure. Synchronet can store roughly a thousand files per directory and many CD-ROMs

have far less than that per directory to accommodate BBS software that have a lower maximum number of files per directory. Each logical directory in Synchronet takes up memory so it is advantageous to keep the number of directories to a minimum when working with large quantities of files.

Synchronet's alternate file paths allow the sysop to add files from multiple physical directories into one logical directory on the BBS. Every logical directory on the BBS has a default storage path where uploaded files will be received and downloaded files are sent from. If you wish to add files to an existing logical directory that are not stored in the default storage path, then you will need to add an alternate file path that points to the location of these files. Then either use ADDFILES with the ".altpath" switch or use the ;ALTUL and ;UPLOAD sysop transfer commands to add the files. You will need to know the number of the alternate file path for which you are adding files from, so note which alternate path number you are uploading from before running ADDFILES or ;UPLOAD.

Alternate file paths are useful for grouping files of a similar subject from different CD-ROMs into a single logical directory in Synchronet, thus freeing the sysop to organize the files based on content rather than physical storage location.

## [10.12] - Internet FTP Server (*v3+ Only*)

### FTP Filename Aliases

You can create a list of file aliases that will appear in your FTP root directory for:

- Quick and easy access to often downloaded user files
- Static filenames that corresponds to a dynamically changing filenames  
e.g. ftp://vert.synchro.net/sbbs\_for\_dos.zip always points to the current versioned filename of Synchronet for DOS (e.g. SBBS230B.ZIP)
- Download access to files on the local disk not in the BBS file database

How? Edit the file **CTRL/FTPALIAS.CFG** (*SBBSCTRL->FTP->Edit->Alias List*). The format is similar to that of the Synchronet Mail Server **ALIAS.CFG**, one alias per line in the format:

#### alias path description

The alias is not case-sensitive and may not contain spaces. This is the filename that will appear in your FTP root directory.

The path element is either the full path and filename to a file on a local file system (e.g. **C:\DOCS\MYFILE.TXT**) or a virtual path to a file in the BBS file database. Virtual paths are specified as **"bbs://lib/dir/filename"** where lib is the library short name, dir is the directory internal code and filename is the actual filename of the referenced file (may be the long filename, but may not contain spaces).

The description is the optional description of the file that will be used in the dynamically generated Auto Index file (if you have this option enabled).

Lines beginning with a ';' are considered comments and are ignored.

Example Alias:

**sbbs\_for\_dos.zip bbs://main/sbbs/sbbs230b.zip Current version for SBBS for DOS**

### Sysop Access

To login with system operator access, you must login with a user account that has a security level of 90 or higher (user #1 usually has level 99) and enter your password as **"pass:sypass"** where pass is your personal password and sypass is the system password.

A successful sysop login will generate a login message stating **"Sysop access granted"**. Logging in with just your personal password will give you normal (non-sysop) access.

Sysop access allows you to mount local file systems and will enable secure BBS database features in the future (such as deleting/renaming files).

### Local File System Access

This is a powerful new feature that allows the sysop to access any and all locally accessible files and file systems via FTP (very convenient for remote administration). The *"Local File System"* checkbox must be checked in the FTP configuration dialog in SBBSCTRL to enable this feature.

To access local file systems, you must first login with sysop access (see "Sysop Access" for details). Once logged in with sysop access, you may

mount the local file system. There are two ways to mount the local file system (and subsequently to revert to the BBS file system):

#### **SMNT (Structure Mount)**

Sysops may use the FTP **"SMNT"** command to mount the local files system. Since most FTP clients do not support the **SMNT** command, you'll probably need to use the "quote" feature of your FTP client to send the command to the FTP server.

The syntax is **"SMNT new\_dir"** where *new\_dir* is the file system specifier for the new directory used in the FTP hierarchy. Example: **"SMNT C:\"** would make the root directory of your local C drive your new current FTP directory (for Windows' command-line ftp, type **"quote SMNT C:\"**).

You can access shared drives on other machines on your local area network by using **"SMNT \\othersys\drive"** where *othersys* is the name of the other computer and drive is the share name of the desired drive.

To switch back to (mount) the BBS file database, use **"SMNT bbs:"**

#### **CD (Change Directory)**

Sysops can mount the local file system using a special case of the **CD** (**CWD**, **XCWD**) command. By prefixing **"local:"** to the new directory, the local file system will be automatically mounted. Example: **"cd local:c:\"**.

Once the local file system is mounted, you can switch between drives and shared drives with the CD command (the **"local:"** specifier is not needed). Example: **"cd d:\"**.

To access a shared drive on other machines: **"cd \\othersys\drive"**.

To switch back to (mount) the BBS file system, use **"cd bbs:"**.

Since some FTP clients do not support the **SMNT** command, nor allow users to enter verbatim FTP commands, the CD method is the easiest and most reliable method of accessing the local file system.

### [Back to Top](#)

---

#### [Copyright](#) © 2000 by [Rob Swindell](#)

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMBLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .

# Synchronet BBS

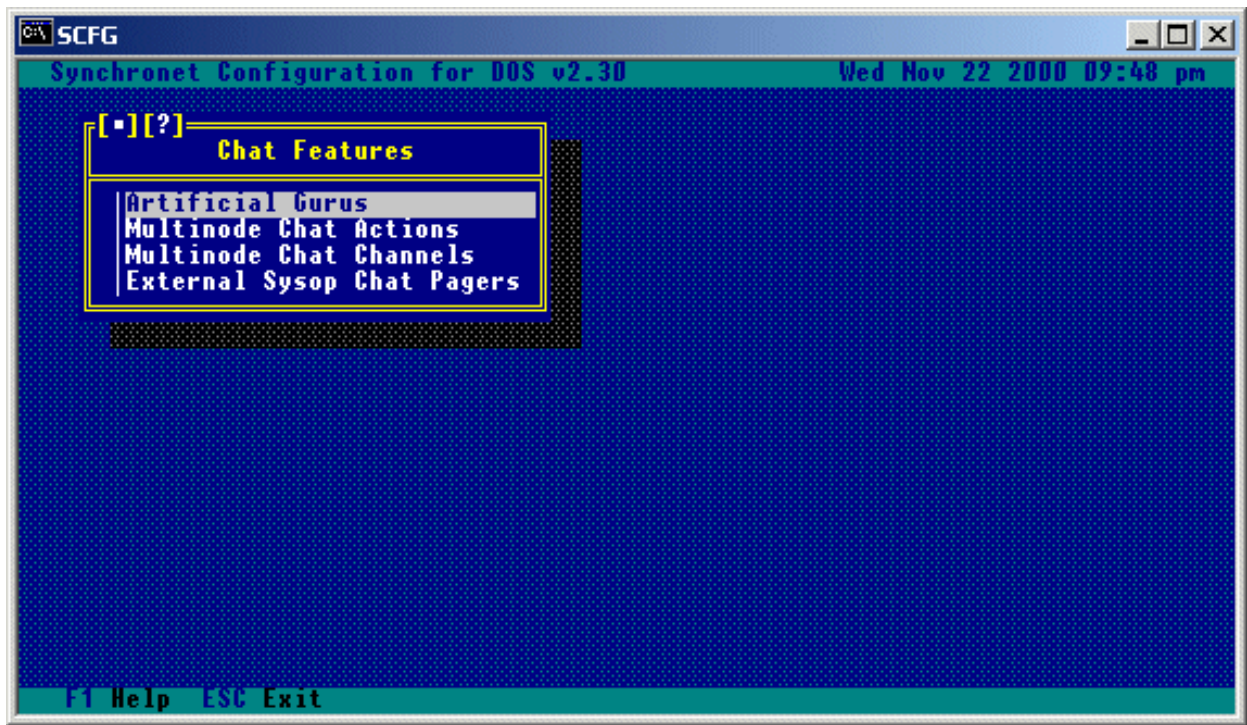
Multinode Bulletin Board System Software

---

[Back to Table of Contents](#)

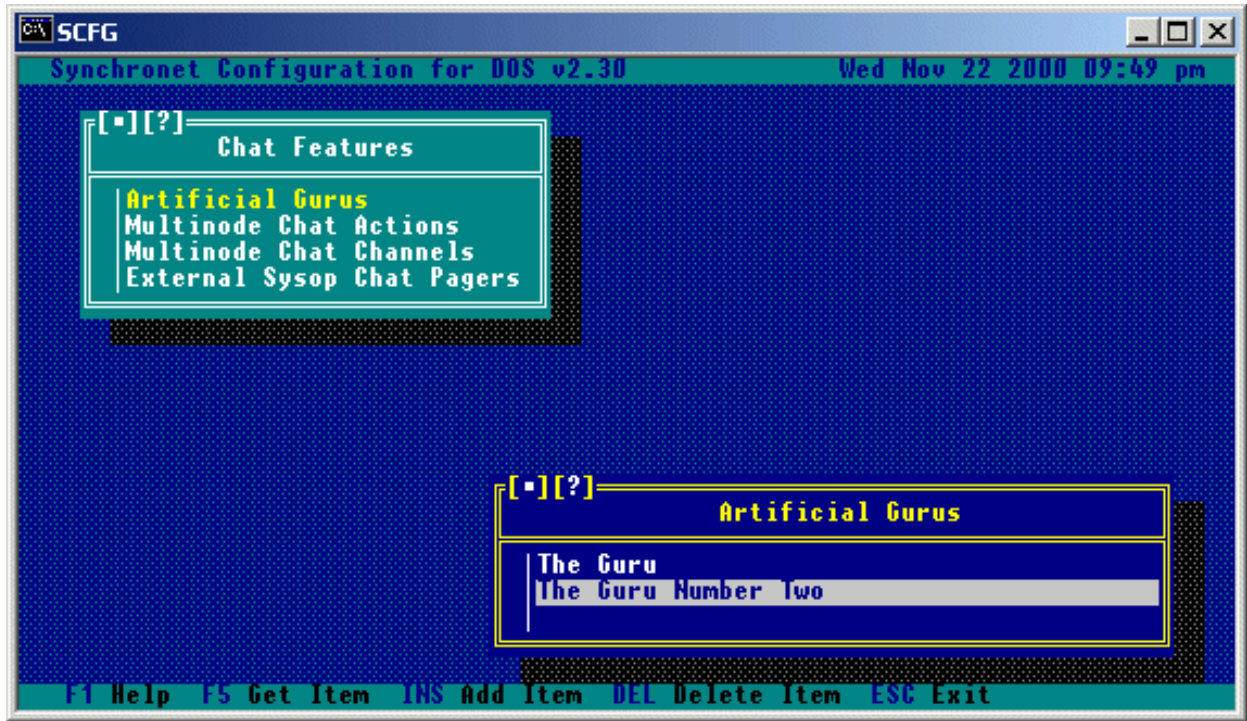
## [11.0] - Chat Features

Selecting the Chat Features option from the SCFG will bring you to the following sub-menu, an explanation of the options shown here follow.



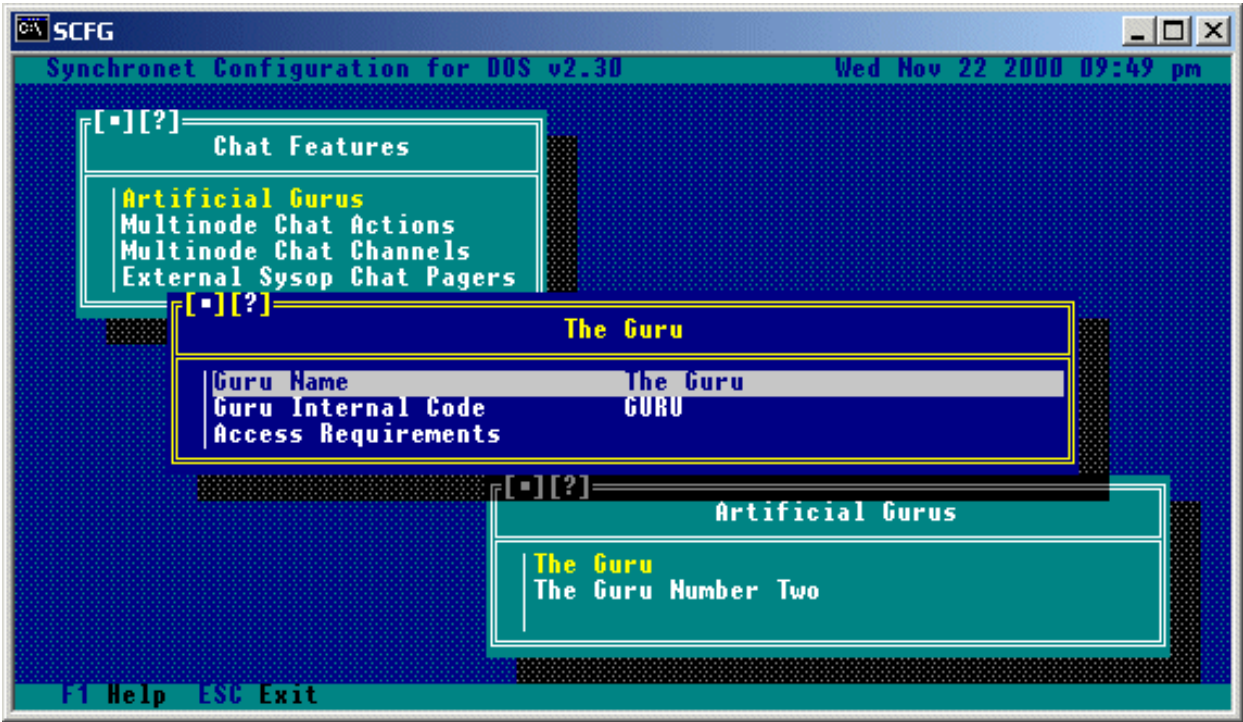
## [11.1] - Artificial Chat Gurus

This selection will give you a sub-menu with a list of available artificial gurus, here you can insert and delete gurus.



Selecting a guru from the menu will bring you to a menu allowing you to configure that guru.





**Guru Name:**  
The name users will see when chatting with this artificial guru.

**Guru Internal Code:**  
This is an 8 character code used internally by Synchronet and should be unique for each guru. This is also the filename which will be used for the intelligence engine for this guru (with the .DAT extension, located in your CTRL directory). See the section on customization for information on customizing chat gurus.

**Access Requirements:**  
Only users meeting the criteria set here will be able to chat with this guru.

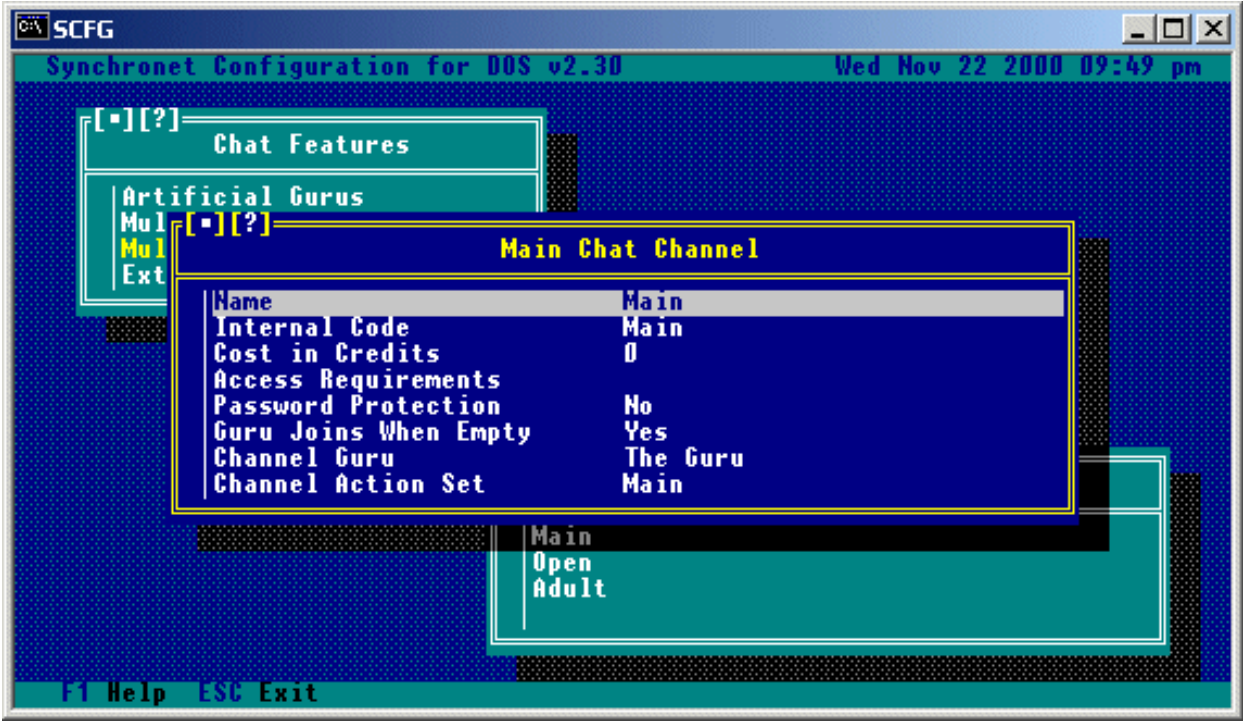
## [11.2] - Chat Actions

Selecting this option will show a list of available chat action sets where you can insert and delete action sets. Selecting an action set from this menu will display a list of the actions contained within that set.

Synchronet comes with a default chat action set, which you can modify, or use as a reference for creating your own action set. Each command response should have two '%s' fields contained in the line, the first will be replaced with the name of the user performing the action, the second with the name of the user to which the action is directed.

## [11.3] - Multinode Chat Channels

Selecting this option will give you a sub-menu. An example of how a chat channel might be set up, along with the options available, is shown:



**Name:**  
This is the name of this chat channel, and what will be displayed to users when viewing a list of available chat channels.

**Internal Code:**  
This is an internal code for SBBS to distinguish this chat channel from the others on the system. This must be a unique name of up to 8 valid DOS filename characters.

**Cost in Credits:**

This is the amount of credits which will be deducted from the users account when joining this particular chat channel.

**Access Requirements:**

A user must meet the criteria set here in order to be able to use (or even see) this chat channel.

**Password Protection:**

If you wish to allow users to be able to lock this channel with a password (only users knowing this password can enter the channel), you should set this option to Yes.

**Guru Joins When Empty:**

Set this option to Yes to have the selected chat guru enter this chat channel when there is only one user in the channel.

**Channel Guru:**

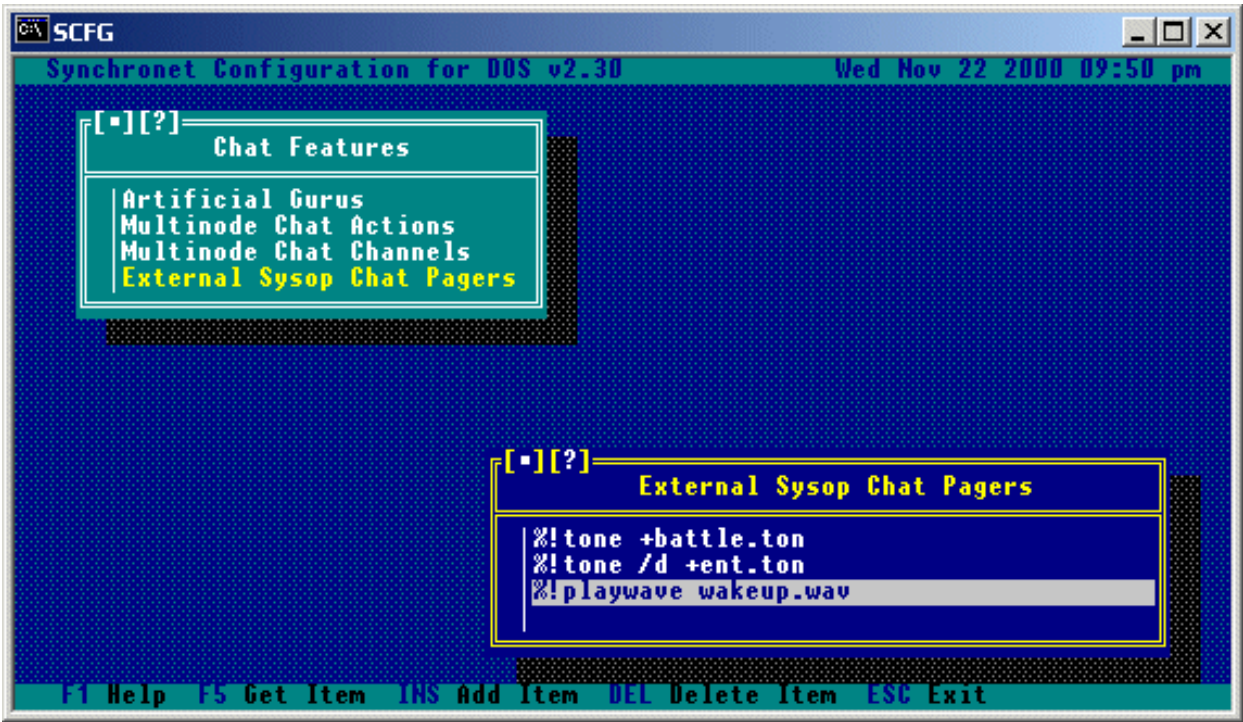
This is the guru which will join this channel (if the above option is set to yes). Selecting this option will give you a list of available chat gurus to choose from.

**Channel Action Set:**

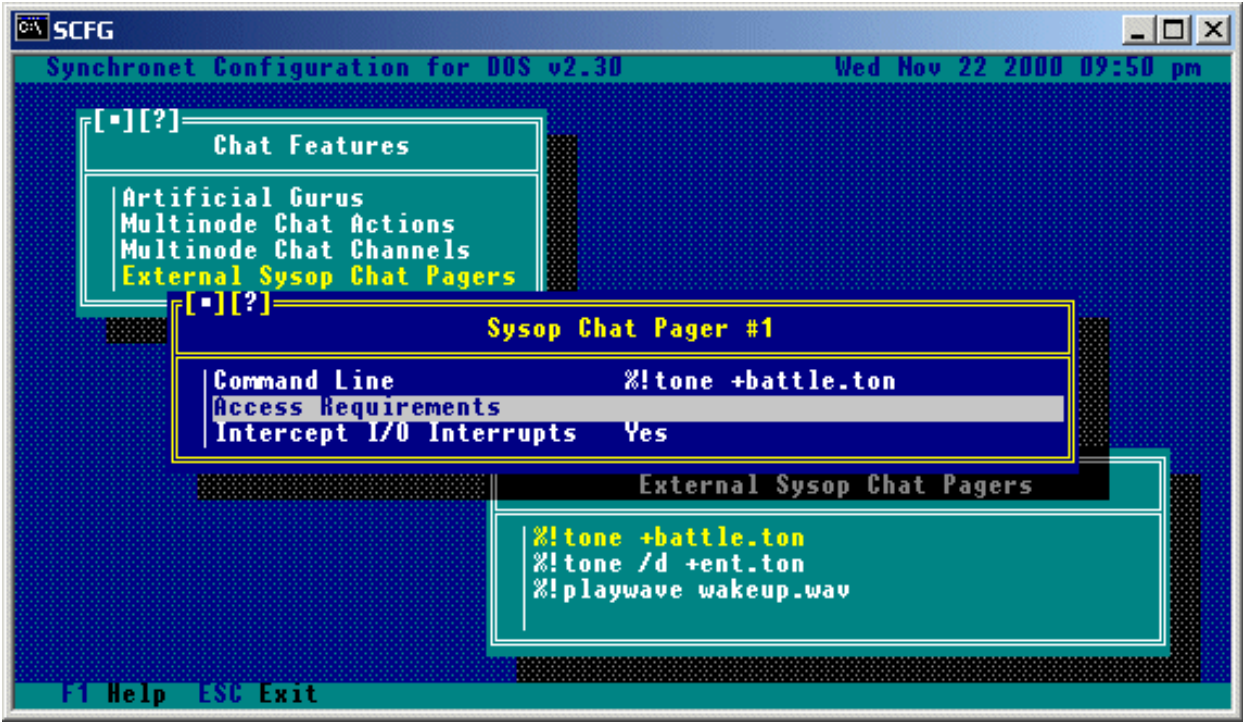
This is the chat action set which will be available to users in this channel. Selecting this option will give you a list of available chat action sets.

**[11.4] - External Sysop Chat Pagers**

This option allows for sysops to have custom chat pages (external to what is available in Synchronet). Sysops can have .TON files played when certain or all users page, or if a sound card is installed, the sysop may have a .WAV or .VOC file play. Selecting this option will give you a list of currently configured chat pagers.



Selecting a pager from the above menu will give you other options for this chat pager.



**Command Line:**

This is the command line necessary to invoke the chat pager.

**Access Requirements:**

This is the list of criteria a user must meet for this pager to affect them.

**Intercept I/O Interrupts:**

If the chat pager specified produces screen output that the user should see (or a series of beeps that the user should hear), this option should be toggled to Yes.

**[Back to Top](#)**

---

**[Copyright](#) © 2000 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)

PO Box 501

Yorba Linda, CA 92885

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .

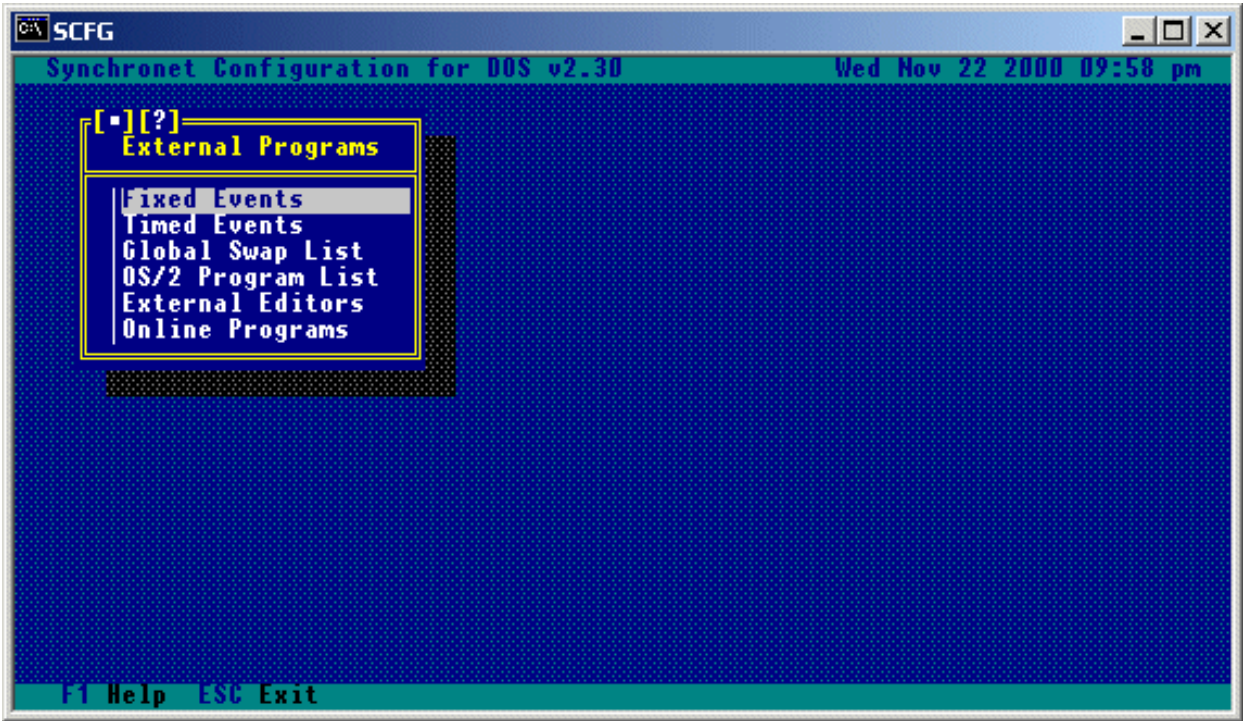
# Synchronet BBS

Multinode Bulletin Board System Software

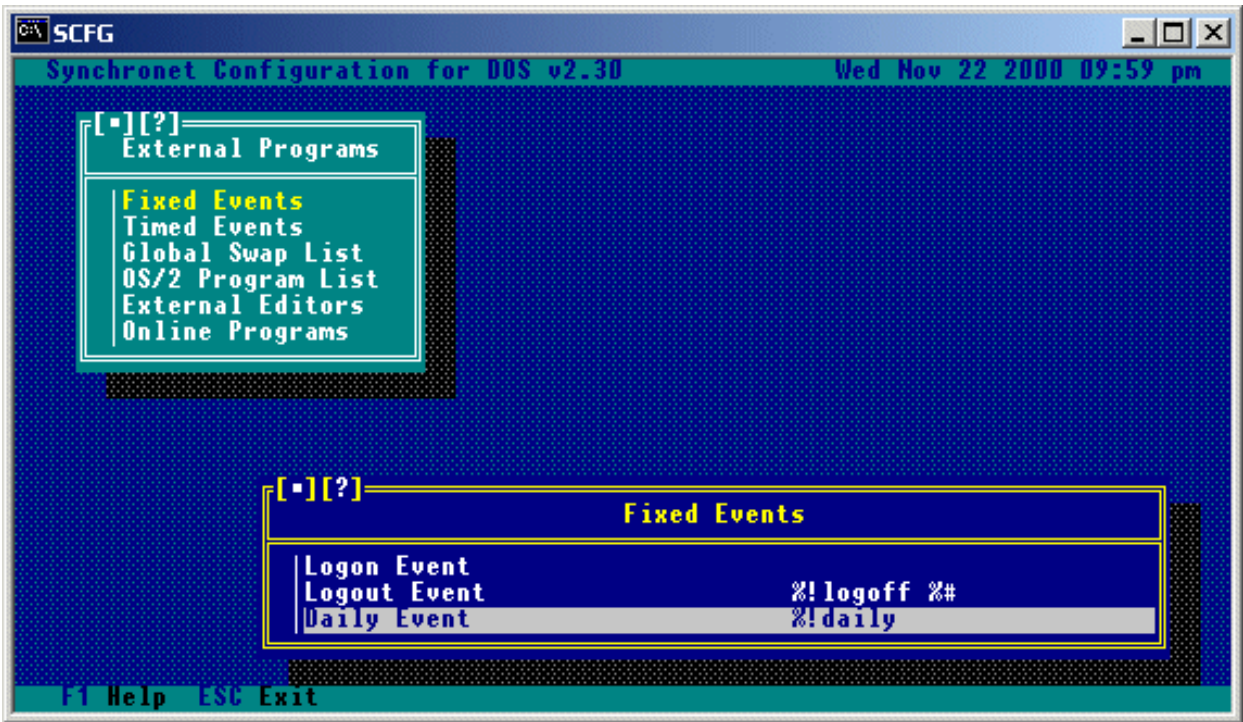
[Back to Table of Contents](#)

## [12.0] - External Program Configuration

Selecting the External Programs option from the SCFG will bring you to the following sub-menu, an explanation of the options shown here follow.



## [12.1] - Fixed Events



### Logon Event:

This is the command line for a program that will execute during the logon sequence of every user. The program cannot have user interaction. The program will be executed after the LOGON message is displayed and before the logon user list is displayed. If you wish to place a program in the logon sequence of users that includes interaction or requires account information, you probably want to use an online external program configured to run as a logon event.

### Logoff Event:

This is the command line for a program that will execute during the logoff sequence of every user. This program cannot have user interaction because it is executed after carrier is dropped. If you wish to have a program execute before carrier is dropped, you probably want to use an online external program configured to run as a logoff event.

### Daily Event:

This is the command line for a program that will run after the first user that logs on after midnight, logs off (regardless of what node).



# [12.2] - Timed Events

Selecting Timed Events from the External Programs menu will give you a list of currently configured timed events. Following is an example of a timed event called MAIN, and the options available for timed events.



**Internal Code:**  
This is an internal code for SBBS to distinguish this event from the others on the system. This must be a unique name of up to 8 valid DOS filename characters.

**Start-up Directory:**  
This is the directory where the event execution should take place. Synchronet will change to this directory before executing the command line for this event.

**Command Line:**  
This is the command line for a program that will run on the specified times and days.

**Execution Node:**  
This is the node that will execute the event.

**Execution Days:**  
These are the day(s) when this timed event will be executed.

**Execution Time:**  
This is the time (in 24hr format) for the above mentioned program to execute.

**Requires Exclusive Execution:**  
You may set this event to run exclusively (all other nodes inactive or taken offline) by selecting this option.

**Force Users Off-line For Event:**  
Setting this to Yes will cause the BBS to force users offline for this event.

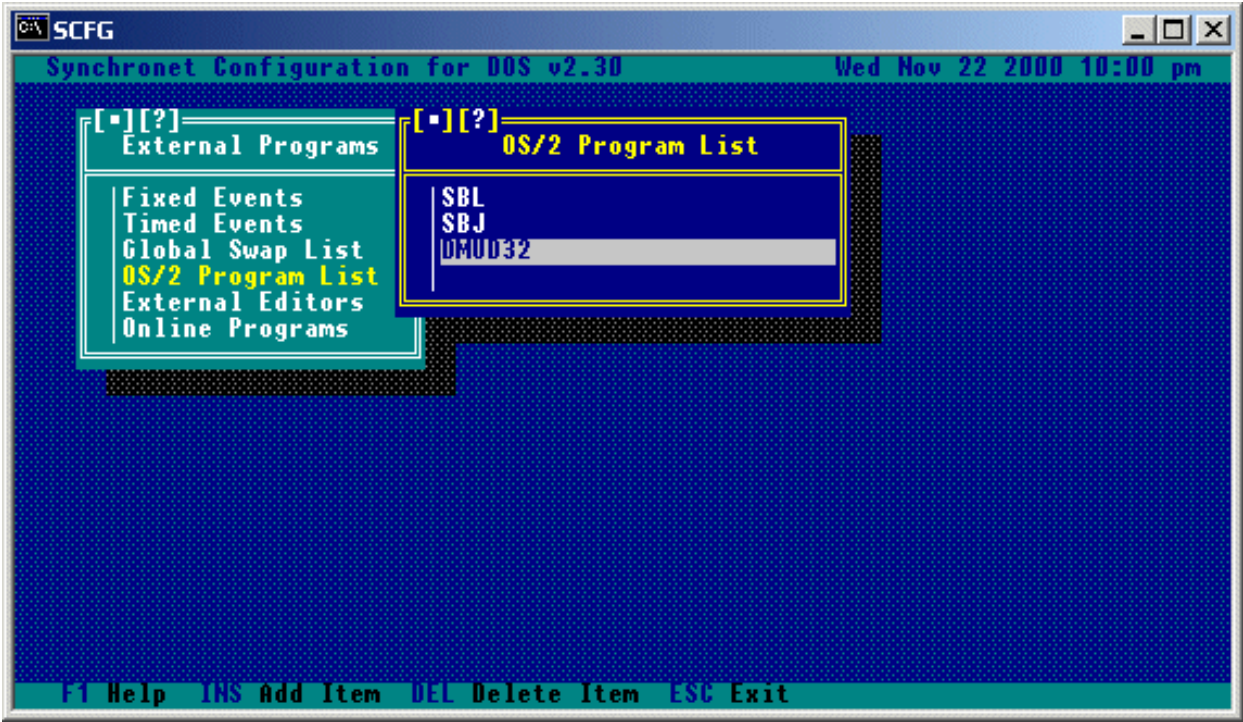
# [12.3] - Global Swap List (v2 for DOS Only)





This is a list of programs which, when run, will cause Synchronet to swap out of memory. This is useful for large executables, or programs which have the possibility of Utilitizing a large amount of memory. Shown is an example of what might be in your swap list. Notice that it is not necessary to specify an extension for the programs unless an extension is specified on the command line to be executed. If a file extension is specified (EXE, COM, or BAT) the extension must be specified on the command line for the program to be found in the swap list.

## [12.4] - Native (32-bit) Program List



When running a node with Synchronet for OS/2 or Win32, all external programs (doors, editors, file viewers, etc) will be assumed DOS programs and executed inside a virtual DOS machine (VDM) automatically unless the program's name is included in this list. Only CMD.EXE (OS/2 command interpreter) should be included with an extension (e.g. .EXE) unless you actually configure your external program command lines in SCFG with extensions (note ZIP and UNZIP do not have specified extensions). All native Win32 and OS/2 programs that you wish to execute from the BBS must be added to this list.

## [12.5] - External Editors

This selection will bring up a list of external editors that are available as alternatives to the Synchronet built-in editor. Editors can be added and deleted with the INS and DEL keys. Selecting an editor allows you to view and edit its settings. The available settings are (example given is for SyncEdit):



### Internal Code:

This is an internal code for SBBS to distinguish this editor from the others on the system. This must be a unique name of up to 8 valid DOS filename characters.

### Local Command Line:

This is the command line to use when the user is online locally.

### Remote Command Line:

This is the command line to use when the user is online remotely.

**Access Requirements:**

These are the requirements that a user must meet to be able to use this external editor.

**Intercept I/O Interrupts:**

If the editor relies on Synchronet for intercepting I/O interrupts, this option should be set to Yes. When set to Yes, you will be able to specify whether or not the editor uses WWIV color codes.

**Quoted Text:**

If you want all the message text to be automatically entered into the message edit/quote file (INPUT.MSG or MSGTMP), set this option to 'All'.

If you want the user to be prompted for which lines to quote before running the editor, set this option to 'Prompt User'.

If you want none of the lines to be automatically quoted, set this option to 'None'. This option is mainly for use with editors that support the QUOTES.TXT drop file (like SyncEdit v2.x).

**QuickBBS Style (MSGTMP):**

If this editor was designed to run on QuickBBS, Remote Access, or SuperBBS (uses a MSGTMP file for editing), set this option to Yes. This option will also create a MSGINF file automatically as well.

**Expand Line Feeds to CRLF:**

If this editor stores new lines as a single line feed character, set this option to Yes.

**BBS Drop File Type:**

If this editor requires a BBS drop file (DOOR.SYS, DORINFO#.DEF, etc) use this option to select the drop file format.

**[12.6] - Online Programs**

This selection will generate a list of configured external program sections. You can add and remove external program sections from this list with the INS and DEL keys. To edit an external program section or the external programs within that section, select it with the arrow keys and hit ENTER.

For each external program section, you can set the Name, Internal Code, and Access Requirements. Selecting "Available External Programs" will give a list of external programs contained in this external program section. Synchronet has been designed to eliminate the need for batch files and drop file conversion programs to run external programs (doors) on your BBS. Synchronet supports many different drop file types (often referred to as drop files) listed below.

**[12.7] - Supported Dropfile Types:**

Software	Write File(s)	Read File(s)
~~~~~		
DOOR32	(v3+ Only)	DOOR32.SYS
GAP	DOOR.SYS	DOOR.SYS
WWIV	CHAIN.TXT	
PCBoard 14.x	PCBOARD.SYS and USERS.SYS	USERS.SYS
RBBS/QuickBBS	DORINFO#.DEF and EXITINFO.BBS	EXITINFO.BBS
Wildcat! 2.x	CALLINFO.BBS	
SpitFire	SFDOORS.DAT	
TriBBS	TRIBBS.SYS	
MegaMail	UTIDOOR.TXT	
Solar Realms	DOORFILE.SR	
Synchronet	XTRN.DAT	MODUSER.DAT

**[12.8] - Call-back Verifiers**

Call-back verification and user upgrade programs for other BBS packages may be used, as long as the program supports the DOOR.SYS, EXITINFO.BBS, USERS.SYS, or MODUSER.DAT file format. In general, Call-back verifiers written for PCBoard, GAP, QuickBBS, or Synchronet will work. To allow a program to modify the user data, you must set the option "Modify User Data" to "Yes". The following user data items may be updated with the following formats:

- DOOR.SYS: Security level, Flag Set #1, Expiration Date, Minutes, Total Files/Bytes Downloaded, and Credits
- EXITINFO.BBS: Security level and Flag Set #1
- USERS.SYS: Security level and Expiration Date
- MODUSER.DAT: Security level, Flag Set #1-#4, Exemptions, Restrictions, Expiration Date, Credits, and Minutes

# [12.9] - Installing a New External Program

Install your new external program into a sub-directory off your XTRN directory.

Example: C:\SBBS\XTRN\TRADEWAR

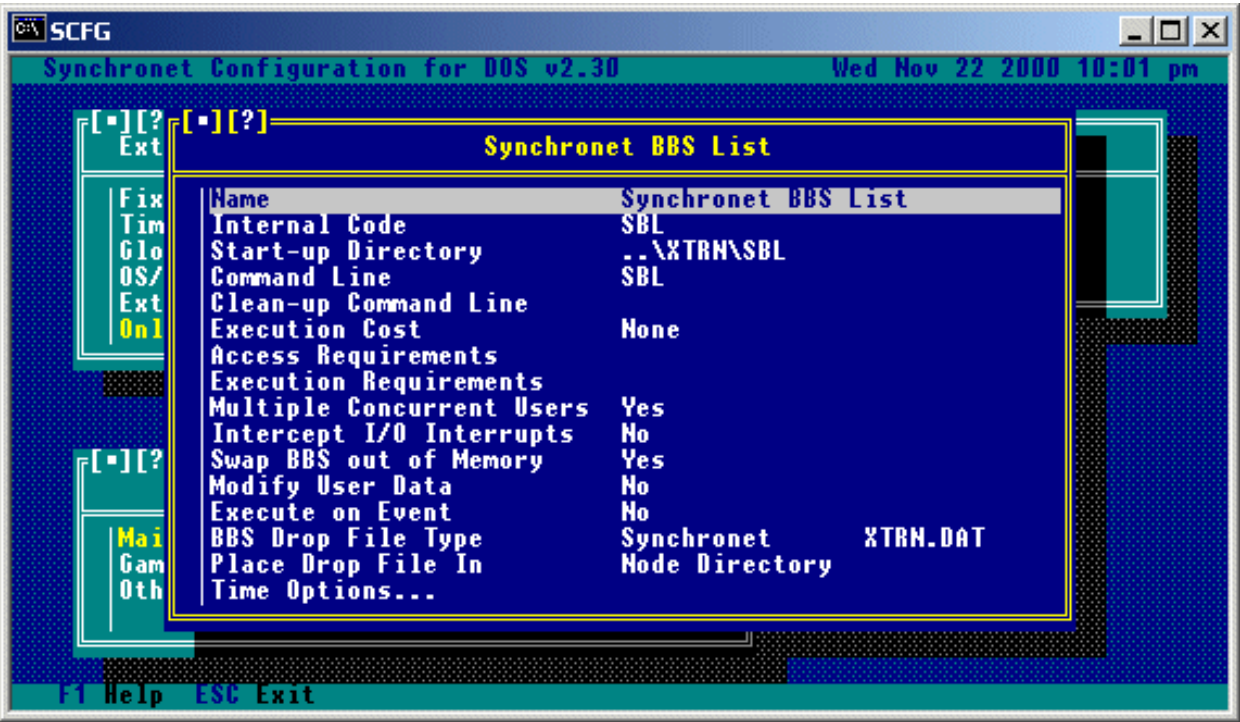
If your program asks you to provide the directory where the BBS software is installed or where the drop file will be located, enter the directory where you have installed the program (C:\SBBS\XTRN\TRADEWAR). Note: this is usually only required for programs that do NOT support multiple nodes.

The full path and file name of the drop file can be specified on the command line by using the %f command line specifier in SCFG.

# [12.10] - Configuring a New External Program

Run SCFG from any NODE directory (or hit 'C' at the Waiting For Call screen). Select "External Programs" and hit enter. Hit END and ENTER for the External Program Sections menu and select an External Program Section to add this new program to. Select "Available External Programs" and hit the Insert key. Enter a description for your new external program.

Now your new external program appears in the list. Hit ENTER to edit the configuration for this program. Synchronet BBS List is given as an example.



**Name:**  
This is the name of the external program that will be shown to users.

**Start-up Directory:**  
The start-up directory will be the current directory when the program is started. Move the lightbar down to "Start-up Directory" and hit ENTER. Enter the path to the directory where you've installed the program.

**Command Line:**  
This is the command required to execute your external program. If the path and filename for the BBS drop file is required on the command line, use %f to represent the path and filename. See Appendix A for a full list of available command line specifiers.

You can launch a Baja module in place of a DOS executable by prepending an asterisk (\*) to the name of the Baja module (e.g. "\*MYMOD"). The module .BIN file must be located in your EXEC directory.

Some programs, when set to SWAP, require ".\" to be prepended to the command line (e.g. ".\MYDOOR") to function properly.

**Clean-up Command Line:**  
If your external program requires a clean-up command line (usually only multi-user programs), select "Clean-up Command Line" and enter the required command line for your program. Note: Most programs do NOT require this option.

**Cost in Credits:**  
If you wish your users to be charged credits when running this program, you should enter the amount to charge the users here. Users with the 'X' exemption will not be charged.

**Extra Minutes:**  
This option will allow you to give extra time to users for use in the external program they are entering. For example, with this option set

to 10, a user having 15 minutes left on the BBS will end up having 25 minutes in the external program they enter. This is especially useful in conjunction with the 'Suspended (Free) Time' option.

**Suspended (Free) Time:**

Setting this option to "Yes" will cause a user's time left online to suspend (not countdown) when they enter the external program. The user's time allowed in the external program, however, will be what they had left on the BBS. To give them more time in the external program, use the above 'Extra Minutes' option.

**Access Requirements:**

If you wish to limit which users have access to (can see) this program, use this option to set those requirements.

**Execution Requirements:**

If you wish to have some users see but not be able to execute this program, use this option to determine what requirements must be met to run the program.

**Multi-user:**

If this program supports multiple simultaneous users, set this option to "Yes".

**Intercept I/O Interrupts:**

If this program does NOT have it's own modem communications I/O and does NOT use a FOSSIL driver, select "Intercept I/O Interrupts" and set it to "Yes". Note: Most programs require this option be set to "No". Programs written specifically for WWIV or Synchronet BBS software, should usually have this option set to "Yes". Consult the documentation for the program if you are unsure. If you do set this option to "Yes", and the program was not written specifically for Synchronet, you'll need to place the following string at the BEGINNING of your command line: "%!dcdwatch %&". You will also be able to select whether or not the program uses WWIV color codes when setting this option to Yes.

Without the use of DCDWATCH, the program will not exit if a user drops carrier (hangs up) while using the program (unless specifically written for Synchronet). Some programs time-out eventually, and exit back to the BBS, but not all.

**Swap BBS out of Memory:**

If this program requires a large amount of free memory, set this option to 'Yes' and Synchronet will swap to XMS, EMS, Extended Memory or Disk (see "Node Toggle Options" for Swap Type selection) to make room for this program to execute. Setting this option to 'Yes' has this same effect as adding this program to the "Global Swap List", but is available as a toggle on this menu for your convenience.

**Modify User Data:**

If this program can modify the current user's data through one of the supported drop file types (see Call-back Verifiers above), set this option to Yes.

**Execute on Event:**

If you want this program to run automatically during user logon, logoff, new user application, or a user's birthday, use this option.

**BBS Drop File Type:**

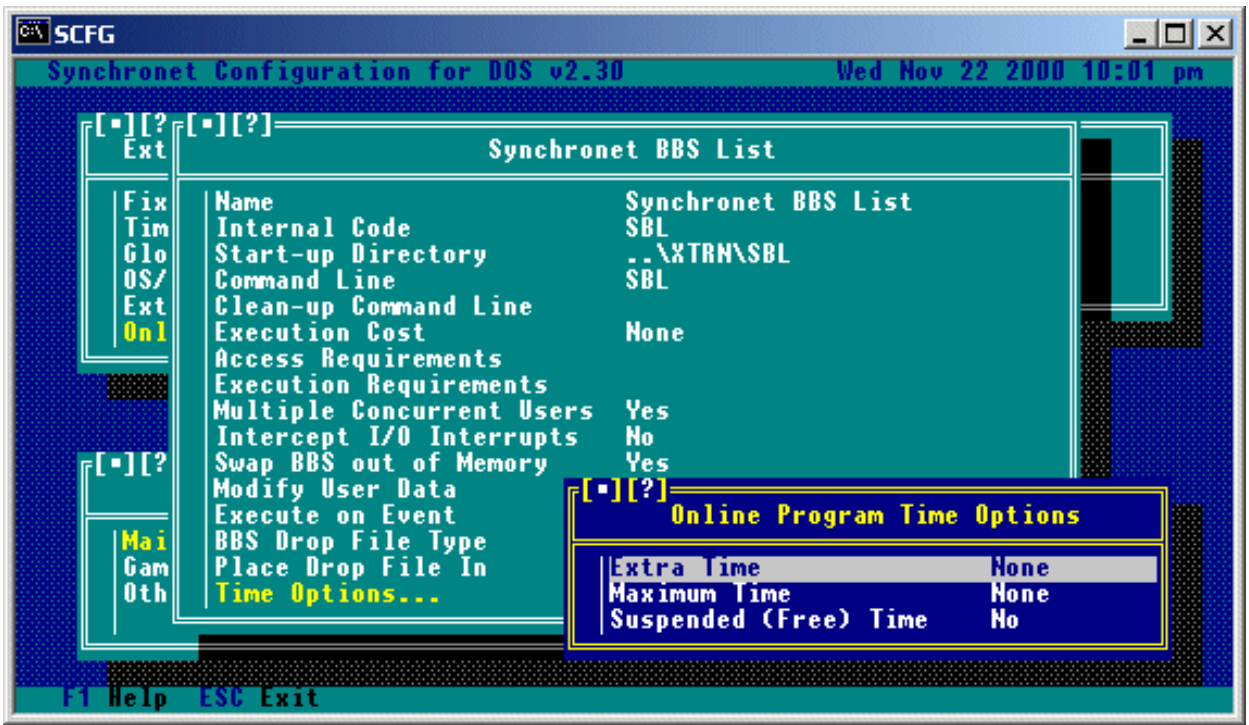
If your program requires a BBS drop file (most do), use this option to select the type of drop file to create.

**Place Drop File In:**

This is the directory where the drop file will be created. In general, for multi-user programs or programs that can have the path and filename of the drop file included on the command line or in a config file, select "Node Directory". For single user programs that require the drop file to be in the current directory, select "Start-up Directory".

**Time Options....:**

Selecting this option will bring you to a sub-menu similar to the following:



**Extra Time:**

This option allows you to give users an extra amount of time (in minutes) which can be used only within this online program.

**Maximum Time:**

This option allows you to set the maximum amount of time a user should be able to spend in this online program (in minutes) PER execution (not per day or per call).

**Suspended (Free) Time:**

When this option is set to 'Yes', a users' time is suspended while they are within the online program. That is, if they enter the door with 60 minutes remaining on the BBS, when they get back to the BBS they will still have 60 minutes of time remaining.

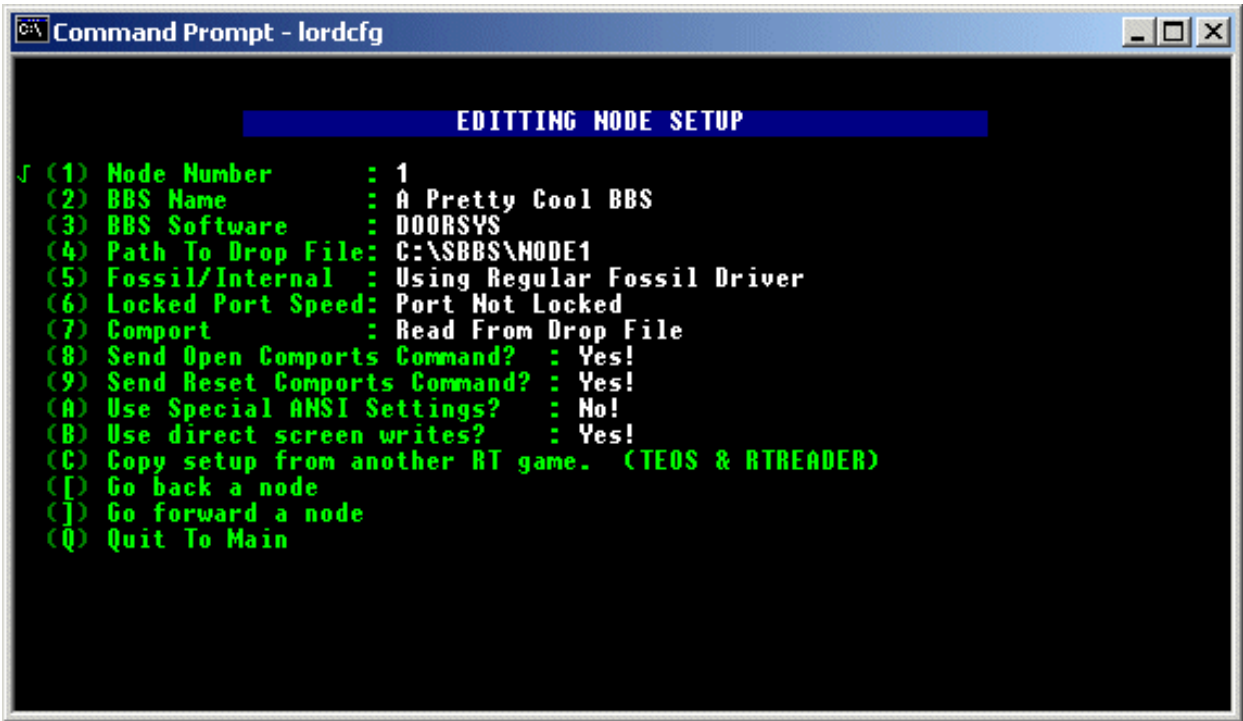
**[12.11] - Examples for some popular programs**

It is advised to install all your doors off the SBBS\XTRN directory (e.g. C:\SBBS\XTRN\BRE) if it's not too late. This will help keep all your doors in an easy to reference point. Also, this is where the pre-configured doors (SBL,SMM) are installed by default.

When configuring your Doors it is best whenever possible to place the dropfiles in your SBBS\NODE# directories and configure the game to look there for the file. This ensures smooth Multinode usage if your door supports multiple users at the same time.

**[12.11.1] - Legend of the Red Dragon (LORD)**

From the LORDCFG program your configuration should look something like this:



You will need to do this for each of your nodes. Be sure to set the 'Path To Drop File' to the node directory of the node you are configuring. The reason we set the 'Locked Port Speed' to 'Port Not Locked' is because you should be locking the speed of your FOSSIL when you load it, once that's done, it can't be altered.

\*\*\* Be Sure to edit **START.BAT** and check for proper paths and filenames!



Now go into your SCFG program and create a new external program. It should look like this when it's all setup:



### [12.11.2] - Usurper

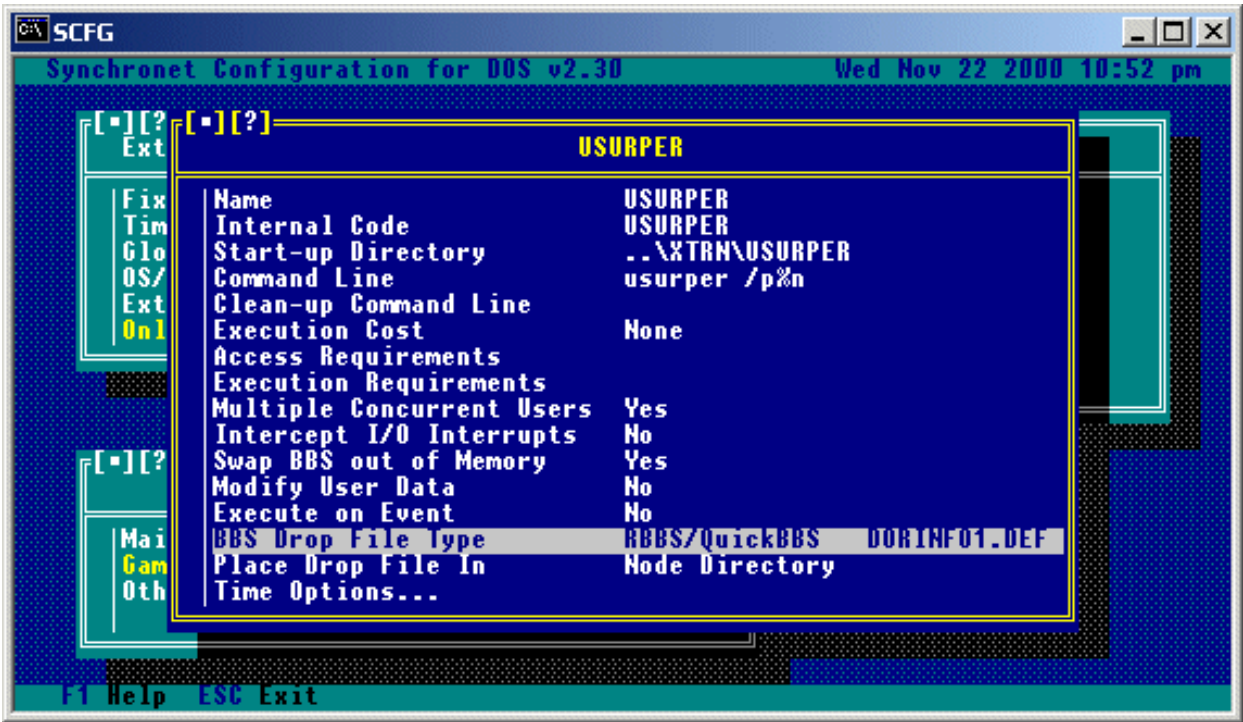
You MUST use a fossil driver in order to use this program. Follow the Usurper documentation for getting the game going.

Next, bring up the USURPER.CFG into a text editor and edit the first few lines (or you can do it through the EDITOR program that comes with Usurper). It should end up looking something like this:

```
Sysop Name Goes Here
BBS Name Goes Here
C:\SBBS\XTRN\USURPER\
DORINFO1.DEF
```

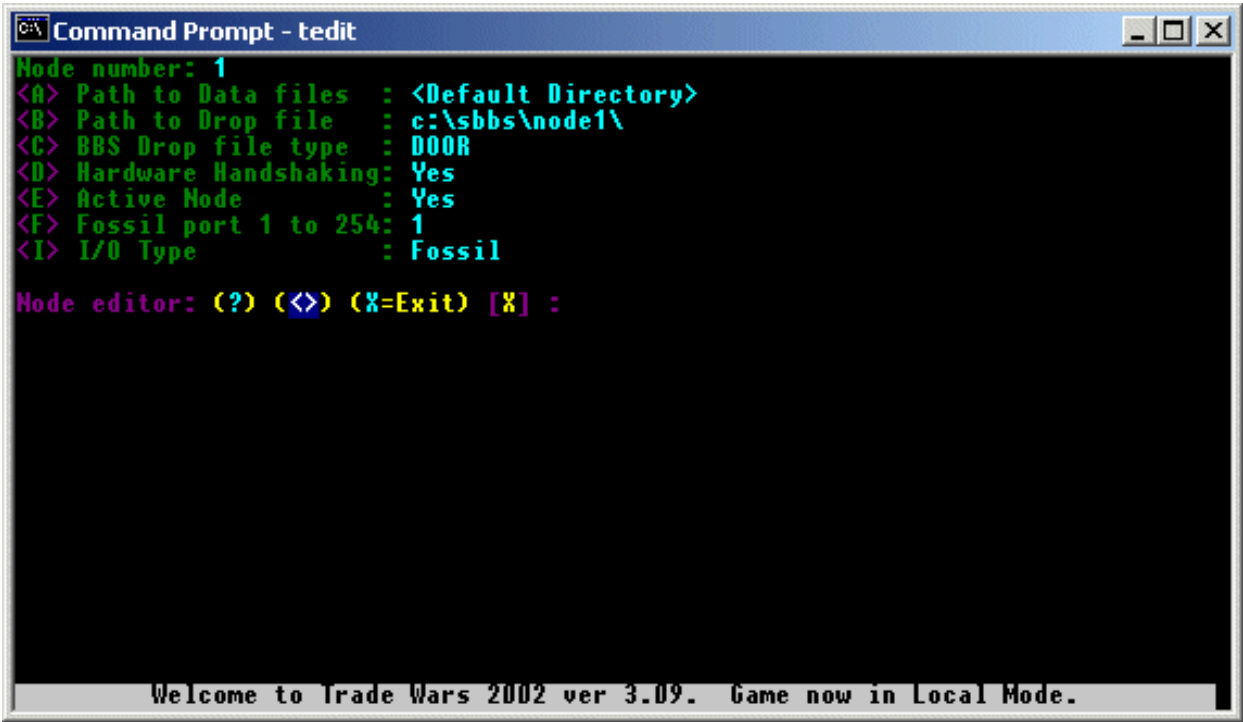
The important things are line 3 and line 4. Line 3 MUST be the path to your Usurper directory (with a trailing backslash) and line 4 must be the type of drop file (I used DORINFO1.DEF like the Usurper documentation recommends).

Now you'll need to go into your SCFG program and add an external program. Set it up to look something like this:



### [12.11.3] - Trade Wars 2002

From the TEDIT program your configuration should look something like this:



You will need to do this for each of your nodes. Be sure to set the 'Path To Drop File' to the node directory of the node you are configuring.

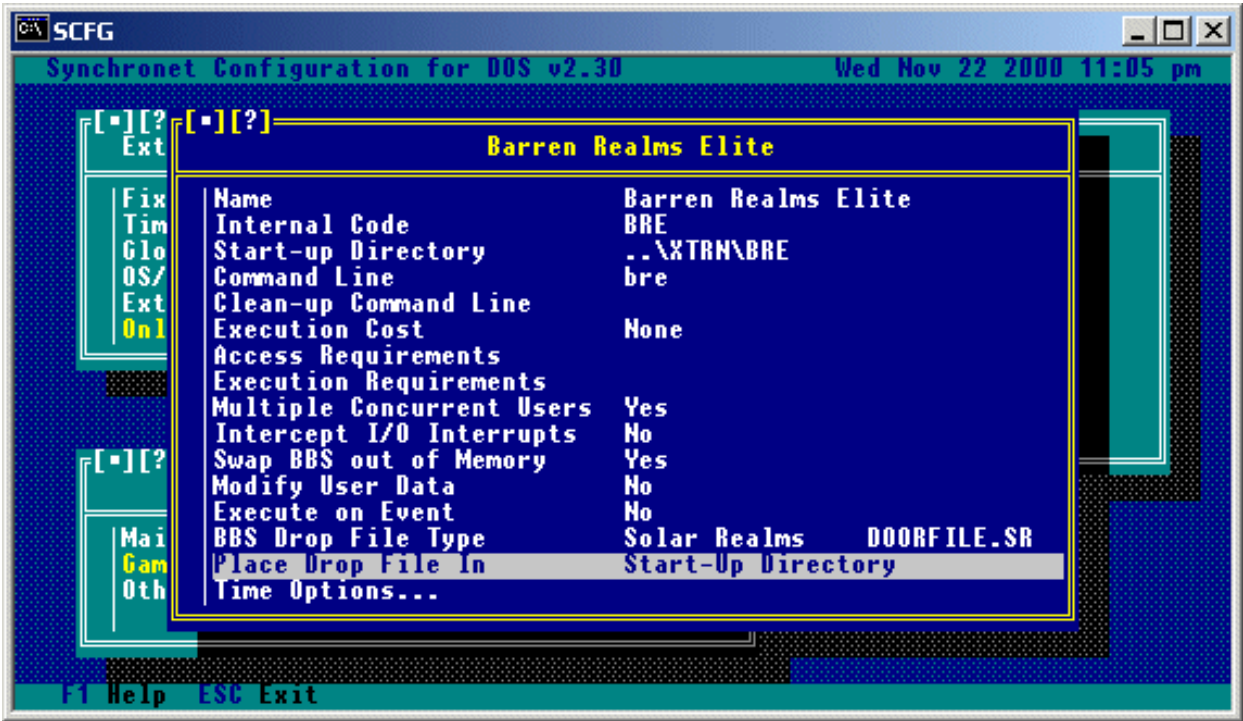
Now go into your SCFG program and create a new external program. It should look like this when it's all setup:



#### [12.11.4] - Barren Realms Elite (BRE)

Install and Configure BRE per the instructions in the BRE documentation.

Now go into your SCFG program and create a new external program. It should look like this when it's all setup:



BRE and the other SR / John Daily games are easy with Synchronet since it supports SR Games native DOORFILE.SR . Just place the DOORFILE.SR in BRE's directory and all is well.

\*\* Note: BRE is not Multiplayer and can only be used on one node at a time. However, we can leave Allow Multiple Users ON in SCFG as BRE will inform the player the game is in use by itself.

# [12.12] - Internet Gateways

This allows users on your system to be connected to other TCP/Telnet services on the Internet through your BBS. Typically, these services would be other telnetable BBSs and MUDs, although really, you can gate to any single-port TCP service (i.e. POP3, but not FTP).

```
TELNET_GATE <"str" || str_var> [mode]
```

The str or str\_var argument is the destination address in the format:

```
<addr>[:port]
```

If port is not specified, the default is 23 (telnet), or 513 (when the TG\_RLOGIN mode is used).

Examples:

```
"vert.synchro.net"
"24.0.193.46"
"retromud.org:3000"
```

The valid mode bits (multiple may be |'d together) are:

TG_ECHO	Turn on telnet echo
TG_CRLF	Expand sole CR to CRLF
TG_LINEMODE	Send entire lines only
TG_NODESINC	Call Nodesync, get msgs, etc.
TG_CTRLKEYS	Interpret ^P ^U ^T, etc locally
TG_PASSTHRU	Pass-through telnet commands/responses
TG_RLOGIN	Use BSD RLogin protocol

This function can be called from any module or shell, but will most likely be used from the external programs menu (even though the telnet gate isn't technically an external program). The following modules are included (in your EXEC directory): TELGATE.BIN, MUDGATE.BIN, UNIXGATE.BIN, and RLOGIN.BIN. Use them as follows:

To set-up an external gateway to another BBS:

```
Command line: "*telgate vert.synchro.net"
```

To set-up an external gateway to a MUD:

```
Command line: "*mudgate batmud.bat.org"
```

To set-up an external gateway to a Trade Wars Game Server (TWGS):

```
Command line: "*rlogin twgs.someserver.com"
```

While using the telnet gateway (and not currently in telnet "Binary" mode), users can press ctrl-] for a menu of telnet gate commands including: Disconnect, Toggle Echo, List Users, and Private Message.

# [12.13] - Troubleshooting External Programs

This section will explain common reasons why an external program may not be functioning properly and what can be done to get the program functioning properly. It is impossible for Digital Dynamics to be familiar with every available external program, therefore if you continue to have problems with a program after reading this section, you should contact the company that wrote the external program you are working with.

IMPORTANT NOTE: Some games will not work properly if you do not put a PATH on the command line (see the L.O.R.D. example), others will (see the Usurper example). If your game is not working at all, try using a path on the command line (e.g.: .\GAMENAME).

PROBLEM : The external program works, but displays the wrong user name.

SOLUTION: Check to see that the program is looking in the proper place for the drop file. Usually you will need to specify the path to look for the drop file on the command line or in the program's setup program or config file(s).

PROBLEM : The program works fine locally, but when people call in they cannot see anything.

SOLUTION: If the game uses a FOSSIL driver, make sure that you have a FOSSIL driver loaded and make sure that you've locked the baud rate at the same rate that you've locked Synchronet's com ports. If the game uses it's own internal com routines (no FOSSIL required), you should be able to pass it IRQ and I/O Address information (either through configuration files, or on the command line) otherwise the game probably will not work on anything other than the standard COM1 and COM2.

NOTE : Normally only games written for Synchronet or WWIV will ever require the 'Intercept I/O Interrupts' option to be set to YES. If a game has it's own internal com routines or uses a FOSSIL set this option to NO.

PROBLEM : The program works fine locally, but when people call in they only see garbled information.

SOLUTION: If the game is using a FOSSIL driver, make sure the locked rate of the FOSSIL is the same as the locked rate in Synchronet.  
If you are running Synchronet under OS/2 and you have "Intercept I/O" set to "Yes" for this program, make sure you are using the SVDMANSI program for ANSI under OS/2.

PROBLEM : The program works fine when only one user is playing, but when a second user tries to enter it doesn't function properly.

SOLUTION: The program is probably not intended for multiple simultaneous users. Be sure to set the 'Multiple Concurrent Users' option to NO if this is the case.

[Back to Top](#)

---

**[Copyright](#) © 2000 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .

# Synchronet BBS

Multinode Bulletin Board System Software

---

[Back to Table of Contents](#)

## [13.0] - Text File Sections

Run the SCFG utility, and select the Text File Sections option from the menu. This selection will produce a list of text file sections that the sysop has created for users to freely view text files. It is requested that all sysops make some or all of the Synchronet documentation available in a text file section. At the very minimum, TEXT\SBBS.MSG should be available for users to review. You can add and remove sections with the INS and DEL keys. The following configuration options are available for each section:

## [13.1] - Text File Options

**Name:**

This is a description of the text file section.

**Access Requirements:**

Use this option to set the security requirements to access this file section.

**Internal Code:**

This is an internal code for SBBS to distinguish this text file section from the others on the system. This must be a unique name of up to 8 valid DOS filename characters.

## [13.2] - Adding Text Files

To add files to a text file section, you'll need to log on to the BBS and enter the section you would like to add files to. Here you'll be given a choice to add files to the section, you will be prompted for the title of the file, and the path/filename. If you do not enter a path for the file, the default path is the Synchronet DATA\TEXT directory PLUS the internal code of the text file section.

[Back to Top](#)

---

**Copyright © 2006 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Document](#) .



# Synchronet BBS

Multinode Bulletin Board System Software

---

[Back to Table of Contents](#)

## [15.1] - CHKSMB - Checks mail/message base for validity

**usage:** `chksmb [/opts] <filespec.SHD>`

opts:

- s - stop after errored message base
- p - pause after errored message base
- q - quiet mode (no beeps while checking)

**WARNING:** All BBS nodes should be DOWNed (offline) or inactive when utilizing the CHKSMB utility.

The purpose of the CHKSMB utility is to insure that mail and/or message bases are valid. If you suspect that your mail, or one of your message bases has become corrupted, you can run CHKSMB. This will inform you of any possible corruption, and display information which may be useful in determining the cause of the corruption.

Example command line:

```
CHKSMB \SBBS\DATA\MAIL \SBBS\DATA\SUBS\*.SHD
```

to check your e-mail and sub-board data files for possible corruption.

Most mail and/or message bases which are corrupted can be reconstructed (fixed) using the SMBUTIL program with the pack option.

Message bases with corrupted index files must be fixed with the FIXSMB program.

## [15.2] - FIXSMB - Rebuild Synchronet Message/Mail Base

**usage:** `fixsmb [/opts] <smb_file>`

opts:

- m - mail format instead of sub-board format

ex: **FIXSMB /M C:\SBBS\DATA\MAIL**  
or: **FIXSMB C:\SBBS\DATA\SUBS\DEBATE**

Use the FIXSMB utility whenever CHKSMB shows that a message base has become corrupted due to problems with the index.

## [15.3] - SMBUTIL - Synchronet Message Base Utility

**usage:** `smbutil [/opts] cmd <filespec.SHD>`

cmd:

- l[n] = list msgs starting at number n
- r[n] = read msgs starting at number n
- v[n] = view msg headers starting at number n
- k[n] = kill (delete) n msgs
- i<f> = import from text file f
- s = display msg base status
- c = change msg base status
- m = maintain msg base - delete old msgs and msgs over max
- p[k] = pack msg base (k specifies minimum packable Kbytes)

opts:

- a = always pack msg base (disable compression analysis)
- f = fast msg creation mode
- d = disable duplicate message checking
- z[n] = set time zone (n=min +/- from UT or 'EST','EDT','CST',etc)

The SMBUTIL can be used for several things, but most importantly, it must be used for maintaining your Synchronet message base (using the 'm' command). This command causes SMBUTIL to mark old messages, and messages over the maximum sub-board limit, as deleted so that their spaces can be used by new messages. If you are using the 'Hyper Allocation' or 'Fast Allocation' method for your message bases, you must also use this utility to pack your message bases (using the 'p' command) periodically. The pack command causes SMBUTIL to compress the message bases removing any unused spaces (messages marked as deleted).

SMBUTIL maintenance and/or packing should be done using your daily event (or after importing from a network such as FidoNet). If your message base is using the 'Fast Allocation' or 'Hyper Allocation' storage methods, you should run maintenance before packing (there is no reason to pack if you do not run maintenance!). If your message base is set up for the 'Self Packing' method, you do NOT need to use SMBUTIL to pack, only to perform maintenance.

Example batch file lines for SMBUTIL would read:

for maintenance and packing (100k or more per sub):

```
SMBUTIL mp100 \sbbs\data\subs\*.shd
```

for maintenance only (self-packing sub-boards):

```
SMBUTIL m \sbbs\data\subs\*.shd
```

**WARNING:** All BBS nodes should be DOWNed (offline) when using SMBUTIL with the pack option! To do this, set the SMBUTIL event to be 'Exclusive' in the SCFG program.

When using the pack command, SMBUTIL will check to see if it is necessary for you to pack your message base, if you want to skip this checking and have it unconditionally pack the message base (useful when attempting to repair a corrupted message base), you should add the '/a' switch to the command line:

```
SMBUTIL /a p \sbbs\data\subs\*.shd
```

The maintenance and pack commands are likely all that you will ever need to SMBUTIL for.

## [15.4] - SMBACTIV - Checks Synchronet Message Base Activity

SMBACTIV.EXE is used to create a list of the number of users that are currently reading each message area (sub-board) on your BBS. If a message area has a very low number of active users, then it may be a waste of disk space and memory on your BBS.

Before SMBACTIV is run, you must have your SBBSNODE environment variable set to point to a valid node directory. Example:

```
SET SBBSNODE=C:\SBBS\NODE1
```

SMBACTIV can be run with no parameters to display a list of all your sub-boards and the number of active users. If you wish to keep this list for reference, it is a good idea to redirect the output to a file or printer. Example:

```
SMBACTIV > PRN
```

You can also specify a maximum number of active users to include in the list of sub-boards as a paramter. Example:

```
SMBACTIV 50
```

would create a list of all sub-boards that have 50 or less active users. To create a list of sub-boards that have no active users, you would use:

```
SMBACTIV 0
```

## [15.5] - ADDFILES - Adds Files to Synchronet File Database

```
usage: addfiles code [.alt_path] [/opts] ["*user"] +list [desc_off] [size_off]
```

```
or: addfiles code [.alt_path] [/opts] ["*user"] file "description"
```

available opts:

```
a  import ASCII only (no extended ASCII)
d  delete list after import
e  do not import extended descriptions
f  include file date in descriptions
t  include today's date in descriptions
i  include added files in upload statistics
n  do not update information for existing files
o  update upload date only for existing files
u  do not update upload date for existing files
z  check for and import FILE_ID.DIZ and DESC.SDI
k  keep original short description (not DIZ)
s  search for files in directory (no file list)
*  use * in place of code for Auto-ADD of FILES.BBS
   use *filename to Auto-ADD a different filename
```

example for importing FILES.BBS format:  
**addfiles games +c:\lists\files.bbs**

example for importing DIRxx format:  
**addfiles games +c:\lists\dir01 33**

#### **SBBSNODE DOS Environment variable:**

The environment variable SBBSNODE must be set prior to running ADDFILES. Add the following line to your AUTOEXEC.BAT:

**SET SBBSNODE=C:\SBBS\NODE1**

Where, the path to NODE1 reflects the complete path of your NODE1 directory.

## **Parameter Explanations:**

#### **dir\_code:**

The dir\_code parameter is the first argument and is the internal code of the Synchronet file directory you are adding the files to. This parameter is not case sensitive. The dir\_code parameter is required. Use \* for Auto-ADD.

#### **.alt\_path:**

If the files are located on an alternate file path, you can specify the alternate file path number by using the ".x" paramter, where 'x' is the number of the alternate path.

#### **/opts:**

If desired, you may specify a list of options. The available options are:

A - Import ASCII character only (ignore any extended ASCII or control chars)  
D - If you want ADDFILES to delete the file list after importing it  
E - If you want ADDFILES to NOT import extended descriptions  
F - Automatically include the file date in the beginning of the description  
T - Automatically include the today's date in the beginning of the description  
I - Include files that are added in the system's upload statistics  
N - Do not update any information for files already in the database  
O - Only update the upload date for files already in the database  
U - Do not update upload date for files already in the database  
Z - Check for and import FILE\_ID.DIZ and DESC.SDI as extended description  
K - Keep original short description (don't override with DIZ)  
S - Search for files in directory (no file list for names and descriptions)

#### **\*user:**

The \*user parameter sets the name of the uploader of the files to the string following the asterisk. An example would be: "\*Digital Man". If an uploader is not specified, files will be added with "-> ADDFILES <-" as the uploader. The quotation marks are only necessary for uploader names of more than one word. The quotation marks are not used as part of the actual name. Example:

```
*Joe   is OK
*Joe Doe is NOT OK
"*Joe Doe"      is OK
```

Multiple uploader names can be used if more than one file or file list is being added. Example:

**ADDFILES GAMES "\*Bob" +FILES.BBS "\*John Doe" TICTAC.ZIP "tic-tac-toe"**

The above command line would add the files listed in FILES.BBS using "Bob" as the uploader, and then add TICTAC.ZIP using "John Doe" as the uploader.

#### **file "description":**

You can specify individual filename and description pairs on the command line to add. The filename is NOT case sensitive and the description IS. The quotation marks are only necessary for descriptions of more than one word. Example:

```
TICTAC.ZIP Tic-Tac-Toe   is OK
TICTAC.ZIP Tic Tac Toe   is NOT OK
TICTAC.ZIP "Tic Tac Toe"      is OK
```

#### **+list [desc\_off] [size\_off]:**

An ASCII text file list can be specified for adding to the database. The format of the file must be as follows (FILES.BBS lists work best):

Filename and description on single line. Number of spaces between filename and description doesn't matter.

Filename must contain a period (.).

Example: TICTAC.ZIP is OK

TICTAC ZIP is NOT OK

TICTAC .ZIP is OK

Lines beginning with control characters or extended ASCII characters are ignored (this includes blank lines).

Up to 58 characters of description are used.

Lines following a filename/description line that begin with space are used as additional information for the description, and the entire description is used as an extended description for the file as well.

The name of the file list is specified on the command line after the plus (+) character. Example:

**ADDFILES GAMES +FILES.BBS**

If the file list is not located in the current directory, you must specify the complete path to the file list:

**ADDFILES GAMES +C:\LISTS\FILES.BBS**

If no filename/description pairs are given on the command line and no +list parameter is given, ADDFILES will attempt to use a file list with the dir\_code as the filename and .LST as the extension.

Example:

**ADDFILES GAMES**

Would add files to the GAMES directory using GAMES.LST as the file list. If GAMES.LST is can not be found, it will look for FILES.BBS in the current directory or in the directory's storage path.

If a file list is specified on the command line, but is not found in the current directory, the storage path for that directory is searched. If the file list exists, it will be imported from there.

**desc\_off:**

If a filelist is specified on the command line, a description offset can be specified as the next argument. If used, this parameter will specify at what column to start importing the descriptions. For PCBoard DIR file format, you should specify a description offset of 33. Example:

**ADDFILES GAMES +DIR10 33**

This offset is also used for the importing of any extended description lines. size\_off:

If a filelist is specified on the command line with a description offset, a file size offset can be specified too. If a file size offset is specified, the disk does not need to be searched for the size of the file to determine the credit value. This is useful for adding lists of files from a CD-ROM drive as it speeds up process since the CD does not need to be scanned for the file size. This parameter is also useful for importing off-line file lists, where the file doesn't actually exist on the drive. For PCBoard DIR file format, this offset should be set to 13. Example:

**ADDFILES GAMES +DIR10 33 13**

## [15.6] - FILELIST - Generate Synchronet File Directory Listings

**Description:**

This utility creates an ASCII text file list of the files in a Synchronet file transfer database. This utility is useful for creating file lists for users to download, exporting into other programs, or for easy statistics reporting by the sysop.

The default output is compatible with the FILES.BBS standard format. Options are available for enhancing the output for your needs.

Understand that this is an EXPORT utility and is not necessary for the normal execution of Synchronet. This utility exports from the binary indexed file databases of Synchronet into ASCII text. If you are looking for a program to IMPORT files from ASCII into Synchronet, you need to look for the ADDFILES utility for Synchronet. If you are just trying to ADD files to your Synchronet database, then you need to read Sysop Commands - specifically the ";UPLOAD" sysop transfer section command.

**Syntax:**

**usage: FILELIST [dir\_code] [switches] outfile**

switches: /LIB name All directories of specified library  
/ALL All directories in all libraries  
/NOT code Exclude specific directory  
/CAT Concatenate to existing outfile  
/PAD Pad filename with spaces  
/HDR Include directory headers  
/CDT Include credit value  
/ULN Include uploader's name  
/ULD Include upload date  
/DFD Include DOS file date  
/DLD Include download date  
/DLS Include total downloads  
/NOD Exclude normal descriptions  
/NOE Exclude normal descriptions, if extended exists  
/EXT Include extended descriptions  
/JST Justify extended descriptions under normal  
/+ Include extended description indicator (+)  
/- Include offline file indicator (-)  
/\* Short-hand for /PAD /HDR /CDT /+ /-

## Parameter Descriptions:

Either the "/LIB name", "/ALL", or <dir\_code> parameter must be included for a list to be generated. If you want to create a list of files for a single file directory, use that directory's internal code as the first parameter on the command line. If you want all directories in a certain library to be in the list, use the "/LIB name" switch - where "name" is the short name of the library you want to list. If the library short name is more than one word, you must put quotes around the name (e.g. /LIB "short name"). If you want to include all the directories on your system in the list, use the "/ALL" parameter.

If you want to exclude a specific directory from a library (when using the "/LIB" or "/ALL" parameters), use the "/NOT" switch and follow the switch with the internal code of the directory to NOT include in the list. If you wish to exclude multiple directories, you must use multiple "/NOT" switches.

If you want the list to be appended to an existing file rather than overwrite the original file, include the "/CAT" switch on your command line.

The default format for file names is non-padded (FILE.EXT). If you want the list to have the filename and extension in separate columns (like the file listings inside Synchronet), use the "/PAD" switch, so that "FILE.EXT" will be displayed as "FILE .EXT".

If you want a small header to be placed at the beginning of each directory, include the "/HDR" switch. The header contains the library short name, the directory long name, and the total number of files in the directory.

If you want the credit value of the file (normally the same as the file size) included in the listing, include the "/CDT" switch on the command line.

If you want the name of the uploader of each file to be included in the list, use the "/ULN" switch on the command line.

If you want the date of the upload (in MM/DD/YY format) to be included in the list, use the "/ULD" switch on the command line.

If you want the DOS file date to be included in the file listing, use the "/DFD" switch on the command line.

If you want the date of the most recent download (in MM/DD/YY format) to be included in the list, use the "/DLD" switch on the command line.

If you want the total number of downloads for each file included in the list, use the "/DLS" switch on the command line.

If you DO NOT want the normal (58 character) description included in the file list, include the "/NOD" switch on the command line.

If you want the normal description to be excluded only if an extended description exists, then use the "/NOE" switch. This is useful for generating file lists from a directory where the original descriptions were imported from an ASCII file list using ADDFILES. When using this switch, you do not have to include the "/EXT" switch.

If you want extended descriptions to be included in the file list, use the "/EXT" switch on the command line.

If you want extended descriptions to be automatically justified under the normal description, include the "/JST" switch on the command line. You do not have to include the "/EXT" switch if you use this switch.



If you want the '+' identifiers for extended descriptions to be included in the list, use the "/"+" switch.

If you want the existence of each file to be verified and non-existent (offline) files to be indicated by a '-', use the "/"- switch. This switch should NOT be used for CD-ROM directories.

To generate a list most closely resembling the internal Synchronet file listing format, use the "/"\* switch. It is the same as including the "/HDR", "/CDT", "/PAD", "/"+" , and "/"- switches.

#### **SBBSNODE Environment Variable:**

Prior to running FILELIST you must set the SBBSNODE environment variable to the path of one of your NODE directories.

Example:

```
SET SBBSNODE=C:\SBBS\NODE1
```

#### **Examples:**

##### **FILES.BBS**

If you want to create a simple FILES.BBS format listing of your GAMES directory (and the directory's internal code is "GAMES"), then use the following command line:

```
FILELIST GAMES FILES.BBS
```

This command line would create the file "FILES.BBS" in your current DOS directory with a list of the filenames and descriptions in your Synchronet GAMES file directory. This file could then be imported into another BBS package or used by another FILES.BBS compatible program for searching, sorting, moving or other file maintenance.

##### **FILELIST.TXT**

If you want to create a list of all the files on your BBS that closely matches the format of the internal Synchronet file listings, use the following command line:

```
FILELIST /* /ALL FILELIST.TXT
```

This command line would create the file "FILELIST.TXT" in your current DOS directory with a list of the filenames, credit values, and descriptions of all the files in your Synchronet file transfer database. A header for each file directory would be included and the filenames would be padded with spaces for easier viewing.

To generate the same list, but include any extended descriptions as well, add the "/EXT" switch after the "/ALL" parameter.

##### **FILESTAT.TXT**

If you wanted to create a list of all the files on your BBS with detailed statistical information, you may want to use a command line similar to the following:

```
FILELIST /* /ALL /NOD /ULN /ULD /DLD /DLS FILESTAT.TXT
```

This command line would create a list of the files with the uploader's name, the date the file was uploaded, the date of the most recent download, and the total number of downloads. The "/NOD" switch would cause the descriptions to be excluded from the list. This list could then be used for statistic reports. It would be a fairly simple programming task to create a utility that read in this generated file and created file popularity graphs or a list of the most valued uploaders. The possibilities are endless.

#### **NOTE:**

The functionality of creating file lists is not limited to only this utility. There are commands to generate file lists (of both New files and All files) from the Temp Directory menu in the Synchronet transfer section. Users can use these commands to generate lists for download immediately. Also, QWK packets generated in Synchronet contain a file named "NEWFILES.DAT" - an ASCII text file containing a list of files uploaded since the user's last logon.

The main advantages of this utility over the internal file list generation capabilities of Synchronet are its output format flexibility and compatibility with the FILES.BBS standard.

## [15.7] - DUPEFIND - Synchronet Duplicate File Finder

DUPEFIND.EXE is used to find duplicate file names in your Synchronet file database. This is most useful for CD-ROM installations, where files may be duplicated on the CD-ROM and your hard disk. Duplicate files can be a waste of memory and disk space on your BBS.

Before DUPEFIND is run, you must have your SBBSNODE environment variable set to point to a valid node directory. Example:

```
SET SBBSNODE=C:\SBBS\NODE1
```

DUPEFIND can be run with no parameters to search all of your file directories in Synchronet for duplicate file names. If you wish to keep this list for reference, it is a good idea to redirect the output to a file or printer. Example:

```
DUPEFIND > PRN
```

You can also specify a starting and ending library number to limit the search to certain libraries. Example:

```
DUPEFIND 1 3
```

would only search file libraries 1 through 3 for duplicate file names.

## [15.8] - DELFILES - Removes Files from Synchronet File Database

**usage: DELFILES <dir\_code or \* for ALL> [switches]**

switches: /LIB name All directories of specified library  
/NOT code Exclude specific directory  
/OFF Remove files that are offline (don't exist on disk)  
/NOL Remove files with no link (don't exist in database)  
/RPT Report findings only (don't delete any files)

**Examples:**

If you wanted to remove ALL files that are offline (don't exist on disk) you would use the command line:

```
DELFILES * /OFF
```

To remove files that exist on the disk but not in the Synchronet file database you would use:

```
DELFILES * /NOL
```

Or to simply remove files that match the criteria specified in the SCFG program you would use:

```
DELFILES *
```

The /NOT parameter is used to exclude certain directories:

```
DELFILES * /NOT GAMES /NOT TEXT
```

Or you can specify a library name rather than a directory name:

```
DELFILES /LIB <library name>
```

Appending /RPT to the command line will cause DELFILES to generate a report of files that would have been removed, but it will not actually remove any files.

## [15.9] - Synchronet MLABELS Utility

The MLABELS utility is used to generate a list of mailing labels from a Synchronet user database. Two basic label forms are supported, single column (e.g. Avery 4145) and double column (e.g. Avery 4143). Specific groups of users to print labels for can be specified by level, flags, exemptions, or restrictions. If the BBS is real name based (no aliases), and the "Company Name" new user question is toggled on in SCFG, then the company name will appear at the top of the address and an ATTN: <User's Name> can optionally be added to the end of the address (with the use of the "/A" command line switch).

Syntax:

```
MLABELS C:\SBBS\DATA\USER [-required] [/options] <output>
```

## User Data Path

The first parameter is the path to your USER.DAT file. The example given is the default location of the USER.DAT file ("\\SBBS\\DATA\\USER").

## Requirements

The second parameter, "-require" is optional. Use this parameter to specify a security requirement for the users to be included. Multiple "-require" parameters can be specified, and the syntax is as follows:

```
-L#   set minimum level to # (default is 0)
-M#   set maximum level to # (default is 99)
-F#<flags> set required flags for flag set # (Default is flag set #1)
-E<flags> set required exemption flags
-R<flags>      set required restriction flags
```

The syntax of the -require option is identical to the ALLUSERS utility, so please see the chapter on ALLUSERS for examples of how to use this parameter.

## Options

The next parameter, if specified, are one or more valid option characters following the slash '/' character. The available option characters are:

```
D   Double column labels
A   Add ATTN: <Alias/Real Name> to the labels
```

## Output

The last parameter is the name of the output file. If you want the output to go immediately to the printer, specify "PRN" as the output file.

Example:

```
MLABELS \\SBBS\\DATA\\USER PRN
```

If you wish to view the labels before you print them, specify a filename.

Example:

```
MLABELS \\SBBS\\DATA\\USER LABELS.TXT
```

## Complete Example

```
MLABELS \\SBBS\\DATA\\USER -L50 -M59 -FP /DA PRN
```

The above command line would create a double wide mailing list containing all users with security levels between 50 and 59 and flag 'P' from flag set #1, and send the output directly to the printer, adding "ATTN: <Alias/Name>" to the end of each label.

## Example label:

```
John Doe
555 Main St.
Small Town, Ny 01234
ATTN: Mr. Anonymous
```

# [15.10] - Synchronet QWKNODES Utility

**usage: qwknodes [/opts] cmds**

```
cmds: r   = create ROUTE.DAT
       u   = create USERS.DAT
       n   = create NODES.DAT
```

```
opts: f   = format addresses for nodes that feed from this system
       a   = append existing output files
       t   = include tag lines in NODES.DAT
       l   = include local users in USERS.DAT
       m#  = maximum message age set to # days
```

The QWKNODES utility is used to scan through all the messages in all of your QWK networked sub-boards and create one or more lists:

## ROUTE.DAT

The DATA\\QNET\\ROUTE.DAT file is automatically created and maintained by SBBS. It includes the routing necessary to get netmail from your BBS to any other QWKnet BBS (using Synchronet's QWKnet extensions) in your QWK network. Entries are automatically added, modified, and removed (when out-dated). You can use QWKNODES to create this file (by scanning your QWK networked message bases), but it isn't normally necessary.

If you are a QWKnet hub and wish to create a ROUTE.DAT for your QWKnet nodes (other BBSs that call your BBS for QWKnet messages), you must include the /F command line option when creating the file (and DO NOT put that ROUTE.DAT in your DATA\QNET directory as the routing information will be incorrect for your system). This is a convenience for your QWKnet nodes which allows them to immediately send routed QWK netmail (without waiting for SBBS to create the ROUTE.DAT automatically while parsing incoming echomail).

#### USERS.DAT

The DATA\QNET\USERS.DAT file (if it exists) is used by SBBS to look-up user names on other BBSs in your QWK network when attempting to send e-mail to an unknown user name. The QWKNODES utility must be used to create this file (most likely via timed event). If the /L command line option is specified when this file is created, it will include the names of users that posted from YOUR BBS as well as all others in the network (not of any use to SBBS, but informative if you wish to create a list of ALL active users in your QWK network). Only users who are active in the message bases (on any of the BBSs) will appear in the USERS.DAT file.

#### NODES.DAT

While this file is informative (list of all QWKnet nodes that have generated messages in your QWKnet sub-boards) it is not used by SBBS currently. If the /T command line option is specified when this file is created, it will include a copy of each node's tagline in the list in addition to their QWK-ID and routing details.

#### /A

The /A option is used to append (concatenate) an existing output file. If this option isn't specified any existing output file (ROUTE.DAT, USERS.DAT, or NODES.DAT) will be truncated and overwritten.

#### /M#

The maximum message age option is used to specify a maximum age of messages (in days) to include in the list of messages scanned to obtain user/node /routing information from (example: /M90 to specify a maximum age of 90 days causing QWKNODES to ignore any messages older than 90 days). The default behavior is all messages (no maximum age).

**Note:** Output files are created in the current directory. While it is possible to create all three output files from one execution of QWKNODES, the U command will cause duplicate entries in the NODES.DAT and ROUTE.DAT files. So it is best to create the USERS.DAT in a separate execution of the QWKNODES utility. Example:

```
QWKNODES RN -> To create ROUTE.DAT and NODES.DAT
QWKNODES U -> To create USERS.DAT
```

## [15.11] - Synchronet ALLUSERS Utility

ALLUSERS can allow a sysop to modify the security settings for a large number of users with one command line. If you're a Synchronet sysop, you may have found yourself at one time or another going through the internal User Editor searching for users with a certain flag or exemption and changing their level, removing an exemption, adding a restriction, etc. The User Editor's ARS search facility is very quick and flexible, but having to hand modify each user, even with the use of macros, can be very tedious with a large user database.

This program attempts to eliminate the monotony of such a task. With a single command line, you can change security levels, add/remove flags, exemptions, or restrictions to some or all of the users in your database. The syntax is pretty simple:

#### Syntax

**ALLUSERS C:\SBBS\DATA\USER -require /modify**

#### User Data Path

The first parameter, "C:\SBBS\DATA\USER" is the directory where your USER.DAT file is located. The example given is the default location of the USER.DAT file. If the USER.DAT is in the current directory, the path should be simply "." for current directory. This parameter is not optional.

#### Requirements

The second parameter, "-require" is optional. Use this parameter to specify a security requirement for the modifications to follow. Multiple "-require"

parameters can be specified, and the syntax is as follows:

```
-L#   set minimum level to # (default is 0)
-M#   set maximum level to # (default is 99)
-F#<flags> set required flags for flag set # (Default is flag set #1)
-E<flags> set required exemption flags
-R<flags> set required restriction flags
```

#### Examples:

```
-L20   indicates that only users with a level of 20 or higher will be modified
-M80   indicates that only users with a level of 80 or lower will be modified
-FC    indicates that only users with flag 'C' from flag set #1 will be modified
-F3G   indicates that only users with flag 'G' from flag set #3 will be modified
-RA    indicates that only users with the 'A' restriction will be modified
```

If multiple requirement parameters are specified, they must each begin with a dash '-' character and be separated by at least one space.

#### Examples:

```
-L50 -M59   indicates that only users between level 50 and 59 will be modified
-RB -F2M    indicates that only users with the 'B' restriction and flag 'M' in
            flag set #2 will be modified
```

Multiple required flags may be specified as well.

#### Example:

```
-F4AC      indicates that only users with both the 'A' and 'C' flags from
            flag set #4 will be modified
-EPLM      indicates that only users with the 'P', 'L', and 'M' exemptions
            will be modified
```

#### Modifications

Next on the command line are one or more modification parameters. Each modification parameter begins with the slash '/' character. The available modification parameters are as follows:

```
/L#   set level to #
/F#[+|-]<flags> add or remove flags from flag set #
/E#[+|-]<flags> add or remove exemption flags
/R#[+|-]<flags> add or remove restriction flags
```

#### Examples:

```
/L20   indicates that all users' levels will be changed to 20
/FA     indicates that flag 'A' from flag set #1 will be added to all users
/F3-G   indicates that flag 'G' will be removed from flag set #3 for all users
/E+L    indicates that exemption 'L' will be added to all users
/R-N    indicates that restriction 'N' will be removed from all users
```

Flag set #1 is the default, so "/F1+A" and "/F+A" are equivalent.  
Addition of flags (+) is the default, so "/F+A" and "/FA" are equivalent.  
Flags and parameters are not case sensitive, so "/FA" and "/fa" are equivalent.

Multiple modification parameters can be used.

#### Example:

```
/L40 /F2-Z   indicates that all users will be set to level 40 and have flag
            'Z' from flag set #2 removed
```

Multiple flags per modification may be specified.

#### Example:

```
/RAB        indicates that both 'A' and 'B' restrictions will be added to
            all users.
```

#### Complete Examples

To change all level 20 users to level 30, use the following command line:

```
ALLUSERS C:\SBBS\DATA\USER -L20 -M20 /L30
```

To give all QWKnet node users the 'M' exemption, use the following command:

```
ALLUSERS C:\SBBS\DATA\USER -RQ /EM
```

To give all users below level 90, the 'B' restriction, use the following:

```
ALLUSERS C:\SBBS\DATA\USER -M89 /RB
```



To give all users with level 50 or higher, the 'M' and 'L' exemptions:

```
ALLUSERS C:\SBBS\DATA\USER -L50 /EML
```

To remove flag 'C' from flag sets #1 and #2 from all users:

```
ALLUSERS C:\SBBS\DATA\USER /F1-C /F2-C
```

## [15.12] - Synchronet AUTONODE Utility

The AUTONODE utility is used for automatically finding an available local node for logon and running that node. You set the first local node number in SCFG->System->Advanced Options->First Local Auto-Node.

In order for the AUTONODE utility to run, you must first set the SBBSCTRL and SBBSNODE environment variables. Add the lines:

```
SET SBBSCTRL=C:\SBBS\CTRL
SET SBBSNODE=C:\SBBS\NODE1
```

to your AUTOEXEC.BAT file. If your CTRL or NODE1 directories are in a different locations, then specify the correct paths accordingly.

This utility is useful for systems that have more than one local node. This eliminates the need for those users to change into an unused node directory and run SBBS. It will automatically find the first unused local node and take them to the logon prompt. When they logoff, they will return to the DOS prompt.

If for some reason, you do not wish the user to automatically go to a logon prompt or to exit back to DOS after logoff, you can specify which command line to use when running SBBS. The default is "SBBS L Q", which tells Synchronet to take the user straight to the logon prompt and then quit back to DOS.

## [15.13] - Synchronet Node Display/Control Utility

The NODE utility can be used to display or control the status of nodes from the DOS prompt or via batch files.

**usage: node [/debug] [action [on|off]] [node numbers] [...]**

actions (default is list):

list	= list status
anon	= anonymous user
lock	= locked
intr	= interrupt
down	= shut-down
rerun	= rerun
event	= run event
nopage	= page disable
noalerts	= activity alerts disable
status=#	= set status value (definition of status values at end of section)
useron=#	= set useron number
action=#	= set action value (definition of action values at end of section)
errors=#	= set error counter
conn=#	= set connection value
misc=#	= set misc value
aux=#	= set aux value
extaux=#	= set extended aux value

Before the NODE utility can be run, the SBBSCTRL environment variable must be set to point to your CTRL directory. Example:

```
SET SBBSCTRL=C:\SBBS\CTRL
```

Place this line in your AUTOEXEC.BAT file to automatically set this variable at boot-up. It is necessary for the NODE utility to find the shared node control files. If your CTRL directory is in a different location, use the correct path for the SET command.

If you want to be able to run the NODE utility from any directory, you must have the NODE.COM program in one of your DOS search directories. Either copy this file into a directory already in your DOS search path, or add your Synchronet EXEC directory to your search path in your AUTOEXEC.BAT. Example:

```
SET PATH=C:\DOS;C:\UTIL;C:\SBBS\EXEC
```

(DOS and UTIL directory are included for example only).

**Running NODE**

To run the NODE utility, you must type the word "NODE", an action (ex: "LOCK"), then a list of nodes to take the action on (ex: "1 2 4 5"). If no node numbers are specified, it is assumed you wanted to perform the action on ALL nodes.

Some actions can have an option "ON" or "OFF" state specified. Typing "NODE LOCK 1" would toggle the "locked" state of node 1. If it were previously off, it would be set to on. To be sure you are setting the state to either on or off rather than toggling, you may specify on or off (e.g. "NODE LOCK ON 1" would set the locked state of node 1 to on).

Some actions are setting a value for the node status. These actions are listed with "=#" after the action word. This indicates that you must specify a numeric value for that status option (e.g. "NODE USERON=1 5" would set the current user number on node 5 to 1).

The optional "/debug" switch can be used to view the current numeric values associated with the node status.

### **Example Usage**

Example #1: To list the status of all your nodes, type

**NODE LIST**

Example #2: To lock node 1, type

**NODE LOCK ON 1**

Example #3: To unlock nodes 1 and 2, type

**NODE LOCK OFF 1 2**

Example #4: To set the status of node 3 to "Offline", type

**NODE STATUS=5 3**

Example #5: To interrupt and lock node 4, type

**NODE INTR ON 4 LOCK ON 4**

Example #6: To clear the error counter of all nodes, type

**NODE ERRORS=0**

Example #7: To toggle the "rerun" status of all nodes, type

**NODE RERUN**

Example #8: To down node 2, type

**NODE DOWN ON 2**

### **Definition of Numeric Values**

Possible Node Status Values:

- 0 Waiting for call
- 1 At logon prompt
- 2 New user applying for access
- 3 User online
- 4 User online in quiet mode
- 5 Offline
- 6 Networking
- 7 Waiting for all nodes to become inactive before running timed event
- 8 Running timed event
- 9 Waiting for timed event node to finish running event

### **Possible Node Action Values:**

- 0 Main Prompt
- 1 Reading Messages
- 2 Reading Mail
- 3 Sending Mail
- 4 Reading G-Files
- 5 Reading Sent Mail
- 6 Posting Message
- 7 Auto-message
- 8 Running External Program (aux=program number)
- 9 Main Defaults Section
- 10 Transfer Prompt
- 11 Downloading File (aux=estimated time of transfer completion)
- 12 Uploading File
- 13 Bi-directional Transfer (aux=estimated time of transfer completion)
- 14 Listing Files

- 15 Logging on
- 16 In Local Chat with Sysop
- 17 In Multi-Chat with Other Nodes
- 18 In Local Chat with Guru
- 19 In Chat Section
- 20 Sysop Activity
- 21 Transferring QWK packet
- 22 In Private Chat (aux=node chatting with)
- 23 Paging another node for Private Chat (aux=node being paged)
- 24 Retrieving file from a sequential device (aux=device num)

**Bits Used in Node Misc:**

- 0 Anonymous User
- 1 Locked for sysops only
- 2 Interrupted - hang up
- 3 Message is waiting for user
- 4 Paging disabled
- 5 Activity Alert disabled
- 6 User data has been updated by another node
- 7 Re-run this node when logoff
- 8 Must run node event after logoff
- 9 Down this node after logoff
- 10 Reset private chat
- 11 Message is waiting for node

## [15.14] - UTI Driver

This UTI driver complies with UTI Driver Specification Rev 2.1 by Kip Compton. This driver is for use with PostLink v1.05+ and possibly MegaMail.

Copy the UTI driver files:

- UTIVER.EXE Returns UTI version number (2)
- UTILIST.EXE Generates list of conferences (sub-boards)
- UTIHIGH.EXE Returns highest message number in a conference
- UTILSTRD.EXE Generates list of message pointers for a user
- UTIIMPRT.EXE Imports messages into a conference
- UTIEXPRT.EXE Exports messages from a conference

into your PostLink directory, or another directory that is in your DOS search path.

The UTIDOOR.EXE program is not part of this driver set, since Synchronet can generate a UTIDOOR.TXT file internally.

You may need to know that the conference identifiers, as well as the name of each conference, is the same as the Synchronet internal code for each conference. This allows you to modify your message base configuration in Synchronet without having to immediately "Update" the network or mail software conference list to avoid a catastrophe. However, you should still "Update" the network or mail software to avoid configuration confusion - especially when deleting sub-boards.

**ERROR LEVELS**

If one of the UTI driver programs exits with an error level, you can define the error with the following table. If the UTI driver program exits with an unlisted error level, please contact Digital Dynamics.

**Level Description**

- 1 Syntax error on command line
- 2 Cannot open/create UTI text file
- 3 Memory allocation error
- 4 Too many messages in a conference to import
- 5 Cannot open Synchronet message data file
- 6 Cannot open Synchronet message index file
- 7 Unrecognized Synchronet sub-board code
- 8 Cannot open Synchronet user name data file
- 9 Cannot find user name in Synchronet user database
- 10 Cannot open Synchronet message pointer index

If you get an error 2, 5, 6, 8, or 10, you may want to double check the number of files handles you are reserving in your CONFIG.SYS file (FILES=# statement) and try increasing it.

If you get an error 3, then you need to make more DOS memory available to the UTI driver and the parent program (i.e. PostLink).

## [15.15] - System/Node Statistics Log Viewing Utility

**Usage:** `slog [path]`

where path is the directory where CSTS.DAB is located. If your SBBSCTRL environment variable is set and no path is specified, it will use the system's statistics file located in the CTRL directory. To set your SBBSCTRL environment variable, add the following line to your AUTOEXEC.BAT:

**SET SBBSCTRL=C:\SBBS\CTRL**

Make sure the path is the correct path for the Synchronet CTRL directory in your configuration.

If you want to list the statistics of a specific node on your system (same as using the ;NLOG command from the main menu, or the 'N' WFC command), specify the path for that node on the command line. Example:

**SLOG C:\SBBS\NODE1**

If a path is not specified and the SBBSCTRL environment variable is not set, the current directory will be searched for CSTS.DAB.

The output of SLOG can be redirected to a file or printer for easy viewing. Example:

**SLOG > PRN**

or

**SLOG > SLOG.TXT**

## [15.16] - Daily Statistics Editor Documentation

### Description:

DSTSEEDIT (Daily Statistics Editor) is used to edit the statistic values of your BBS that are stored in the DSTS.DAB file in your CTRL directory. There is also a separate DSTS.DAB file in each node directory which stores the statistic values for that individual node. This utility can be used to edit either of the two statistic types: System or Node.

### Usage:

To edit your system's statistics, you can either run DSTSEEDIT with your CTRL directory as the current directory, or run DSTSEEDIT with the path of the CTRL directory as an argument.

Example: **DSTSEEDIT C:\SBBS\CTRL**

To edit an individual node's statistics, you can either run DSTSEEDIT with the node's directory as the current directory, or run DSTSEEDIT with the node's directory as an argument.

Example: **DSTSEEDIT C:\SBBS\NODE1**

## [15.17] - TOTALS: External Programs Credit Gain/Loss Log Totaling Utility

The TOTALS.COM utility is used to generate a total gain/loss of multiple log files created by SBBS external programs that adjust user credits and log the adjustments in text files. Each file containing one line that contains the gain/loss value in credits. It is a positive value if it had net winnings (took more credits than it gave), and negative value if it had a net loss (gave away more credits than it took). The Synchronet external programs that currently generate such logs (by running the program with the /L option) are Synchronet Blackjack, Dice War, and Domain Poker.

**usage: totals <log files [...]>**

examples:

**totals \*.log**  
**totals 06\*.log 07\*.log 080192.log**

## [15.17] - ANS2ASC

ANSI to Synchronet Ctrl-A message format conversion utility

This utility will convert basic non-animated ANSI files into Synchronet Ctrl-A code files that can be displayed to color or monochrome ansi users with attractive results as well as non-ansi users. This allows you to use an ANSI drawing program (such as TheDraw) to create your menus in ANSI and convert the ANSI file to Ctrl-A message format for your .ASC or .MSG version. Or you can

use the converted file for all users by keeping the ANSI (.ANS) version in a directory other than TEXT\MENU.

If, for example, you create a main menu with an ANSI drawing program that you want your ASCII and monochrome ANSI users to see with attractive output, use ANS2ASC to create the .MON and .ASC versions.

Type: **ans2asc main.ans main.mon**  
in your TEXT\MENU directory to create the monochrome version.

Type: **ans2asc main.ans main.asc**  
in your TEXT\MENU directory to create the ASCII version.

If you rename MAIN.ANS or move into a directory other than TEXT\MENU, then you will not need the .MON version as the .ASC version will be used for all terminal types.

This utility is also useful for creating colorful system, newuser, feedback and other Synchronet .MSG files for display. For example, create TEXT\SYSTEM.ANS with your favorite ANSI editor and then convert to .MSG by typing: **ans2asc system.ans system.msg**

Be aware that ANSI animation codes are NOT supported since there aren't Ctrl-A equivalents. The following ANSI codes (preceded by <ESC>[]) are supported:

**ANSI Code    Ctrl-A Code    Explanation (\* indicates not supported by IBM)**

2J	L	Clear Screen
#C	7Fh-FFh	Move cursor right # columns
0m	N	Normal attribute
1m	H	High intensity
2m	N	Low intensity *
3m	I	Italic *
4m	I	Underline *
5m	I	Blink
6m	I	Rapid Blink *
7m	H	Reverse Video *
8m	E	Concealed text *
30m	K	Foreground black
31m	R	Foreground red
32m	G	Foreground green
33m	Y	Foreground yellow
34m	B	Foreground blue
35m	M	Foreground magenta
36m	C	Foreground cyan
37m	W	Foreground white
40m	0	Foreground black
41m	1	Foreground red
42m	2	Foreground green
43m	3	Foreground yellow
44m	4	Foreground blue
45m	5	Foreground magenta
46m	6	Foreground cyan
47m	7	Foreground white

## [15.18] - ASC2ANS

Converts Synchronet Ctrl-A file to ANSI escape sequences.

This utility allows you to convert files that use Ctrl-A codes into ANSI files. (the opposite of the ANS2ASC utility).

The syntax is: **asc2ans main.asc main.ans**

The extension for the input file will most likely be .ASC or .MSG and the output file should be .ANS.

This utility is also useful for viewing files with Synchronet Ctrl-A codes from the DOS prompt.

Example: **asc2ans main.asc con**

This will read from the Ctrl-A file MAIN.ASC and write to your screen using ANSI escape sequences.

[Back to Top](#)

---

**Copyright © 2000 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.



Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)

PO Box 501

Yorba Linda, CA 92885

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .



[Back to Table of Contents](#)

## [17.1] - Customization: Menus and Text Files

One of the easiest and possibly the most obvious ways to customize or personalize your BBS is to change the look of the menus. Menus are merely ASCII text files (with optional ctrl-a codes or ANSI escape sequences) stored in the TEXT\MENU directory. The filenames are descriptive of the menu subject and the extensions represent the content of the file. The possible extensions and their meanings are:

<b>RIP</b>	Contains RIPscrip escape sequences for use with RIPterm
<b>WIP</b>	Contains WIP escape sequences for use with DC-Term
<b>ANS</b>	Contains ANSI escape sequences suitable for color display
<b>MON</b>	Contains ANSI escape sequences suitable for monochrome display
<b>ASC</b>	Contains no ANSI

All of the above file types can contain ctrl-a codes, and only the ASC file must exist. If a user has color ANSI, the ANS file will be displayed; if it doesn't exist the ASC file will be displayed. If a user has monochrome ANSI, the MON file will be displayed; if it doesn't exist the ANS file will be displayed; and if it doesn't exist the ASC file is then displayed.

A user without ANSI will always be displayed the ASC file.

To edit files with ANSI escape sequences, it is usually preferable to use a utility designed for such a task. TheDraw is quite popular for this use.

To edit files with ctrl-a codes, you can use any editor that allows the input of ctrl characters, but you won't see the attributes till you view the file within Synchronet. You can, however, use the Synchronet internal editor (;EDIT from the main menu) and it will display the attributes as you edit the file. The Synchronet editor limits the line length to 79 characters which may not be sufficient for lines with multiple ctrl-a codes.

The best way to edit files with Ctrl-A codes is to first convert them to ANSI with MSG2ANS.EXE (see the Utility reference for more information). Then edit with an ANSI editor (such as TheDraw). Then convert back to Ctrl-A format using ANS2MSG.EXE (see the Utility reference for more information).

### Menu Files Description

~~~~~	
ALLMAIL	Sysop's reading all mail on system menus
ATTR	Ctrl-A code menu for use within the Synchronet internal editor
BATCHXFR	Batch transfer menu
BATDPROT	Batch download transfer protocols
BATFLAG	Batch flag command key menu
BATUPROT	Batch upload transfer protocols
BIPROT	Bidirectional transfer protocols
CHAT	Chat section menu
DLPROT	Download transfer protocols
E-MAIL	E-mail section menu
EDITOR	Synchronet internal editor commands and line editing keys
EXEMPT	Exemption flag descriptions for use within User Edit
FLAGS1	Flag set #1 descriptions for use within User Edit
FLAGS2	Flag set #2 descriptions for use within User Edit
FLAGS3	Flag set #3 descriptions for use within User Edit
FLAGS4	Flag set #4 descriptions for use within User Edit
LOGOFF	Logoff ('O' command) screen
LOGON	Logon screen (LOGON2, LOGON3, ... LOGON9 also supported)
MAILREAD	Reading mail menu
MAIN	Main section menu
MAINCFG	Main configuration menu
MAININFO	Main information menu
MSGSCAN	Message reading/scanning menu
MULTCHAT	Multinode chat menu
PRIVCHAT	Private node-to-node chat menu
QWK	QWK Packet menu
RESTRICT	Restriction flag descriptions for use within User Edit
SENTMAIL	Reading sent mail menu
SYSMAILR	Sysop additional commands while reading mail
SYSMAIN	Sysop additional commands for main section
SYSMSCAN	Sysop additional commands while message reading/scanning
SYSSMAIL	Sysop additional commands while reading sent mail
SYSXFER	Sysop additional commands for transfer section

TEMPXFER	Temporary directory commands menu
TRANSFER	Transfer section menu
UEDIT	User Edit menu
ULPROT	Upload transfer protocols
WFC	Waiting for call menu
XFERCFG	Transfer section configuration menu
XFERINFO	Transfer section information menu

**Optional Menus**

The following files must be created in the TEXT\MENU directory if you wish to use them.

GRPS	Message Group listing
SUBSx	Message Sub-board listing, 'x' is the group number
LIBS	File Library listing
DIRSx	File Directory listing, 'x' is the library number
XTRN_SEC	External Program Sections
XTRNx	External Program listing, 'x' is the section number
TEXT_SEC	Text File Section listing
TEXTx	Text File listing, 'x' is the text file section number
CHAN	Multinode Chat Channels
TMESSAGE	Displays when the user enters the transfer section
TPOLICY	Transfer policy (&T transfer section command)

**Other Optional Message Files**

If you want to have an information file displayed for sub-boards when using the 'I' scanning command and 'IS' main menu command, create a file for the sub-board in the DATA\SUBS directory using the internal code for that sub-board as the name and .MSG as the extension. Example:

**DATA\SUBS\GENERAL.MSG**

If you want to create in information file for file directories to be displayed with the 'ID' transfer section command, create a file for the directory in the DATA\DIRS directory using the internal code as the name and .MSG as the extension. Example:

**DATA\DIRS\GAMES.MSG**

If you want to create a custom file listing header for a file directory, create a file in the DATA\DIRS directory using the internal code as the name and .HDR as the extension. Example:

**DATA\DIRS\GAMES.HDR**

**Colors**

To modify some of colors of the BBS you can edit the ATTR.CFG file which is located in the CTRL directory. The file contains one color per line and is commented as to the use of the color. The colors are represented with ctrl-a code attribute letters/numbers. The number of colors you can change with this method is very small.

**Text Files**

There are some text files that are displayed to users at different points in the system. The locations and descriptions follow (\* indicates optional):

TEXT\ANSWER.ANS	Answer message for ANSI users
TEXT\ANSWER.ASC	Answer message for non-ANSI users
TEXT\ANSWER.RIP	Answer message for RIP users
TEXT\SYSTEM.MSG	Description of the system and its configuration
TEXT\NEWUSER.MSG	Displayed to new users (usually system rules)
TEXT\FEEDBACK.MSG	Displayed before new users write validation feedback
TEXT\NUPGUESS.MSG	*Displayed to callers attempting to guess the NUP (after failing to guess correctly)
TEXT\TOOSLOW.MSG	*Displayed to users logging onto a node at less than the minumum configured connect rate for that node
TEXT\BADCID.MSG	*Displayed to users calling from a number contained in CID.CAN
TEXT\BADNAME.MSG	*Displayed to new users trying to use a name contained in NAME.CAN
TEXT\BADPHONE.MSG	*Displayed to new users using a number contained in PHONE.CAN
TEXT\BADFILE.MSG	*Displayed to user trying to upload filename contained in FILE.CAN
TEXT\QWK\HELLO	*Included in QWK packets
TEXT\QWK\BBSNEWS	*Included in QWK packets
TEXT\QWK\BLT-0.?	*Included in QWK packets (? must be number)

**Logon Message Flow Chart**

The order of the % specifiers (if they exist) in a TEXT.DAT line cannot be altered. The display of %s specifiers can be suppressed by changing the '%s' to '%.0s'. Another way to suppress the display of specifiers is to enclose them between Ctrl-A( and Ctrl-A). Any text between Ctrl-A( and Ctrl-A) would only be displayed to users of level 90 or higher. To suppress the display to all users, put the text/specifiers between Ctrl-ACtrl-Z and Ctrl-A) (assuming that the Z flag from flag set #1 is not set on any user accounts).

**\*\* WARNING \*\***

Make a backup of the TEXT.DAT file before you edit it. If you damage the file syntax when editing it, Synchronet may execute erroneously or even fail to initialize.

## [17.1.2] - Node Action Text

The node action text can be over-ridden by editing the NodeAction lines in the TEXT.DAT file (see previous section for details on TEXT.DAT). The node action text is what is displayed on the node status line when a node is in use. (i.e. instead of "Node 1: So-and-so uploading at 14400bps", you could make it say whatever you like).

You can also include the following optional specifiers (in this order):

**%s** User's name or alias  
**%u** User's security level  
**%u** User's age  
**%c** User's sex (gender, M or F)  
**%s** User's computer type  
**%s** User's note  
**%s** Date user was first online  
**%u** Auxiliary value (chat channel, door number, paged node, etc)  
**%u** Connection rate (in bps)

If you include any of the above specifiers, you must also include any of the specifiers above it. The order of the specifiers cannot be changed. If you wish to suppress the display of a %s specifier, use "%.0s" instead of "%s". To suppress other specifiers, see the previous section for details.

## [17.1.3] - Trash Can Files

Trash can files are used to be able to reject text during certain sequences on the BBS. For example, if you don't want a user to be able to log on with a certain name or use a certain phone number, you would place the information you don't want them to use into the appropriate trash can file. Synchronet looks for trash can files in the \SBBS\TEXT directory, certain ones may already exist, others will need to be created if you wish to use them. The names of the various trash can files and their function are as follows:

**TEXT\CID.CAN** You MUST have caller ID capabilities in your area (as well as a modem which can support those capabilities) in order to use this file. When a call comes in to the BBS who's caller ID phone number matches a phone number in this file, the call will be rejected.

**TEXT\NAME.CAN** When a user attempts to use a name contained in this file as the name for their user account, they will be told that s/he cannot use that name.  
If the file **TEXT\BADNAME.MSG** exists, this will be displayed to the user.

**TEXT\FILE.CAN** When a user uploads a file to the transfer section who's filename matches one of the names contained in this file, the user will be told that s/he cannot upload the file.  
If the file **TEXT\BADFILE.MSG** exists, this will be displayed to the user.

**TEXT\PHONE.CAN** When a user attempts to use a phone number contained in this file as a phone number for their user account, they will be told that s/he cannot use that phone number. This file can also be used by the Synchronet Callback Verifier program.  
If the file **TEXT\BADPHONE.MSG** exists, this will be displayed to the user.

The trash can files also allow special key characters to be used within them, the tilde '~' character means "contained within" the carrot '^' character means "beginning with", the '!' character means "negate the match logic" and the '\*' character means "begins with or ends with" depending on the position in the line. For example:

**sysop** in the name.can would mean users could not use the name "sysop".

**sysop^** or **sysop\*** would mean users could not use names beginning with the word "sysop", like "sysopa" or "sysops" etc.

**\*sysop** or **sysop\*** would mean users could not use names ending with the word "sysop", like "thesysop" or "your\_sysop" etc.

**sysop~** would mean users could not use names that have the word "sysop" anywhere in them, like "imthesysop" or "mesysosphere".

**!sysop~** would mean users *must* use names that have the word "sysop"



anywhere in them, like "imthesysop" or "mesysosphere".

These key characters can be used in any of the trash can files.

## [17.2] - Customization: Message Variables

### Using Message Variables

Message Variables (also called @-Codes) are a way to customize text files in Synchronet to display information about the user online or the BBS. If Synchronet encounters an @-Code in a text file (i.e.: TEXT.DAT, menus, etc.), it will replace the @-Code in the file with the information that it corresponds with. @-Codes in e-mail messages and posts will only be expanded if they were posted locally (not networked) by user #1. TEXT.DAT lines that have % specifiers will not expand @-Codes unless the % specifiers are removed.

For example, placing the following line in a text file displayed to a user named Fred Jones living on 100 Maple Street:

**Hi @ALIAS@, you live at @ADDR1@ don't you?**

Would result in the user seeing:

**Hi Fred Jones, you live at 100 Maple Street don't you?**

It's that simple. The following is a list of the @-Codes that Synchronet will recognize. Remember that the Code NAME must begin and end with an @ symbol. The two columns after the description of the @-Code show which other BBS programs also support that @-Code (either PCBoard or Wildcat).

The @-Codes with "[...]" following the name indicate that you can have the variable displayed with padding. If the @-Code name ends in "-L" the variable will be left justified and "-R" indicates it will be right justified. If for example, the user's name is Bob, the text string "\_\_@NAME-L@" would display to the user as "\_\_Bob\_\_" and the text string "\_\_@NAME-R@" would display to as "\_\_Bob\_\_". If you want the string to be padded longer, you can add extra characters to the end of the @-Code name and before the terminating '@' sign. So for example, "\_\_@NAME-L#####@" would display as "\_\_Bob\_\_" (the "\_\_" segments of the above text examples are only to demonstrate where the padding begins and ends).

\*\*\* Synchronet Supported @-Codes \*\*\*  
( \* Indicates Synchronet specific )

Note: All codes must be uppercase and sandwiched between @ symbols  
(e.g. @USER@).

### System Information

Code	Description	Ver
BBS	Name of BBS	
BOARDNAME	Name of BBS	
CONF	Name of current Group and Sub-board	
CONFNUM	Number of current Group and Sub-board	
CONN	Connection description (modem type, "Telnet", "Local")	2.3c
DATE	Current system date	
DIR	Current file directory short description	
DIR-L[...]	" " padded and left justified	
DIR-R[...]	" " padded and right justified	
DIRL	Current file directory long description	
DIRL-L[...]	" " padded and left justified	
DIRL-R[...]	" " padded and right justified	
DN	Number of current file directory	
DL	" " padded and left justified (4 chars wide)	
DR	" " padded and right justified (4 chars wide)	
FIDOADDR	System's primary FidoNet address	2.3c
FREESPACE	Free disk space available for uploads	
GRP	Current message group short description	
GRP-L[...]	" " padded and left justified	
GRP-R[...]	" " padded and right justified	
GRPL	Current message group long description	
GRPL-L[...]	" " padded and left justified	
GRPL-R[...]	" " padded and right justified	
GN	Number of current message group	
GL	" " padded and left justified (4 chars wide)	
GR	" " padded and right justified (4 chars wide)	
INETADDR	System's Internet Address (as configured in SCFG)	2.3c
LASTCALLERNODE	Name of user last on this node	
LASTCALLERSYSTEM	<same as LASTCALLERNODE>	
LIB	Current file library short description	
LIB-L[...]	" " padded and left justified	
LIB-R[...]	" " padded and right justified	
LIBL	Current file library long description	

LIBL-L[...]	" " padded and left justified	
LIBL-R[...]	" " padded and right justified	
LN	Number of current file library	
LL	" " padded and left justified (4 chars wide)	
LR	" " padded and right justified (4 chars wide)	
LOCAL-IP	System's IP address	3.0b
LOCATION	System location (city, state)	2.3c
NOACCESS	Why user was denied access (last false ARS)	
NODE	Number of current node	
NODE###	Status of node number ###	
NUMCALLS	<same as STATS.LOGONS>	
NUMDIR	Number of current library and directory	
PREVON	<same as LASTCALLERNODE>	
QUESTION	Current Yes/No question (for TEXT\MENU\YESNO.*)	
QWKID	System's QWK BBS-ID	2.3c
REV	Software revision (single letter)	
STATS.LOGONS	Total logons during history of system	3.0b
STATS.LTODAY	Total logons today	3.0b
STATS.TIMEON	Total time used during history of system (in minutes)	3.0b
STATS.TTODAY	Total time used today (in minutes)	3.0b
STATS.ULS	Total uploads today	3.0b
STATS.ULB	Total bytes uploaded today	3.0b
STATS.DLS	Total download today	3.0b
STATS.DLB	Total bytes downloaded today	3.0b
STATS.PTODAY	Total posts today	3.0b
STATS.ETODAY	Total e-mails sent today	3.0b
STATS.FTODAY	Total feedbacks sent today	3.0b
STATS.NUSERS	Total number of new users today	3.0b
SUB	Current message sub-board short description	
SUB-L[...]	" " padded and left justified	
SUB-R[...]	" " padded and right justified	
SUBL	Current message sub-board long description	
SUBL-L[...]	" " padded and left justified	
SUBL-R[...]	" " padded and right justified	
SN	Number of current message sub-board	
SL	" " padded and left justified (4 chars wide)	
SR	" " padded and right justified (4 chars wide)	
SYSDATE	Current system date	
SYSOP	Name of System Operator	
SYSTIME	Current system time	
TCALLS	Total number of logons for system	
TFILE	Total number of files on system	
TIME	Current system time	
TMSG	Total number of messages on system	
TNODE	Total number of nodes on system	
TUSER	Total number of user slots on system	
VER	BBS version number	
WHO	Display status of all active nodes	

User Information

Code	Description	Ver
-----		
ADDR1	User's street address	
ALIAS	User's name or alias	
BAUD	User's connect rate (DCE) in bps	
BDATE	User's birthdate (MM/DD/YY)	
BPS	<same as BAUD>	
BYTELIMIT	User's free credits per day	
BYTESLEFT	User's total credits	
CALLS	Total number of logons for user	
CID	Caller's Caller-ID info or IP address	2.3c
CITY	User's city	
COMPANY	User's company name or real name	
CPU	User's computer type (v2.x) or hostname (v3.x)	
DATA	<same as PHONE>	
DATAPHONE	<same as PHONE>	
DAYBYTES	Number of free credits used today by user	
DLBYTES	Total bytes downloaded by user	
DLFILES	Total files downloaded by user	
DLKLIMIT	User's total credits (in kilobytes)	
DOWNK	Total kilobytes downloaded by user	
DOWNNS	<same as DLFILES>	
EXDATE	User's expiration date (MM/DD/YY)	
EXPDATE	<same as EXDATE>	
EXPDAYS	Days left before user expires	
FIRST	User's first name/alias	
FIRSTREAL	User's first real/company name	
FROM	User's location (City, State)	
HANDLE	User's chat handle	
HOMEPHONE	<same as PHONE>	
HOST	<same as CPU>	2.3c
IP	<same as CID>	2.3c
KBLEFT	User's total credits (in kilobytes)	
KBLIMIT	User's free credits per day (in kilobytes)	

LAST	User's last name (alias)
LASTDATEON	Date of user's last logon (MM/DD/YY)
LASTNEW	Date of user's last new file scan (MM/DD/YY)
LASTON	Date and time of user's last logon
LASTTIMEON	Time of user's last logon (HH:MM am)
LASTREAL	User's last real/company name
LEFT	<same as MINLEFT>
MAILW	Number of mail messages waiting for current user
MAILW:x	Number of mail messages waiting for current user #x
MAILP	Number of pending mail messages sent by current user
MAILP:x	Number of pending mail messages sent by current user #x
MAXDK	<same as KBLIMIT>
MEMO	Date of user's last password modification
MEMO1	User's note
MEMO2	<same as COMPANY>
MINLEFT	User's time left in minutes
MSGLEFT	Total number of messages posted by user
MSGREAD	Number of messages read by user this call
MSGSLLEFT	<same as MSGLEFT>
NAME	User's name or alias
NAME-L[...]	User's name (padded and left justified)
NAME-R[...]	User's name (padded and right justified)
NEWFILETIME	Date and time of user's last new file scan
NUMTIMESON	<same as CALLS>
PHONE	User's phone number (###-###-####)
REAL	User's real first name
SEC	User's security level
SECURITY	<same as SEC>
SINCE	Date of user's first call (MM/DD/YY)
STATE	User's state (from location)
TIMELEFT	<same as MINLEFT>
TIMELIMIT	Maximum time per call in minutes
TIMEON	Time used this call in minutes
TIMEUSED	<same as TIMEON>
TLEFT	Time left (H:MM:SS)
TPERC	Time allowed per call (H:MM:SS)
TPERD	Time allowed per day (H:MM:SS)
TUSED	Time used this call (H:MM:SS)
UPBYTES	Total bytes uploaded by user
UPFILES	Total files uploaded by user
UPK	Total kilobytes uploaded by user
UPS	<same as UPFILES>
USER	User's name or alias (same as ALIAS)
ZIP	User's zip/postal code

Display

Code	Description	Ver
-----		
AUTOMORE	Toggle automatic pausing	
BEEP	Generate a beep	
BELL	<same as BEEP>	
CLS	Clear screen	
CRLF	Carriage return/line-feed pair	3.0b
MENU:filename	Display a menu file (from TEXT\MENU directory)	
MSGREPLY	Command key to reply to last message	3.0b
MSGREREAD	Command key used to re-read last message	3.0b
NOPAUSE	<Same as POFF>	
MORE	<same as PAUSE>	
PAUSE	Immediately produces a [Hit a key] prompt	
PON	Toggles automatic screen pause for everyone	
POFF	Toggles automatic screen pause for everyone	
TYPE:filename	Display a specific filename	
UP	Move cursor up one row (ANSI)	3.0b
UP:n	Move cursor up n rows (ANSI)	3.0b
DOWN	Move cursor down one row (ANSI)	3.0b
DOWN:n	Move cursor down n rows (ANSI)	3.0b
RIGHT	Move cursor right one column (ANSI)	3.0b
RIGHT:n	Move cursor right n columns (ANSI)	3.0b
LEFT	Move cursor left one column (ANSI)	3.0b
LEFT:n	Move cursor left n columns (ANSI)	3.0b
GOTOXY:x,y	Move cursor to x/y (1-based) (ANSI)	3.0b
PUSHXY	Save current cursor position (ANSI)	3.0b
POPHY	Restore saved cursor position (ANSI)	3.0b

Miscellaneous

There a few special Synchronet specific @-Codes which require a parameter (following the colon and before the terminating @ symbol):

Code	Description
-----	
HANGUP	Immediately disconnect user
SETSTR:STR	Sets the current Baja command string to STR
EXEC:MODNAME	Execute a loadable (Baja) module, EXEC\MODNAME.BIN
TYPE:FILENAME	Display a specific filename (must specify path and file ext.)

MENU:FILENAME     Display a menu file (from TEXT\MENU with automatic file ext.)

Synchronet command line specifiers may be used in the FILENAME parameter to the TYPE: @-Code allowing symbolic replacment for specific Synchronet directories (%!, %z, %k, %j, etc).

Examples:

@EXEC:MYMOD@  
@TYPE:%zSYSTEM.MSG@  
@MENU:YESNO@

[17.3] - Customization: Message Color Codes

Synchronet supports six different Color Code formats. When Synchronet encounters one of these Color Codes in a message, it changes the text following the Color Code to the specified color. Support of the formats which are not native to Synchronet (WWIV, Celerity, Renegade, PCBoard, and Wildcat) can be toggled on and off from the Synchronet Configuration utility (System->Message Options->Extra Attribute Codes). The non-Synchronet color codes (Extra Attribute Codes) only affect the text that is displayed on the SAME LINE. When using Synchronet color codes, the new color is retained from line to line until another color code is processed or the end of the text is reached.

Synchronet Format

The native Synchronet Color Code format (preferred) consists of a Control-A followed by a singe character. The following is a list of valid Control-A Color Codes:

Foreground Background	
Black	K 0
Red	R 1
Green	G 2
Yellow	Y 3
Blue	B 4
Magenta	M 5
Cyan	C 6
White	W 7

Attribute Description	
High	H High Intensity
Blink	I Blinking
Normal	N No Special Attributes (Normal)
Pause	P Insert a Pause Prompt into message
Pause Reset	Q Reset the line counter for the auto screen-pause
Delay	, Insert a Tenth Second Delay into message
Delay	; Insert a Half Second Delay into message
Delay	. Insert a Two Second Delay into message
Date	D Display the system date
Time	T Display the system time
Cls	L Insert a Form Feed (Ctrl-L, Clear Screen) into message
Clreol	> Clear to End of Line (leave cursor in current position)
Bckspc	< Non-destructive backspace (Ctrl-H)
CR	[ Carriage return (Ctrl-M)
LF	] Line feed (Ctrl-J)
Ctrl-A	A Send an actual Ctrl-A character
Sync	S Synchronize output with remote system
EOF	Z End of displayable text in this file

Normal - Same as 'N' but only sends ANSI codes if the (minus) High Intensity, Blinking, or Background attribute is set.

Normal \_ Same as 'N' but only sends ANSI codes if the (underscore) Blinking or Background attribute is set.

Synchronet also supports Special Control-A codes used to hide text from users not meeting certain criteria (i.e.: Security Level or Flags from Flag Set #1). The following is a list of Special Control-A codes, and a brief description of each code's usage:

Code Description	
^A thru ^Z Only display the following text to users with the corresponding flag A through Z turned on (from Flag Set #1).	
!	Toggle the text display off/on for users of less than level 10.
@	" 20.
#	" 30.
\$	" 40.

```
% " " 50.
^ " " 60.
& " " 70.
* " " 80.
( " " 90.
) Restore the displaying of text to ALL users.

"<filename>      Display contents of <filename> (from your TEXT directory)
```

High Bit (greater than ASCII 127) Used for cursor right positioning.

### WWIV Format

Synchronet also supports Color Codes which are native to WWIV BBS software. These codes consist of a Control-C followed by a number (0 through 7):

Code	Color
0	Normal
1	High Intensity Cyan
2	High Intensity Yellow
3	Normal Magenta
4	High Intensity White with Blue Background
5	Normal Green
6	High Intensity Blinking Red
7	High Intensity Blue
8	Low Intensity Blue
9	Low Intensity Cyan

### Celerity Format

Synchronet also supports Color Codes which are native to Celerity BBS software. These codes consist of a pipe symbol '|' followed by a letter (case sensitive):

Code	Color (foreground)
k	Normal Black
b	Normal Blue
g	Normal Green
c	Normal Cyan
r	Normal Red
m	Normal Magenta
y	Brown
w	Normal White
d	High Intensity Black
B	High Intensity Blue
G	High Intensity Green
C	High Intensity Cyan
R	High Intensity Red
M	High Intensity Magenta
Y	Yellow
W	High Intensity White
S	* Swap foreground and background

Example: "|b|S|W" would set the current color to high intensity white on a blue background.

Note: Due to conflicting escape sequences (namely, the pipe character), Celerity color codes are not supported when using RIP terminal mode.

### Renegade Format

Synchronet also supports Color Codes which are native to Renegade BBS software. These codes consist of a pipe symbol '|' followed by a number (0-23):

Code	Color
0	Normal Black
1	Normal Blue
2	Normal Green
3	Normal Cyan
4	Normal Red
5	Normal Magenta
6	Brown
7	Normal White
8	High Intensity Black
9	High Intensity Blue
10	High Intensity Green
11	High Intensity Cyan
12	High Intensity Red
13	High Intensity Magenta
14	Yellow
15	High Intensity White
16	Background Black



- 17 Background Blue
- 18 Background Green
- 19 Background Cyan
- 20 Background Red
- 21 Background Magenta
- 22 Background Brown
- 23 Background White

Example: "|15|17" would set the current color to high intensity white on a blue background.

Note: Due to conflicting escape sequences (namely, the pipe character), Renegade color codes are not supported when using RIP terminal mode.

PCBoard/Wildcat Format

Two of the Color Code formats which Synchronet supports (PCBoard and Wildcat) use similar, yet cryptic, methods of displaying colors. The PCBoard method uses the format "@X<Background><Foreground>", and Wildcat uses the format "@<Background><Foreground>@". The following is a list of the Background and Foreground choices available:

<Background>	Color	Attribute	<Foreground>	Color	Attribute
0	Black	Normal	0	Black	Normal
1	Blue	"	1	Blue	"
2	Green	"	2	Green	"
3	Cyan	"	3	Cyan	"
4	Red	"	4	Red	"
5	Magenta	"	5	Magenta	"
6	Brown	"	6	Brown	"
7	White	"	7	White	"
8	Black	Blinks Foreground	8	Black	High Intensity
9	Blue	"	9	Blue	"
A	Green	"	A	Green	"
B	Cyan	"	B	Cyan	"
C	Red	"	C	Red	"
D	Magenta	"	D	Magenta	"
E	Brown	"	E	Yellow	"
F	White	"	F	White	"

Example: "@1F@" in Wildcat format and "@X1F" in PCBoard format would set the current color to high intensity white on a blue background.

[17.4] - SIF Questionnaire File

An automatic new user SIF questionnaire can be specified in SCFG->System. If a newuser SIF is specified, all users who logon and don't have a copy of the answered questionnaire data in their user file will be given the questionnaire upon logon. The sysop can view the answered questionnaire from User Edit with the '#' command. For convenience, the sysop can create a second (abbreviated) SIF file for his own use in viewing user's answers. The two SIF files (input and output) should be identical with the exception of what is in the 'text' portion.

format:

<STX>text<ETX>mode[mod] [l] [r] [x] [.n] ["str"]

element descriptions:

- STX is the ASCII code for start of text (ASCII 2 / Ctrl-B)
- ETX is the ASCII code for end of text (ASCII 3 / Ctrl-C)
- text is any number of ASCII characters - Synchronet Ctrl-A codes supported
- mode text input mode desired for this field. Possible mode values are:
  - c single character
  - s string of characters
- mod optional mode modifier. Possible mode modifiers are:
  - n numeric characters only
  - u input converted to uppercase
  - f forced word capitalization ('s' mode only)
- l input line will be displayed (inverse bar of maximum input length)
- r a carriage return / line feed pair will be appended to this field in the data buffer. Only use this field if you want the data buffer or file to be more readable. All data is on one line otherwise.
- x maximum string length allowed (required for non-template 's' mode)

**n**            minimum string length allowed (only applicable with 's' input mode)

- str**    1: in 's' modes, a template string that defines what will be displayed at the prompt and what type of characters the user can input. All characters other than 'N', 'A' or '!' are printed at the prompt. Occurances of 'N', 'A' or '!' define which type of character the user can input for each character position. 'N' allows the user only to enter a numeric character, 'A' allows only alphabetic, and '!' allows any character. Popular templates are "NNN-NNN-NNNN" for phone number input or "NN/NN/NN" for date input.
- 2: in 'c' modes, a string that defines which characters the user will be allowed to input (not case sensitive), usually used for multiple choice answers. Most common allowed characters are "ABCD..." or "1234...". If this string is specified in 'c' input mode, 'u' and 'n' have no effect and input will be converted to uppercase automatically.

**Example 1:**

```
<BOT>
Enter string: <EOT>sulr8.3
```

Prints the prompt, "Enter string: ", then a line of 8 blue spaces (an input bar, if you like), would convert all of user's input to uppercase, allow the user to input a maximum of eight characters, a minimum of three and append a CRLF onto the end of the data field.

**Example 2:**

```
<BOT>
A> First Answer
B> Second Answer
C> Third Answer

Which: <EOT>c"ABC"
```

Prints "A> First Answer" "B> Second Answer" etc... then allows the user to input one character, either A,B, or C. No other characters will be accepted as input.

**Example 3:**

```
<BOT>
Enter phone number: <EOT>s"NNN-NNN-NNNN"
```

Prints "Enter phone number: ", then allows the user to input only numbers in the 'N' character positions, and automatically skips over the '-' characters.

See **EXAMPLE.SIF** in the **SBBS\TEXT** directory for more information.

# [17.5] - GURU.DAT

The Synchronet Guru is an artificial intelligence engine that users can chat with for entertainment or educational purposes. You may wish to fool the users into believing the guru is a live human, or tell them up-front that it's just a program.

You may have up to 500 different guru available to chat with on your BBS, each with its own "personality" and "intelligence". You add additional gurus in SCFG->Chat Options->Artificial Gurus. You can specify access requirements (see the ARS chapter for details) for each guru, allowing you to have specific gurus for different groups of users or allow users to choose which guru they want to chat with.

The gurus of your BBS can be "taught" to respond to keywords and phrasing that your users use when chatting with him. The default guru's "brain" is a file named GURU.DAT that is kept in the CTRL directory. It is a special data file that contains logic expressions and lists of responses. Before you edit the GURU.DAT file, be sure you understand exactly what you are doing, as the neuro system that interprets the GURU.DAT file does not handle syntax errors very well. The basic structure of the GURU.DAT is as follows:

**(expression)**  
**response**  
**response**  
**(expression)**  
**response**  
**response**  
**()**  
**response**

**response**  
**response**

You may include as many Expression/Response sets as you like, as long as the file size does not exceed 64k or the amount of available memory. Each expression contains one or more string of characters that The Guru may respond to and logic operators. The string must be in all uppercase and may not contain the following characters: ~^|&()

If the expression just contains one string (e.g. (HELLO)) and that string is used in the users input, The Guru will pick a random response from the list that follows that expression. The Guru will only use one response for each line input by the user, so as soon as a "true" expression is encountered, a response is made and the evaluation of the user's line is complete.

All expressions are evaluated from the top of the file down, so if a true expression is encountered toward the top of the file, all of the remaining expressions are ignored until the next evaluation. Notice that the last set of responses is preceded by a pair of empty parenthesis. This is an "always true" expression and should always be the last expression in the file. Omitting this fall-through expression is a syntax error. If all the previous expressions are evaluated as false, then a response will be picked from the set following the fall-through expression.

The simplest form of an expression is just a string of uppercase letters (with or without spaces). If the string is followed by a tilde '~', the string will be evaluated as true even if the string is embedded in another string (e.g. if the user types "XhelloX", an expression of (HELLO) would evaluate as false, but an expression of (HELLO~) would evaluate as true).

You can also specify that the string must be the beginning of the users input line by following the string with a caret '^'symbol (e.g. if the user types "I said, Hello!", an expression of (HELLO) would evaluate to true, but an expression of (HELLO^) would be false).

An expression can contain multiple strings connected with logic symbols. The valid logic symbols are & (and) and | (or) (e.g. if you have the expression (HELLO&GURU) the user must type both "hello" and "guru" in the input line in order for the expression to be true. If you have the expression (HELLO|HI), it will be evaluated as true if the users includes either "hello" or "hi" in his input string). Nested evaluations are supported (e.g. the expression, (GURU&(HELLO|HI)) will evaluate as true if the user inputs either "guru" and "hello", or "guru" and "hi").

Expressions may also contain AR strings within square brackets ([ and ] ). The expression (HELLO&GURU&[LEVEL 20]) would evaluate as TRUE only if the user typed the words "HELLO" and "GURU" and had a level of 20 or higher. See the ARS chapter for details on the AR string syntax and possible keywords.

**RESPONSES:**

Each expression can be followed by up to 100 responses and each response can be up to 512 bytes long. Responses can not contain the characters ( or ) and may only span several lines if the last character of each continued line is a back-slash '\'. Responses are picked at random from the group below the first expression that is evaluated as true. The more responses you have to each expression, the less likely The Guru is to repeat himself. The Guru can also respond with information about the current user or perform an action. To initiate these special responses, you must precede a valid response variable with a back-quote (`) character. The valid response variables and their definitions are as follows:

- A User's alias (name, if Aliases not allowed)
- B User's birth date
- C User's computer type
- D User's download bytes
- G Guru's name
- H Hang up on the user (immediately)
- I System's QWK ID
- J Current day of the month
- L User's security level
- M Current month
- N User's note (location, if Aliases not allowed)
- O Sysop's name
- P User's phone number
- Q Quit chat
- R User's real name (address, if Aliases not allowed)
- S System name
- T Current time
- U User's upload bytes
- W Current day of the week
- Y Current year
- Z User's zip/postal code
- \$ User's credits
- # User's age

! Toggle The Guru's typing mistakes Off/On  
\_ Pause in response

Three of the above response variables only have effect when the user is chatting with The Guru in the "Local" mode and not from multinode chat. These are the 'Q'uit chat (which is the only means of the exiting without hitting Alt-G locally), '!' Toggle typing mistakes, and '\_' pause in response.

e.g. The expression/response pair:

(HELLO)  
Hello there, `a...

would display "Hello there, Joey..." if Joey were to say "hello" to The Guru.

See **CTRL\GURU.DAT** for more detailed examples of GURU.DAT programming.

[Back to Top](#)

---

**Copyright © 2000 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .

# Synchronet BBS

## Multinode Bulletin Board System Software

---

[Back to Table of Contents](#)

## [18.1] - Appendix A: Command Line Specifiers

Specifiers (argument replacement) for use with command lines in SCFG and many string-related Baja functions.

**%a** User alias or name (may require surrounding with "'s)  
**%b** Baud (DTE) rate  
**%c** Connect description  
**%d** Connect (DCE) rate  
**%e** Estimated CPS rate\*10  
**%f** File path/name (example: C:\SBBS\NODE1\DOOR.SYS)  
**%g** Temp directory (example: C:\SBBS\NODE1\TEMP\  
**%h** HardWare flow control (Y/N)  
**%i** COM port IRQ line (if UART) or channel number (if Int 14h driver)  
**%j** Data directory (example: C:\SBBS\DATA\  
**%k** Control directory (example: C:\SBBS\CTRL\  
**%l** Maximum lines per message  
**%m** Minutes in the user's minute bank  
**%n** Node directory (example: C:\SBBS\NODE1\  
**%o** Sysop's name (as specified in SCFG)  
**%p** COM port number (0 if local node)  
**%q** System's QWK ID (as specified in SCFG)  
**%r** Rows on user's terminal screen  
**%s** File specification (example: C:\FILES\\*.ZIP) or current command string  
**%t** Time (in seconds) user has left online  
**%u** UART I/O address in hex or F=FOSSIL, B=PC BIOS, E=PS/2, D=DigiBoard  
**%v** Synchronet version (example: 200a is Version 2.00 Revision a)  
**%w** Timeslice API types supported (decimal bit field)  
**%x** User's command shell (internal code)  
**%y** COMSPEC environment variable or DOS Command Interpreter (if SBBS4OS2)  
**%z** Text directory (example: C:\SBBS\TEXT\  
**%!** EXEC directory (example: C:\SBBS\EXEC\  
**%#** Node number (same as SBBSNNUM environment variable)  
**%\*** Node number (0 padded to 3 digits)  
**\$\$** User's credits  
**%%** Percent symbol  
**%&** Used for DCDWATCH utility  
**%?** Platform descriptions (e.g. Win32 or Linux) **(v3+ Only)**  
**%1** User number  
**%2** User number (0 padded to 2 digits)  
**%3** User number (0 padded to 3 digits)  
**%4** etc...

## [18.3] - Appendix C: SBBS Environment Variables

During operation, Synchronet may write to several different environment variables which can be utilized by other programs. Some of these environment variables are also used by the utility programs that are included with Synchronet as well as third party programs written for Synchronet. Following is a list of environment variables and a brief explanation of them:

**SBBSCTRL** This variable contains the path of the Synchronet CTRL directory. This variable is NOT set by (and not required to run) SBBS, it must be set manually prior to running any programs which require it (the Synchronet NODE utility, for example).

**SBBSNODE** This environment variable contains the path of the node directory of the current node. This variable is set by SBBS at the time it is executed. Some off-line utilities may require this environment variable be set during boot-up (AUTOEXEC.BAT), in which case it should point to your NODE1 directory. SBBS.EXE will automatically reset this variable to point to the current node directory, for any online programs it may execute.

**SBBSNNUM** This variable contains the NUMBER of the active node. This variable is set by SBBS at the time it is executed.

**SBBSFILENAME** This variable contains the NAME of the file currently being processed (up to 12 characters including '.', not padded with spaces). This variable is set by Synchronet during 'Testable

Filetype' events. The same information is written to the file SBBSFILE.NAM in the current node directory.

NOTE: Upon returning from a 'Testable Filetype' event, Synchronet will read in information contained in the file SBBSFILE.NAM located in the current node directory to determine if the filename has changed during the testing process.

**SBBSFILEDESC** This variable contains the ONE LINE DESCRIPTION of the file currently being processed (up to 58 characters). This variable is set by Synchronet during 'Testable Filetype' events. The same information is written to the file SBBSFILE.DES in the current node directory.

NOTE: Upon returning from a 'Testable Filetype' event, Synchronet will read in information contained in the file SBBSFILE.DES located in the current node directory to determine if the file description has changed during the testing process.

**DSZLOG** This variable is automatically set by Synchronet to point to the file PROTOCOL.LOG in the current node directory. If a transfer protocol has been configured (in SCFG) to support DSZLOG, then this log file will be read in by Synchronet after the transfer is complete, to determine which files were successfully transferred and which files were not.

## [18.5] - Appendix E: Node Status Display

Anywhere within Synchronet's Telnet/RLogin interface, you can list the users currently online with the Ctrl-U command.

From the Main, Transfer, Multinode Chat, or Private Chat sections you can list the current status of all the nodes with the '/L' command.

You can display the status of all nodes from the DOS command line (or batch file) with the NODE utility.

### Node Status Line:

The format of each node status line is as follows:

Node n: username action connection (flags) [sysflags]

Where:

n = node number  
username = user's name  
action = what the user is currently doing  
connection = how the user is connected (bps rate if remote)  
flags = letters representing special modes of the node:  
A - Activity Alert Disabled:  
User of this node will not be notified of other users logging on or off other nodes.  
L - Node is locked for sysop use:  
Node is temporarily reserved for sysop use only.  
M - Message waiting for node:  
An unreceived node-to-node message is waiting for this node.  
P - Page disabled:  
User of this node does not wish to allow users of other nodes to page him for chat or send him node-to-node messages.

sysflags = modes that are sysop settable/viewable only:

A - Anonymous:  
User on this node is in anonymous mode.  
D - Down:  
This node will be shutdown as soon as possible.  
E - Event:  
This node will run it's daily event (if one is specified) before waiting for another call.  
I - Node will be interrupted:  
Node connection will be terminated as soon as possible.  
Q - Quiet:  
User on this node is in quiet mode.  
R - Rerun:  
This node will Rerun when user logs off.  
U - User data update:  
This user's data has been modified by another node and needs to be read from



disk.

If the node is not in use, the possible status are:  
Offline: Node is not currently running Synchronet  
Waiting for Call: Node is waiting for a call.  
Networking: Node is currently executing network functions.  
New User Logging On: A user is logging on as new.  
Waiting for all nodes to become inactive.  
Running external event.  
Waiting for node # to finish external event.

## [18.6] - Appendix F: Log File Line Type Specifiers

Each line in the daily caller log has a line type specifier (the first 2 characters). The specifiers and their descriptions follow:

!! Critical error  
!\* User auto-deleted  
!% User expired  
!= New day for statistics  
!: Ran event  
@ Modem connection established  
@- Modem connection terminated  
@! Modem error  
@\* Caller-ID information  
@R RIP support detected  
@W WIP support detected  
++ Logon  
+! Logon error  
N New user begins application  
N+ New user finishes application  
N! New user fails application  
S+ Sysop enabling function  
S- Sysop disabling function  
S! Sysop security error  
P+ Posted message  
P- Removed message  
P! Posting error  
E Read E-mail  
E- Deleted E-mail  
E+ Sent E-mail  
EN Sent NetMail  
E! E-mail error  
C Chat function  
U+ Upload  
U- Removed Upload  
U! Attempted upload  
D- Download  
D! Attempted download  
X- External program execution  
T- Read text file  
M+ Moved message  
Q! QWK error  
L! Log already exists (possible crash)  
\$+ Gained credits  
\$- Lost credits  
\*+ Gained minutes  
\*- Lost minutes

To view today's caller log, hit 'L' from the WFC screen or ";LOG" from the main menu. Yesterday's caller log can be viewed by hitting 'Y' at the WFC screen, or ";YLOG" from the main menu. The log files are stored in the DATA\LOGS directory with a file for each day named in the format: MMDDYY.LOG.

## [18.7] - Appendix G: File Formats

XTRN.DAT

This is the drop file that Synchronet creates for Synchronet specific external programs. It is an ASCII text file with the format as follows:

Sample Data	Description
~~~~~	~~~~~
Digital Man	User name
Vertrauen	System name
digital man	System operator
The Guru	System guru
..\CTRL\	CTRL directory
..\DATA\	DATA directory
5	Total nodes on system
5	Current node number
19448	Time left online (in seconds)

Yes  
24  
10770335  
99      Security Level  
         -unused-  
12/31/69  
M      Sex (Gender)  
1  
714-529-9525  
0  
3      COM port IRQ (or channel number)  
2f8      COM port I/O address (in hex)  
2400  
No  
No  
ATQ0V0E0M1X4&C1&D2H0  
  
ATV1E1X4  
ATDT  
ATH1M0  
ATA  
795154132  
11  
Global War  
Trade Wars 2002  
Food Fight!  
Pit Fiend ] [  
Dice War  
Synchronet Blackjack!  
Domain Poker  
Phantasia  
Emperor  
Synchronet Upgrade Door  
Credit Card Order Door  
AB D F J      User's Flags #1  
  BC    G JK N      User's Flags #2  
A            L            T  
          D  
2b43cfd0  
PO Box 501      User's Address  
Yorba Linda, Ca      User's Location  
92686      User's Zip/Postal Code  
A    E      User's Flags #3  
  B   EFG      User's Flags #4  
1      Timeslice API types  
Rob Swindell      User's real name or company name  
14400      User's DCE rate  
..\EXEC\      EXEC directory (BBS executable files)  
..\TEXT\      TEXT directory (TEXT files)  
TEMP\      TEMP directory  
VERT      System's QWK ID  
22C8      Node toggle options (in hex)

ANSI ? (Yes/No/Mono)  
Lines per screen  
Credits  
  
Birthdate  
  
User number  
User phone number  
COM port (0 if no modem or local)  
[1]  
[2]  
COM port DTE rate  
Modem uses hardware flow ctrl (Y/N)  
Modem locked at DTE rate (Y/N)  
Modem initialization string  
Modem special init string  
Modem terminal mode init string  
Modem dial prefix  
Modem off-hook string  
Modem answer string  
Address of Modem Status Register  
Number of External Programs  
Names of External Programs (or blank  
                                 if user doesn't have access)  
  
User's Exemptions  
User's Restrictions  
Expiration Date (Unix format in hex)

~~~~~

[1]: IRQ line if UART COM port type or Channel (zero based) if Int 14h type  
[2]: I/O base address if UART COM port type or:

F = FOSSIL  
B = PC BIOS  
E = PS/2 BIOS  
D = DigiBoard

MODUSER.DAT

This is an optional file created by external programs to modify the data of the current user. It is an ASCII text file with the format:

| Sample Data                                                                                        | Description                          |
|----------------------------------------------------------------------------------------------------|--------------------------------------|
| ~~~~~                                                                                              | ~~~~~                                |
| -500                                                                                               | Credit Adjustment                    |
| 60      Security Level<br>-unused-                                                                 |                                      |
| AB E I      Flags #1 to add<br>G      Flags #2 to add                                              |                                      |
| A            P    T    Exemptions to add<br>Restrictions to add                                    |                                      |
| 2b43cfd0                                                                                           | Expiration Date (Unix format in hex) |
| 0      Add these number of minutes to bank<br>XYZ    Flags #3 to add<br>TUV    Flags #4 to add     |                                      |
| J      Flags #1 to remove<br>L      Flags #2 to remove<br>Flags #3 to remove<br>Flags #4 to remove |                                      |

D       Exemptions to remove  
A       Restrictions to remove

      If any of the lines in the file are blank, that user item is not  
modified. The security level cannot exceed 89.

NOTE: Technical specifications and C source code for Synchronet configuration  
and data files can be downloaded from Vertrauen.

[Back to Top](#)

---

**[Copyright](#) © 2006 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation,  
library, executable, and [source code](#) files, all of which are covered by the\_  
[GNU General Public License](#) with the exception of the following portions covered by  
the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the\_  
[Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today.  
Synchronet Version 3 is not Public Domain software.

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Document](#) .

# Synchronet BBS

## Multinode Bulletin Board System Software

---

[Back to Table of Contents](#)

## [19.0] - Glossary

### Alias:

Nickname that a user is known as. The use of aliases is common on BBSes. Synchronet allows the sysop to disallow the use of them, forcing all users to be known by their real names only, if desired.

### ANSI:

American National Standards Institute. References made to ANSI are actually referring to the ANSI X3.64 terminal definition and the related escape sequences used to change cursor positioning and text attributes.

### Archive:

A file that contains multiple (possibly compressed) files, that can be expanded. Archived files usually have a specific extension that specifies what type of archive utility was used to create it.

The most popular archive utilities are PKZIP (.ZIP files), LHARC (.LZH), and ARJ (.ARJ files).

### ASCII:

American Standard Code of Information Interchange. A 7 bit binary code used to represent letters, numbers, symbols, and control codes. Supported by almost every computer and terminal manufacturer. See Extended ASCII.

### Baja:

Baja (pronounced bä'hä) is a tool used to create command shells and modules (exec/\*.bin files) for the Telnet/RLogin experience of Synchronet BBS software. Baja source files have a .src filename suffix.

Baja is also the name of the programming language used to create or modify the contents of Baja source (.src) files.

See [baja.html](#) for more details.

### Baud:

An obsolete term defining the number of signal-level changes per second. The CCITT now prefers to use "symbols per second". For example a V.22bis connection transfers only 600 symbols per second, but in this protocol each symbol can represent up to four values; thus, you have an effective transmission rate of 2,400 bps. V.32 is a 2,400 symbol-per-second (9,600-bps) modulation protocol.

### BBS:

Bulletin Board System. A system configured to accept user logins via modem or network to access public and private messages. Many BBSes also incorporate file transfers (the uploading and downloading of program and data files), information services, online entertainment, and more. On a multinode BBS, all nodes must use the same live database for users, messages, and file transfers. Plural: BBSes.

### BinkD:

BinkD (Binkley Daemon) is a FLO-style FidoNet mailer that uses the BinkP protocol (FSP-1011) to transfer files over the Internet.

### BinkP:

BinkP (BinkD Protocol) is a TCP/IP protocol (described in FSP-1018) for transferring files between FidoNet mailers over the Internet (on TCP port 24554). This protocol was created for BinkD and is also supported by such mailers as Argus, Radius, Taurus, BForce, Internet Rex, and BeeMail.

### BPS:

Bits Per Second or Bit Rate. The rate of data transmitted between modems. For every byte of data (8 bits), a start and stop bit are added for a total of ten bits per data byte.

### Chat:

Online real-time communication between users either in line by line (multinode chat) or key by key (private chat). Also IRC.

### COM Port:

RS-232 communications port on an IBM PC compatible computer through which digital signals are exchanged between it and the modem (or other

peripheral). The interface is either a 25 or 9 pin male connector.

**Command Line:**

The complete syntax used for the execution of a program. This includes the program path and name to execute and any parameters that may be required by the program for proper execution. Command lines configured in SCFG can use special command line specifiers for variable parameter replacement. See [Appendix A](#) for more information.

**Compression:**

See Archive.

**Conference Mail:**

See EchoMail.

**Co-sysop:**

BBS user with additional privileges to enable partial system maintenance. Co-sysops on a Synchronet system would usually have a security level in the range 80-89 and have an exemption flag for each sysop function he is given rights to.

**CR:**

Carriage Return. This character represents the end of a line of text and is usually initiated with the ENTER key on most keyboards.

**CrashMail:**

Referring to FidoNet NetMail being sent immediately and directly (not routed) to the BBS or network address of the destination person. A more accurate term would be Crash NetMail, or NetMail with Crash status.

**Decompression:**

See Extraction.

**Directory:**

A section within a file library that contains files for uploading or downloading. Also known as a file area.

**Download:**

Transferring a file from a BBS or other host system to a remote (client) system.

**Doors:**

See External Programs.

**DCE:**

Data Communications Equipment. Dial-up modems that establish and control the data link via the telephone network.

**DCE Rate:**

The data transfer rate between two modems.

**DTE:**

Data Terminal Equipment. The device that generates or is the final destination of data - the computer.

**DTE Rate:**

The data transfer rate between the computer and the modem.

**Echo:**

The term "Echo" or "Echo Conference" is often used to refer to a sub-board where messages are distributed across a message network. The term actually comes from FidoNet EchoMail - the technology used to distribute sub-board messages across FidoNet.

**EchoMail (Conference Mail):**

Sub-board messages echoed across FidoNet in compressed packets.

**EchoMail Program ("Tosser"):**

EchoMail programs packetize FidoNet EchoMail (and sometimes NetMail) messages and archive the packets into bundles which are transferred with a FidoNet mailer. SBBSecho, TosScan, Squish, GEcho, and FreeMail are examples of EchoMail programs.

**E-mail or Electronic Mail:**

Private multiple line messages between users that are stored on a BBS until the receiver deletes them.

Network E-mail (NetMail) is e-mail that is sent between systems on a network.

Internet E-mail is a form of NetMail that is sent to the recipient's Internet Mail Server using the SMTP protocol.

**Escape Sequence:**

A sequence of characters usually preceded by a control code to perform attribute changes and cursor positioning on a terminal.  
See ANSI.

**Exemptions:**

Extended privileges given to users to circumvent access limitations or provide access to certain sysop functions.  
See User Edit for more information.

**External Programs:**

Programs (binary executables or scripts) that the BBS executes for added functionality. External programs are used for archive manipulation, file transfers, games, databases, text editors, virus scanning, backups, and more. Often referred to as "doors" or "chains".

**Extraction:**

The splitting (and possible decompression) of an archived file into the original set of multiple files.  
See Archive.

**FidoNet:**

Long-standing global message network for BBS operators (sysops) and users.

**File Transfer Protocol:**

See Transfer Protocol.

**Finger:**

Internet service that provides information about the users on a particular computer (on TCP port 79).

**Flag:**

Security Flag: one of 26 possible switches labeled A through Z. Flags are used to represent specific sysop-defined security access (privileges or restrictions) for a user. See User Edit for more information.

**FOSSIL:**

Fido/Opus/Seadog Serial Interface Layer. A specification for x86-compatible computers to use an x86 interrupt (14h) interface for serial communications.

A FOSSIL driver is a device driver that provides an int14h interface for applications (typically, 16-bit DOS doors) to use to communicate with the remote user.

Synchronet for Windows includes FOSSIL drivers for Win9x and NT-based operating systems (no extra configuration required).

Traditional DOS FOSSIL drivers (e.g. BNU, ADF) are not needed when using Synchronet for Windows.

**Front-end Mailer:**

An EMSI-compatible FidoNet message front-end. FrontDoor, SEAdog, Binkley, and D'bridge are examples of front-end mailers.

Modern TCP/IP FidoNet mailers (e.g. BinkD, Argus, Internet Rex) run concurrently with the BBS, so they are not *ront-end* mailers.

**FTP:**

A TCP/IP protocol for accessing and transferring files between systems (on TCP port 21).

**Gopher:**

An Internet server document browsing and searching system, a precursor to HTTP, that lets you search and retrieve texts on the Internet (on TCP port 70).

**Group or Message Group:**

A group of message sub-boards with a similar subject matter.

**Hardware Flow Control:**

The modem's use of the CTS (Clear to Send) line to control the flow of data to from the computer to the modem.

**HTTP:**

Hyper Text Transfer Protocol. HTTP is the actual TCP-based protocol that enables Web browsing (on TCP port 80).

**Internet:**

A network of computer networks which operates world-wide using a common set of communications protocols.

**IP:**

Internet Protocol. The lowest-level protocol in the Internet TCP/IP protocol suite.

**IP Address:**

Internet Protocol Address. An IP Address is a 32-bit (4 byte) number, represented in dotted-decimal notation (e.g. 123.45.67.89).



**IRC:**

Internet Relay Chat. A TCP/IP protocol used for distributed, multi-server, multi-channel communication ("chat") between Internet clients.

**LAN or Local Area Network:**

A group of personal computers connected in a local environment for the purpose of sharing data, applications, and peripherals.

**Level or Security Level:**

A decimal value in the range of 0 to 99 that determines a user's security level on Synchronet BBS. A user's level determines how long he can stay online per call, total time per day, total logons a day, maximum number of lines per message, which Message Groups, Sub-boards, External Programs, General Text File Sections, Transfer Libraries, and Directories the user can access.

**Library or Lib:**

A group of transfer directories with a similar subject matter.

**Login:**

The act of connecting to a service and exchanging user credentials (e.g. user name and password) to obtain access.

**Logon:**

The act of entering a BBS system through a valid user account. In Synchronet, a *logon* follows a *login* (the initial connection and exchange of user credentials).

**Message:**

File stored on the system created by a user that may contain ASCII text, color/attribute codes (e.g. Ctrl-A codes), and ANSI escape sequences. Messages are either public (posted on a sub-board) or private (E-mail sent to a single user).

**Message Network:**

Two or more BBSes sharing public message sub-boards (Aka Echoes or conferences) where messages posted on one BBS get distributed to the other BBSes on the network. There are many different network technologies used for the distribution of networked messages. Synchronet supports the most popular technologies; QWK, FidoNet, and PostLink.

**Multinode:**

System that operates with multiple simultaneous access paths to the same database of messages and other resources.

**Multitask:**

The act of performing multiple tasks seemingly simultaneously.

**Modem:**

A device that transmits/receives computer data through a communications channel such as radio or telephone lines. Modems modulate, or transform, digital signals from a computer into an analog form that can be carried successfully on a phone line. Modems also demodulate signals received from the phone link back to digital signals before passing them to the receiving computer.

**NetMail:**

A personal message sent to a specific person on a specific BBS or at a specific network address through a message network. Most commonly used in reference to FidoNet NetMail. QWK Netmail and Internet E-mail are other forms of NetMail.

**Network:**

Connection of two or more computers to facilitate the sharing of resources. See LAN and Message Network.

**NNTP:**

The protocol used by client and server software to carry USENET postings back and forth over a TCP/IP network (e.g. the Internet).

**NUP or New User Password:**

A password that the sysop has determined as a requirement before a new user can apply for access.

**Online:**

The state of a user when he is currently using a BBS.

**Offline:**

The state of a BBS or BBS Node when it is not able to receive users and the state of a user when he is not currently using the BBS.

**POP3:**

Post Office Protocol version 3. A TCP/IP protocol used by Internet e-mail clients to check and download mail from a TCP/IP mail server

(on TCP port 110).

**Post:**

The act of a user writing and saving a message in a sub-board.

**Protocol:**

A system of rules and procedures governing communications between two or more devices. Protocols vary, but communicating devices must follow the same protocol in order to exchange data. The format of the data, readiness to receive or send, error detection and error correction are some of the operations that may be defined in protocols. See Transfer Protocol.

**QWK Packet:**

A single compressed file, usually in PKZIP format, that contains new messages, E-mail, sysop bulletins, and a list of new files that can be downloaded by a user for use with an offline message reader. The filename is the BBS ID followed by a .QWK extension (regardless of the compression method). The developer of this packet format was Mark Herring (Sparky) of Sparkware, for use with his Qmail Door (external program). There are many popular offline message readers that support the QWK format. QWK packets may also be used for message networking (i.e. QWK Networking).

**REP Packet:**

A QWK reply packet. Also, a single compressed file, usually in PKZIP format, that contains e-mail or posts from the user that he/she created with an offline message reader. The filename is the BBS ID followed by a .REP extension. The packet must be uploaded by the user before the messages and e-mail can be sent to the destination users or posted.

**Restrictions:**

Flags that a sysop can place on a user to restrict the user from certain features of a BBS.

**RS-232:**

Interface standard developed by the Electronic Industries Association (EIA) to define the signals and voltages used when data is exchanged between a computer or terminal and a modem or serial printer. Data is usually transmitted via a cable with a 9 or 25 pin connector.

**SCFG:**

Synchronet configuration program. See [system\\_config.html](#) for details.

**Serial Port:**

See COM Port.

**SMTP:**

Simple Mail Transfer Protocol is a TCP-based protocol used to send and receive email (on TCP port 25).

**Sub-board:**

A section within a message group that contains multiple messages posted by users on a specific topic. Also referred to as a conference, forum, or special interest group (SIG).

**Sysop:**

System Operator. A person who participates in the maintenance or management of a BBS. In Synchronet, sysops are defined as users with a security level of 90 or higher.

**TCP:**

Transmission Control Protocol. TCP works with IP to ensure that packets travel reliably on the Internet. This is the method by which most Internet activity takes place.

**Text File Sections:**

Areas for the storage of text files that the sysop wants users to have the ability to read. Often referred to as general text file sections. Common text files would be information about the BBS, ANSI artwork, and documents on debatable subjects. Text files placed in text file sections do not get purged as public messages do and are not part of the transfer section, so credits and transfer access are not required.

**Transfer Protocol:**

A protocol designed to govern the transmission of files between two computer systems. BBS transfer protocols are usually specific to modem transmissions. The most common of which are Xmodem, Ymodem, and Zmodem. Most communications programs contain built-in protocol support and stand-alone transfer protocol programs (e.g SexyZ) are also available.

**Throughput:**

The effective rate of data flow for a file transfer, measured in bits per second. Throughput depends on the connect rate and the

error-control and data-compression protocols, if any.

**UART:**

Universal Asynchronous Receiver Transmitter. The IC (Integrated Circuit) that controls the serial port I/O. You must have a UART for each COM port in your computer. The most common UARTs for IBM PCs are NS8250s and NS16450s. If you are using a high-speed (9600bps or higher) modem with your COM port and having communication problems, quite often the only solution is to replace your UART for that COM port with a buffered UART, usually an NS16550AFN. More modern UARTs are being integrated with other peripheral controller ICs into a single chip (or chip-set). This design does not allow for the replacement of the actual UART. Internal modems have their own built-in UART.

**UDP:**

User Datagram Protocol. A communications protocol for the Internet network layer, transport layer, and session layer, which makes it possible to send a datagram message from one computer to an application running in another computer. Like TCP (Transmission Control Protocol), UDP is used with IP (the Internet Protocol). Unlike TCP, UDP is connectionless and does not guarantee reliable communication; the application itself must process any errors and check for reliable delivery.

**Upload:**

Transferring a file from a remote computer to a BBS or other host system.

**User to User Transfer:**

An upload that is sent to a particular user or set of users. These transfers are only allowed if the sysop creates a sub-board with a short name of "User". The sysop should set the access level to 90 and the upload level to something in the user range to allow users to upload to the directory, but not be able to list the contents of the directory. A user performs a user to user upload with the '/U' command from the transfer menu, and the destination user(s) can download the file with the '/D' command.

**[Back to Top](#)**

---

**[Copyright](#) © 2006 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Document](#) .

## Synchronet Baja® Documentation

### Table of Contents

1. **Introduction to BAJA**
  - 1.1 [Introduction](#)
  - 1.2 [Source Files](#)
  - 1.3 [BAJA.EXE](#)
  - 1.4 [Modules](#)
2. **BAJA Programming Basics**
  - 2.1 [Programming Syntax](#)
  - 2.2 [Variables](#)
  - 2.3 [Arguments](#)
  - 2.4 [C Strings](#)
  - 2.5 [Integer Constants](#)
  - 2.6 [Command Key Constants](#)
  - 2.7 [Include Files](#)
  - 2.8 [Macro Definitions](#)
  - 2.9 [Global Variable Declarations](#)
3. **Programming Style and Rules**
  - 3.1 [Programming Style](#)
  - 3.2 [Suggested Rules](#)
4. **BAJA Function Reference**
  - 4.1 [Function Definitions](#)
  - 4.2 [String Manipulation](#)
  - 4.3 [Integer Manipulation](#)
  - 4.4 [Logic/Control Flow](#)
  - 4.5 [Display](#)
  - 4.6 [Input](#)
  - 4.7 [Miscellaneous](#)
  - 4.8 [Mail](#)
  - 4.9 [Message Base](#)
  - 4.10 [File Base](#)
  - 4.11 [Chat](#)
  - 4.12 [Information](#)
  - 4.13 [File I/O](#)
  - 4.14 [File System](#)
  - 4.15 [Directory System](#)
5. **System Variables**
  - 5.1 [System Variables](#)
6. **Function Reference**
  - 6.1 [Quick Function Reference](#)

## [1.1] - Introduction

Baja (pronounced bä'hä) is a high-level development tool used to create "command shells" and "loadable modules" for Synchronet BBS software. The programmable command and menu structure (PCMS) technology in Synchronet version 2 allows sysops to modify the user interface for the BBS by replacing or modifying command shells. Synchronet version 2.0 shipped with six stock command shells (Classic and Novice Synchronet shells as well as emulations of PCBoard, Wildcat, Major, and Renegade BBS packages). These shells can be modified, removed, or replaced with ease. Up to 500 shells can be made available to the users of the BBS with configurable security access to each shell via ARS (see the Synchronet Sysop manual for details on ARS).

Synchronet loadable modules are used to extend or customize the abilities of the BBS. Some modules are automatically loaded during specific events (login, logon, logoff, etc) while others may be executed at the request of a user (much like a door or other internal menu option).

### Note:

If you come across an undefined term in this document, please read through the document to possibly find the term defined later or used again to clarify its meaning by context, or see the glossary in the Synchronet sysop manual.

[Back to Table of Contents](#)

## [1.2] - Source Files

Command shells and modules begin their life as a text source file (normally with a .SRC extension) or group of text files (including .INC files). These source files (and .INC files) can be edited with any ASCII text editor (e.g. EDIT, QEDIT, EDLIN, etc). The contents of the text file (often referred to as "source code") must conform to the Baja language specifications defined in this document. An extremely basic example of a command shell source file:

```
# EXAMPLE1.SRC

# Label for later "goto" commands
:TOP

# Clear the screen
CLS

# Show menu of commands
MNEMONICS "~Comment to Sysop\r\n"
MNEMONICS "~Goodbye (Logoff)\r\n"

# Show command prompt
PRINT "\r\nCommand: "

# Get command key
GETCMD "CG"

# Comment to sysop?
COMPARE_KEY C
IF_TRUE
    SETSTR "1"
        MAIL_SEND_FEEDBACK
        GOTO TOP
    END_IF

# Logoff?
COMPARE_KEY G
IF_TRUE
    LOGOFF
    GOTO TOP
END_IF
```

The syntax of the above text will be explained later in this document.

For more examples of Baja source code, please see the \*.SRC files in your Synchronet EXEC directory.

[Back to Table of Contents](#)

## [1.3] - BAJA.EXE

After the source file has been created, BAJA.EXE is used to compile the .SRC file into a binary file with a .BIN extension that Synchronet can interpret and execute. If the source filename is EXAMPLE1.SRC, then the command line would be:

```
BAJA EXAMPLE1
```

The compiler's output:

```
C:\$BBS\EXEC>baja example1.src
BAJA v2.10 · Synchronet Shell/Module Compiler · Developed 1995-97 Rob Swindell
Compiling EXAMPLE1.SRC...
Resolving labels...
Done.
C:\$BBS\EXEC>
```

This creates the binary file EXAMPLE1.BIN.

Warning: Do not attempt to edit, view, or print .BIN files.

If the compile fails, an error message will be displayed with the filename that contains the error and the specific line number followed by a colon and the offending code.

Baja source (.SRC), include (.INC), and binary (.BIN) files as well as the Baja program itself are normally stored in the Synchronet EXEC directory. This is where Synchronet loads command shells and modules from, so the .BIN files must be created in or copied to this directory before they can be used.

After a command shell has been compiled, it must be added to the list of available command shells in SCFG (Synchronet Configuration program) to be available for the users to use. If using a multinode BBS, the other nodes

must be re-run before the new command shell will be available on those nodes.

If modifying an existing command shell, SCFG does not need to be run and nodes do not have to be re-run since command shell binary files are dynamically loaded for each logon.

[Back to Table of Contents](#)

## [1.4] - Modules

Baja modules are very similar to command shells with the main difference being that they are not automatically loaded upon logon as command shells are (with the exception of login and logon modules). Instead, modules must be loaded from a command shell with the EXEC\_BIN function described later in this document or launched as an external program or event configured in SCFG. Binary module files, like command shell binaries, must be located in the Synchronet EXEC directory.

Modules can be used to consolidate identical portions of multiple command shells into one binary that can be loaded from each command shell so that if a modification needs to be made to that portion, then only one source file needs to be modified and recompiled as opposed to making the same change to every command shell and recompiling them all. An example of this use is the STR\_CMDS (string commands) module that is launched from multiple command shells.

Another use for modules is for third-party add-ons. Rather than distributing Baja source that must be merged into command shell source by sysops, developers can distribute module binaries that can be loaded directly from a command shell with a very small modification to the existing shells.

To load a Baja module in place of a DOS executable in SCFG, prepend an asterisk (\*) to the name of the module (e.g. "\*MYMOD") for the command line. If an argument is included on the command line (e.g. "\*MYMOD %A") the contents will be automatically placed into the current command string for possible use by the module or functions there in.

To load a Baja module while displaying a message, menu file, or TEXT.DAT line (excluding lines containing % variables) use "@EXEC:MYMOD@", where MYMOD is the name of the module to execute. This is useful for executing modules in places where they cannot normally be loaded by event or command line. @-codes are only expanded in posts and e-mail if posted locally by user #1 (the sysop).

[Back to Table of Contents](#)

## [2.1] - Programming Syntax

### Basics:

Execution always begins at the top of the source code and continues downward until a function is executed which changes the flow of execution. Only one function is allowed per line. The total length of each line cannot exceed 1000 characters.

### Whitespace:

All whitespace (TAB and SPACE characters) at the beginning of each line are ignored. The first printable character of a line is considered the first character. Completely blank lines are ignored.

### Comments:

If the first character of a line is the pound symbol (#), the line is ignored. These are normally used for text comments (remarks) about the following source code line(s). Comments may be included on the same lines as some (not all) functions, but in general it is considered best to keep comments on their own lines as to not interfere with proper Baja syntax. Comments are more easily read if there is a space after the pound symbol (#) and before the comment text begins.

### Labels:

If the first printable character of a line is a colon (:), the line is considered a label to be used for the GOTO or CALL functions. Labels are not case sensitive. There should be no space between the colon and the name of the label. Label names may not contain spaces. Duplicate label names within the same module or shell are disallowed.

### Case Sensitivity:

Most elements of Baja syntax are not case sensitive (where upper or lower case is significant). Functions, and variable names (for example) are not case sensitive. Character constants ('A' for example) are case sensitive.

[Back to Table of Contents](#)

## [2.2] - Variables:



Variable names are not case sensitive and must begin with an alphabetic character or an underscore (\_). Variable names can be any length and may contain alpha, numeric, and underscore characters only. The variable name "str" is reserved to represent the current command string and should not be used as a user defined variable name.

When referencing a variable name defined by the system or another module (global), the "!GLOBAL" compiler directive must be used to declare the variable or the compiler will not recognize the variable name and exit with a syntax error. Global variable declarations should appear near the top of your source file.

Baja modules that use variables require SBBS v2.2 or later.

## [2.3] - Arguments:

When a function requires one or more arguments, there must some amount of white space between the function name and each argument.

In function definitions (in this document), required arguments are listed in angled brackets (< >) and optional arguments are listed in square brackets ([ ]). The actual bracket symbols are NOT used in the source code. They are strictly for documentary purposes.

When a string variable argument is specified in the definition of a function, any string variable name (either local or global) may be used. If str is used in the source code as the string variable name, the current command string will be used in place of an actual user or system string variable.

An argument of # (pound sign) indicates an integer constant. The actual pound sign is not used in the argument. See the section on Integer Constants for more details.

An argument of "str" (with double quotes) indicates a string constant argument enclosed in double quotation marks. To place a double quotation mark inside a string constant (as part of the actual string constant), use \" instead of just " where you want the quotation mark to appear in the string.

An argument of "cstr" (with double quotes) indicates a C string constant argument enclosed in double quotation marks. See the next section on C strings for the definition of the contents.

See the beginning of the Functions section for the complete definition of all argument types and how they're specified in function definitions.

## [2.4] - C Strings:

Some character string arguments are defined as being C type (cstr). This indicates the contents of the string use the same special/control character escape sequences as defined for the printf() function in the ANSI definition of the C programming language. The only exception to the ANSI definition, is that \### defines a character value in decimal (base 10), not octal (base 8). All escape sequences begin with the backslash (\) character. The valid escape sequences are:

|      |                                |
|------|--------------------------------|
| \\   | Backslash                      |
| \?   | Question mark                  |
| \'   | Single quote                   |
| \"   | Double quote                   |
| \r   | Carriage return                |
| \n   | Line feed                      |
| \t   | Horizontal tab                 |
| \b   | Backspace                      |
| \a   | Bell                           |
| \f   | Formfeed                       |
| \v   | Vertical tab                   |
| \### | Character value in decimal     |
| \x## | Character value in hexadecimal |

The text string must be enclosed in double quote (") symbols. Example:

```
"Hello, world.\r\n"
```

To include an actual back-slash (\) character in a C string, you must prefix the back-slash with another back-slash. Example:

```
"C:\README.TXT"
```

would be an invalid path and filename.

```
"C:\\README.TXT"
```

would be the correct C string to represent C:\README.TXT.

## [2.5] - Integer Constants:

Many functions allow or require the use of integer constants (as opposed to integer variables). An integer constant is a number that is "hard-coded" into the resulting executable file and will remain constant (never change) from that point on.

You can specify integer constants in many ways:

| Representation        | Range                        | Examples      |
|-----------------------|------------------------------|---------------|
| Decimal (base 10)     | -2147483648 to 2147483647    | 0, 10, -25    |
| Hexadecimal (base 16) | 0x0 to 0xffffffff            | 0x7f, 0xA69B  |
| Octal (base 8)        | 00 to 037777777777           | 0377, 0505    |
| Bit value             | .0 to .31                    | .0, .1, .15   |
| ASCII value           | ' ' to ' '                   | 'A', ' ', '%' |
| Special char          | '\r' to '\v' (see C Strings) | '\r', '\n'    |

Constants can be modified during specification (no variables may be used in the modification). No space is allowed between the numeric constant and the modifier symbol. Valid modifiers:

| Symbol | Operation                   | Example  |
|--------|-----------------------------|----------|
| +      | Addition                    | 10+2     |
| -      | Subtraction                 | 20-8     |
| *      | Multiplication              | 4*3      |
| /      | Division                    | 36/3     |
| %      | Modulus (remainder)         | 53%8     |
| &      | Bit-wise AND                | 695&0x1f |
|        | Bit-wise OR                 | 0x10 8   |
| ~      | Bit-wise NOT                | ~3       |
| ^      | Bit-wise XOR (exclusive OR) | 0xff^12  |
| >>     | Right shift                 | 03700>>4 |
| <<     | Left shift                  | 255<<2   |

## [2.6] - Command Key Constants:

Some functions require the use of a command key constant. A command key constant is a keyword or symbol representing a valid command selected by a user.

The simplest way to specify a command key constant, is to simply type the key that you wish to use for a specific command. Example:

```
CMDKEY A
```

Would use the 'A' key for the command key constant. Simply specifying a letter does not indicate case sensitive (i.e. the above example could have been "CMDKEY a" with the same results).

To specify a specific case (upper or lower), you must prefix the command key with a single quote symbol ('). Example:

```
CMDKEY 'a
```

To specify a control character (like Ctrl-H, which is a backspace), you must prefix the control character (in upper case) with a caret (^) symbol. Example:

```
CMDKEY ^H
```

Valid control character specifications:

- ^A Attribute/color code (internal line editor)
- ^B Beginning of line (internal line editor)
- ^C Abort
- ^D Delete word right (internal line editor)
- ^E End of line (internal line editor)
- ^F Forward cursor (Right arrow key)
- ^G Beep
- ^H Backspace
- ^I Tab
- ^J Line feed (Down arrow key)
- ^K Ctrl-key help menu
- ^L Form feed (Clear screen)
- ^M Carriage return (Enter key)
- ^N Next word (internal line editor)
- ^O Temp pause toggle
- ^P Private message
- ^Q Un-pause
- ^R Redraw line (internal line editor)
- ^S Pause

```
^T Time info
^U List users online
^V Center line (internal line editor)
^W Delete word left (internal line editor)
^X Delete line (internal line editor)
^Y Delete to end-of line (internal line editor)
^Z Raw input mode toggle
^] Back cursor (Left arrow key)
^[ Escape (Esc key)
^^ Up cursor (Up arrow key)
```

You may also specify special control characters (carriage return, form feed, etc) be prepending a backslash (i.e. `\r`, `\f`, etc). See the section on C strings for details.

Some command key input functions allow the use of slash-commands (command keys prepended with a slash character, i.e. `/A`). The two character slash-commands are the only multi-character commands supported by command key functions. Example:

```
CMDKEYE /A
```

Is valid, but:

```
CMDKEYE -A
```

Is not valid.

To test specifically for digit (0-9) command keys and slash-digit command keys (`/0-9`), use `DIGIT` and `EDIGIT` respectively. Example:

```
CMDKEYE DIGIT
```

To test if the user hit a key from 0 to 9, or:

```
CMDKEYE EDIGIT
```

To test if the user hit a slash-key combination from `/0` to `/9`.

You may, of course, also test for a specific digit command key:

```
CMDKEYE 1
and
CMDKEYE /1
```

are both valid command key constants.

To specify `#` for a command key, use `'#` (otherwise `#` will be interpreted as the beginning of a comment).

To specify `SPACE` bar for the command key, use `' '`, example:

```
CMDKEY ' '
```

Extended ASCII characters (Ç for example) are not valid command keys.

[Back to Table of Contents](#)

## [2.7] - Include Files

Source code segments that are to be shared among multiple command shells or modules can be placed into a separate text file (preferably with a `.INC` extension) and included in the source code for multiple modules by using the `"!INCLUDE"` compiler directive to specify the filename to include. Example:

```
!INCLUDE MYDEFS.INC
```

This is useful when including files supplied by Digital Dynamics or third party developers to pre-define constants (macros), common sub-routines, or variable definitions. The following include files are distributed with Baja (and should not be modified):

```
ERRNO.INC Definitions for possible values of _ERRNO system variable
FILE_IO.INC Constants for use with file I/O functions
DIR_ATTR.INC Constants for use with SET_FILE_ATTR & GET_FILE_ATTR functions
USERDEFS.INC Synchronet User constants for use with various functions
NODEDEFS.INC Synchronet Node constants for use with various functions
SBBSDEFS.INC Synchronet system variable declarations and constants
              (automatically includes USERDEFS.INC and NODEDEFS.INC)
```

Take care not to include the same file more than once in the same source code. It is suggested you put all of your `!include` statements near the top of your source file. If you include a file that in-turn includes other files (`SBBSDEFS.INC`, for example, includes `USERDEFS.INC` and `NODEDEFS.INC`), you do not need to include the other files. If you `!include SBBSDEFS.INC`, you DO NOT need to `!include NODEDEFS.INC` and `USERDEFS.INC` as well. Take care not to nest include files too deeply or the compilation will fail.

This compiler directive is very similar to the "#include" directive used by C compilers, but if a specific path is not specified, only the current directory will be searched for the file to include.

[Back to Table of Contents](#)

## [2.8] - Macro Definitions

Commonly used character strings and integer constants can be specified as macros by using the "!DEFINE" compiler directive to specify the macro name to use and the value of the macro. Example:

```
!DEFINE MAXIMUM_USERS 250
```

Now, any place you might want to use the number 250 in your Baja source, you can use the macro "MAXIMUM\_USERS" instead. This makes changing the value of the numeric or character string constant very easy.

This compiler directive is very similar to the "#define" directive used by C compilers with the main exception being that the macro name is not case sensitive.

[Back to Table of Contents](#)

## [2.9] - Global Variable Declarations

When using variables defined by the system (see System Variables) or another shell or module (see GLOBAL\_STR and GLOBAL\_INT functions), you must first declare these variables using the "!GLOBAL" compiler directive. Example:

```
!GLOBAL SOME_GLOBAL_VAR
```

Multiple global variables may be declared with one !GLOBAL statement:

```
!GLOBAL SOME_GLOBAL_VAR ANOTHER_GLOBAL_VAR
```

If you attempt to use a global variable without first declaring it, the compiler will fail to compile the source with an "expected variable name" syntax error message.

This compiler directive is very similar to the "extern" keyword used by C compilers.

[Back to Table of Contents](#)

## [3.1] - Programming Style

As stated in the previous section, white space characters at the beginning of a line are ignored and have no effect on the resulting compilation. But, whitespace can be very useful in making the source code readable by indenting conditional blocks of code, and using blank lines to separate blocks of related code. Comments are also very helpful in making the code readable.

Compare the following source to the example given in the Source Files chapter:

```
:TOP
CLS
MNEMONICS "~Comment to Sysop\r\n"
MNEMONICS "~Goodbye (Logoff)\r\n"
PRINT "\r\nCommand: "
GETCMD "CG"
COMPARE_KEY C
IF_TRUE
SETSTR "1"
MAIL_SEND_FEEDBACK
GOTO TOP
END_IF
COMPARE_KEY G
IF_TRUE
LOGOFF
GOTO TOP
END_IF
```

Without the whitespace and the comments, the code appears much more cryptic (the control flow is hard to follow). But the code will compile into exactly the same binary file as the earlier example.

Consider this more elaborate example using indentation only:

***With indentation:***

```
COMPARE ARS FILE CMDS=0
```

```
IF_TRUE
  COMPARE_USER_MISC UM_ASK_NSCAN
  IF_TRUE
    YES_NO "\r\n\r\nSearch all libraries for new files"
    IF_TRUE
      FILE_NEW_SCAN_ALL
    END_IF
  END_IF
END_IF
```

**Without indentation:**

```
COMPARE_ARS FILE_CMDS=0
IF_TRUE
COMPARE_USER_MISC UM_ASK_NSCAN
IF_TRUE
YES_NO "\r\n\r\nSearch all libraries for new files"
IF_TRUE
FILE_NEW_SCAN_ALL
END_IF
END_IF
END_IF
```

As you can see without the indentation, you can get lost in nested IF statements trying to understand how the code will execute.

While you may have known quite well what you intended to accomplish and how you expected the code to execute at the time of writing it, it would be much harder to understand how the code was expected to execute at a later date (or by another person) if a consistent indentation style was not used.

[Back to Table of Contents](#)

## [3.2] - Suggested Rules of Style

There isn't a "right" or "wrong" way to use indentation, blank lines, or comments, but we will present a set of suggested "rules" here. If you decide you prefer another style, that is fine. The most important thing is that you are consistent in your use of whatever style you choose.

### Top of File Comments

Place a comment as the very first line of the file with the name of the .SRC file. Example:

```
# EXAMPLE.SRC
```

Then a block of comment lines that describe what the shell or module's purpose is, who programmed it, when, where, etc.

### Label Comments

Labels normally represent the beginning of a sub-routine and should be well commented. A graphical line to show the beginning of a new sub-routine is often helpful. Example:

```
##### My Sub-routine #####
:MY-SUB
```

### CMD\_HOME Comments

The CMD\_HOME function marks the beginning of a command/menu loop and should be well commented. Example:

```
#####
# MAIN MENU #
#####
CMD_HOME
```

### Conditional Indentation

Whenever a block of code is conditionally executed (only executed under specific conditions), the block of code should be indented one tabstop past the block header (IF\_TRUE, IF\_FALSE, ELSE, SWITCH, CASE, CMDKEY, CMDSTR, etc). The block of code to be indented includes the block terminator (END\_IF, END\_CMD, END\_CASE, END\_SWITCH, etc). Example:

```
GETCMD "ABCD"

CMDKEY A
  COMPARE_ARS LEVEL 50
  IF_TRUE
    PRINT "You have level 50 or higher.\r\n"
  ELSE
```

```
        PRINT "You have level 49 or lower.\r\n"
    END_IF
END_CMD
```

Separate Code Blocks

Use blank lines to separate medium to large code blocks from other code (most especially CMDKEY, CMDKEYS, CMDSTR, and SWITCH code blocks). Example:

```
GETCMD "ABCD"

CMDKEY A
    PRINT "You hit A.\r\n"
END_CMD

CMDKEY B
    PRINT "You hit B.\r\n"
END_CMD
```

End of File Comment

Place a comment at the end of the file. Example:

```
# End of EXAMPLE.SRC
```

[Back to Table of Contents](#)

# [4.1] - Function Definitions

The remainder of this document is for defining the correct use of all Baja functions. If you wish to look up a specific function by subject, please refer to the table of contents or the quick reference at the end of the document.

Argument Types

Each function definition will begin with the name of the function followed by all possible arguments (parameters) and the type of each argument. The possible argument types are:

Type      Description

```
"str"           String constant (contained in double quotes)
"cstr"          C string constant (contained in double quotes)
txt  String constant (no quotes)
#   Integer constant (the # symbol is not part of the syntax)
str_var  String variable name or 'str' for current command string
int_var  Integer variable name
any_var  Any string or integer variable name
char  Character constant (single character, no quotes)
key  Command key constant
keylist  List of command key constants (no space between each key)
```

Each argument type is contained in either angled brackets (< and >) to indicate the argument is required, or square brackets ([ and ]) to indicate the argument is optional. An argument of [...] indicates a variable number of optional arguments of the last specified type. Example:

```
    SPRINTF <str_var> <"cstr"> <any_var> [any_var] [...]
```

This function definition indicates that the SPRINT function requires a minimum of three arguments: a string variable, a C string constant, and one or more variables of either integer or string type.

Required Include Files

Following the function argument types, may be one or more required include (.INC) files. If any files are listed on the same line as the function (to the right of the page), these files must be included at the top of your source file to use this function arguments correctly. Example:

```
    !INCLUDE NODEDEFS.INC
```

Do not include the same file more than once in the same source file. Once is enough, and it should be done so near the top of your source (.SRC) file.

Argument Names

If any arguments are allowed for a function, a list of argument names will follow the text "args:" below the function name and argument list. This will help in referencing the individual arguments in the function description.

[Back to Table of Contents](#)

# [4.2] - String Manipulation Functions



Functions used for manipulating string variables (global or local) and/or the current command string.

**STR <str\_var> [str\_var] [...]**  
**args: new\_vars**

Define one or more local string variables for later use. The new\_vars arguments are the names of the variables to define. These names must not be currently used by other variables (local or global).

The variable name "str" is used by other functions to refer to the current command string, and therefore cannot be used as a user defined variable name.

After definition, the variable names can be used by other functions. The variables will each be initialized to a null string. The variables can only be referenced in the current shell or module. When the current shell or module is terminated, the variable names and their contents are lost. If you wish to retain a variable name and its contents between shells or modules, use a global variable instead of a local variable.

**GLOBAL\_STR <str\_var> [str\_var] [...]**  
**args: new\_vars**

Define one or more global string variables for later use. The arguments are the names of the variables to define. Each variable name must not be currently used by another variable (local or global). Global variable names and their contents are not disposed of until the current user logs off the BBS.

**SET <str\_var> <"cstr">**  
**args: dest value**

Writes to the string variable dest, the contents of the C string specified as value. This function is the Baja equivalent of the standard C strcpy() function with the exception that the source must be a string constant. Example:

```
STR USER_NAME
SET USER_NAME "John Doe"
```

When writing to the current command string, use the SETSTR function instead of this function for best optimization.

**COPY <str\_var> <any\_var>**  
**args: dest source**

Copies the source variable (string or integer) into the specified destination variable. This function is the Baja equivalent of the standard C strcpy() function with the exception that the source must be a variable, but may be either of integer or string type. Example:

```
# define two variables
STR VAR1 VAR2
# copy some text into the first variable
SET VAR1 "Some Text"
# copy that variable into the second variable
COPY VAR2 VAR1
# display the contents of second variable
PRINT VAR2
```

If the source variable is an integer variable, it is converted into a decimal (base 10) string before the copy is executed. Use SPRINTF if you want to convert an integer variable into a string using hexadecimal (base 16) or octal (base 8) representation instead of decimal.

**SWAP <str\_var> <any\_var>**  
**args: var1 var2**

Exchanges the contents of two string variables, or a string variable and an integer variable. The first argument (var1) is the name of a string variable and the second argument (var2) is the name of a string or integer variable. If var2 is an integer variable, var1 will be set to the decimal representation of the integer variable contents and var2 will be set the integer representation of var1 (specified in hex, octal, or decimal). Example:

```
INT I
STR S

# set S to 100h (256)
SET S "0x100"
# set I to 100
SET I 100
# swap them
SWAP S I
# I now equals 256 and S is "100"
```

**STRCAT <str\_var> <str\_var or "cstr">**  
**args: dest source**

Concatenates (appends) the contents of a string variable (specified as dest) with the contents of a C string or string variable specified as source. This function is the Baja equivalent of the standard C strcat() function.

**SPRINTF** <str\_var> <"cstr"> <any\_var> [any\_var] [...]  
args: dest format variables

Writes the specified variables into the string variable (dest) based on the specified format. This function is the same as PRINTF except that the output is written into the dest string variable instead of the console. This function is the Baja equivalent of the standard C sprintf() function with the exception that all the arguments following the format must be variables (no constants allowed). See the PRINTF function definition for details.

**TIME\_STR** <str\_var> <int\_var>  
args: dest time

Writes to the string variable (specified as dest) the time and date stored in the integer variable specified by time. This function is the Baja equivalent to the standard C ctime() function. Example:

```
# Display current time
INT T
TIME T
TIME_STR STR T
PRINTF "Current time: %s\r\n" STR
```

Would display the current date/time similar to:

Current time: Thu Aug 31 1995 08:34 am

**DATE\_STR** <str\_var> <int\_var>  
args: dest time

Writes to the string variable (specified as dest) the date stored in the integer variable specified by time (obtained with TIME or similar function) in MM/DD/YY format or DD/MM/YY format if European date format is enabled in SCFG. Example:

```
# Display current date
INT T
TIME T
DATE_STR STR T
PRINTF "Current date: %s\r\n" STR
```

Would display the current date similar to:

Current date: 08/31/95

**SECOND\_STR** <str\_var> <int\_var>  
args: dest seconds

Writes to the string variable (specified as dest) the number of seconds (in HH:MM:SS format) stored in the integer variable specified by seconds. Example:

```
# Display number of seconds elapsed
INT START END
TIME START
PRINT "Wait a period of time, then hit a key..."
GETKEY
TIME END
SUB END START
SECOND_STR STR END
PRINTF "\r\n%s time elapsed.\r\n" STR
```

**FTIME\_STR** <str\_var> <"cstr"> <int\_var>  
args: dest format time

Writes to the string variable (specified as dest) the date and time information from the integer variable (specified as time) formatted according to string constant specified as format. The format characters are described below:

```
%a abbreviated weekday name
%A full weekday name
%b abbreviated month name
%B full month name
%c date and time
%d day of month (01-31) (zero-padded)
%H hour (24-hour clock) (00-23)
%I hour (12-hour clock) (01-12)
%j day of the year (001-366)
%m month (01-12)
%M minute (00-59)
%p AM or PM
%S second (00-59)
%U week number of the year (Sunday as 1st day of week) (00-52)
```

```
%w weekday (0-6, Sunday is 0)
%W week number of the year (Monday as 1st day of week) (00-52)
%x date
%X time
%y year without century (00-99)
%Y year with century
%Z time zone name, if any
%% % (percent symbol)
```

This function is the Baja equivalent of the standard C strftime() function.  
Example:

```
# Display current date and time
INT T
TIME T
FTIME_STR STR "Current_Time:%A,%B-%d-%Y,%I:%M %p" T
PRINTF "%s\r\n" STR
```

**SHIFT\_STR [str\_var] <#>**  
**args: string len**

-----  
This function ("Shift String") shifts the specified string variable if specified (current command string if not specified) to the left 'len' number of characters.

```
# Displays the string "ELLO"
SETSTR "HELLO"
    SHIFT_STR 1
PRINT STR
```

This function is most often used to parse command string arguments or "stacked" commands.

**STRIP\_CTRL [str\_var]**  
**args: string**

-----  
Removes all control characters (and Ctrl-A codes) from specified string variable. If string is not specified, then the current command string is processed.

**TRUNCSP [str\_var]**  
**args: string**

-----  
Removes (truncates) all white space characters from end of specified string variable or current command string (if string not specified).

**STRUPR [str\_var]**  
**args: string**

-----  
Converts specified string argument to all uppercase characters. This function is the Baja equivalent to the standard Cstrupr() function.

**STRLWR [str\_var]**  
**args: string**

-----  
Converts specified string argument to all lowercase characters. This function is the Baja equivalent to the standard Cstrlwr() function.

**SETSTR <"cstr">**  
**args: text**

-----  
Writes to current command string, specified constant text string. This function should be used in place of "SET STR" for best optimization.

**REPLACE\_TEXT <#> <"cstr">**  
**args: num text**

-----  
This function is used to replace a default internal text string (read from the CTRL\TEXT.DAT file) in a shell or module. The first argument (num) is the text number to replace (see TEXT.DAT for the number of the text you wish to replace). The second argument (text) is the C type string to use in place the current one. All text replacements are lost (back to default TEXT.DAT) when the user logs off. Example, to replace the "[Hit a key]" prompt with a different string:

```
REPLACE_TEXT 563 "\ln\lh\lb{\lwPAUSE\lb} "
```

When replacing a TEXT.DAT string that contains %, %d, or other % specifiers, you will need to change them to %s, %d, etc (without changing the order).

If you wish to globally replace a string (for all shells and modules), you may want to just edit the CTRL\TEXT.DAT file directly instead.

**LOAD\_TEXT <"str">**  
**args: filename**

-----  
Loads an alternate TEXT.DAT file (from CTRL directory, automatically assuming

a .DAT extension for "filename"). Only those text items that are different from the current TEXT items will be changed. All changes will only be in effect until the current user logs off or the effects are reversed with the REVERT\_TEXT function. The base filename should be something other than just "TEXT" (since that is the default TEXT.DAT that is loaded by SBBS).

This function is useful for making multiple languages available to your users.

```
REVERT_TEXT <# or ALL>
  args: num
-----
```

This function is used to reverse the effects of a previous call to REPLACE\_TEXT or LOAD\_TEXT (reverts back to default TEXT.DAT). Example:

```
REPLACE_TEXT 563 "\ln==[Hit a Key]== "
PAUSE
REVERT_TEXT 563
```

If the argument is the word ALL, then all text strings are reverted to the original TEXT.DAT entries.

[Back to Table of Contents](#)

## [4.3] - Integer Manipulation Functions

Functions used for manipulating integer variables (global or local).

```
INT <int_var> [int_var] [...]
args: new_vars
-----
```

Define one or more local integer variables for later use. The new\_vars arguments are the names of the variables to define. These names must not be currently used by other variables (local or global).

After definition, the variable names can be used by other functions. The variables will each be initialized to a 0. The variables can only be referenced in the current shell or module. When the current shell or module is terminated, the variable names and their contents are lost. If you wish to retain a variable name and its contents between shells or modules, use a global variable instead of a local variable.

```
GLOBAL_INT <int_var> [int_var] [...]
  args: new_vars
-----
```

Define one or more global integer variables for later use. The arguments are the names of the variables to define. Each variable name must not be currently used by another variable (local or global). Global variable names and their contents are not disposed of until the current user logs off the BBS.

```
SET <int_var> <#>
args: dest      value
-----
```

Writes to the integer variable dest, the integer constant specified as value.

```
INT I
SET I 100
```

```
ADD <int_var> <any_var or #>
args: dest value
-----
```

Adds the value specified in the second argument to the integer variable specified as dest.

```
SUB <int_var> <any_var or #>
args: dest value
-----
```

Subtracts the value specified in the second argument from the integer variable specified as dest.

```
MUL <int_var> <any_var or #>
args: dest value
-----
```

Multiplies the first argument by the value specified in the second argument and places the result into dest.

```
DIV <int_var> <any_var or #>
args: dest value
-----
```

Divides the first argument by the value specified in the second argument and places the result into dest.

```
MOD <int_var> <any_var or #>
args: dest value
-----
```

Divides the first argument by the value specified in the second argument and places the remainder into dest.

**AND <int\_var> <any\_var or #>**  
**args: dest value**

-----  
Bit-wise ANDs the value specified in the second argument with the integer variable specified as dest.

**OR <int\_var> <any\_var or #>**  
**args: dest value**

-----  
Bit-wise ORs the value specified in the second argument with the integer variable specified as dest.

**NOT <int\_var> <any\_var or #>**  
**args: dest value**

-----  
Bit-wise ANDs the unary opposite of the value specified in the second argument with the integer variable specified as dest.

**XOR <int\_var> <any\_var or #>**  
**args: dest value**

-----  
Bit-wise exclusive ORs the value specified in the second argument with the integer variable specified as dest.

**COPY <int\_var> <any\_var>**  
**args: dest source**

-----  
Copies the source variable (string or integer) into the specified destination variable. Example:

```
# define two variables
INT INT1 INT2
# place a value into the first variable
SET VAR1 5
# copy that variable into the second variable
COPY VAR2 VAR1
# display the contents of second variable
PRINTF "VAR2=%ld\r\n" VAR2
```

If the source variable is an string variable, it is converted into an integer before the copy is executed.

**SWAP <int\_var> <any\_var>**  
**args: var1 var2**

-----  
Exchanges the contents of two integer variables, or a string variable and an integer variable. The first argument (var1) is the name of an integer variable and the second argument (var2) is the name of a string or integer variable.

**RANDOM <int\_var> <#>**  
**args: dest num**

-----  
Places a random number between 0 and num-1 into dest.

**TIME <int\_var>**  
**args: dest**

-----  
Places the current time (in unix/time\_t format) into dest. This function is the Baja equivalent of the standard C time() function.

**STRLEN <int\_var> <str\_var>**  
**args: dest string**

-----  
Places the length (in characters) of the specified string into dest. This function is the Baja equivalent of the standard C strlen() function.

**DATE\_INT <int\_var> <str\_var>**  
**args: dest date**

-----  
Converts a date string (in MM/DD/YY format) into an integer and places it into dest.

**CRC16 <int\_var> <str\_var>**  
**args: dest string**

-----  
Calculates the 16-bit CRC of the specified string and places the result into dest.

**CRC32 <int\_var> <str\_var>**  
**args: dest string**

-----  
Calculates the 32-bit CRC of the specified string and places the result into dest.

**CHKSUM <int\_var> <str\_var>**  
**args: dest string**

-----

Calculates the CHKSUM of the specified string and places the result into dest.

**CHARVAL <int\_var> <str\_var>**  
**args: dest string**

Places the ASCII value of the first character in the specified string into dest.

[Back to Table of Contents](#)

## [4.4] - Logic/Control Flow Functions

**GOTO <txt>**  
**args: label**

This function is used to force the control flow to immediately jump to another part of the code. A valid label name must follow GOTO as an argument. This function is very similar to the GOTO command used in the BASIC and C programming languages or the JMP instruction used in the Assembly programming language.

**CALL <txt>**  
**args: label**

This function is used to execute a sub-routine. The sub-routine must be defined with the specified label somewhere in the current module. When a RETURN function is executed in the sub-routine, control flow will return to the next line following the CALL function that started the sub-routine. The CALL function is very similar to the GOSUB command used in the BASIC programming language or the CALL instruction used in the Assembly programming language. At least one RETURN function must exist in the sub-routine. Sub-routines should not use a GOTO function to "jump out" of the sub-routine.

**RETURN**

This function is used to exit a sub-routine and return to the point in the code from which it was called with the CALL function. If a RETURN function is executed without a prior CALL, then the shell or module is immediately terminated. If a shell is terminated with a user online, they are immediately logged off. If a module is terminated, execution control returns to the shell or module that loaded it.

**SETLOGIC <TRUE or FALSE or GREATER or LESS>**  
**args: logic**

This function is used to set the current logic state to TRUE (same as EQUAL), FALSE (same as NOT\_EQUAL), GREATER, or LESS. The logic state is set by many functions, and tested by the IF\_... functions to control the flow of execution.

**COMPARE <any\_var> <any\_var or "cstr" or #>**  
**args: var cmp**

This function compares any two variables, or a variable and a constant. If the cmp argument is an integer variable or integer constant and the var variable is a string variable, then cmp is converted to a string representation of the cmp variable before the comparison is made. If the cmp argument is a string variable or string constant and the var variable is an integer variable, then cmp is converted to an integer before the comparison is made. The current logic state is set to the result of the comparison: TRUE (same as EQUAL), FALSE (as an NOT\_EQUAL), GREATER, or LESS.

**IF\_TRUE**

If the current logic state is set to TRUE, then following block of code (ending in an END\_IF or ELSE function) will be executed. Otherwise, the interpreter will skip the code until an ELSE or END\_IF function is found and resume execution from that point. This function is the same as IF\_EQUAL.

**IF\_FALSE**

This function is identical to the IF\_TRUE function except that it works in the opposite manor in regards to the current logic state. This function is the same as IF\_NOT\_EQUAL.

**IF\_GREATER**

If the current logic state is set to GREATER by a subsequent COMPARE function, the following IF\_... block will be executed.

**IF\_GREATER\_OR\_EQUAL**

If the current logic state is set to GREATER or EQUAL, the following IF\_... block will be executed. This function is the same as IF\_EQUAL\_OR\_GREATER.



## **IF\_LESS**

-----

If the current logic state is set to LESS by a subsequent COMPARE function, the following block will be executed.

## **IF\_LESS\_OR\_EQUAL**

-----

If the current logic state is set to LESS or EQUAL, the following code block will be executed. This function is the same as IF\_EQUAL\_OR\_LESS.

## **ELSE**

----

This function marks the end of an IF\_... block of code and begins an alternate block of code to be executed if the condition is not met for the IF\_... function. The block of code is terminated with an END\_IF function.

## **END\_IF**

-----

This function is used to terminate an IF\_... or ELSE code block. Every IF\_... function must be followed by an ELSE or END\_IF instruction, and every ELSE function must be followed by an END\_IF.

## **SWITCH <int\_var>**

-----

### **CASE <#>**

-----

### **END\_CASE**

-----

### **DEFAULT**

-----

### **END\_SWITCH**

-----

The SWITCH function is used in conjunction with the CASE, DEFAULT, END\_CASE, and END\_SWITCH functions to replace multiple COMPARE/IF\_.../ELSE statements.

Example:

```
INT I
RANDOM I 3
COMPARE I 0
IF_EQUAL
  PRINT "Zero"
ELSE
  COMPARE I 1
  IF_EQUAL
    PRINT "One"
  ELSE
    COMPARE I 2
    IF_EQUAL
      PRINT "Two"
    END_IF
  END_IF
END_IF
```

Could be replaced with:

```
INT I
RANDOM I 3
SWITCH I
  CASE 0
    PRINT "Zero"
  END_CASE
  CASE 1
    PRINT "One"
  END_CASE
  CASE 2
    PRINT "Two"
  END_CASE
  DEFAULT
    PRINTF "Random returned %ld" I
  END_CASE
END_SWITCH
```

If the SWITCH argument value is equal to one of the specified CASE arguments, then that CASE code block is executed until an END\_CASE function is reached. If an END\_CASE function does not exist in the code block, the execution will continue through (fall-through) the following CASE or DEFAULT functions until an END\_CASE or END\_SWITCH function is reached.

If the SWITCH argument does not match any of the CASE arguments, the DEFAULT code block will be executed until an END\_CASE or END\_SWITCH function is reached. If a DEFAULT function is not specified, execution will skip to the END\_SWITCH function. Only one DEFAULT code block should be specified per SWITCH/END\_SWITCH block, if used at all. This function set is the Baja equivalent of the standard C switch(), case:, default:, and break keywords.

## **CMD\_HOME**

-----

This function (called "Command Home") is very similar to a label. It is a convenience function for programming menu commands. It sets a return point (or anchor, if you will) for the automatic looping on a specific menu. Execution control is returned to the point in the code where the last CMD\_HOME was specified by the END\_CMD function.

## **CMDKEY <key>**

**args: cmd**

-----

This function (called "Command Key") is like a combination of the COMPARE\_KEY and IF\_TRUE functions. It is usually immediately preceded by a GETCMD, GETKEY, GETSTR or similar function.

If the current command key matches the command key constant (cmd), then the block of code following (terminated with an END\_CMD function) will be executed. When the END\_CMD function is executed, execution control returns to the most recently executed CMD\_HOME function. If a RETURN or GOTO function will be executed before the END\_CMD function, then a CMD\_POP function must be executed to tell the interpreter to "forget" the previous CMD\_HOME function. Even if execution control will cause the END\_CMD function to never be executed, it still must be present to define the code block so that it can be skipped if the CMDKEY comparison is false.

## **CMDKEYS <keylist>**

**args: cmds**

-----

This function is identical to the CMDKEY function with the exception that the argument is a list of valid keys and if the current command key matches any of those listed, the following code block is executed.

## **CMDSTR <"cstr">**

**args: cmd**

-----

This function (called "Command Str") is identical to the CMDKEY function with the exception that the argument is a C type string. It is usually immediately preceded by a GETSTR or similar function.

## **END\_CMD**

-----

This function is used to mark the end of a CMDKEY, CMDKEYS, or CMDSTR code block.

## **CMD\_POP**

-----

This function tells the interpreter to "forget" the most recently executed CMD\_HOME function. It should only be used when a CMDKEY, CMDKEYS, or CMDSTR code block contains a GOTO or RETURN function.

## **COMPARE\_KEY <key>**

**args: cmd**

-----

This function is identical to the CMDKEY function in regards to the key argument. The current command key is set by the GETCMD, GETKEY, GETSTR or similar functions. If the current command key matches the key argument, the current logic state is set to TRUE, otherwise it is set to FALSE.

As with all COMPARE functions, it is usually immediately followed by an IF\_TRUE or IF\_FALSE function.

## **COMPARE\_KEYS <keylist>**

**args: cmds**

-----

This function is identical to the CMDKEYS functions in regards to the keylist argument. If the current command key is contained in the keylist, then the current logic state is set to TRUE, otherwise it is set to FALSE.

## **COMPARE\_STR <"cstr">**

**args: string**

-----

This function compares the current command string (set by the SETSTR, GETSTR, or similar functions) against the C type string argument and sets the current logic state to TRUE or FALSE accordingly. The comparison is not case sensitive. This function is the Baja equivalent to the standard C strcmp() function.

## **COMPARE\_WORD <"cstr">**

**args: string**

-----

This function is identical to the COMPARE\_STR function with the exception that the current command string must only match the string argument for as many characters as included in the string argument.

If for example, the current command string is "OPEN BLACKJACK" then the line:

```
COMPARE_WORD "OPEN"
```

would set the current logic state to TRUE since the first 4 characters of each string match, but the line:

```
COMPARE_STR "OPEN"
```

would set the current logic state to FALSE since the strings do not entirely match.

```
COMPARE_ARS <txt>
  args: ars
-----
```

This function compares the requirement string argument ars against the information on the current user and sets the current logic state to either TRUE or FALSE based on the result of the comparison. See the ARS chapter of the Synchronet sysop manual for details on the syntax of ARS.

Example:

```
COMPARE_ARS LEVEL 60
IF_TRUE
    PRINT "You have level 60 or higher.\r\n"
END_IF
```

```
COMPARE_STRN <#>  <str_var> <str_var or "cstr">
  args: num  str1      str2
-----
```

This function is used to compare the first num characters of strings str1 and str2. This function is the Baja equivalent of the standard C strnicmp() function. If the two strings are the same (ignoring case) for the first num characters, the current logic state is set to TRUE, otherwise it is set to FALSE.

```
COMPARE_SUBSTR <str_var> <str_var or "cstr">
  args: str1  str2
-----
```

This function is used to check if str2 is contained in str1 (case sensitive). This function is the Baja equivalent of the standard C strstr() function. If str2 is located in str1, the current logic state is set to TRUE, otherwise it is set to FALSE. This following example would set the current logic state to TRUE:

```
SETSTR "This is a test"
COMPARE_SUBSTR STR "test"
```

[Back to Table of Contents](#)

## [4.5] - Display Functions

Display functions are used to display output to the user either locally, remotely, or both.

```
PRINT <"cstr" or var>
args:  string
-----
```

This function sends a string of characters to the user (both locally and remotely, if applicable). The string to be printed can extend beyond the 80th column of the screen, but when printing multiple lines, multiple consecutive PRINT statements are preferred.

Example:

```
PRINT "hello, world\r\n"

or:

STR my_var
SET my_var "hello, world\r\n"
PRINT my_var
```

The string can contain color codes using Synchronet Ctrl-A codes. See the Synchronet sysop manual for the definition of Ctrl-A codes and their usage. If you are using an ASCII editor that does not support control characters, a "\1" can be used in place of the Ctrl-A (smiley face) character.

Variables can also be included in the string (example: user's name, sysop's name, BBS name, etc) by using Synchronet supported @-codes and command line % specifiers. See the Synchronet sysop manual for the definition of @-codes and command line % specifiers. The %s command line specifier, when used, is replaced with the current command string and %f is replaced with the complete path of the current file directory. Be careful not to confuse command line % specifiers (%a, %b, etc) with printf % specifiers (described later). They can actually be used together in the same string (supported by PRINTF functions) by using double % for the command line % specifiers (e.g. %%s instead of %s).

```
PRINTF <"cstr"> [any_var] [...]
args: format  variables
-----
```

This function is similar to print, except that defined variables can be placed (formatted) into to displayed text output. This function is the Baja

equivalent of the standard C printf() function. The format argument is a C type string with C printf() % specifiers for variable formatting.

The valid specifiers are:

|         |                                                               |
|---------|---------------------------------------------------------------|
| %%      | Percent symbol                                                |
| %s      | String variable                                               |
| %#s     | String variable, minimum # chars wide, right justified        |
| %-#s    | String variable, minimum # chars wide, left justified         |
| %.#s    | String variable, maximum # chars wide                         |
| %#. #s  | String variable, min and max width specified, right justified |
| %-#. #s | String variable, min and max width specified, left justified  |
| %ld     | Integer variable, decimal - use lx for hexadecimal (lX=caps)  |
| %#ld    | Integer variable minimum # chars wide right justified         |
| %-#ld   | Integer variable minimum # chars wide left justified          |
| %0#ld   | Integer variable minimum # digits wide zero padded            |

Multiple variable names may be specified after the str argument, but each variable MUST be properly represented in the str argument with the correct PRINTF specifier (either %ld, %lx, or %s).

Example:

```
STR s
SET s "hello, world"
PRINTF "%s~\r\n"      s
PRINTF "%10s~\r\n"    s
PRINTF "%10s~\r\n"    s
PRINTF "%-10s~\r\n"   s
PRINTF "%15s~\r\n"    s
PRINTF "%-15s~\r\n"   s
PRINTF "%15.10s~\r\n" s
PRINTF "%-15.10s~\r\n" s
```

Would produce the following display:

```
~hello, world~
~hello, world~
~hello, wor~
~hello, world~
~hello, world  ~
~      hello, wor~
~hello, wor      ~
```

And:

```
INT i
SET i 1234
PRINTF "%ld~\r\n"      i
PRINTF "%3ld~\r\n"     i
PRINTF "%6ld~\r\n"     i
PRINTF "%-6ld~\r\n"    i
PRINTF "%06ld~\r\n"    i
PRINTF "%06lx~\r\n"    i
PRINTF "%06lX~\r\n"    i
```

Would produce the following display:

```
~1234~
~1234~
~ 1234~
~1234 ~
~001234~
~0004d2~
~0004D2~
```

```
PRINT_LOCAL <"cstr">
args: string
-----
```

This function works identical to the PRINT function with the exception that the output is only displayed locally (on the BBS console). Ctrl-A codes and @-codes are not supported.

```
PRINT_REMOTE <"cstr">
args: string
-----
```

This function works identical to the PRINT function with the exception that the output is only displayed remotely (on the user's terminal). If the current user is not logged on remotely, then no output is performed. Ctrl-A codes and @-codes are not supported.

```
PRINTSTR
-----
```

This function works identical to the PRINT function with the exception that this function does not accept a string argument, and instead uses the current command string (set with SETSTR, GETSTR, or similar function) as the character string to print.

Example:

```
SETSTR "hello, world\r\n"
PRINTSTR
```

#### **PRINTKEY**

-----  
This function displays the current command key. The command key is normally obtained with the GETKEY, GETCMD, or similar function.

#### **MNEMONICS <"cstr">** **args: string**

-----  
This function works identical to the PRINT function with the exception that the output is automatically colorized and characters following a tilde (~) are considered a mnemonic command character (hot key) and are high-lighted or placed in parentheses if the user's terminal does not support ANSI. If Ctrl-A codes are included in the str argument, then the automatic colorization and high-lighting is not used.

Example:

```
mnemonics "Select: ~Good, ~Bad, or ~Indifferent: "
```

#### **CLS**

---  
This function clears the current text screen.

#### **CRLF**

----  
This function displays a carriage return/line feed (new line). It is equivalent to PRINT "\r\n", but requires less memory and executes faster.

#### **PRINTFILE <"str" or str\_var> [#]** **SBBSDEFS.INC** **args: file mode**

-----  
Displays the contents of the file to the user (both locally and remotely, if applicable).

Valid mode parameters:

P\_NOABORT Disallows abortion of display with Ctrl-C  
P\_SAVEATR Retain attributes (colors) after display  
P\_NOATCODES Don't expand @-code variables/actions  
P\_OPENCLOSE Open and close immediately (before display)

Multiple mode parameters can be specified by joining them together with the OR (|) symbol.

Example:

```
!INCLUDE SBBSDEFS.INC
PRINTFILE "C:\TEXT\HELP.TXT" P_NOABORT|P_OPENCLOSE
```

#### **PRINTTAIL <str\_var> <#> <#> SBBSDEFS.INC** **args: file mode x**

-----  
Displays the last x number of lines of file. The valid mode parameters are the same as those shown for the PRINTFILE function. If you don't want any of the mode values, use 0 for the mode.

#### **PRINTFILE\_STR**

-----  
This function is identical to the PRINTFILE function, with the exception that there is no argument, the current command string is used for the path and filename instead.

#### **PRINTFILE\_LOCAL <"str">** **args: file**

-----  
This function works identical to the PRINTFILE function, with the exception that the contents are only displayed locally.

#### **PRINTFILE\_REMOTE <"str">** **args: file**

-----  
This function works identical to the PRINTFILE function, with the exception that the contents are only displayed remotely. If the user is not logged on remotely, then no output is performed.

#### **LIST\_TEXT\_FILE**

-----  
This function works similar to the PRINTFILE function, with the exception that the system password is required when run remotely and the user is prompted to enter the path and name of the file to list.

#### **EDIT\_TEXT\_FILE**

-----  
This function is generally for sysop use only. It will automatically prompt the user (if remote) for the system password, and then prompt them for the path and name of the file to edit.

**PAUSE**

-----

This function produces a "[Hit a key]" prompt and waits for the user to hit a key.

**MENU <"str">**

**args: file**

-----

This function is similar to the PRINTFILE function with the exception that the TEXT\MENU directory is assumed to be the root path for the file and the file's extension (.ASC, .ANS, .MON, or .RIP) should not be specified. This function will automatically display the correct file based on the what the user's terminal supports and which menu files are available. To display a menu file in a different directory, a sub-directory can be specified before the filename. For example, do display the MAIN.\* menu file in the TEXT\MENU\SIMPLE directory, you would use the following:

MENU "SIMPLE\MAIN"

To display the MAIN.\* menu file in the TEXT\MENU directory, you would use the following:

MENU "MAIN"

You can also change the default menu directory with the SET\_MENU\_DIR function. Using the SET\_MENU\_FILE function, you can override the location and filename for MENUs that are displayed from within core functions of Synchronet.

**NODELIST\_ALL**

-----

This function displays the current status of all nodes.

**NODELIST\_USERS**

-----

This function displays the current status of all nodes that are in use.

**USERLIST\_SUB**

-----

This function displays all users that have access to the current message sub-board.

**USERLIST\_DIR**

-----

This function displays all users that have access to the current file directory.

**USERLIST\_ALL**

-----

This function displays all users that have active accounts on the BBS.

**USERLIST\_LOGONS**

-----

This function displays a list of the users who have logged on today.

**YES\_NO <"cstr">**

**args: question**

-----

This function displays the question argument with automatic colorization followed by the text " (Y/n) ? " and waits for the user to hit 'Y', 'N' or CR. If CR (enter) is hit, then 'Y' is assumed. If the answer to the prompt is 'Y', then the current logic state is set to TRUE, otherwise it is set to FALSE.

Example:

```
YES_NO "Logoff Now"
IF_TRUE
    LOGOFF
END_IF
```

**NO\_YES <"cstr">**

**args: question**

-----

This function is identical to the YES\_NO function with the exception that the string is followed by text " (y/N) ? " and if CR is hit, then 'N' is assumed. If the answer to the prompt is 'N', then the current logic state is set to TRUE, otherwise it is set to FALSE.

Example:

```
NO_YES "Logoff Now"
IF_FALSE
    LOGOFF
END_IF
```

**READ\_SIF <"str">**

**args: file**

-----

This function displays the contents of a SIF questionnaire data file (path and filename specified in the current command string). The data is read from the SIF data file and displayed through the SIF questionnaire file:



"TEXT\file.SIF".

The original SIF data file is created with the CREATE\_SIF function. The SIF file used to create and display the data file do not have to be the same SIF file, but should have the same order and format of data types.

See the Synchronet Sysop Manual for details on the SIF file format.

**SAVELINE**

-----  
This function saves the current line of text on the console to an internal Synchronet buffer for later re-display with the RESTORELINE function.

**RESTORELINE**

-----  
This function re-displays a previously saved line of text to the local and remote console (if applicable).

[Back to Table of Contents](#)

[4.6] - Input Functions

Input functions are used to obtain input data from the user (most often from the user's keyboard).

**INKEY**

-----  
Checks to see if the user has hit a key. If the user has hit a key, the current command key is set to the hit key and the current logic state is set to TRUE, otherwise the logic state is set to FALSE. This function does not wait for a user to hit a key.

**GETKEY**

-----  
Waits for a single key from the user. The key is converted to upper case (if a letter) and becomes the current command key for functions such as CMDKEY, COMPARE\_KEY, etc. The key is not echoed (displayed). Use the PRINTKEY function to display a key accepted with this function.

**GETKEYE**

-----  
This function is identical to the GETKEY function with the exception that "slash commands" are allowed. That is, if the user hits the slash (/) key, then the slash is echoed and the system waits for another key from the user. The second key is converted to upper case (if a letter) and the slash/key combination becomes the current command key for functions such as CMDKEY, COMPARE\_KEY, etc. The second key is not echoed (displayed). Use the PRINTKEY function to display a key accepted with this function.

**GETCMD <"cstr">**  
**args: keys**

-----  
This function is very similar to the GETKEY function, except that the 'keys' argument specifies the only valid key to be accepted. The keys arguments is a C style string, so carriage return (ENTER) is specified with a \r. If, for example, you wanted to allow the user to hit A, B, C or ENTER, then you would use the following:

```
GETCMD "ABC\r"
```

When an allowed key is hit, the key is converted to upper case (if a letter) and echoed (displayed) followed by a CRLF automatically. The key then becomes the current command key for functions such as CMDKEY, COMPARE\_KEY, etc.

Ctrl-C will always be allowed, in which case the command key becomes ^C.

**GETSTR [str\_var] [#] [#] SBBSDEFS.INC**  
**args: string max mode**

-----  
This function ("Get String"), is used to accept a string of characters from the user. The 'max' argument specifies the maximum length allowed for the string (in characters). 'max' should not exceed 79 characters for most uses. This function allows the user to enter any valid ASCII characters to form command words, sentences, etc. Control will not return to the module until the user either hits CR (ENTER), Ctrl-C, or Ctrl-V or is disconnected. The resultant string can then be used in functions such as CMDSTR, COMPARE\_STR, COMPARE, SHIFT\_STR, XTRN\_EXEC, FILE\_DOWNLOAD, etc.

If string is not specified, the current command string is written to. If max is not specified, the value 128 is used.

The valid mode parameters (defined in SBBSDEFS.INC) are:

| Mode        | Description                        |
|-------------|------------------------------------|
| ~~~~ ~~~~~~ |                                    |
| K_UPPER     | Converts all letters to upper case |

```
K_UPRLWR Upper/Lower case automatically
K_NUMBER Allow numbers only
K_WRAP Allows word wrap
K_MSG Allows Ctrl-A codes and Ctrl-G (beep)
K_SPIN Spinning cursor
K_LINE Input line (inverse color)
K_EDIT Edit string passed
K_NOCRLF Don't print CRLF after string input
K_ALPHA Only allow alphabetic characters
K_LOWPRIO low priority input
K_NOEXASC No extended ASCII allowed
K_E71DETECT Detect E-7-1 terminal type
K_AUTODEL Auto-delete text (used with EDIT)
K_NOECHO Don't echo input to local or remote console
```

Multiple mode parameters can be used by ORing them together with the | symbol.  
Example:

```
GETSTR STR 25 K_UPRLWR|K_EDIT|K_AUTODEL
```

```
GETLINE [str_var] [#]
args: string max
-----
```

This function is identical to GETSTR with the exception that an inverse line is displayed to the user (if the user has an ANSI terminal) to show the maximum length of the input string.

```
GETSTRUPR [str_var] [#]
args: string max
-----
```

This function ("Get String Upper") is identical to GETSTR with the exception that the string is automatically converted to uppercase as the user types.

```
GETNAME [str_var] [#]
args: string max
-----
```

This function is identical to GETSTR with the exception that the string is automatically converted to "Capitalized" words (as in a proper noun) as the user types.

**GETFILESPEC**  
-----

This function is specifically used to accept a file specification (name and/or wildcard characters) from the user. It will automatically display a prompt similar to:

```
Filespec [*.]*:
```

before waiting for the user to input the file specification. If the user hits CR, \*.\* is assumed to be the file specification. If a period is not included in the string, a ".\*" is automatically appended (so CR, "\*" and ".\*" all result in the same string: ".\*"). If the user enters "A\*", the string becomes "A\*.\*". If the user enters "A\*." the string remains unchanged. If the user hits Ctrl-C, the current logic state is set to FALSE. Otherwise, the logic state is set to TRUE.

Example:

```
GETFILESPEC
IF_TRUE
    FILE_DOWNLOAD
END_IF
```

**GETLINES**  
-----

This function only has an effect if the user is logged on remotely, has ANSI terminal capability and their screen lines set to "Auto-detect". It sends an ANSI sequence to detect the number of screen lines in the text window of the remote terminal and waits for a response.

```
GETNUM [any_var] <#>
args: dest max
-----
```

This function is similar to GETSTR with the exception that it only accepts numbers as input and automatically stops accepting input if the number being input would exceed max (maximum of 32767). The dest variable (current command string if not specified) will contain the number and the current logic state will be TRUE unless the user hit 'Q', Ctrl-C, or enter instead of a number.

**GET\_TEMPLATE <"str">**  
-----

This function is much like GETLINE except that an input template is passed as an argument to this function. The template is useful when requiring the user to input a string of a fixed format, like a date, for example. The characters in the template have special meaning: N indicates only a number may be used in this position, A indicates only an alphabetic character, and !

indicates any character may be used in that position. Characters other than N, A, or ! are automatically displayed and used in the current command string in those positions. The first character of the template must be either N, A, or !. An example (date format):

```
GET_TEMPLATE "NN/NN/NN"
```

would require the user to input two numbers, then automatically skip the slash character, two more numbers, skip the slash, then two more numbers.

Another example (Canadian postal code):

```
GET_TEMPLATE "ANA NAN"
```

would require the user to input a letter, then a number, another letter, automatically skip the space, another number, another letter, and one more number. No other characters (symbols or otherwise) would be allowed in the string using the above template example.

**CHKSYSPASS**  
-----

This function ("Check System Password") displays an SY: prompt to the user and waits for the user to input the system password (with no echo). If the password is correct, the current logic state is set to TRUE, otherwise it is set to FALSE.

**CREATE\_SIF <"str">**  
**args: file**  
-----

This function is used to present the user with a SIF questionnaire and store their answers in a data file. The str argument is the filename of a .SIF file (without the .SIF extension) located in your Synchronet TEXT directory. The user's answers are stored in a data file whose path and filename are specified in the current command string. The data file should be specified with the user's number to ensure unique data filenames. Example:

```
SETSTR "C:\\SBBS\\USER%4.DAT"  
CREATE_SIF "NEWUSER"
```

This would read the file TEXT\\NEWUSER.SIF and the user would fill-out the questionnaire with all answers being stored in the file C:\\SBBS\\USER####.DAT (where #### would be the user's number padded with zeros to four digits). The questionnaire could then be read back with the answers filled in.  
Example:

```
SETSTR "C:\\SBBS\\USER%4.DAT"  
READ_SIF "NEWUSER"
```

See the Synchronet Sysop Manual for details on the SIF questionnaire format.

[Back to Table of Contents](#)

# [4.7] - Miscellaneous Functions

**ONLINE**  
-----

Assume online execution (default mode), auto-return from module if user is disconnected or logged off.

**OFFLINE**  
-----

Assume offline execution, no carrier and no one online locally. If used (for OFFLINE modules), must be the FIRST executed function of the module.

**LOGIN <"cstr">**  
**args: pwprompt**  
-----

Process login attempt using the current command string as the user name/number and pwprompt for the password prompt. This function may also prompt for the system password with the SY: prompt if the user has level 90 or higher.

**LOGON**  
-----

Logon the currently logged-in user. This function must be called before the LOGIN module terminates if the user is to be allowed online. This function, in turn, executes the LOGON module, if one is configured in SCFG.

**LOGOFF**  
-----

This function prompts the user if they really wish to log off. If they answer Yes, then any logoff events are executed and the LOGOFF menu file is displayed before they are disconnected.

**LOGOFF\_FAST**  
-----

This function immediately drops carrier and logs the user off. It differs

from the HANGUP function in that it performs a SYNC function, then it immediately drops carrier, as opposed to the HANGUP function which does not perform a SYNC operation and has a buffer flush delay before actually dropping carrier.

#### **LOGOUT**

-----

Logout the currently logged-on user. This function is used solely for re-logging-in an already connected user.

#### **NEWUSER**

-----

Begins the new user application process.

#### **SET\_MENU\_DIR <"str">**

**args: dir**

-----

This function sets the default menu directory for the MENU function (normally ..\TEXT\MENU). This function is useful for globally changing the directory where MENU files will be retrieved from (including internal Synchronet functions that can not be modified with a shell). The following is an example of how to use this function along with the MENU function:

```
# The next line would display ..\TEXT\MENU\MAIN.*
MENU "MAIN"

# Change the directory for menus

MENU_DIR "..\TEXT\MYMENUS"

# This next line would now display ..\TEXT\MYMENUS\MAIN.*
MENU "MAIN"
```

#### **SET\_MENU\_FILE <"str">**

**args: path\_and\_filename**

-----

This function is used to OVERRIDE the filename specified for a menu displayed in an internal Synchronet function. Example:

```
SET_MENU_FILE "..\TEXT\MYMENUS\EMAIL.RIP"
MAIL_READ
```

#### **SYNC**

----

This function synchronizes the local and remote displays (waits for the remote terminal to catch up), displays any internode messages waiting for this node, and synchronizes the current node status with the shared node database file. It should be called before all command prompts are displayed.

#### **ASync**

-----

This function is identical to the SYNC function with the exception that any pending output (menu, text file, etc.) can be aborted by hitting any key.

#### **RIOSync**

-----

This function synchronizes the remote console (waits for output buffers to be sent or aborted).

#### **PUT\_NODE**

-----

Force flush (write) of current node information (who's online, etc) to disk.

#### **PAUSE\_RESET**

-----

This function resets the line counter used for the automatic screen pause Feature.

#### **CLEAR\_ABORT**

-----

This function clears the abort (Ctrl-C) status flag.

#### **UNGETKEY**

-----

This function puts the current command key into the input buffer to be the next key retrieved with GETKEY, GETCMD, GETSTR, or similar functions.

#### **UNGETSTR**

-----

This function puts the entire contents of the current command string into the input buffer.

#### **HANGUP**

-----

This function immediately disconnects and logs the user off.

#### **EXEC <"str">**

**args: cmd**

-----

This function calls the operating system to spawn the 'cmd' command line specified. Remote users will not see or be able to interact with the program executed unless the program is aware of the remote user and is specifically designed to be run remotely (i.e. DOORs).

If an internal DOS command is to be executed, a command processor must be loaded. Example:

```
EXEC "COMMAND /C COPY C:\FILES\NEWFILE.TXT C:\DOWNLOADS"
```

Synchronet's command line specifiers can also be used:

```
EXEC "%!dsz portx %u,%i sz file.zip"
```

See the Synchronet sysop manual, appendix A for a list of valid specifiers. When used from a command shell or module, %f will be replaced with the path to the current file directory, and %s will be replaced with the current command string.

**EXEC\_INT <"str">**  
**args: cmd**

-----

This function ("Execute with I/O intercepted") is identical to the EXEC function with the exception that DOS I/O interrupts (screen output and keyboard input) is intercepted and redirected to the COM port if the user is logged on remotely. This allows programs that use standard DOS I/O for screen output to be displayed remotely. Example:

```
EXEC_INT "COMMAND /C DIR C:\FILES\*.TXT"
```

**EXEC\_BIN <"str">**  
**args: name**

-----

This function loads and executes a Baja module. A .BIN file with the filename specified as the 'name' argument must exist in the Synchronet EXEC directory for this function. If for example, you want a shell to load and execute MYMOD.BIN from your EXEC directory, you would add the following to one or more command shells:

```
EXEC_BIN "MYMOD"
```

**EXEC\_XTRN <"str">**  
**args: code**

-----

This function executes the online external program specified by it's internal code with the 'code' argument. If an online external program is not found with a matching internal code or the user does not have access to that program, this function does nothing.

**LOG <"cstr">**  
**args: string**

-----

This function writes the C type 'str' argument to the system's activity log file for the current date if the user is logged on remotely.

**LOGSTR**

-----

This function is identical to the LOG function with the exception that a string argument is not used, but instead the current command string is written to the system's activity log file.

**LOGKEY**

-----

This function writes the current command key to the system's activity log file for the current date if the user is logged on remotely.

**LOGKEY\_COMMA**

-----

This function is identical to the LOGKEY function, with the exception that a comma (,) is prepended to the command key to separate it visually from other logged command keys.

**NODE\_STATUS <#>**      **NODEDEFS.INC**  
**args: status**

-----

This function is used to set the current node status. The valid status parameters (defined in NODEDEFS.INC) are:

| Status       | Description                  |
|--------------|------------------------------|
| ~~~~~        | ~~~~~                        |
| NODE_WFC     | Waiting for caller           |
| NODE_LOGON   | User logging on              |
| NODE_NEWUSER | New user applying for access |
| NODE_INUSE   | User online                  |
| NODE_QUIET   | User online in quiet mode    |
| NODE_OFFLINE | Offline (down)               |
| NODE_NETTING | Networking                   |

NODE\_EVENT\_WAITING Waiting for another node to run timed event  
NODE\_EVENT\_RUNNING Running timed event  
NODE\_EVENT\_LIMBO Waiting for all nodes to be downed

Only one status parameter may be used.

**NODE\_ACTION <#>            NODEDEFS.INC**  
**args: action**

-----  
This function is used to set the current node action. The valid action parameter (defined in NODEDEFS.INC) are:

| Action    | Description                                 |
|-----------|---------------------------------------------|
| ~~~~~     | ~~~~~                                       |
| NODE_MAIN | Main Prompt                                 |
| NODE_RMSG | Reading Messages                            |
| NODE_RMAL | Reading Mail                                |
| NODE_SMAL | Sending Mail                                |
| NODE_RTXT | Reading G-Files                             |
| NODE_RSML | Reading Sent Mail                           |
| NODE_PMSG | Posting Message                             |
| NODE_AMSG | Auto-message                                |
| NODE_XTRN | Running External Program                    |
| NODE_DFLT | Main Defaults Section                       |
| NODE_XFER | File Transfer Prompt                        |
| NODE_DLNG | Downloading File                            |
| NODE_ULNG | Uploading File                              |
| NODE_BXFR | Bidirectional Transfer                      |
| NODE_LFIL | Listing Files                               |
| NODE_LOGN | Logging on                                  |
| NODE_LCHT | In Local Chat with Sysop                    |
| NODE_MCHT | In Multi-Chat with Other Nodes              |
| NODE_GCHT | In Local Chat with Guru                     |
| NODE_CHAT | In Chat Section                             |
| NODE_SYSP | Sysop Activity                              |
| NODE_TQWK | Transferring QWK packet                     |
| NODE_PCHT | In Private Chat                             |
| NODE_PAGE | Paging another node for Private Chat        |
| NODE_RFSD | Retrieving file from seq dev (node.aux=dev) |

Only one action parameter may be used.

**INC\_MAIN\_CMDS**

-----  
This function increments the "main menu command" counter that can be used in ARS comparisons.

**INC\_FILE\_CMDS**

-----  
This function increments the "file menu command" counter that can be used in ARS comparisons.

**COMPARE\_USER\_MISC <#>            USERDEFS.INC**  
**args: attrs**

-----  
This function is used to compare the current user's miscellaneous attributes to the attrs argument specified. If all the attrs specified are ENABLED for the current user, then the current logic state is set to TRUE, otherwise it is set to FALSE. One or more of the following attributes (defined in USERDEFS.INC) can be specified:

| Attribute     | Description                                    |
|---------------|------------------------------------------------|
| ~~~~~         | ~~~~~                                          |
| UM_EXPERT     | Expert menu mode                               |
| UM_DELETED    | Deleted user                                   |
| UM_INACTIVE   | Inactive user                                  |
| UM_AUTOTERM   | Automatic terminal type detection              |
| UM_ANSI       | ANSI terminal                                  |
| UM_COLOR      | Color terminal (ANSI)                          |
| UM_RIP        | RIP compatible terminal                        |
| UM_WIP        | WIP compatible terminal                        |
| UM_NO_EXASCII | Extended ASCII not supported                   |
| UM_UPAUSE     | Automatic screen pause                         |
| UM_SPIN       | Spinning cursor                                |
| UM_ANFSCAN    | Automatic new file scan                        |
| UM_CLRSCRN    | Clear screen between messages                  |
| UM_QUIET      | Default to quiet mode (if 'Q' exempt)          |
| UM_BATCHFLAG  | Batch flagging mode enabled                    |
| UM_NETMAIL    | Forward E-mail to NetMail                      |
| UM_CURSUB     | Remember current sub-board and directory       |
| UM_ASK_NSCAN  | Ask for new-scans at logon                     |
| UM_ASK_SSCAN  | Ask for your message scan at logon             |
| UM_COLDKEYS   | Cold keys (CR required - opposite of Hot Keys) |
| UM_EXTDESC    | Extended file descriptions in listings         |

Multiple attributes can be specified by ORing them together with the OR (|)



symbol. Example:

```
COMPARE_USER_MISC UM_ANSI|UM_COLOR
```

COMPARE\_USER\_CHAT <#>

USERDEFS.INC

args: attrs

-----  
This function is used to compare the current user's chat attributes to the attrs argument specified. If all the attrs specified are ENABLED for the current user, then the current logic state is set to TRUE, otherwise it is set to FALSE. One or more of the following attributes (defined in USERDEFS.INC) can be specified:

| Attribute   | Description                                                 |
|-------------|-------------------------------------------------------------|
| ~~~~~       | ~~~~~                                                       |
| CHAT_ECHO   | Echo users input in multinode chat                          |
| CHAT_ACTION | Action commands enabled in multinode chat                   |
| CHAT_NOPAGE | This user can't be paged by other users                     |
| CHAT_NOACT  | This user will not receive node activity (logon/off) alerts |
| CHAT_SPLITP | This user prefers split-screen private node-to-node chat    |

Multiple attributes may be specified by ORing them together with the OR (|) symbol. Example:

```
COMPARE_USER_CHAT CHAT_NOACT|CHAT_NOPAGE
```

COMPARE\_USER\_QWK <#>

USERDEFS.INC

args: attrs

-----  
This function is used to compare the current user's QWK attributes to the attrs argument specified. If all the attrs specified are ENABLED for the current user, then the current logic state is set to TRUE, otherwise it is set to FALSE. One or more of the following attrs (defined in USERDEFS.INC) can be specified:

| Attribute   | Description                                              |
|-------------|----------------------------------------------------------|
| ~~~~~       | ~~~~~                                                    |
| QWK_FILES   | Include new files list in QWK packet                     |
| QWK_EMAIL   | Include unread e-mail/netmail messages in QWK packet     |
| QWK_ALLMAIL | Include all mail in QWK packet                           |
| QWK_DELMAIL | Automatically delete packed e-mail after download        |
| QWK_BYSELF  | Include messages in QWK packet from self                 |
| QWK_EXPCTLA | Expand Ctrl-A codes in messages to ANSI                  |
| QWK_RETCTLA | Retain (keep) Ctrl-A codes in messages                   |
| QWK_ATTACH  | Include file attachments in QWK packet                   |
| QWK_NOINDEX | Do not include .NDX files in QWK packet                  |
| QWK_TZ      | Include time zone (@TZ) in messages                      |
| QWK_VIA     | Include message path (@VIA) in messages                  |
| QWK_NOCTRL  | Do not include control files (DOOR.ID, CONTROL.DAT, etc) |

Multiple attributes may be specified by ORing them together with the OR (|) symbol. Example:

```
COMAPARE_USER_QWK QWK_TZ|QWK_VIA
```

COMPARE\_NODE\_MISC <#>

NODEDEFS.INC

args: attrs

-----  
This function is used to compare the current node's misc attributes to the attrs argument specified. If all the attrs specified are ENABLED for the current node, then the current logic state is set to TRUE, otherwise it is set to FALSE. The valid attributes (defined in NODEDEFS.INC) are:

| Attribute  | Description                                         |
|------------|-----------------------------------------------------|
| ~~~~~      | ~~~~~                                               |
| NODE_ANON  | Anonymous node                                      |
| NODE_LOCK  | Locked node (sysop's and N exempt users only)       |
| NODE_INTR  | Interrupted node (log user off ASAP)                |
| NODE_MSGW  | Short message waiting for this node                 |
| NODE_POFF  | Not available for paging                            |
| NODE_AOFF  | No node activity alerts                             |
| NODE_UDAT  | User data has been updated by another node          |
| NODE_RRUN  | Re-run this node after user logs off                |
| NODE_EVENT | Run node daily event after user logs off            |
| NODE_DOWN  | Down this node after user logs off                  |
| NODE_NMSG  | Node message waiting for this node                  |
| NODE_EXT   | Extended status information available for this node |

Multiple attributes may be specified by ORing them together with the OR (|) symbol. Example:

```
COMPARE_NODE_MISC NODE_DOWN|NODE_INTR
```

TOGGLE\_USER\_MISC <#>

USERDEFS.INC

args: attrs

-----

This function is used to toggle specific attributes on or off for the current user. The valid attrs for the argument are identical to those used for the COMPARE\_USER\_MISC function.

You cannot specify if you want the attribute turned ON or OFF with this command. Use in combination with the COMPARE\_USER\_MISC function to specifically enable or disable a user's attribute. Example to turn the NETMAIL misc attr OFF for the current user:

```
        !INCLUDE USERDEFS.INC
COMPARE_USER_MISC UM_NETMAIL
        IF_TRUE
        TOGGLE_USER_MISC UM_NETMAIL
        END_IF
```

**TOGGLE\_USER\_CHAT <#>       USERDEFS.INC**  
**args: attrs**

-----  
This function is used to toggle specific chat attributes on or off for the current user. The valid attrs for the argument are identical to those used for the COMPARE\_USER\_CHAT function.

You cannot specify if you want the attribute turned ON or OFF with this command. Use in combination with the COMPARE\_USER\_CHAT function to specifically enable or disable a chat attribute. Example to turn the Split Screen Private Chat (CHAT\_SPLITP) attribute OFF for the current user:

```
COMPARE_USER_CHAT CHAT_SPLITP
        IF_TRUE
        TOGGLE_USER_CHAT CHAT_SPLITP
        END_IF
```

**TOGGLE\_USER\_QWK <#>       USERDEFS.INC**  
**args: attrs**

-----  
This function is used to toggle specific QWK attributes on or off for the current user. The valid attrs for the argument are identical to those used for the COMPARE\_USER\_QWK function.

You cannot specify if you want the attr turned ON or OFF with this command. Use in combination with the COMPARE\_USER\_QWK function to specifically enable or disable a QWK attribute.

**TOGGLE\_NODE\_MISC <#>       NODEDEFS.INC**  
**args: attrs**

-----  
This function is used to toggle the miscellaneous attributes associated with the current node. The valid attrs for the argument are identical to those used for the COMPARE\_NODE\_MISC function.

**TOGGLE\_USER\_FLAG <char> <char>**  
**args: set   flag**

-----  
This function is used to toggle the state of one of the current user's flags. The first argument is a single character indicating the flag set. Valid flag sets are 1, 2, 3, 4, or E for exemption, or R for restriction. The second argument is the actual flag character to toggle (A through Z).

**ADJUST\_USER\_CREDITS <#>**  
**args: adj**

-----  
This function is used to add or subtract credits from the current user. The 'adj' argument must be either a positive or negative number (in Kilobytes). Example, to subtract 100K in credits from the current user:

```
ADJUST_USER_CREDITS -100
```

**ADJUST\_USER\_MINUTES <#>**  
**args: adj**

-----  
This function is used to add or subtract minutes from the current user's minute bank. The 'adj' argument must be either a positive or negative number. Example, to subtract 60 minutes from the current user:

```
ADJUST_USER_MINUTES -60
```

**SET\_USER\_LEVEL <#>**  
**args: level**

-----  
This function changes the current user's security level to the number passed as the argument.

**SET\_USER\_STRING <#>       USERDEFS.INC**  
**args: user\_string**

-----  
This function changes one of the current user's text string fields to the

value of the current command string. The user\_string argument must be one of the following (defined in USERDEFS.INC):

| User_string          | Description                                       |
|----------------------|---------------------------------------------------|
| ~~~~~                | ~~~~~                                             |
| USER_STRING_ALIAS    | User's alias or real name on real name based BBSs |
| USER_STRING_REALNAME | User's real name or company name                  |
| USER_STRING_HANDLE   | User's chat handle                                |
| USER_STRING_COMPUTER | User's computer type description                  |
| USER_STRING_NOTE     | User's note/Caller-ID                             |
| USER_STRING_ADDRESS  | User's street address                             |
| USER_STRING_LOCATION | User's city, state                                |
| USER_STRING_ZIPCODE  | User's zip/postal code                            |
| USER_STRING_PASSWORD | User's password                                   |
| USER_STRING_BIRTHDAY | User's birthday (MM/DD/YY or DD/MM/YY)            |
| USER_STRING_PHONE    | User's phone number                               |
| USER_STRING_MODEM    | User's modem type description                     |
| USER_STRING_COMMENT  | User's secret sysop comment                       |

**USER\_EVENT <#>            SBBSDEFS.INC**  
**args: event\_type**

-----  
This function forces immediate execution of all online events for the specified event\_type. The valid event\_types (defined in SBBSDEFS.INC) are:

| Event_type     | Description                   |
|----------------|-------------------------------|
| ~~~~~          | ~~~~~                         |
| EVENT_LOGON    | Execute during logon sequence |
| EVENT_NEWUSER  | Execute during newuser app.   |
| EVENT_BIRTHDAY | Execute on birthday           |
| EVENT_LOGOFF   | Execute during normal logoff  |

**AUTO\_MESSAGE**  
-----

This function allows the user to read and write to the auto-message. If the user has the 'W' restriction, they will not be allowed to write to the auto-message.

**USER\_DEFAULTS**  
-----

This function takes the user to the default configuration menu where they can modify their default settings (e.g. Terminal type, Command Shell, etc). If the user selects a new command shell while in this function, the new shell is loaded and executed immediately (from the top) instead of resuming with the shell that originally called this function.

**USER\_EDIT**  
-----

This function is used to edit the user information for the user name or number specified in the current command string. Since almost all user information can be changed and other users can be edited from this section, this is normally reserved as a "sysop only" operation.

**TEXT\_FILE\_SECTION**  
-----

This function takes the user to the General Text File section of the system.

**XTRN\_EXEC**  
-----

This function is identical to the EXEC\_XTRN function with the exception that the external program's internal code is contained in the current command string instead of an argument.

**XTRN\_SECTION**  
-----

This function takes the user to the external program section of the system.

**MINUTE\_BANK**  
-----

This function allows the user to convert credits into minutes, or deposit unused free minutes (if allowed) in their minute bank.

**CHANGE\_USER**  
-----

This function prompts the user for the system password and if entered correctly, then prompts them for the user name or number to change into. This is normally reserved as a "sysop only" operation. After changing into a "non-sysop" user, the temp sysop status is set so the user may change back into the original or another user.

**ANSI\_CAPTURE**  
-----

This function enables the capture of ANSI terminal escape sequences when the local capture (Alt-L) is in use. This is normally reserved as a "sysop only" operation.

**FINDUSER**

-----

Scans the user database for the user name or partial name in the current command string. If the name is found, the logic state is set to TRUE. If the name is not found, it is set to FALSE.

#### **SELECT\_SHELL**

-----

This function displays a list of the available command shells and allows the user to choose a command shell. If a shell is not selected, the logic state is set to FALSE.

#### **SET\_SHELL**

-----

This function attempts to set the current user's shell to the internal code contained in the current command string. If the internal code is not valid or the user does not have access to that shell, the logic state is set to FALSE.

#### **SELECT\_EDITOR**

-----

This function displays a list of the available external editors and allows the user to choose an editor. If an editor is not selected, the logic state is set to FALSE.

#### **SET\_EDITOR**

-----

This function attempts to set the current user's editor to the internal code contained in the current command string. If the internal code is not valid or the user does not have access to that external editor, the logic state is set to FALSE.

#### **TRASHCAN <"str"> args: file**

-----

This function opens the file TEXT\file.CAN and searches for the current command string. If the string is found in the file the current logic state is set to TRUE and if the file TEXT\BADfile.MSG exists, it is automatically displayed to the user. If the string is not found, the logic state is set to FALSE.

If a line in the .CAN file ends in a tilde (~), the contents of the line may be found anywhere in the tested string and result in a TRUE logic state.

If a line in the .CAN file ends in a caret (^), the contents of the line must begin the tested string to result in a TRUE logic state.

#### **GETTIMELEFT**

-----

Checks the user's current time left online and disconnects the user (with a message) if they're out of time.

#### **MSWAIT <#> args: num**

-----

Suspends execution for num milliseconds (gives up time-slices).

#### **SEND\_FILE\_VIA <char> <"str" or str\_var> args: prot path\_and\_filename**

-----

Immediately begins a transfer of a file from the local disk (path\_and\_filename) to the remote user using the specified protocol (prot must match the command key of one of the configured protocols in SCFG).

#### **RECEIVE\_FILE\_VIA <char> <"str" or str\_var> args: prot path\_and\_filename**

-----

Immediately begins a transfer of a file from the remote user using the specified protocol (prot must match the command key of one of the configured protocols in SCFG). The path\_and\_filename variable may be set to a directory name for protocols that don't require a specified filename (i.e. Ymodem, Zmodem, etc - NOT Xmodem). [SBBS 2.3]

[Back to Table of Contents](#)

## [4.8] - Mail Functions

The following functions are related to private E-mail.

#### **MAIL\_READ**

-----

This function retrieves and displays mail waiting in the mail-box of the current user, if any.

#### **MAIL\_READ\_SENT**

-----

This function retrieves and displays mail that the current user has sent and is still pending deletion in the mail-box(es) of the recipient(s).

**MAIL\_READ\_ALL**  
-----

This function retrieves and displays all the e-mail on the system. This function is normally reserved as a "sysop only" operation.

**MAIL\_SEND**  
-----

This function attempts to send mail to the user specified by the current command string. If the user name or number specified in the string cannot be located, the operation is aborted and the current logic state is set to FALSE. If the user name or number is located, the current logic state is set to TRUE. Example usage:

```
PRINT "Send mail to who? "  
GETNAME 25  
MAIL_SEND
```

**MAIL\_SEND\_FILE**  
-----

This function is identical to the MAIL\_SEND function with the exception that a file attachment is included with the mail message.

**MAIL\_SEND\_BULK**  
-----

This function is used to send bulk (multiple destination) mail. The function is normally reserved as a "sysop only" operation, but some sysops may want some or all users to have the ability to send bulk mail.

**MAIL\_SEND\_FEEDBACK**  
-----

This function is identical to the MAIL\_SEND function with the exception that "Re: Feedback" is placed at the top of the mail message.

**MAIL\_SEND\_NETMAIL**  
-----

This function prompts the user for a user name and netmail address. If the user just hits CR, the current logic state is set to FALSE, otherwise a NetMail operation is attempted and the logic state is set to TRUE.

**MAIL\_SEND\_NETFILE**  
-----

This function is identical to the MAIL\_SEND\_NETMAIL function with the exception that if the NetMail file attachments are allowed, this function will allow the user to send an attached file to a netmail message (for FidoNet or Internet).

[Back to Table of Contents](#)

# [4.9] - Message Base Functions

The following functions are related to the system's message base.

**MSG\_SET\_AREA**  
-----

This function uses the current command string to search all sub-boards the current user has access to for an internal code matching the current command string. If the internal code is found, that sub-board becomes the current sub-board and the current logic state is set to TRUE, otherwise it is set to FALSE and the current sub-board remains unchanged.

**MSG\_SET\_GROUP**  
-----

This function uses the current command string to search all message groups the current user has access to for a short name matching the current command string. If the short name is found, that group becomes the current group and the current logic state is set to TRUE, otherwise it is set to FALSE and the current group remains unchanged.

**MSG\_SELECT\_AREA**  
-----

This function prompts the user to select a message group (if the user has access to more than one) and sub-board. If the user aborts the selection, the current logic state is set to FALSE and the current sub-board is unchanged. Otherwise, the logic state is set to TRUE and the selected sub-board becomes the current sub-board.

**MSG\_SHOW\_GROUPS**  
-----

This function displays a list of message groups the current user has access to.

**MSG\_SHOW\_SUBBOARDS**  
-----

This function displays a list of sub-boards within the current message group that the user has access to.

**MSG\_GROUP\_UP**

-----

This function moves the current message group up by one. If already at the highest group, then it wraps to the first group.

**MSG\_GROUP\_DOWN**

-----

This function is the opposite of the MSG\_GROUP\_UP function.

**MSG\_SUBBOARD\_UP**

-----

This function moves the current message sub-board up by one. If already at the highest sub-board in the current group, it wraps to the first sub-board in the current group.

**MSG\_SUBBOARD\_DOWN**

-----

This function is the opposite of the MSG\_SUBBOARD\_UP function.

**MSG\_GET\_SUB\_NUM**

-----

This function accepts a number from the user to change the current sub-board.

**MSG\_GET\_GRP\_NUM**

-----

This function accepts a number from the user to change the current message group.

**MSG\_READ**

-----

This function is used to read messages in the current sub-board. If the read is aborted, the current logic state is set to FALSE, otherwise it is set to TRUE.

**MSG\_POST**

-----

This function attempts to post a message on the current sub-board. If the user does not have posting access on the current sub-board or the post is aborted, the current logic state is set to FALSE, otherwise it is set to TRUE.

**MSG\_QWK**

-----

This function puts the user in the QWK message packet section.

**MSG\_PTRS\_CFG**

-----

This function allows the user to manipulate their new-message scan pointers.

**MSG\_PTRS\_REINIT**

-----

This function resets the user's new-message scan pointers to their value at logon.

**MSG\_NEW\_SCAN\_CFG**

-----

This function allows the user to manipulate their new-message scan configuration.

**MSG\_NEW\_SCAN**

-----

This function performs a new message scan on either the current sub-board, current message group, or all sub-boards.

**MSG\_NEW\_SCAN\_ALL**

-----

This function performs a new message scan on all sub-boards.

**MSG\_NEW\_SCAN\_SUB**

-----

This function performs a new message scan on the current sub-board. If the new-scan is aborted, the current logic state is set to FALSE.

**MSG\_CONT\_SCAN**

-----

This function performs a continuous new message scan on either the current sub-board, current message group, or all sub-boards.

**MSG\_CONT\_SCAN\_ALL**

-----

This function performs a continuous new message scan on all sub-boards.

**MSG\_BROWSE\_SCAN**

-----

This function is identical to the MSG\_NEW\_SCAN function, with the exception that if there are no new messages in a sub-board, the last message posted will be displayed.

**MSG\_BROWSE\_SCAN\_ALL**

-----



This function performs a browse scan on all sub-boards.

**MSG\_FIND\_TEXT**  
-----

This function prompts for a search string from the user and searches the current sub-board, current group, or all sub-boards for the text.

**MSG\_FIND\_TEXT\_ALL**  
-----

This function performs a text search on all sub-boards.

**MSG\_YOUR\_SCAN\_CFG**  
-----

This function allows the user to configure their un-read personal message scan.

**MSG\_YOUR\_SCAN**  
-----

This function performs an un-read personal message scan on the current sub-board, current message group, or all sub-boards.

**MSG\_YOUR\_SCAN\_ALL**  
-----

This function performs an un-read personal message scan on all sub-boards.

[Back to Table of Contents](#)

## [4.10] - File Base Functions

**FILE\_SET\_AREA**  
-----

This function uses the current command string to search all directories the current user has access to for an internal code matching the current command string. If the internal code is found, that directory becomes the current directory and the current logic state is set to TRUE, otherwise it is set to FALSE and the current directory remains unchanged.

**FILE\_SET\_LIBRARY**  
-----

This function uses the current command string to search all libraries the current user has access to for a short name matching the current command string. If the short name is found, that library becomes the current library and the current logic state is set to TRUE, otherwise it is set to FALSE and the current library remains unchanged.

**FILE\_SELECT\_AREA**  
-----

This function prompts the user to select a file library (if the user has access to more than one) and directory. If the user aborts the selection, the current logic state is set to FALSE the current directory is unchanged. Otherwise, the logic state is set to TRUE and the selected directory becomes the current directory.

**FILE\_SHOW\_LIBRARIES**  
-----

This function displays a list of all file libraries the current user has access to.

**FILE\_SHOW\_DIRECTORIES**  
-----

This function displays a list of all directories within the current library that the user has access to.

**FILE\_LIBRARY\_UP**  
-----

This function increments the current file library. If already at the last library, it will wrap to the first library.

**FILE\_LIBRARY\_DOWN**  
-----

This function is the opposite of the FILE\_LIBRARY\_UP function.

**FILE\_DIRECTORY\_UP**  
-----

This function increments the current file directory. If already at the last directory in the current library, it will wrap to the first directory of the library.

**FILE\_DIRECTORY\_DOWN**  
-----

This function is the opposite of the FILE\_DIRECTORY\_UP function.

**FILE\_GET\_DIR\_NUM**  
-----

This function waits for the user to enter a number to be used to change the current file directory.

**FILE\_GET\_LIB\_NUM**  
-----

This function waits for the user to enter a number to be used to change the current file library.

**FILE\_LIST**  
-----

This function displays the contents matching the filespec contained in the current command string of the current file directory. If the listing is aborted, the current logic state is set to FALSE, otherwise it is set to TRUE. Example:

```
GETFILESPEC
IF_TRUE
    FILE_LIST
END_IF
```

**FILE\_LIST\_EXTENDED**  
-----

This function displays the extended information about the files in the current directory that match the filespec contained in the current command string.

**FILE\_VIEW**  
-----

This function views the contents of the files that match the filespec in the current command string.

**FILE\_UPLOAD**  
-----

This function attempts to upload to the current file directory. If the user cannot upload for some reason, the current logic state is set to FALSE. Otherwise, it is set to TRUE.

**FILE\_UPLOAD\_USER**  
-----

This function allows a user to upload a file specifically to another user, provided the sysop has created a "User" file transfer directory for user-to-user transfers. If the user cannot upload to the user directory or one doesn't exist, the current logic state is set to FALSE, otherwise it set to TRUE.

**FILE\_UPLOAD\_SYSOP**  
-----

This function allows the user to upload a file to the invisible sysop directory. If the user does not have upload access to the sysop directory or a sysop directory has not been created, the current logic state is set to FALSE, otherwise, it is set to TRUE.

**FILE\_DOWNLOAD**  
-----

This function attempts to download all files matching the filespec contained in the current command string.

**FILE\_DOWNLOAD\_USER**  
-----

This function attempts to download any user-to-user files pending for the current user. If there are no files waiting for the user or the user does not have download access to the user directory, the current logic state is set to FALSE, otherwise it is set to TRUE.

**FILE\_DOWNLOAD\_BATCH**  
-----

This function attempts to download any files in the batch download queue. If there are no files in the queue or the user answers No to the question if they wish to download the files now, then the current logic state is set to FALSE. If there are files in the queue and the user answers Yes to the question if they wish to download the files, then the logic state is set to TRUE.

**FILE\_REMOVE**  
-----

This function attempts to remove any files the user has uploaded (unless R exempt) that match the filespec contained in the current command string.

**FILE\_BATCH\_ADD**  
-----

This function attempts to add the filename contained in the current command string to the batch download queue. If the filename is not found, the current logic state is set to FALSE.

**FILE\_BATCH\_ADD\_LIST**  
-----

This function opens the list file specified in the current command string and attempts to add each filename (one filename per line) to the current user's batch download queue. After the list file is read, it is automatically deleted.

**FILE\_BATCH\_CLEAR**  
-----

This function clears the current batch download queue.

#### **FILE\_BATCH\_SECTION**

-----

This function takes the user to the Batch/Bi-dir transfer menu where they can edit the batch queues and initiate batch or bi-directional file transfers.

#### **FILE\_TEMP\_SECTION**

-----

This function takes the user to the temporary archive menu where they can extract archives, create file listings, and compress temporary files for download.

#### **FILE\_NEW\_SCAN**

-----

This function scans the current directory, current library, or all directories for new files.

#### **FILE\_NEW\_SCAN\_ALL**

-----

This function scans all directories for new files.

#### **FILE\_FIND\_TEXT**

-----

This function prompts the user to enter a search string which is used in a text search of all file descriptions in the current directory, current library, or all directories.

#### **FILE\_FIND\_TEXT\_ALL**

-----

This function prompts the user for a search string to use in a text search of file descriptions in all directories.

#### **FILE\_FIND\_NAME**

-----

This function prompts the user for a filespec and searches the current directory, current library, or all directories for files matching the specification.

#### **FILE\_FIND\_NAME\_ALL**

-----

This function prompts the user for a filespec and searches all directories for files matching that specification.

#### **FILE\_PTRS\_CFG**

-----

This function allows the user to change their new file scan date/time. If the pointer is changed, the current logic state is set to TRUE, otherwise it is set to FALSE.

#### **FILE\_SET\_ALT\_PATH**

-----

This function sets the current upload path to the alternate path number contained in the current command string. This function is normally reserved as a "sysop only" operation, used to upload files on CD-ROM in most cases.

#### **FILE\_RESORT\_DIRECTORY**

-----

This function is used to resort (if directory sort order or type has changed) and compress (if any deleted file slots exist) the current directory, current library or all directories. All other nodes must be inactive for this function to work. The current command string must be set (with SETSTR, GETSTR, or similar function) to "ALL" (resort all directories), "LIB" (resort current library), or "" (current directory only).

#### **FILE\_GET**

-----

This function is used to remotely download a file that is not in the file database. This function is normally reserved as a "sysop only" operation. The path and filename to be downloaded are contained in the current command string.

#### **FILE\_SEND**

-----

Same as FILE\_GET, but doesn't require system password.

#### **FILE\_PUT**

-----

This function is used to remotely upload a file to the system's hard disk, but not place the file in the file database. This is normally reserved as a "sysop only" operation. The path and filename to be uploaded are contained in the current command string.

#### **FILE\_UPLOAD\_BULK**

-----

This function is used to do a local bulk upload of files that are in the physical DOS directory, but not already in the file database. This is normally

reserved as a "sysop only" operation. If the current command string is set to "ALL", then all directories will be scanned, "LIB" then only the current library, and "" indicates only the current directory.

**FILE\_FIND\_OLD**  
-----

This function is used to search directories for files that have not be downloaded since the current new file scan date (set with FILE\_PTRS\_CFG). For each file that is found, the user is prompted if they wish to remove the file. This is normally reserved as a "sysop only" operation. If the current string is set to "ALL", then all directories will be scanned "LIB" indicates only the current library, and "" indicates only the current directory.

**FILE\_FIND\_OPEN**  
-----

This function searches for file records that are currently open. This function should not be used with other nodes in use. For each open file that is found, the user is prompted if they wish to close the file record. As with the FILE\_FIND\_OLD function, the current command string must contain "ALL", "LIB", or "" to specify the search range.

**FILE\_FIND\_OFFLINE**  
-----

This function searches for files that are in the database, but not in the physical DOS directory. For each missing file that is found, the user is prompted if they wish to move or remove the file. This normally reserved as a "sysop only" operation. As with the FILE\_FIND\_OLD function, the current command string must contain "ALL", "LIB", or "" to specify the search range.

**FILE\_FIND\_OLD\_UPLOADS**  
-----

This function searches for files that were uploaded before the current new scan date (set with FILE\_PTRS\_CFG). For each file that is found, the user is prompted if they wish to remove the file. This is normally reserved as a "sysop only" operation. As with the FILE\_FIND\_OLD function, the current command string must contain "ALL", "LIB", or "" to specify the search range.

[Back to Table of Contents](#)

## [4.11] - Chat Functions

The following functions are chat related in nature.

**PAGE\_SYSOP**  
-----

This function is used to page the sysop at the local console.

**PAGE\_GURU**  
-----

This function initiates chat with the guru who's internal codes is in the current command string. Example (using SMARTGUY as the internal code):

```
SETSTR "SMARTGUY"
PAGE_GURU
```

**PRIVATE\_CHAT**  
-----

This function is used to initiate private node-to-node chat.

**PRIVATE\_MESSAGE**  
-----

This function is used for private internode messages, telegrams, or chat. Same as the Ctrl-P initiated function.

**CHAT\_SECTION**  
-----

This function takes the user immediately to the chat menu where they can join multinode chat and perform various other chat related operations.

[Back to Table of Contents](#)

## [4.12] - Information Functions

**INFO\_SYSTEM**  
-----

This function displays various system information.

**INFO\_SUBBOARD**  
-----

This function displays information about the current message sub-board.

**INFO\_DIRECTORY**  
-----

This function displays information about the current file directory.

**INFO\_USER**  
-----

This function displays information about the current user.

**INFO\_VERSION**  
-----

This function displays version information about the system software.

**INFO\_XFER\_POLICY**  
-----

This function displays the system's transfer policy.

**GURU\_LOG**  
-----

This function displays the GURU.LOG file from the DATA directory and prompts the user if they wish to delete the log. This is normally reserved as a "sysop only" operation.

**ERROR\_LOG**  
-----

This function displays the ERROR.LOG file from the DATA directory and prompts the user if they wish to delete the log and/or clear the error counters for the nodes. This is normally reserved as a "sysop only" operation.

**SYSTEM\_LOG**  
-----

This function displays the system's detailed activity log for the current date. This is normally reserved as a "sysop only" operation.

**SYSTEM\_YLOG**  
-----

This function displays the system's detailed activity log for the previous day. This is normally reserved as a "sysop only" operation.

**SYSTEM\_STATS**  
-----

This function displays cumulative system usage statistics. This is normally reserved as a "sysop only" operation.

**NODE\_STATS**  
-----

This function displays cumulative usage statistics for the current node. This is normally reserved as a "sysop only" operation.

**SHOW\_MEM**  
-----

This function displays the amount of free memory available. This is normally reserved as a "sysop only" operation.

[Back to Table of Contents](#)

# [4.13] - File I/O Functions

These functions allow Baja modules to read (input) from and write (output) to disk files. If any of the following functions fail, the current logic state will be set to FALSE and the system variable \_ERRNO will contain the reason for the failure. All file I/O functions require SBBS v2.2 or later.

**FOPEN <int\_var> <#> <"str" or str\_var> FILE\_IO.INC**  
**args: handle access path\_and\_filename**  
-----

This function opens or creates a disk file (path\_and\_filename) for read and/or write access. This function is the Baja equivalent of the standard C fopen() function. No file I/O operations can be performed on a file until it has been opened with this function. The 'handle' argument must be a defined integer variable name. The 'access' argument is the requested open access. The access flags (defined in FILE\_IO.INC) are:

| Access Flag | Description                                        |
|-------------|----------------------------------------------------|
| O_RDONLY    | Read Only                                          |
| O_WRONLY    | Write Only                                         |
| O_RDWR      | Read and write                                     |
| O_CREAT     | Create (create if doesn't exist)                   |
| O_APPEND    | Append (writes to end of file)                     |
| O_TRUNC     | Truncate (truncates file to 0 bytes automatically) |
| O_EXCL      | Exclusive (only open/create if file doesn't exist) |
| O_DENYNONE  | Deny None (shareable, for use with record locking) |

To specify multiple access flags, place an OR symbol (|) between each. Example:

O\_WRONLY|O\_TRUNC

If the file does not exist it will be created if the O\_CREAT access was specified. If the O\_CREAT and O\_EXCL access flags are set then the open will fail if the file already exists.

Example:

```
!INCLUDE FILE_IO.INC

INT file
FOPEN file O_RDWR|O_CREAT "%jFILENAME.EXT"
```

Would open the file FILENAME.EXT in the BBS data directory (%j is the Synchronet command line % specifier for the data directory) for reading and writing, creating it if it didn't already exist. All later accesses to the file must use the same integer variable ("file" in the above example). If the open is successful, the current logic state is set to TRUE, otherwise it is set to FALSE.

**FCLOSE <int\_var>**  
**args: handle**

-----  
All files opened with the FOPEN function remain open until either this function (FCLOSE) is used to close the file (using the same unmodified 'handle') or the current shell or module is terminated. This function is the Baja equivalent of the standard C fclose() function. A maximum of 10 files can be concurrently opened by a shell or module at any given time.

**FREAD <int\_var> <any\_var> [int\_var or #]**  
**args: handle dest length**

-----  
This function is used to read from a disk file (specified by 'handle') into a string or integer variable (specified by 'dest') from the current file position. This function is the Baja equivalent of the standard C fread() function.

The optional 'length' argument specifies the number of bytes to read. It must be between 1 and 4 if the dest argument is an integer variable and between 1 and 1024 if the dest argument is a string variable. If 'length' is not specified, then 4 bytes is assumed for integer variables or the current string length for string variables (or 128 if an uninitialized or 0 length string variable is specified). The current file position will be automatically moved forward by the number of bytes successfully read from the file.

If the FSET\_ETX function has been used to set the end-of-text character to something other than 0 and the dest argument is a string variable, the string will be terminated at the first end-of-text character read from the file from the current position.

It is suggested that a 'length' argument always be specified for string variable destinations, though it is not usually necessary for integer variable destinations.

**FWRITE <int\_var> <any\_var> [int\_var or #]**  
**args: handle source length**

-----  
This function is used to write an integer or string variable ('source') to a disk file ('handle') at the current file position. This function is the Baja equivalent of the standard C fwrite() function. If the optional 'length' argument is specified, it will determine the number of bytes written to the file. If 'source' is a string variable and the number of bytes to be written exceeds the string length, the current end-of-text character will be used to fill the extra bytes. If 'length' is not specified, 4 bytes will be used for integer 'source' variables and the current string length will be used for string 'source' variables. The current file position will be automatically moved forward by the number of bytes successfully written to the file.

**FFLUSH <int\_var>**  
**args: handle**

-----  
This function forces any pending writes to the disk file ('handle'). All files opened by Baja modules use buffered I/O for performance. Closing an open file flushes any pending output as well. You may wish to use this function in multinode applications where it is important that other nodes have immediate access to the data written to a file while it is still open. This function is the Baja equivalent of the standard C fflush() function. [SBBS 2.3]

**FGET\_LENGTH <int\_var> <int\_var>**  
**args: handle length**

-----  
This function gets the current length (in bytes) of the file pointed to by 'handle', and writes the value to the integer variable 'length'. This function is the Baja equivalent of the standard C filelength() function.

**FSET\_LENGTH <int\_var> <int\_var>**  
**args: handle length**

-----  
This function changes the length of the file pointed to by 'handle' to the value contained in the integer variable 'length'. If the file is shortened, any bytes beyond the specified length are lost. If the file is extended, the unused bytes will be automatically padded with 0. This function is the Baja



equivalent of the standard C chsize() function.

**FGET\_TIME <int\_var> <int\_var>**  
**args: handle time**

-----  
This function gets the current date/time stamp of the file pointed to by 'handle' and writes this value into the 'time' variable. This function is the Baja equivalent of the standard C getftime() function.

**FSET\_TIME <int\_var> <int\_var>**  
**args: handle time**

-----  
This function sets the current date/time stamp of the file pointed to by 'handle' to the value contained in the 'time' variable. This function is the Baja equivalent of the standard C setftime() function.

**FEOF <int\_var>**  
**args: handle**

-----  
This function checks the file specified by 'handle' to see if the current file position is at the end of the file. If it is currently at the end of the file, the current logic state is set to TRUE. If it is not at the end of the file or the file 'handle' is not currently open, the logic state is set to FALSE. This function is the Baja equivalent of the standard C feof() function.

**FGET\_POS <int\_var> <int\_var>**  
**args: handle offset**

-----  
This function gets the current file position (in bytes from the beginning of the file) for the file specified by 'handle' and writes the value into the integer variable 'offset'. This function is the Baja equivalent of the standard C ftell() function.

**FSET\_POS <int\_var> <int\_var or #> [#] FILE\_IO.INC**  
**args: handle offset whence**

-----  
This function is used to reposition (seek) within the file specified by 'handle' to the offset specified by 'offset' from 'whence', if specified. If 'whence' is not specified the 'offset' will be considered from the beginning of the file. This function is the Baja equivalent of the standard C fseek() function. The valid values of whence (defined in FILE\_IO.INC) are:

Whence Description  
~~~~~ ~~~~~~  
SEEK\_SET From the beginning of the file  
SEEK\_CUR From current file position  
SEEK\_END From end of file

The 'offset' (not offspring) variable can be either an integer variable or an integer constant. It may be negative (to move the file position backwards), positive (forwards), or 0 (to specify no offset from 'whence').

Example:

```
!INCLUDE FILE_IO.INC

# Open file
INT file
FOPEN file O_RDWR|O_CREAT "TEXT.TXT"
IF_FALSE
    # Failed!
    RETURN
END_IF

# Go to end of file
FSET_POS file 0 SEEK_END

# Go to beginning of file
FSET_POS file 0

# Go to offset 256 in file
FSET_POS file 256

# Seek forward 100 bytes
FSET_POS file 100 SEEK_CUR

# Seek backward 50 bytes
FSET_POS file -50 SEEK_CUR

FCLOSE file
```

**FLOCK <int\_var> <int\_var or #>**  
**args: handle length**

-----  
This function is used to lock a file record of 'length' bytes from the current file position of the file pointed to by 'handle'. If the record cannot after 10 retries (one second apart), the current logic state will be set to FALSE. All records that are locked should be subsequently unlocked before the

file is closed or any user interaction or lengthy processing is begun. This function is the Baja equivalent of the C locking(int,LK\_LOCK,long) function.

**FUNLOCK <int\_var> <int\_var or #>**  
**args: handle      length**

-----  
This function is used to unlock a file record previously locked by the current module. The current file position of 'handle' must be the start of the record and the 'length' must be the same length as originally specified to the FLOCK function. This function is the Baja equivalent of the C locking(int,LK\_UNLCK,long) function.

**FPRINTF <int\_var> <"cstr"> [any\_var] [...]**  
**args: handle      format    variables**

-----  
This function is very similar to the PRINTF and SPRINTF functions. The exception is that the output of the formatted string is written to the current file position of the file specified by 'handle'. This function is the Baja equivalent of the standard C fprintf() function.

Example:

```
# Variables
INT file length time
# Open for append
FOPEN file O_WRONLY|O_CREAT|O_APPEND "TEXT.TXT"
IF_FALSE
    # Failed!
    RETURN
END_IF

FGET_LENGTH file length
FGET_TIME file time
TIME_STR str time
FPRINTF file "Length: %ld    Time: %s\r\n" length str
FCLOSE file
```

**FREAD\_LINE <int\_var> <any\_var>**  
**args: handle      dest**

-----  
Reads a single line from the open file specified by 'handle' into the destination variable specified by 'dest'. If the destination variable is a string variable, the terminating carriage return and line feed (if present in the file) will be included in the string variable. You can use the TRUNCSP function to remove them if needed.

If the read is successful, the current logic state is set to TRUE, otherwise it is set to FALSE. This function is the Baja equivalent of the standard C fgets() function.

**FSET\_ETX <#>**  
**args: etx**

-----  
This function is used to set the end-of-text character for the current shell or module. The default end-of-text character is 0. It is used by the FREAD and FWRITE functions to pad/terminate string variables when read from and writing to disk files. The 'etx' argument must be a number from 0 to 255. Many of SynchroNet's data files (USER.DAT, for example) use 3 (appears as ^C or a heart) for an end-of-text character. If you intend to read or write from any files that use a non-zero end-of-text character, you'll want to use this function before using FREAD or FWRITE.

[Back to Table of Contents](#)

## [4.14] - File System Functions

These functions allow Baja modules to perform operations on unopened files.

**CHKFILE <"str" or str\_var>**  
**args: path\_and\_filename**

-----  
This function checks for the existence of the 'path\_and\_filename' specified. The path\_and\_filename should contain path and possibly drive specifications (i.e. CHKFILE C:\MYDIR\MYFILE.TXT). If the file exists the current logic state is set to TRUE, otherwise it is set to FALSE.

**REMOVE\_FILE <str\_var>**  
**args: path\_and\_filename**

-----  
This function attempts to remove the specified path\_and\_filename. If the removal was successful, the current logic state is set to TRUE, otherwise it is set to FALSE. This function is the Baja equivalent of the standard C unlink() and remove() functions.

**RENAME\_FILE <str\_var> <str\_var>**  
**args: source      dest**

-----  
This function attempts to rename the file specified as 'source' to the name specified in 'dest'. If a different path is specified for 'dest', the drive letter must be the same. If the renaming is successful, the current logic state is set to TRUE, otherwise it is set to FALSE. This function is the equivalent of the standard C rename() function.

**COPY\_FILE <str\_var> <str\_var>**  
    **args: source    dest**

-----  
This function attempts to copy the file specified as 'source' to the path AND filename specified in 'dest'. If the copy is successful, the current logic state is set to TRUE, otherwise it is set to FALSE.

**MOVE\_FILE <str\_var> <str\_var>**  
    **args: source    dest**

-----  
This function attempts to move the file specified as 'source' to the path AND filename specified in 'dest'. If the source and dest paths are on the same drive, this function is the same as a RENAME\_FILE. If the source and dest paths are on separate drives, then the this is the same as a COPY\_FILE with the exception that the source file is automatically deleted. If the move is successful, the current logic state is set to TRUE, otherwise it is set to FALSE.

**GET\_FILE\_ATTRIB <int\_var> <str\_var>    DIR\_ATTR.INC**  
    **args: dest    file**

-----  
This function retrieves the directory attributes for the file and writes them into the dest variable. This function is the Baja equivalent of the C \_chmod(char \*,0) function.

**SET\_FILE\_ATTRIB <int\_var> <str\_var>    DIR\_ATTR.INC**  
    **args: attrib    file**

-----  
This function sets the directory attributes for the specified file according to the attrib flags. This function is the Baja equivalent of the C \_chmod(char \*,1) function. Valid attrib flags (defined in DIR\_ATTR.INC) are:

Flag    Description  
~~~~~  ~~~~~  
FA\_RDONLY Read Only  
FA\_HIDDEN Hidden  
FA\_SYSTEM System  
FA\_LABEL Volume Label  
FA\_DIREC Directory (or subdirectory)  
FA\_ARCH Archive

Multiple flags may be specified by ORing the flags together with the OR (|) symbol. Example:

```
!INCLUDE DIR_ATTR.INC
INT I
SETSTR "MYFILE.TXT"
SET I FA_HIDDEN|FA_SYSTEM
SET_FILE_ATTRIB I STR
```

# End of DIR\_ATTR.INC

**GET\_FILE\_TIME <int\_var> <str\_var>**  
    **args: dest    file**

-----  
Retrieves the current date/time stamp of the specified file and writes it into the specified dest variable. If the date/time stamp could not be read (file doesn't exist perhaps), the value -1 is written to the dest variable.

**GET\_FILE\_LENGTH <int\_var> <str\_var>**  
    **args: dest    file**

-----  
Retrieves the current length of the specified file and writes it into the specified dest variable. If the file doesn't exist, the value -1 is written to the dest variable.

## [4.15] - Directory System Functions

These functions allow Baja modules to perform operations on directories and subdirectories of the local file system.

**MAKE\_DIR <str\_var>**  
    **args: dir**

-----  
This function attempts to create the directory specified as dir. If the directory is successfully created, the current logic state is set to TRUE, otherwise it is set to FALSE and the system variable \_ERRNO will contain the reason. This function is the Baja equivalent of the standard C mkdir() function.

**CHANGE\_DIR <str\_var>**  
**args: dir**

-----  
This function attempts to change the current directory to the directory specified in dir. Since Synchronet assumes the current directory is the current NODE directory at all times, this is a very dangerous function and should not be used unless absolutely necessary. If the directory is changed successfully, the current logic state is set to TRUE, otherwise it is set to FALSE and the system variable \_ERRNO will contain the reason. This function is the Baja equivalent of the standard C chdir() function.

**REMOVE\_DIR <str\_var>**  
**args: dir**

-----  
This function attempts to remove the directory specified as dir. If the removal is unsuccessful (files in the directory, directory doesn't exist, etc.) the current logic state will be set to FALSE and the system variable \_ERRNO will contain the reason. This function is the Baja equivalent of the standard C rmdir() function.

**OPEN\_DIR <int\_var> <str\_var>**  
**args: handle dir**

-----  
This function attempts to open the directory specified as dir (should end in a '.'). If the directory is successfully opened, the handle is set to the directory handle and the current logic state is set to TRUE, otherwise the logic state is set to FALSE. This function is the Baja equivalent of the standard C opendir() function.

**READ\_DIR <int\_var> <str\_var>**  
**args: handle filename**

-----  
This function reads the next file name from the open directory handle and puts it into the string variable specified as filename. If the read is unsuccessful (no more files in directory) the current logic state is set to FALSE and the system variable \_ERRNO will contain the reason. This function is the Baja equivalent of the standard C readdir() function.

**REWIND\_DIR <int\_var>**  
**args: handle**

-----  
This function rewinds the open directory handle to the first directory entry (file name). This function is the Baja equivalent of the standard C rewinddir() function.

**CLOSE\_DIR <int\_var>**  
**args: handle**

-----  
This function closes a previously opened directory handle. This function is the Baja equivalent of the standard C closedir() function.

[Back to Table of Contents](#)

## [5.1] - System Variables

System variables are global variables that are initialized and/or used by Synchronet during operation. They should not be defined in your Baja module. You must declare these variables by including (with !INCLUDE) SBBSDEFS.INC or by declaring (with !GLOBAL) the individual variable names you wish to use.

These may be used anywhere a user-defined global variable may be used, with the exception that many variables may not be written to (modified directly by a shell or module). In some cases, there are specific Baja functions for modifying otherwise "unmodifiable" system variables. These "Read Only" system variables are indicated here with an asterisk (\*).

### String Variables

|                  |                                                 |
|------------------|-------------------------------------------------|
| -----            |                                                 |
| _USERON.ALIAS    | *Alias/Real Name of user online                 |
| _USERON.NAME     | *Real/Company Name of user online               |
| _USERON.HANDLE   | *Chat Handle of user online                     |
| _USERON.COMP     | *Computer description for user online           |
| _USERON.NOTE     | *Note for user online                           |
| _USERON.ADDRESS  | *Address of user online                         |
| _USERON.LOCATION | *Location of user online                        |
| _USERON.ZIPCODE  | *Zip/Postal Code of user online                 |
| _USERON.PASS     | *Password of user online                        |
| _USERON.BIRTH    | *Birthdate of user online                       |
| _USERON.PHONE    | *Phone Number of user online                    |
| _USERON.MODEM    | *Modem/Connection description of user online    |
| _USERON.NETMAIL  | *NetMail Forwarding Address for user online     |
| _USERON.TMPEXT   | *Temp File Extension (i.e. ZIP) for user online |
| _USERON.COMMENT  | *Sysop Comment for user online                  |
| CONNECTION       | Current connection description                  |

\_CAP\_FNAME Capture file name  
\_CID Caller-ID information  
\_COMSPEC \*Path to operating system command interpreter

## Integer Variables

-----  
\_USERON.NUMBER \*Number of user online  
\_USERON.ULS \*Total number of files uploaded by user online  
\_USERON.DLS \*Total number of files downloaded by user online  
\_USERON.POSTS \*Total number of posts sent by user online  
\_USERON.EMAILS \*Total number of e-mails sent by user online  
\_USERON.FBACKS \*Total number of feedbacks sent by user online  
\_USERON.ETODAY \*Total number of e-mails sent today by user online  
\_USERON.PTODAY \*Total number of posts sent today by user online  
\_USERON.TIMEON \*Total minutes used (excluding this call) by user online  
\_USERON.TEXTRA \*Extra time obtained by uploads during previous calls  
\_USERON.LOGONS \*Total number of logons by user online  
\_USERON.TTODAY \*Total time used today (previous calls)  
\_USERON.TLAST \*Total time used during last call  
\_USERON.LTODAY \*Total number of logons today  
\_USERON.XEDIT \*External editor used by user online  
\_USERON.SHELL \*Command shell used by user online  
\_USERON.LEVEL \*Security level of user online  
\_USERON.SEX \*Sex (gender) of user online  
\_USERON.ROWS \*Number of screen lines for user online  
\_USERON.PROT \*Default download protocol of user online  
\_USERON.LEECH \*Total leech downloads detected for user online  
\_USERON.MISC \*Miscellaneous attributes for user online  
\_USERON.QWK \*QWK settings for user online  
\_USERON.CHAT \*Chat settings for user online  
\_USERON.FLAGS1 \*Flag set #1 for user online  
\_USERON.FLAGS2 \*Flag set #2 for user online  
\_USERON.FLAGS3 \*Flag set #3 for user online  
\_USERON.FLAGS4 \*Flag set #4 for user online  
\_USERON.EXEMPT \*Exemption flags for user online  
\_USERON.REST \*Restriction flags for user online  
\_USERON.ULB \*Total bytes uploaded by user online  
\_USERON.DLB \*Total bytes downloaded by user online  
\_USERON.CDT \*Total credits for user online  
\_USERON.MIN \*Current minutes in minute bank for user online  
\_USERON.FREECDT \*Free credits left today for user online  
\_USERON.FIRSTON \*Date/time of first call for user online  
\_USERON.LASTON \*Date/time of last call for user online  
\_USERON.EXPIRE \*Date/time of expiration for user online  
\_USERON.PWMOD \*Date/time of last password modification for user online  
\_USERON.NS\_TIME \*Date/time of last new-file scan for user online  
\_CUR\_RATE \*Current DCE rate  
\_CUR\_CPS \*Estimated CPS rate of current connection  
\_DTE\_RATE Current DTE rate  
\_LNCNTR Line counter (for auto-screen pause)  
\_TOS Top-Of-Screen? (1=Yes, 0=No)  
\_ROWS Number of screen rows for current user  
\_AUTOTERM Automatic terminal detection results  
\_CONSOLE Console attributes  
\_ANSWERTIME Time phone was answered  
\_LOGONTIME Time of successful logon  
\_NS\_TIME Current new-file scan pointer  
\_LAST\_NS\_TIME Time of last new-file scan  
\_ONLINE Online? (1=LOCAL, 2=REMOTE, 0=No)  
\_SYS\_STATUS System status attributes  
\_SYS\_MISC System toggle options (SCFG)  
\_SYS\_PSNUM PostLink Site Number (SCFG)  
\_SYS\_TIMEZONE \*System time zone (in SMB format, SCFG)  
\_SYS\_PWDAYS \*Days between forced password changes (0=disabled, SCFG)  
\_SYS\_DELDAYS \*Days to preserve deleted user slots (SCFG)  
\_SYS\_AUTODEL \*Number of days before deleting inactive users (SCFG)  
\_SYS\_NODES \*Total configured nodes (SCFG)  
\_SYS\_EXP\_WARN \*Days before expiration to warn user (SCFG)  
\_SYS\_LASTNODE \*Last displayable node (SCFG)  
\_SYS\_AUTONODE \*First auto-node (SCFG)  
\_NODE\_SCRNLEN \*Screen length of this node (0=auto-detect, SCFG)  
\_NODE\_SCRNBLANK \*Minutes between screen blanks (0=disabled, SCFG)  
\_NODE\_MISC Node toggle options (SCFG)  
\_NODE\_VALUSER \*Number of user to send newuser feedback to (SCFG)  
\_NODE\_IVT \*Bits determining which time-slice APIs to use (SCFG)  
\_NODE\_SWAP \*Bits determining which swap types to use (SCFG)  
\_NODE\_MINBPS \*Minimum connect rate allowed for this node (SCFG)  
\_NODE\_NUM \*Number of this node  
\_NEW\_LEVEL \*Security level to give new users (SCFG)  
\_NEW\_FLAGS1 Flag set #1 to give new users (SCFG)  
\_NEW\_FLAGS2 Flag set #2 to give new users (SCFG)  
\_NEW\_FLAGS3 Flag set #3 to give new users (SCFG)  
\_NEW\_FLAGS4 Flag set #4 to give new users (SCFG)  
\_NEW\_EXEMPT Exemption flags to give new users (SCFG)  
\_NEW\_REST Restriction flags to give new users (SCFG)

\_NEW\_CDT Credits to give new users (SCFG)  
\_NEW\_MIN Minutes to give new users (SCFG)  
\_NEW\_SHELL \*Command shell to default to for new users (SCFG)  
\_NEW\_MISC Default settings for new users (SCFG)  
\_NEW\_EXPIRE \*Automatically set newuser expiration days (SCFG)  
\_NEW\_PROT \*Default download protocol for new users (SCFG)  
\_EXPIRED\_LEVEL \*Default security level to give expired users (SCFG)  
\_EXPIRED\_FLAGS1 Default flag set #1 to give expired users (SCFG)  
\_EXPIRED\_FLAGS2 Default flag set #2 to give expired users (SCFG)  
\_EXPIRED\_FLAGS3 Default flag set #3 to give expired users (SCFG)  
\_EXPIRED\_FLAGS4 Default flag set #4 to give expired users (SCFG)  
\_EXPIRED\_EXEMPT Default exemption flags to give expired users (SCFG)  
\_EXPIRED\_REST Default restriction flags to give expired users (SCFG)  
\_MIN\_DSPACE \*Minimum disk space (in k) to allow uploads (SCFG)  
\_CDT\_MIN\_VALUE \*Minutes per 100k of credits (SCFG)  
\_CDT\_PER\_DOLLAR Credits per dollar conversion rate (SCFG)  
\_LEECH\_PCT \*Leech detection percentage (0=disabled, SCFG)  
\_LEECH\_SEC \*Minimum elapsed seconds before leech detection (SCFG)  
\_NETMAIL\_COST Credit cost to send FidoNet netmail (SCFG)  
\_NETMAIL\_MISC \*Toggle options for FidoNet netmail (SCFG)  
\_INETMAIL\_COST Credit cost to send Internet netmail (SCFG)  
\_INETMAIL\_MISC Toggle options for Internet netmail (SCFG)  
\_LOGON\_ULB Total bytes uploaded this call  
\_LOGON\_DLB Total bytes downloaded this call  
\_LOGON\_ULS Total files uploaded this call  
\_LOGON\_DLS Total files downloaded this call  
\_LOGON\_POSTS Total posts sent this call  
\_LOGON\_EMAILS Total emails sent this call  
\_LOGON\_FBACKS Total feedback sent this call  
\_POSTS\_READ Total posts read this call  
\_LOGFILE \*File handle of currently open NODE.LOG  
\_NODEFILE \*File handle of currently open NODE.DAB  
\_NODE\_EXT \*File handle of currently open NODE.EXB  
\_TIMELEFT Number of seconds left online for current user  
\_MAX\_MINUTES Max number of minutes allowed in minute bank (SCFG)  
\_MAX\_QWKMSGs Max number of messages allowed per QWK packet (SCFG)  
\_UQ Bits determining which questions to ask newusers (SCFG)  
\_ERRORLEVEL \*Error level of most recently executed external program  
\_ERRNO \*Current DOS error code (see ERRNO.INC for values)

[Back to Table of Contents](#)

## [6.1] - Quick Function Reference

### [String Manipulation Functions](#)

-----  
STR <str\_var> [str\_var] [...]  
GLOBAL\_STR <str\_var> [str\_var] [...]  
SET <str\_var> <"cstr">  
COPY <str\_var> <any\_var>  
SWAP <str\_var> <any\_var>  
STRCAT <str\_var> <str\_var or "cstr">  
SPRINTF <str\_var> <"cstr"> <any\_var> [any\_var] [...]  
TIME\_STR <str\_var> <int\_var>  
DATE\_STR <str\_var> <int\_var>  
SECOND\_STR <str\_var> <int\_var>  
FTIME\_STR <str\_var> <"cstr"> <int\_var>  
SHIFT\_STR [str\_var] <#>  
STRIP\_CTRL [str\_var]  
TRUNCSP [str\_var]  
STRUPR [str\_var]  
STRLWR [str\_var]  
SETSTR <"cstr">  
REPLACE\_TEXT <#> <"cstr">  
LOAD\_TEXT <"str">  
REVERT\_TEXT <# or ALL>

### [Integer Manipulation Functions](#)

-----  
INT <int\_var> [int\_var] [...]  
GLOBAL\_INT <int\_var> [int\_var] [...]  
SET <int\_var> <#>  
ADD <int\_var> <any\_var or #>  
SUB <int\_var> <any\_var or #>  
MUL <int\_var> <any\_var or #>  
DIV <int\_var> <any\_var or #>  
MOD <int\_var> <any\_var or #>  
AND <int\_var> <any\_var or #>  
OR <int\_var> <any\_var or #>  
NOT <int\_var> <any\_var or #>  
XOR <int\_var> <any\_var or #>  
COPY <int\_var> <any\_var>  
SWAP <int\_var> <any\_var>  
RANDOM <int\_var> <#>



```
TIME <int_var>
STRLEN <int_var> <str_var>
DATE_INT <int_var> <str_var>
CRC16 <int_var> <str_var>
CRC32 <int_var> <str_var>
CHKSUM <int_var> <str_var>
CHARVAL <int_var> <str_var>
```

### Logic/Control Flow Functions

-----

```
GOTO <txt>
CALL <txt>
RETURN
SETLOGIC <TRUE or FALSE or GREATER or LESS>
COMPARE <any_var> <any_var or "cstr" or #>
IF_TRUE
IF_EQUAL
IF_FALSE
IF_NOT_EQUAL
IF_GREATER
IF_GREATER_OR_EQUAL
IF_LESS
IF_LESS_OR_EQUAL
ELSE
END_IF
SWITCH <int_var>
CASE <#>
DEFAULT
END_CASE
END_SWITCH
CMD_HOME
CMDKEY <key>
CMDKEYS <keylist>
CMDSTR <"cstr">
END_CMD
CMD_POP
COMPARE_KEY <key>
COMPARE_KEYS <keylist>
COMPARE_STR <"cstr">
COMPARE_WORD <"cstr">
COMPARE_ARS <ars>
COMPARE_STRN <#> <str_var> <str_var or "cstr">
COMPARE_SUBSTR <str_var> <str_var or "cstr">
```

### Display Functions

-----

```
PRINT <"cstr" or any_var>
PRINTF <"cstr"> <any_var> [any_var] [...]
PRINT_LOCAL <"cstr">
PRINT_REMOTE <"cstr">
PRINTSTR
PRINTKEY
MNEMONICS <"cstr">
CLS
CRLF
PRINTFILE <"str" or str_var> [#]
PRINTTAIL <str_var> <#> <#>
PRINTFILE_STR
PRINTFILE_LOCAL <"str">
PRINTFILE_REMOTE <"str">
LIST_TEXT_FILE
EDIT_TEXT_FILE
PAUSE
MENU <"str">
NODELIST_ALL
NODELIST_USERS
USERLIST_SUB
USERLIST_DIR
USERLIST_ALL
USERLIST_LOGONS
YES_NO <"cstr">
NO_YES <"cstr">
READ_SIF <"str">
SAVELINE
RESTORELINE
```

### Input Functions

-----

```
INKEY
GETKEY
GETKEYE
GETCMD <"cstr">
GETSTR [str_var] [#] [#]
GETLINE [str_var] [#]
GETSTRUPR [str_var] [#]
```

GETNAME [str\_var] [#]  
GETFILESPEC  
GETLINES  
GETNUM [any\_var] <#>  
GET\_TEMPLATE <"str">  
CHKSYSPASS  
CREATE\_SIF <"str">

Miscellaneous Functions

-----  
ONLINE  
OFFLINE  
LOGIN <"cstr">  
LOGON  
LOGOFF  
LOGOFF\_FAST  
LOGOUT  
NEWUSER  
SET\_MENU\_DIR <"str">  
SET\_MENU\_FILE <"str">  
SYNC  
ASYNCR  
RIOSYNCR  
PUT\_NODE  
PAUSE\_RESET  
CLEAR\_ABORT  
UNGETKEY  
UNGETSTR  
HANGUP  
EXEC <"str">  
EXEC\_INT <"str">  
EXEC\_BIN <"str">  
EXEC\_XTRN <"str">  
LOG <"cstr">  
LOGSTR  
LOGKEY  
LOGKEY\_COMMA  
NODE\_STATUS <#>  
NODE\_ACTION <#>  
INC\_MAIN\_CMDS  
INC\_FILE\_CMDS  
COMPARE\_USER\_MISC <#>  
COMPARE\_USER\_CHAT <#>  
COMPARE\_USER\_QWK <#>  
COMPARE\_NODE\_MISC <#>  
TOGGLE\_USER\_MISC <#>  
TOGGLE\_USER\_CHAT <#>  
TOGGLE\_USER\_QWK <#>  
TOGGLE\_NODE\_MISC <#>  
TOGGLE\_USER\_FLAG <char> <char>  
ADJUST\_USER\_CREDITS <#>  
ADJUST\_USER\_MINUTES <#>  
SET\_USER\_LEVEL <#>  
SET\_USER\_STRING <#>  
USER\_EVENT <#>  
UTO\_MESSAGE  
USER\_DEFAULTS  
USER\_EDIT  
TEXT\_FILE\_SECTION  
XTRN\_EXEC  
XTRN\_SECTION  
MINUTE\_BANK  
CHANGE\_USER  
ANSI\_CAPTURE  
FINDUSER  
SELECT\_SHELL  
SET\_SHELL  
SELECT\_EDITOR  
SET\_EDITOR  
TRASHCAN <"str">  
GETTIMELEFT  
MSWAIT <#>  
SEND\_FILE\_VIA <char> <"str" or str\_var>  
RECEIVE\_FILE\_VIA <char> <"str" or str\_var>

Mail Functions

-----  
MAIL\_READ  
MAIL\_READ\_SENT  
MAIL\_READ\_ALL  
MAIL\_SEND  
MAIL\_SEND\_FILE  
MAIL\_SEND\_BULK  
MAIL\_SEND\_FEEDBACK  
MAIL\_SEND\_NETMAIL

MAIL\_SEND\_NETFILE

**Message Base Functions**

-----  
MSG\_SET\_AREA  
MSG\_SET\_GROUP  
MSG\_SELECT\_AREA  
MSG\_SHOW\_GROUPS  
MSG\_SHOW\_SUBBOARDS  
MSG\_GROUP\_UP  
MSG\_GROUP\_DOWN  
MSG\_SUBBOARD\_UP  
MSG\_SUBBOARD\_DOWN  
MSG\_GET\_SUB\_NUM  
MSG\_GET\_GRP\_NUM  
MSG\_READ  
MSG\_POST  
MSG\_QWK  
MSG\_PTRS\_CFG  
MSG\_PTRS\_REINIT  
MSG\_NEW\_SCAN\_CFG  
MSG\_NEW\_SCAN  
MSG\_NEW\_SCAN\_ALL  
MSG\_NEW\_SCAN\_SUB  
MSG\_CONT\_SCAN  
MSG\_CONT\_SCAN\_ALL  
MSG\_BROWSE\_SCAN  
MSG\_BROWSE\_SCAN\_ALL  
MSG\_FIND\_TEXT  
MSG\_FIND\_TEXT\_ALL  
MSG\_YOUR\_SCAN\_CFG  
MSG\_YOUR\_SCAN  
MSG\_YOUR\_SCAN\_ALL

**File Base Functions**

-----  
FILE\_SET\_AREA  
FILE\_SET\_LIBRARY  
FILE\_SELECT\_AREA  
FILE\_SHOW\_LIBRARIES  
FILE\_SHOW\_DIRECTORIES  
FILE\_LIBRARY\_UP  
FILE\_LIBRARY\_DOWN  
FILE\_DIRECTORY\_UP  
FILE\_DIRECTORY\_DOWN  
FILE\_GET\_DIR\_NUM  
FILE\_GET\_LIB\_NUM  
FILE\_LIST  
FILE\_LIST\_EXTENDED  
FILE\_VIEW  
FILE\_UPLOAD  
FILE\_UPLOAD\_USER  
FILE\_UPLOAD\_SYSOP  
FILE\_DOWNLOAD  
FILE\_DOWNLOAD\_USER  
FILE\_DOWNLOAD\_BATCH  
FILE\_REMOVE  
FILE\_BATCH\_ADD  
FILE\_BATCH\_ADD\_LIST  
FILE\_BATCH\_CLEAR  
FILE\_BATCH\_SECTION  
FILE\_TEMP\_SECTION  
FILE\_NEW\_SCAN  
FILE\_NEW\_SCAN\_ALL  
FILE\_FIND\_TEXT  
FILE\_FIND\_TEXT\_ALL  
FILE\_FIND\_NAME  
FILE\_FIND\_NAME\_ALL  
FILE\_PTRS\_CFG  
FILE\_SET\_ALT\_PATH  
FILE\_RESORT\_DIRECTORY  
FILE\_SEND  
FILE\_GET  
FILE\_PUT  
FILE\_UPLOAD\_BULK  
FILE\_FIND\_OLD  
FILE\_FIND\_OPEN  
FILE\_FIND\_OFFLINE  
FILE\_FIND\_OLD\_UPLOADS

**Chat Functions**

-----  
PAGE\_SYSOP  
PAGE\_GURU  
PRIVATE\_CHAT

PRIVATE\_MESSAGE  
CHAT\_SECTION

### Information Functions

-----  
INFO\_SYSTEM  
INFO\_SUBBOARD  
INFO\_DIRECTORY  
INFO\_USER  
INFO\_VERSION  
INFO\_XFER\_POLICY  
GURU\_LOG  
ERROR\_LOG  
SYSTEM\_LOG  
SYSTEM\_YLOG  
SYSTEM\_STATS  
NODE\_STATS  
SHOW\_MEM

### File I/O Functions

-----  
FOPEN <int\_var> <#> <"str" or str\_var>  
FCLOSE <int\_var>  
FREAD <int\_var> <any\_var> [int\_var or #]  
FWRITE <int\_var> <any\_var> [int\_var or #]  
FFLUSH <int\_var>  
FGET\_LENGTH <int\_var> <int\_var>  
FSET\_LENGTH <int\_var> <int\_var or #>  
FGET\_TIME <int\_var> <int\_var>  
FSET\_TIME <int\_var> <int\_var>  
FEOF <int\_var>  
FGET\_POS <int\_var> <int\_var>  
FSET\_POS <int\_var> <int\_var or #> [#]  
FLOCK <int\_var> <int\_var or #>  
FUNLOCK <int\_var> <int\_var or #>  
FPRINTF <int\_var> <"cstr"> [any\_var] [...]  
FREAD\_LINE <int\_var> <any\_var>  
FSET\_ETX <#>

### File System Functions

-----  
CHKFILE <"str" or str\_var>  
REMOVE\_FILE <str\_var>  
RENAME\_FILE <str\_var> <str\_var>  
COPY\_FILE <str\_var> <str\_var>  
MOVE\_FILE <str\_var> <str\_var>  
GET\_FILE\_ATTRIB <int\_var> <str\_var>  
SET\_FILE\_ATTRIB <int\_var> <str\_var>  
GET\_FILE\_TIME <int\_var> <str\_var>  
GET\_FILE\_LENGTH <int\_var> <str\_var>

### Directory System Functions

-----  
MAKE\_DIR <str\_var>  
CHANGE\_DIR <str\_var>  
REMOVE\_DIR <str\_var>  
OPEN\_DIR <int\_var> <str\_var>  
READ\_DIR <int\_var> <str\_var>  
REWIND\_DIR <int\_var>  
CLOSE\_DIR <int\_var>

### [Back to Table of Contents](#)

---

## **Copyright © 2000 by Rob Swindell**

Synchronet BBS Software (Synchronet) Version 3 is comprised of several documentation, library, executable, and source code files, all of which are covered by the GNU General Public License with the exception of the following portions covered by the GNU Lesser General Public License: SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the Public Domain by Digital Dynamics in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

Rob Swindell  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .

# JavaScript in Synchronet

Synchronet uses [Mozilla's JavaScript engine](#) for its preferred local scripting environment.

You can learn about the core JavaScript language and object model from the following documents:

- [Core JavaScript Guide](#)
- [Core JavaScript Reference](#)
- [More Mozilla JavaScript documents](#)

JavaScript is an established, mature scripting language syntactically similar to C++ and Java.

[JavaScript is not Java.](#)

The ECMA and ISO standards organizations have standardized the core JavaScript language in [ECMA-262](#) (ECMAScript) and ISO-16262.

## Baja and JavaScript

[Baja](#) is the original scripting language of Synchronet (introduced in v2.0), used to create Synchronet-specific *modules* and *command shells*. Baja was originally designed as a simple BASIC-like language for controlling the display of menus and command prompts, accepting commands from the user and passing on control to high-level BBS functions. Over the years, Baja has been extended and enhanced to allow a high-level of functionality, but it was never going to reach the power and flexibility of JavaScript.

For example Baja modules and command shells, see the `.src` files in your Synchronet **exec** directory.

For the foreseeable future Baja modules and command-shells will continue to be supported in Synchronet, but sysops and developers are encouraged to use JavaScript instead of Baja moving forward. With very few exceptions, everything that can be done in a Baja module can be done in a JavaScript module, and much much more.

Eventually, all the stock command-shells and external modules will be converted to JavaScript.

## JavaScript Files

JavaScript files are just ASCII text files. They are normally named with a `.js` file extension and located in your Synchronet **exec** directory or with a `.ssjs` and located in your Synchronet **web** hierarchy. JavaScript files do not need to be compiled. JavaScript files are loaded into memory at the time of execution, so a change to a JavaScript file will take effect the next time that file is executed (no recycling of servers is normally required).

For example JavaScript modules and services, see the `.js` files in your Synchronet **exec** directory.

Modified stock `.js` files should be placed in your Synchronet **mods** directory to prevent over-writing by future upgrades.

## Integration

JavaScript files can be executed from:

- **Terminal Server**  
as a timed event, external program (door), login/logon/newuser module, basically anywhere a Baja module or executable can be launched
- **Web Server**  
dynamically generates HTML files  
see `web/root/*.ssjs`
- **FTP Server**  
dynamically generates HTML index files  
see `exec/ftp-html.js` and `exec/ftp-web-html.js`
- **Services**  
Synchronet services (both static and dynamic) are usually written in JavaScript  
see `exec/*service.js` and `ctrl/services.ini`
- **Mail Server**  
inbound mail processors may be written in JavaScript  
see `exec/mailproc_util.js` and `ctrl/mailproc.ini`
- **JSexec**  
some script files may be executed outside of Synchronet (e.g. as a CGI script or daemon) using `jsexec`  
examples: `ircd.js`, `newslink.js`

From the Terminal Server, a JavaScript file is executed on a native command-line by placing a question mark (?) at the beginning of the command-line before the JavaScript file name (in SCFG). It is not necessary to specify the `.js` portion of the file name on the command-line. For example, the command-line to execute the file `exec/newslink.js` would be `"?newslink"` or `"?newslink.js"`.

From within a Baja module, a JavaScript file may be executed using the following Baja code:

`exec "?modname"` where *modname* is the JavaScript file.

## Object Model (methods and properties)

Synchronet has its own constantly evolving JavaScript object model, not to be confused with the Document Object Model (DOM) used in web browsers. In order to fully understand the capabilities of JavaScript modules in Synchronet, you must familiarize yourself with [Core JavaScript](#) as well as the [Synchronet JavaScript Object Model](#).

\$Id: js.html,v 1.9 2011/10/21 02:35:04 rswindell Exp \$





# Synchronet's FidoNet Packet Tosser Documentation

## Table of Contents

[Introduction](#)  
[Terminology](#)  
[Installation](#)  
[AREAS.BBS Format](#)  
[Configuration](#)  
[Running SBBSecho](#)  
[Command Line Switches](#)  
[Area Manager Remote Commands](#)

## Introduction

SBBSecho is a full-blown FTN echomail program (tosser/scanner) for Synchronet BBS version 2.0 and higher. SBBSecho is level III implementation of the SMB v2.00 specification including support for HyperAllocated and LZH compressed message bases.

It is intended to replace FTSC-1 (\*.MSG) compatible echomail programs (GEcho, Squish, FastEcho, Alexi/Mail, etc.) and SBBSFIDO with a single program incorporating the standard features of echomail programs without the .MSG phase (consuming disk space and time).

You will not need to use SBBSFIDO (\*.MSG import/export utility for Synchronet) or any echomail programs after you have installed SBBSecho.

[Back to Table of Contents](#)

## Terminology

### FTN

**FidoNet Technology Network:** Any network using FidoNet standards for addressing, mail packets, mail sessions, node lists, etc.

### Zones, Nets, Nodes, and Points?

FTN node addresses are like phone numbers, they are made up of multiple components (four to be exact). The components are: zone, net, node, and point. Each component is specified by a decimal (base-10) number, separated by symbols (no spaces):

#### Zone:Net/Node.Point

The zone represents the continent (if FidoNet) or the network number (if other FTN network). All FidoNet nodes in North America have a zone 1 address. When the zone is specified in an address, it is the first component and must be followed by a colon. If the zone is not present in an address, the local system's zone is assumed.

The net represents the network number of the FTN node. Duplicate net numbers may exist between zones. If the net number is not present, the local system's net is assumed.

The node number specifies an exact FTN node within a network. The node number is the only required element of an FTN node address.

The point is an optional component which specifies a sub-node that does not directly receive mail and is also not listed in the main FTN node list, but instead gets all its mail from its boss-node (zone:net/node.0). When the point is not specified, 0 (zero) is assumed (i.e. 1:2/3 and 1:2/3.0 are identical) which indicates the system is not a point node address.

A 2D (2 dimensional) address refers to an FTN address containing just the net and node numbers (i.e. 103/705).

A 3D (3 dimensional) address refers to an FTN address containing the zone, net, and node numbers (i.e. 1:103/705), specifically excluding the point number if it exists.

A 4D (4 dimensional) address refers to an FTN address containing the zone,

net, node, and optional point numbers (i.e. 1:103/705.1).

A 5D (5 dimensional) address refers to an FTN address consisting of a standard 3D or 4D address with an appended "@domain" (i.e. 1:103/705@fidonet.org).

### **Attach or FLO Mailer?**

If you are using FrontDoor, InterMail, D'bridge, SEAdog, Dutchie, or any other ArcMail \*.MSG attach-style mailer, you are using what we will refer to as an "Attach Mailer".

If you are using BinkleyTerm, Portal of Power, or any other FLO/CLO/HLO/DLO style mailer, you are using what we will refer to as a "FLO Mailer".

Both types are supported equally by SBBSecho.

### **NetMail**

Point-to-point (usually person-to-person) direct or routed messages.

### **EchoMail**

Group or conference messages of a particular subject matter. Usually distributed on a regional or continental (e.g. FidoNet Zone 1 backbone) scale. FTN style echomail areas have a unique area tag (name) associated with them to distinguish each area from the others.

### **Packet**

An FTN packet is a group of one or more messages contained in a single uncompressed file. Packets may contain echomail and/or netmail messages. Packets usually have a .PKT extension, although outbound NetMail packets for FLO Mailers will have .?UT extensions (where ? is either O, C, D, or H). The first eight characters of the filename may be anything, but are usually decimal digits representing the date and time the packet was created. SBBSecho creates temporary outbound packet files with a .PK\_ extension and then renames them to .PKT when they're completed. If you find any .PK\_ files in your outbound directory, don't worry. SBBSecho will find them the next time it is run and continue packing them and sending them on their way.

### **Bundle**

An FTN bundle is a single file archive of one or more (usually compressed) packets. Bundles will have file extensions where the first two characters represent the day of the week the bundle was created (MO, TU, WE, TH, FR, SA, and SU) and the third character of the extension is a number or letter. The first eight characters of the filename may be anything, but are usually hexadecimal digits representing the FTN node address (or relative address) of the system that created the bundle. SBBSecho changes the file extension of bad inbound bundles to .MO\_, .MO-, or .MO.

### **Areafix/Area Manager**

Areafix is a synonym for area manager (the very first FTN area manager program was called AreaFix). Area manager capabilities (remote adding/removing of areas, changing compression type, etc) are built into SBBSecho, so therefore no external area manager program is required. If you are not an FTN hub, then the area manager portion of SBBSecho will probably not get any use on your system.

[Back to Table of Contents](#)

## **Installation**

To begin, copy the SBBSECHO.EXE, ECHOCFG.EXE, and AREAMGR.HLP files into your Synchronet EXEC directory (usually C:\SBBS\EXEC), copy the SBBSECHO.CFG file into your Synchronet CTRL directory (usually C:\SBBS\CTRL), and copy the SBBSECHO.DOC file into your Synchronet DOCS directory (usually C:\SBBS\DOCS).

If you plan on using the 32-bit DOS, OS/2, or Windows 95/NT version of SBBSecho, you will need to put the appropriate SBBSECHO.EXE file into your Synchronet EXEC directory (unzip OS2.ZIP for the OS/2 version, DOS4G.ZIP for the 32-bit DOS version, or WIN32.ZIP for the Windows 95/NT version). If you get memory allocation errors running the 16-bit DOS version or desire improved performance, you will want to run one of the 32-bit versions.

Add the SBBSCtrl and SBBSNode environment variables to your AUTOEXEC.BAT. Example:

```
SET SBBSCtrl=C:\SBBS\CTRL
SET SBBSNode=C:\SBBS\NODE1
```

Note: The node number used for the SBBSNode environment variable is not important, use your NODE1 directory.

These environment variables must be present for SBBSECHO and ECHOCFG to

function correctly. Type "SET" at the DOS prompt to be sure they are listed correctly.

You will also need to modify your existing MAILER.BAT (or FDRUN.BAT, IMRUN.BAT, etc) file to accomodate the use of SBBSecho, or replace it with the MAILER.BAT included with SBBSecho. If this is your first time setting up FidoNet, you will not currently have a MAILER.BAT file set up, so use the one that is included and modify it so that it reflects the proper drive letters and directories.

Next, SBBSecho expects to find the file AREAS.BBS in your Synchronet DATA directory (usually C:\SBBS\DATA) unless you have overridden the default name and/or location using the ECHOCFG program. The format of this file is very similar to the standard AREAS.BBS (which many echomail programs use, or can at least export their native area file to this format - i.e. GEcho's GSETUP).

SCFG can also export your sub-boards to AREAS.BBS, but you're probably better off modifying the AREAS.BBS file you're already using (if you are) or converting your current echomail program's area file to AREAS.BBS. This is because SCFG assumes the sub-board short name will be the same as the area's tag (as shown in your echo list), but this may not always be the case - unless you imported an AREAS.BBS or FIDONET.NA file into SCFG to begin with).

[Back to Table of Contents](#)

## AREAS.BBS Format

Lines beginning with a semicolon (;) are considered comments, and are ignored.

Each line is in the following format:

**CODE TAG UPLINKS**

*Where:*

**CODE** is the internal code of the sub-board. Any unrecognized internal codes are considered "Passthru" (not imported to your BBS).

**TAG** is the FidoNet tag as shown in your echo list (i.e. FIDONET.NA). If this is '\*', then it will be considered a "badecho" area and will receive all messages for areas not otherwise specified in this file.

**UPLINKS** is a list of FidoNet addresses which you wish to export mail to. At the very least, your hub's address should be listed here. Your address should NOT be listed here. If multiple addresses are specified, you should specify the full 3D address for each.

**AREAS.BBS Example:**

```
SBBS SYNCHRONET 1:3615/50
SYNCDATA SYNCDATA 1:3615/50
SYNC_SYS SYNC_SYSOPS 1:3615/50
```

The amount of spacing between each element in the line is not important. Each line can be up to 1000 characters in length.

So if your OLD AREAS.BBS file is in .MSG format (for example):

```
C:\FD\ECHO\SBBS SYNCHRONET 1:3615/50
```

It must be changed to match the example given above!

[Back to Table of Contents](#)

## Configuration

In order to set up SBBSecho for your system you must run the included ECHOCFG program. It should already be located in your EXEC directory. You may want to put your EXEC directory in your DOS search path so you can execute SBBSECHO and ECHOCFG from any drive and directory on your system.

Multiple configuration files can be used (but isn't usually required) for multi-mailer systems. The default configuration filename is SBBSECHO.CFG located in your Synchronet CTRL directory. You can override this by specifying the name and location of the configuration file on the SBBSECHO and ECHOCFG command lines. Example:

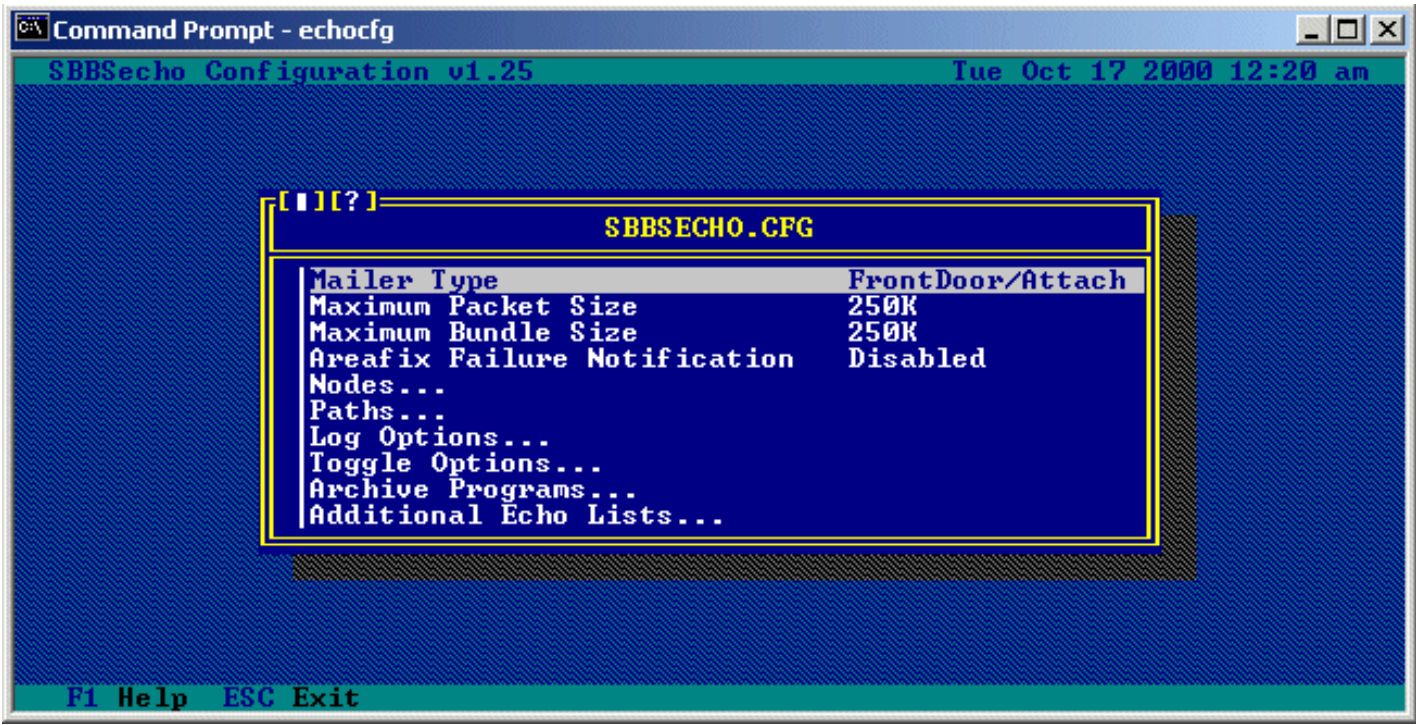
```
ECHOCFG D:\ECHO1.CFG
SBBSECHO D:\ECHO1.CFG
```

If you do not specify a configuration file, ECHOCFG and SBBSECHO will use SBBSECHO.CFG located in your Synchronet CTRL directory.

Upon running ECHOCFG you will be brought to the main menu of the configuration program. The path and filename of the configuration file being modified will appear at the top of the menu. Following are screen captures of available menus



within the configuration program and an explanation of the options contained on each of them.



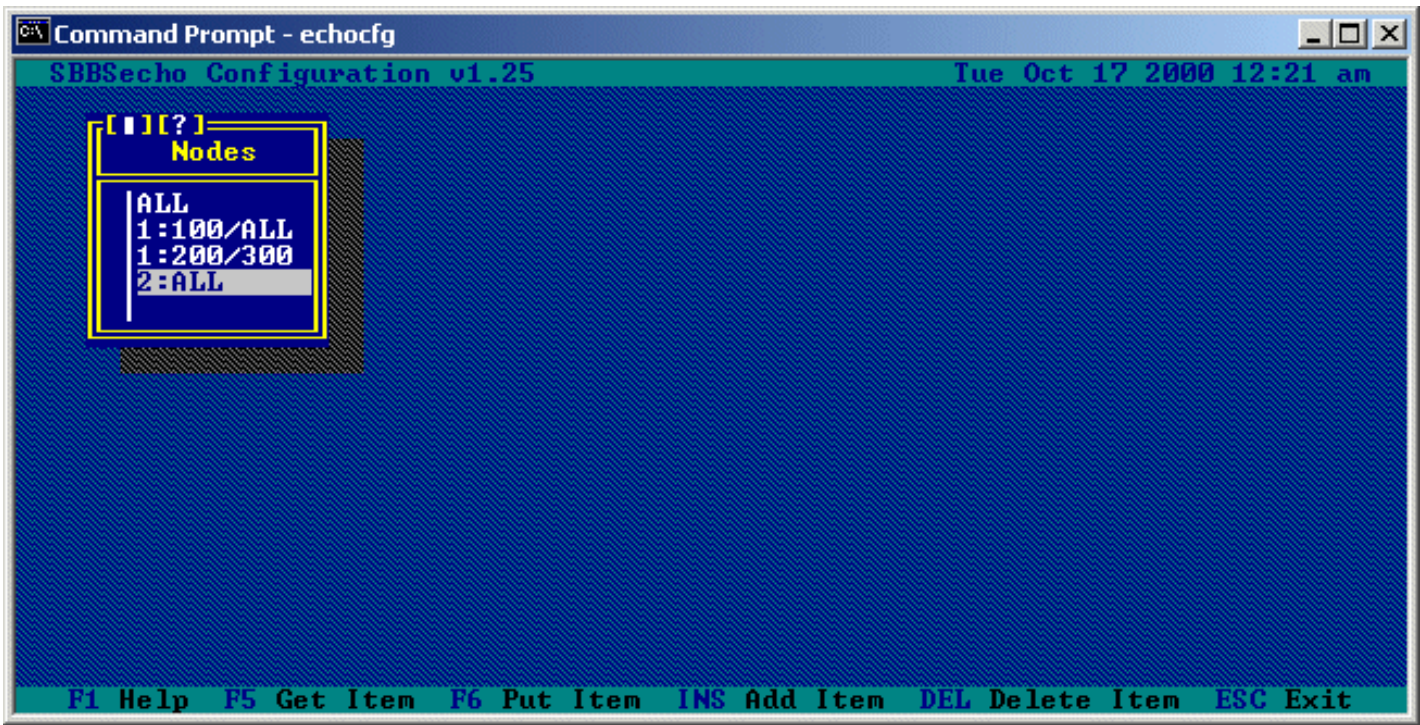
**Mailer Type:**  
Selecting this option will toggle between the mailer types supported by SBBSecho, either FrontDoor (message attach) type mailers or Binkley (FLO file) type mailers. Choose the one that matches your front-end mailer type.

**Maximum Packet Size:**  
This option allows you to set the maximum size of each outgoing echomail packet.

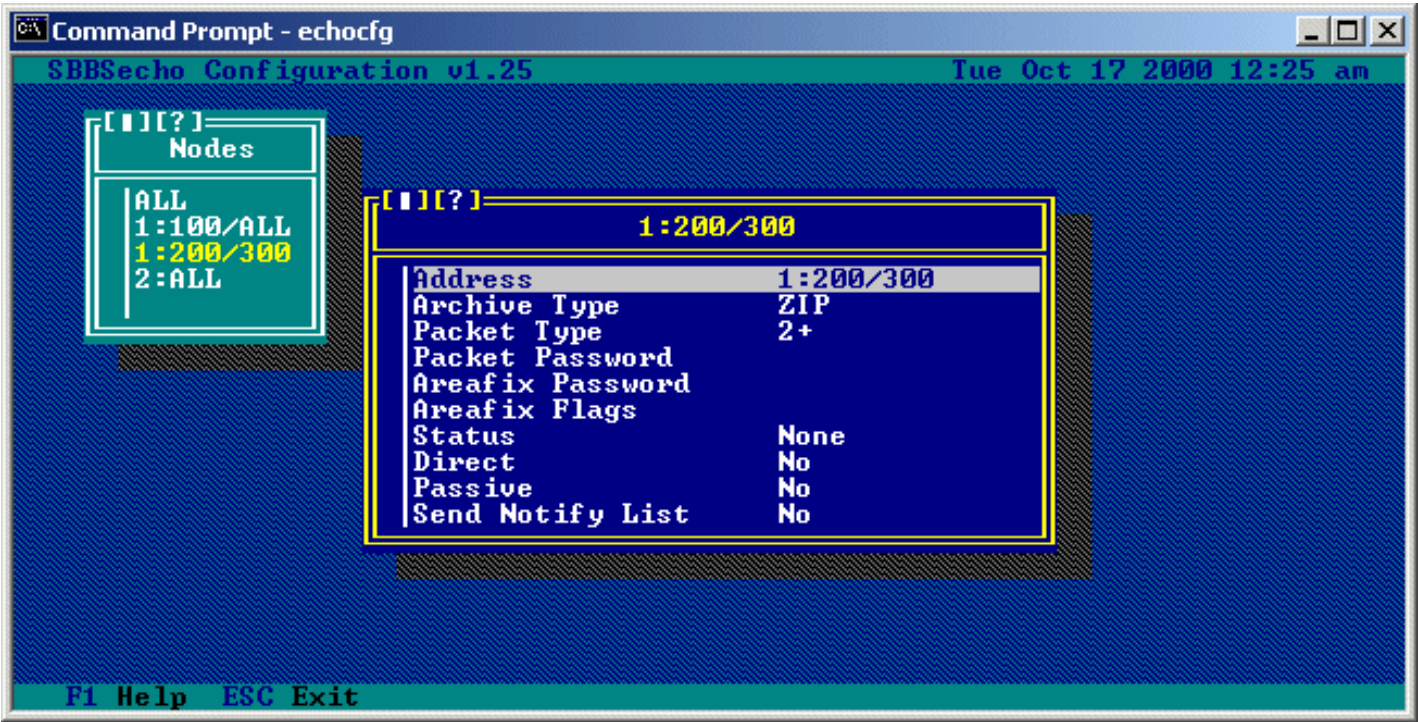
**Maximum Bundle Size:**  
Mail packets are normally packed into what are called "bundles" (unless a node is set up to receive uncompressed mail packets). This option allows you to specify the maximum size of each outgoing mail bundle.

**Areafix Failure Notification:**  
This is the user number of the person where notification of Areafix failures should be sent. Such failures include nodes which are not configured for areafix, nodes using incorrect areafix passwords, and the like. Setting this to 0 will disable this option.

**Nodes...**  
This option allows you to add, remove, and configure nodes that you will be sending mail to. Selecting this option will bring you to a sub-menu which will look similar to the following:

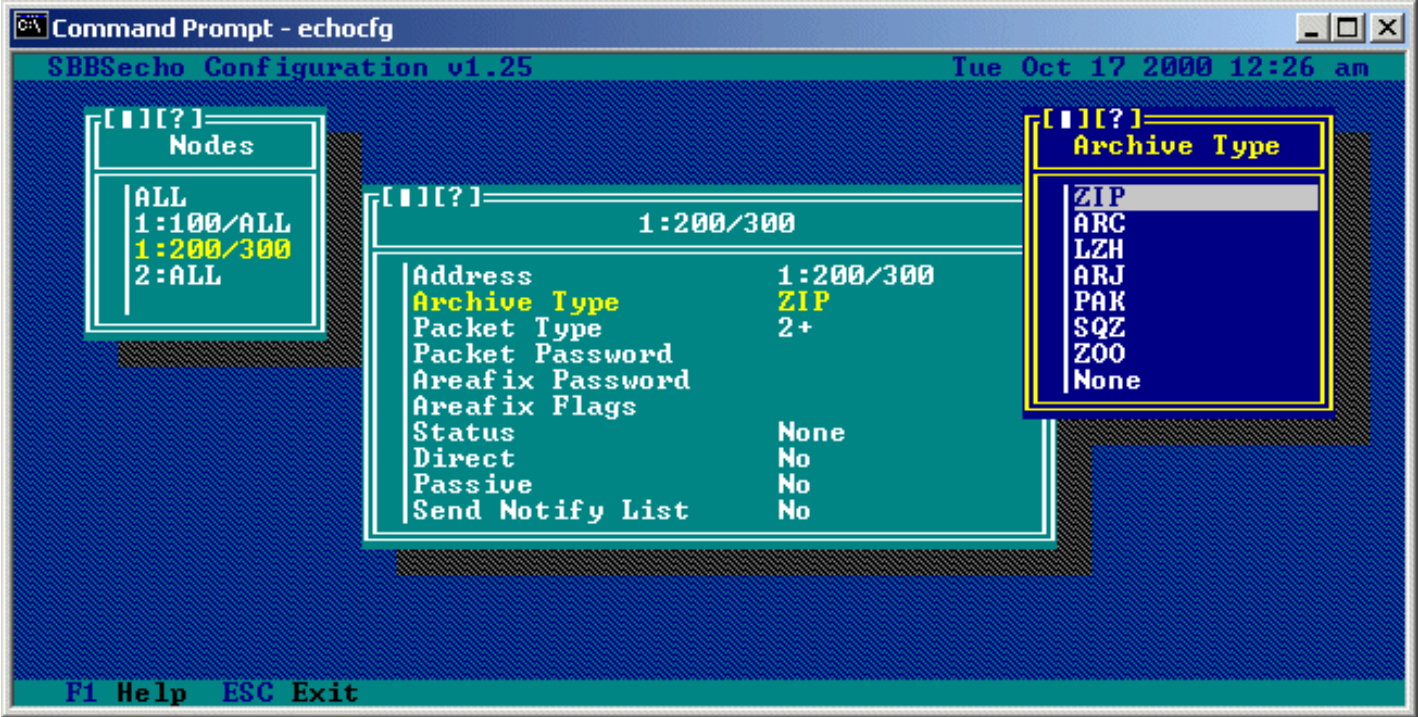


At the Nodes... sub-menu, pressing the INSert key will allow you to add a new node, pressing the DElete key will remove the currently highlighted node number, and pressing ENTER on the currently highlighted node will allow you to edit the options for that particular node from a menu like the following (note that the node number you are editing appears at the top of the window):



**Address:**  
This is the address of the node you are editing, selecting it will allow you to change this to a different address. Using the "ALL" wildcard in place of one of the address components will allow you to configure settings for all nodes that meet that specification (e.g. all nodes in zone 1 can be specified as 1:ALL or all nodes in zone 1, net 103 can be specified as 1:103/ALL).

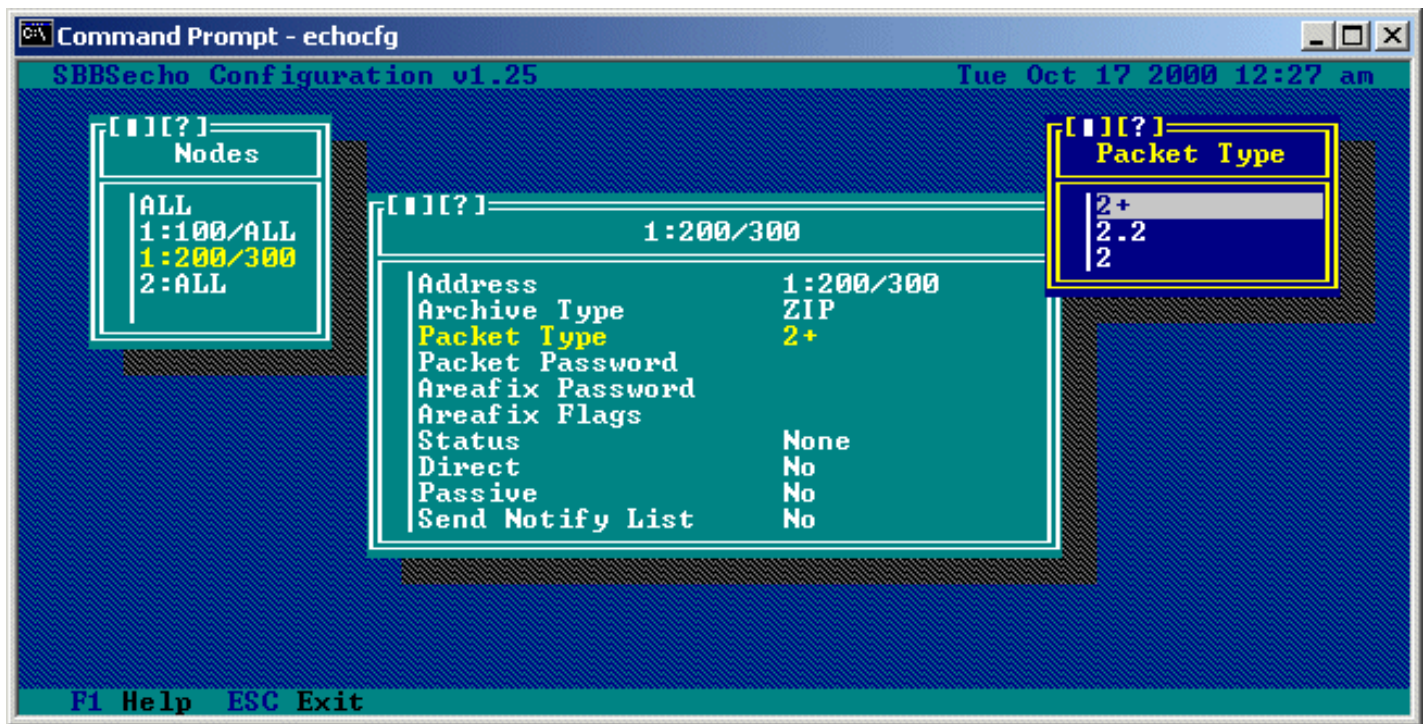
**Archive Type:**  
This is the compression type to be used when packing mail bundles for this node. Selecting this option will allow you to choose from a menu of currently configured archive types, like the following:



Selecting "None" will specify that this node is to receive uncompressed packets (no bundles).

**Packet Type:**  
This is the packet type that will be used when creating mail packets for this node. The default packet type used by SBBSecho is 2+. If you are a "point" address (e.g. 1:100/100.1) you should use either a type 2+ or 2.2 packet since type 2 packets do not support point numbers. Selecting this option will allow you to choose from a menu of currently supported packet types:





**Packet Password:**

This is a password that will be placed into each outgoing mail packet for this node. Passwords are normally used for extra security when sending and receiving mail packets. This node must also have the same packet password defined for your address and SBBSecho must be set up for secure operation (set in the 'Toggle Options' sub-menu) in order for this feature to function properly.

**Areafix Password:**

This is the password that will be required by this node (in the subject field) when it sends messages to the area manager (AreaFix).

**Areafix Flags:**

When additional echo lists have been defined (from the 'Additional Echo Lists...' sub-menu) these flags determine which echo lists can be used by this node when processing area manager add requests.

**Status:**

This option determines the netmail status that will be set when SBBSecho sends out an areafix message or a file attach. Selecting this option toggles between None, Crash, and Hold status.

**Direct:**

When set to 'Yes' this option will add a Direct kludge line to messages that SBBSecho sends out (or create DLO/DUT files for FLO mailers).

**Passive:**

Setting this option to 'Yes' will prevent messages from being sent to this node without the need for altering the AREAS.BBS file. This is useful for temporarily shutting off the messages to this node. This option can be toggled on and off remotely via an area manager request.

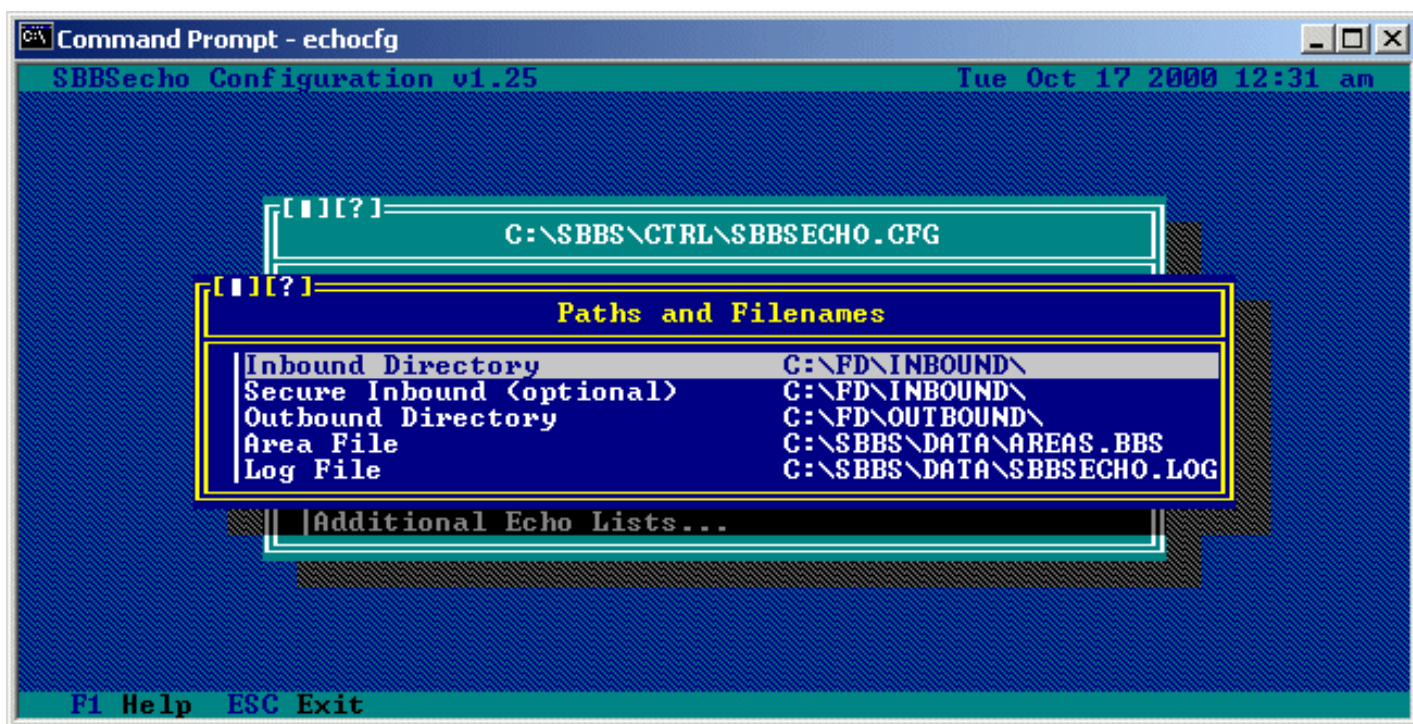
**Send Notify List:**

This determines whether or not this node is sent a Notify List when using that command line option in SBBSecho. A Notify List is a netmail sent to the system operator of the node showing options set for the node as well as connected areas.

**Paths...**

This option allows you to configure the paths and filenames which are used by SBBSecho. Selecting this option will bring you to the following sub-menu:





#### Inbound Directory:

This is the directory where SBBSecho should look for inbound mail packets. This is normally taken from whatever has been set in the SCFG program.

#### Secure Inbound (optional):

This is an optional directory where SBBSecho should look for secure inbound mail packets.

#### Outbound Directory:

This is the directory where SBBSecho will place outgoing mail packets and bundles.

#### Area File:

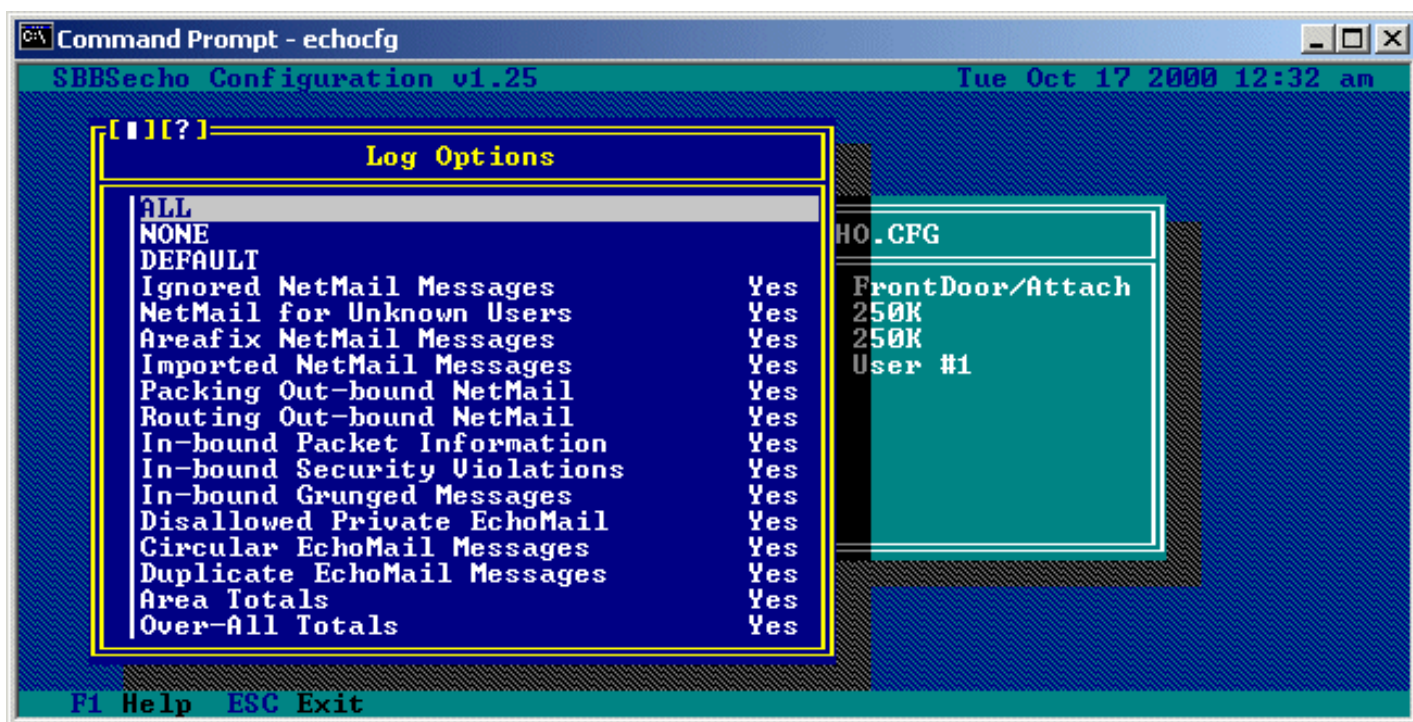
This is the path and filename of the file that SBBSecho will use as it's AREAS.BBS file. By default SBBSecho looks for the file AREAS.BBS in the data directory defined in SCFG.

#### Log File:

This is the path and filename of the file that SBBSecho will use when logging events. By default SBBSecho uses the file SBBSECHO.LOG in the data directory defined in SCFG.

#### Log Options...

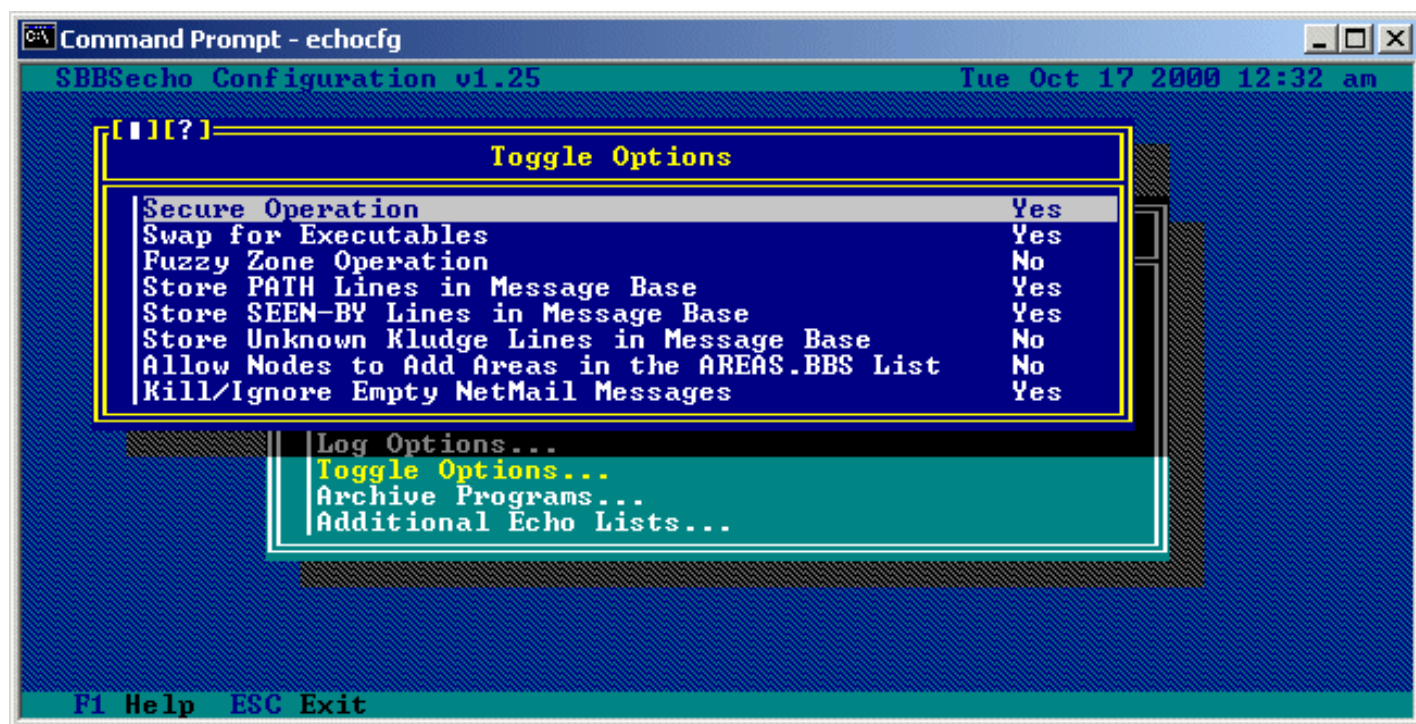
This list allows you to toggle what events SBBSecho should place into the logfile. Selecting this option will bring you to the following sub-menu:



Selecting ALL, NONE, or DEFAULT will toggle all of the options to 'Yes', 'No', or to their default states, respectively.

#### Toggle Options...

These options allow you to toggle various features in SBBSecho. Selecting this option will bring you to the following sub-menu:



#### Secure Operation:

When set to 'Yes', SBBSecho will compare the origin of any incoming mail packets to the nodes in the AREAS.BBS file as messages are imported. If a packet password has been defined for that node, it will also be compared to the password contained in the mail packet. Packets and messages failing this security will not be imported.

#### Swap for Executables:

SBBSecho relies on external executable programs for extracting and compressing mail bundles. This option tells SBBSecho to swap out of memory when running any of these executables, giving the executable programs more free memory to run in. This option is only used in the 16-bit DOS flavor of SBBSecho.

#### Fuzzy Zone Operation:

Some mail programs do not create netmail messages with zone information (INTL kludge line) or may only do so when sending between zones. This is a problem for systems that receive netmail for multiple addresses with different zones (AKAs with different zone numbers). Setting this option to "Yes" allows SBBSecho to guess what the correct originating and destination zone is based on the net and node portions of the destination address in netmail message.

#### Store PATH Lines in Message Base:

When set to 'Yes', SBBSecho will store the PATH lines from incoming echomail in the Synchronet message base headers (not the body text). This option is useful for troubleshooting routing/duplicate message problems.

#### Store SEEN-BY Lines in Message Base:

When set to 'Yes', SBBSecho will store the SEEN-BY lines from incoming echomail in the Synchronet message base headers (not the body text). This option is useful for troubleshooting routing/duplicate message problems.

#### Store Unknown Kludge Lines in Message Base:

When set to 'Yes', SBBSecho will store any unknown kludge lines from incoming echomail in the Synchronet message base headers. This option is useful for troubleshooting problems.

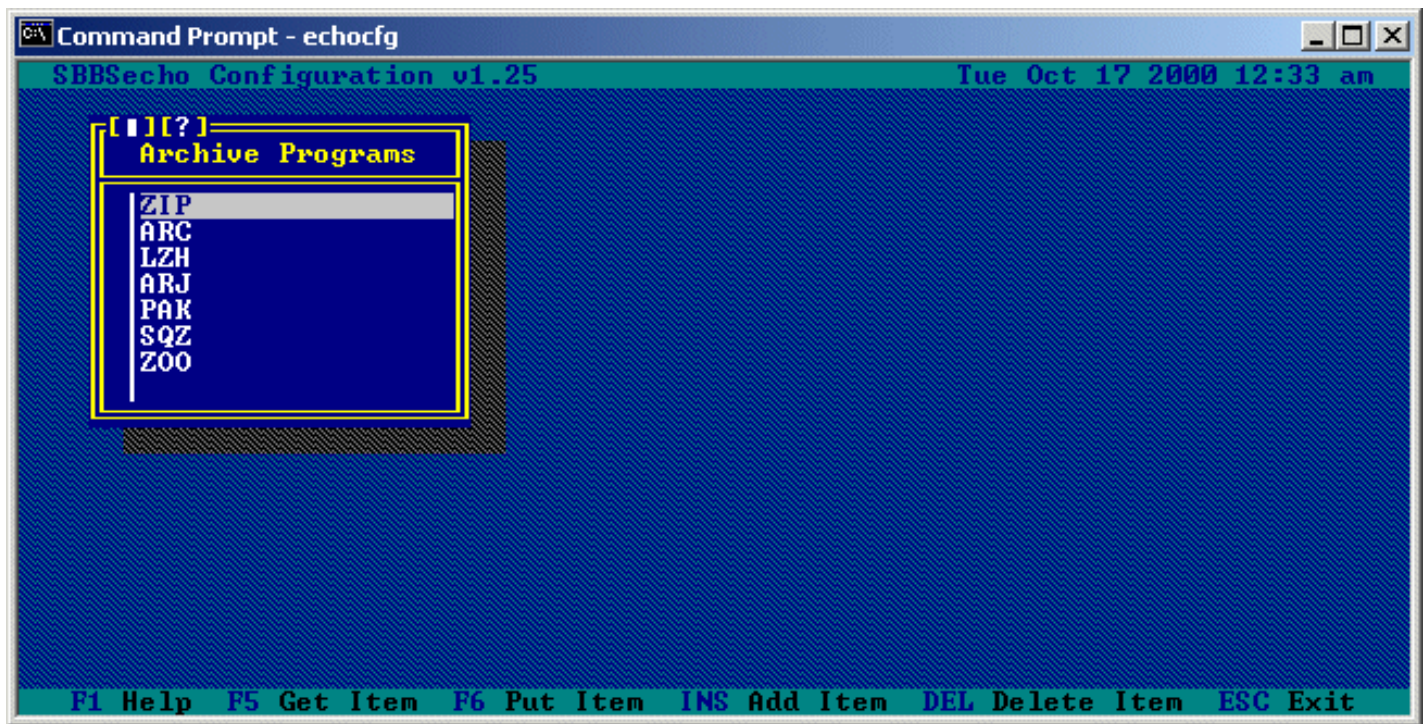
#### Allow Nodes to Add Areas in the AREAS.BBS List:

When set to 'Yes', SBBSecho will allow nodes to add areas (via area manager) which are listed in the AREAS.BBS list. When set to 'No', SBBSecho will only allow nodes to add areas from any additionally configured echo lists which they have access to.

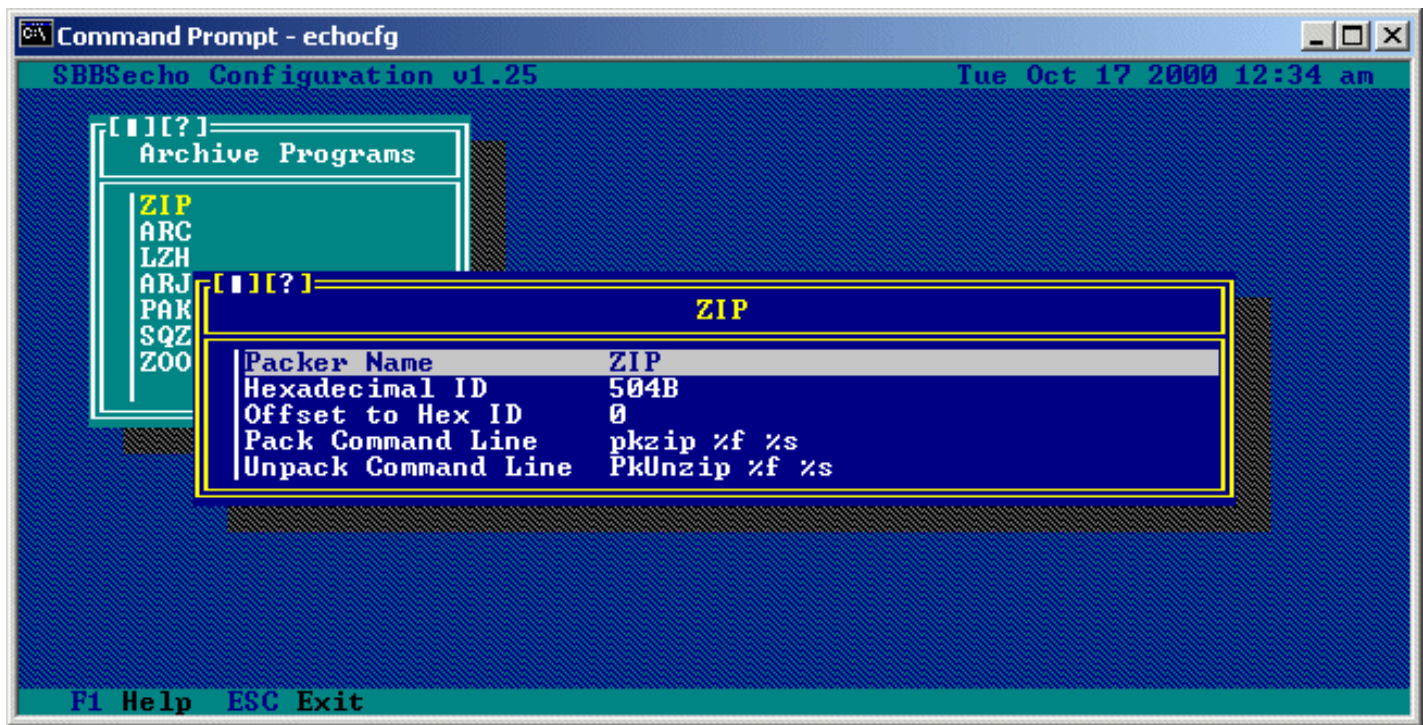
#### Archive Programs...

This option allows you to add or remove archive programs from SBBSecho. Selecting this option will bring you to the following menu:





At the Archive Programs... sub-menu, pressing the INSert key will allow you to add a new archive program. Pressing the DElete key will remove the currently highlighte archive program. And pressing ENTER on the currently highlighte archive program will allow you to edit the options for that particular program from a menu like the following (note that the name of the archive program you are editing appears at the top of the window):



**Packer Name:**  
This is the name that will be used to reference this particular archiving program. This is also the name that should be used by nodes using areamanger to change their compression type remotely.

**Hexadecimal ID:**  
This is a hexadecimal identifier that SBBSecho should look for when determining the compression type used on incoming bundles. In this example we are looking for the ID 'PK', '50' is the hexadecimal value for the letter 'P' and '4B' is the hexadecimal value for the letter 'K'.

**Offset to Hex ID:**  
This is the byte offset (from the beginning of the file) where the hexadecimal ID for this archive program can be located. In this example we are looking at an offset of 0 bytes from the beginning of the file.

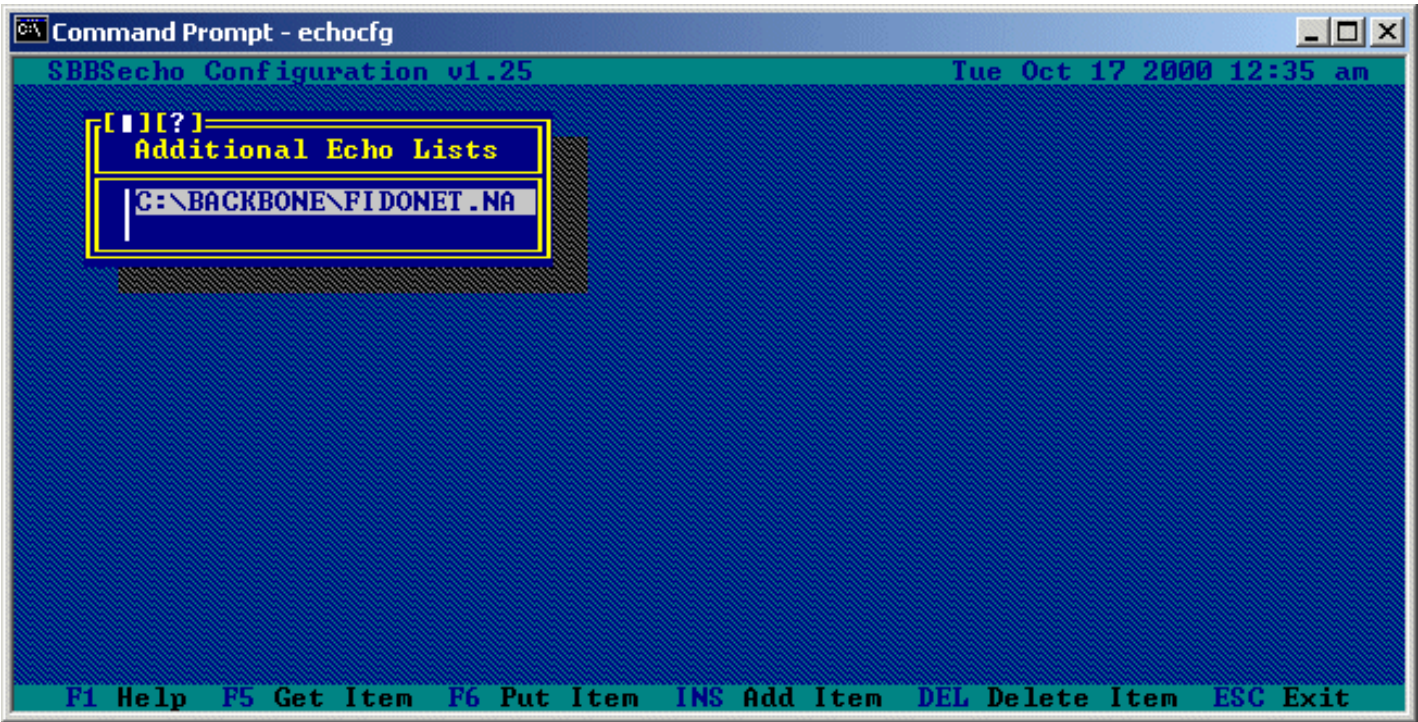
**Pack Command Line:**  
This is the command line used by this archiving program for compressing files. The '%f' command line specifier will expand to the name of the compressed file, the '%s' command line specifier will expand to the name of the file being compressed.

**Unpack Command Line:**  
This is the command line used by the archiving program for Uncompressing files. The '%f' command line specifier will expand to the name of the compressed file, the '%s' command line specifier will expand to the path where the file is being extracted to.

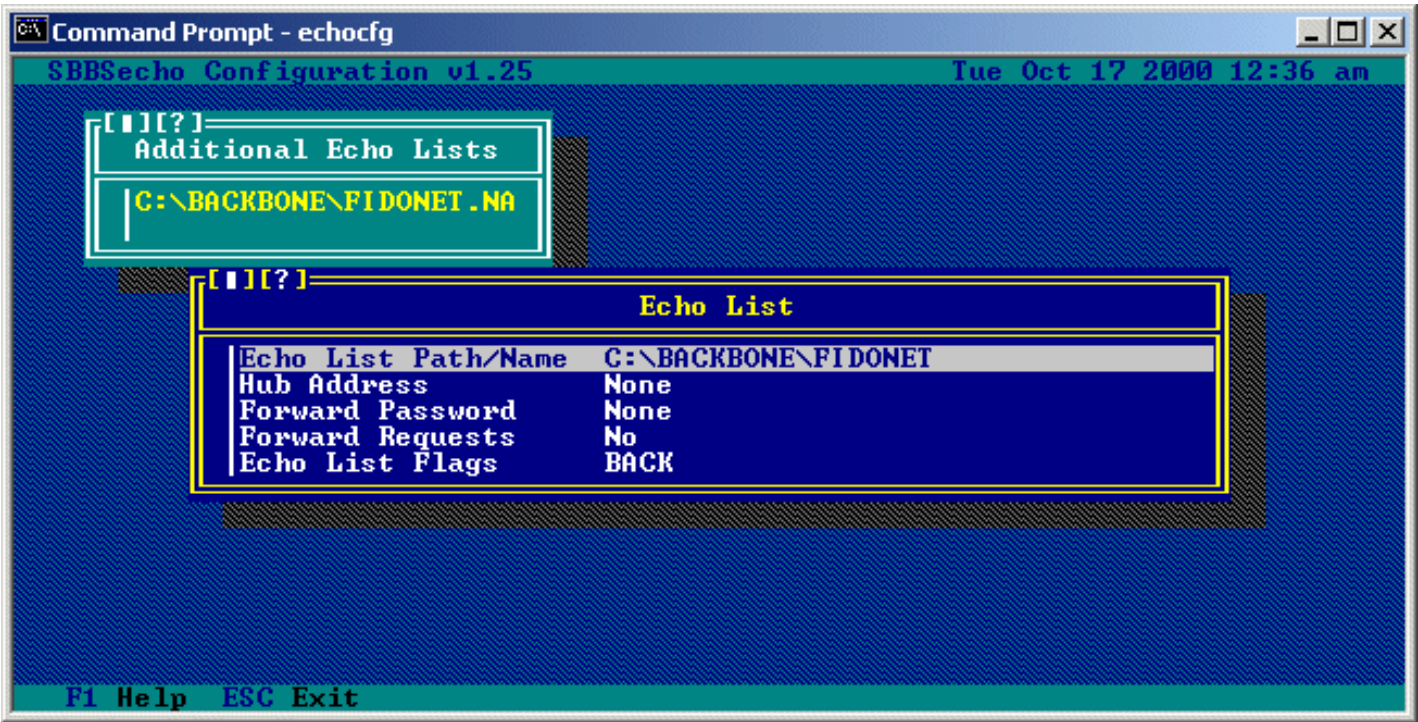
**Additional Echo Lists...**  
This option allows you to add and remove additional echo lists which

can be used by SBBSecho for area manager add requests. Normally these will be used in addition to your AREAS.BBS file. If you have the toggle option 'Allow Nodes to Add Areas in the AREAS.BBS List' set to 'No', you MUST create at least one additional echo list if you wish to allow other nodes to add areas via area manager requests.

Selecting this option will bring you to a sub-menu listing any additional echo lists you currently have defined:



Pressing the INSert key will allow you to add a new echo list, pressing the DElete key will remove the currently highlighted echo list, and pressing ENTER on the currently highlighted echo list will allow you to edit information about that list:



**Echo List Path/Name:**  
This is the full path and filename of the echo list you are defining. This list should contain the areatag names of areas, one per line, with any comments separated from the areatag by at least one space.

**Hub Address:**  
This is the address of the hub of the conferences contained in this list. If an area is remotely added from this list (using Areafix) this address is automatically added to the AREAS.BBS file.

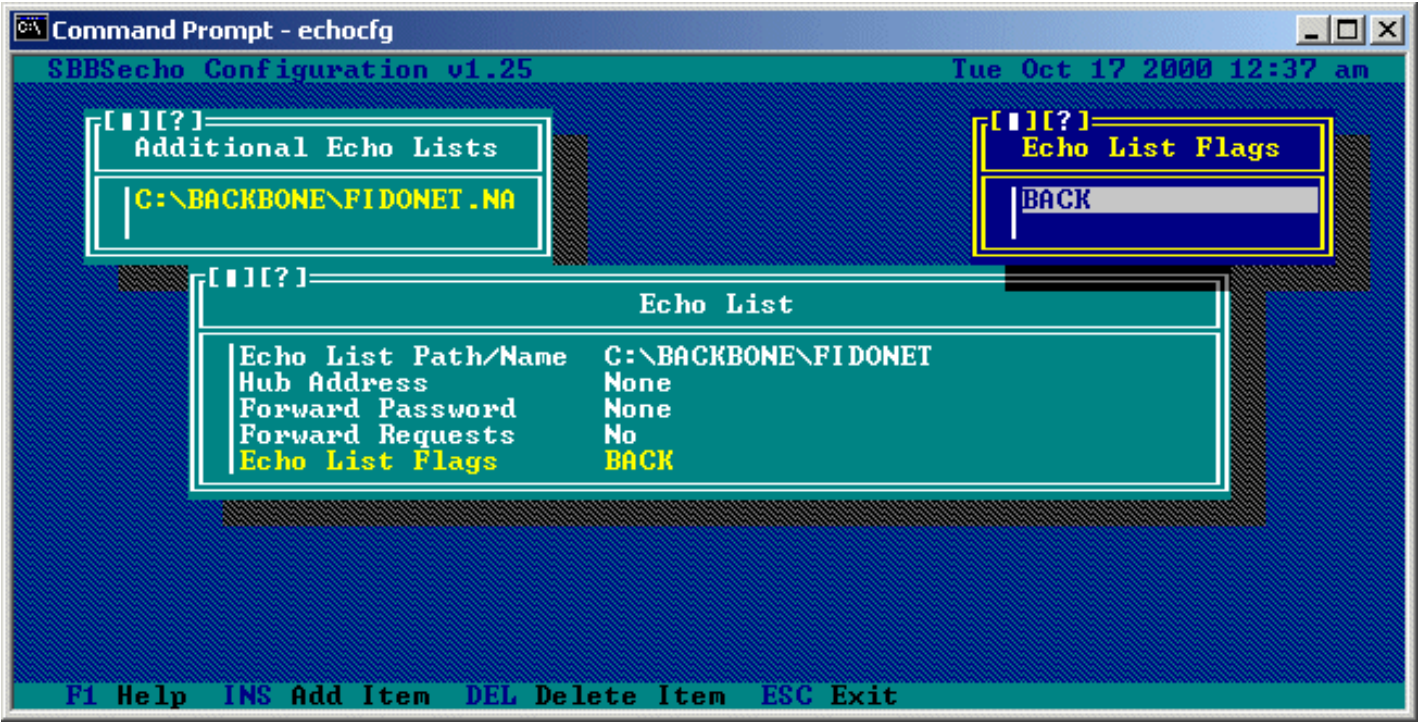
**Forward Password:**  
This is the area manager password to use when forwarding requests.

**Forward Requests:**  
Setting this option to 'Yes' will cause SBBSecho to send a request to the Hub Address specified to turn on an area from this list. This happens when users remotely add areas using Areafix and is not necessary if you are already receiving the conferences in this list (i.e.: from Planet Connect).

**Echo List Flags:**  
These are the flags required for a node to be able to gain access to this particular echo list. These flags are defined for each node from the 'Nodes...' sub-menu.



Selecting this option will bring you to a sub-menu where you may add and remove flags for this echo list.



[Back to Table of Contents](#)

## Running SBBSecho

The command line for executing SBBSECHO is very similar to SBBSFIDO, although some command line switches have been eliminated.

We suggest you use the following command lines:

For importing:

**C:\SBBS\EXEC\SBBSECHO /LES!**

For exporting (Attach Mailer):

**C:\SBBS\EXEC\SBBSECHO /LIN**

For exporting (FLO Mailer):

**C:\SBBS\EXEC\SBBSECHO /LINF**

Remove all references to SBBSFIDO from your batch files (if you have any). See the included MAILER.BAT as an example.

[Back to Table of Contents](#)

## Command Line Switches

**usage: sbbsecho [cfg\_file] [/switches] [sub\_code]**

where: cfg\_file is the filename of config file (default is ctrl\sbbsecho.cfg)  
sub\_code is the internal code for a sub-board (default is ALL subs)

SBBSecho will execute normally (for Attach Mailer operation) with NO command line switches required. Command lines switches are used to cause SBBSecho to \_do\_ or \_not\_ do specific operations. When used, command line switches are prepended by a slash (/) character. Multiple switches can be specified in one grouping (multiple /switch sets are supported, but not required). Switch letters are not case sensitive. The available switch letters and their descriptions are:

- P: Do not import inbound packets or extract inbound bundles
- X: Do not delete inbound packets (\*.PKT) after they're imported
- N: Do not import inbound netmail (packetized netmail and \*.MSG)
- D: Do not delete inbound netmail (\*.MSG) after they're imported
- I: Do not import inbound echomail
- E: Do not export outbound echomail
- M: Ignore echomail pointers (export all locally created echomail messages)
- U: Update echomail pointers only (export no echomail messages)
- T: Do not update echomail pointers (test export)
- H: Export all echomail messages (including messages imported from FidoNet)
- J: Ignore received bit on netmail (import even though flagged as received)
- L: Create log file (data\sbbsecho.log)
- R: Create report of import totals (text\sbbsecho.msg)
- B: Import locally created netmail too (ignore the local flag)
- A: Export ASCII characters only (override individual sub-board ex-ASCII strip)
- F: Create packets for outbound netmail (necessary for FLO Mailer operation)
- G: Generate notify lists
- Y: Import netmail for unknown users to sysop (user #1)
- O: Import all netmail regardless of destination address (unless flagged local)

S: Import private echomail override (strip private flag)  
=: Change existing tear lines to === when exporting  
!: Notify users via telegram of received echomail messages

[Back to Table of Contents](#)

## AREAFIX Commands

Remote area manager, or areafix, commands are used by uplinks to be able to turn echo areas off and on, list currently connected areas, and more.

The following text is from the file AREAMGR.HLP which, after installation of SBBSecho, should be located in your Synchronet EXEC directory. It lists the area manager commands available to the nodes which have been configured using the ECHOCFG program:

Address all requests to 'SBBSecho' or 'AreaFix' (without quotes).  
Your Area Manager password goes on the subject line.

In the body of the message to Area Manager:

```
[+]<areaname>      Connect an area
-<areaname>      Disconnect an area
%HELP           Request this message
%LIST          Request a list of areas available to you
%QUERY         Request a list of areas to which you are connected
%UNLINKED      Request a list of areas to which you are not connected
%COMPRESSION <type> Change the compression type (ARC/ARJ/LZH/PAK/SQZ/ZIP/ZOO)
%PASSWORD <password> Change your AreaMgr password
%RESCAN        Request a rescan of newly added areas
%ACTIVE        Reconnect all temporarily disconnected areas
%PASSIVE       Temporarily disconnect all connected areas
%FROM <address> Remote maintenance, must be the first command
%+ALL          Connect all available areas
%-ALL          Disconnect all areas
[---]          Everything below the tear line is ignored
```

NOTE: A compression type of NONE is also supported for uncompressed packets.

[Back to Table of Contents](#)

---

### [Copyright](#) © 2000 by [Rob Swindell](#)

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .



## Configuring Synchronet Version 3: Answers to Frequently Asked Questions

-----

\$Id: v3cfgfaq.txt,v 1.8 2009/02/21 22:51:55 rswindell Exp \$

Q: DOS programs (e.g. editors, door games) won't run on my Windows Vista-32 system running Synchronet-Win32.

A: There are 2 new requirements for 16-bit DOS programs to work under Synchronet-Win32 on Windows Vista-32:

1. You must have SBBSEEXEC.DLL rev 38 or later (this is included in SBBS v3.14a).
2. You must copy SBBSEEXEC.DLL to your Windows "System32" folder.

Q: DOS programs won't run on my 64-bit Windows system.

A: Microsoft does not include Virtual DOS Machine (NTVDM) support in their 64-bit operating systems. Synchronet for Windows relies on NTVDM for 16-bit DOS program support, so you need to either run a 32-bit Windows (e.g. re-install, use dual-boot, or run in a Virtual Machine) or forego DOS programs on your BBS.

Q: I want to create custom menus. Is there any detailed documentation or samples regarding custom menus on Synchronet?

A: Yes, see the customization chapter in the Synchronet Sysop Manual (<http://synchro.net/docs/customization.html>), the Baja Manual (<http://synchro.net/docs/baja.html>), or Synchronet JavaScript page (<http://synchro.net/docs/js.html>) for details.

Q: How do I get on the Synchronet BBS List (<http://synchro.net/sbbslist.html>)?

A: The best way is to first join DOVE-Net, then run the Synchronet BBS List (SBL) door on \*your\* BBS and add an entry for your BBS. This entry should be automatically exported to the DOVE-Net SYNCDATA echo which will then be propagated to Vertrauen and every other BBS on DOVE-Net. The sbbslist.html page is automatically generated on Vertrauen every night at midnight (Pacific time), so wait a day or so for your entry to appear on the list. If it doesn't appear, verify that that your BBS entry was properly exported (as a message) to the SYNCDATA message area. The events that run SMB2SBL and SBL2SMB to import and export BBS entries should be configured by default. If they are not, see [xtrn/sbl/sbl.doc](#) for more details.

Q: How do I get my BBS listed in the Instant-Message capable BBS list (<ftp://ftp.synchro.net/sbbsimsg.lst>)?

A: First get your BBS on the Synchronet BBS List ([sbbslist.html](http://synchro.net/sbbslist.html)), and if your BBS is running the Synchronet SMTP server on port 25 (like it should) and the Synchronet Finger Service on TCP and UDP port 79, your BBS should automatically appear in the instant message list. If it is not, then verify that your SMTP and Finger services can be reached remotely (the ports aren't being blocked by a firewall or your ISP). This list is generated nightly on Vertrauen at the same time as the Synchronet BBS List (<http://synchro.net/sbbslist.html>). See [docs/imsdocs.txt](#) for more details.

Q: How do I know what ports Synchronet needs opened through my firewall?

A: The answer depends on what Synchronet servers and services you have enabled and whatever ports you have configured Synchronet to use for each. See [http://synchro.net/docs/tcpip\\_faq.html](http://synchro.net/docs/tcpip_faq.html) for details.

Q: How do I join DOVE-Net?

A: If you've installed v3.10 or later, Telnet to [vert.synchro.net](http://vert.synchro.net) and create a new account for your QWKnet transfers (using your system's QWK-ID for the user name), then edit your "Call-out Command Line" in SCFG->Networks->QWK->Hubs->VERT: replace "YOURPASS" with the password you used when creating your QWKnet user account. Do NOT change anything else in the command-line! That's it! For more details, see [docs/dove-net.txt](#).

Q: What is my system's QWK-ID?

A: This is normally an abbreviation of your BBS's name that YOU determine and configure in SCFG->Message Options. You may want to look at NODES.DAT in DOVE-NET.ZIP for any conflicts with other DOVE-Net systems' QWK-IDs.

Q: How do I get my FTP server to work behind my firewall/NAT/router?

A: See [http://synchro.net/docs/tcpip\\_faq.html](http://synchro.net/docs/tcpip_faq.html) for details.

Q: How do I disable the New User Feedback requirement?

A: In sbbs v3.10-win32, run the Configuration Wizard and un-check the "Require new user feedback" checkbox. Otherwise, set SCFG->Nodes->Node 1 ->Advanced Options->Validation User to "0".

Q: What is the "!QWKTOMSG failed" or "Duplicate message" warnings in my Telnet/Event log while importing DOVE-Net QWK packets into my message areas?

A: This warning indicates a duplicate message in your QWK packet was detected. Duplicate messages in the SYNCDATA echo are normal and you can disable the duplicate message checking for that echo in SCFG->Message Areas->DOVE-Net

->SYNCDATA and the warnings should go away.

Q: How do I join/setup a FidoNet Technology Network (FTN)?

A: Use SBBSecho (included with Synchronet) to toss (export) and scan (import) your FidoNet bundles, packets, and NetMail (see <http://synchro.net/docs/sbbsecho.html> for details).

You will also need a TCP/IP FTN transport program like Binkd, Internet Rex, Argus, Beemail, etc. Traditional Front-End Mailers (FrontDoor, InterMail, D'bridge, BinkleyTerm, etc) are NOT normally used for TCP/IP-based FTN transfers.

See <http://net229.darktech.org/fidosetup/> for instructions on setting up Synchronet with the Argus 32-bit FidoNet mailer for Windows.

I, personally, use Binkd for Win32 or Unix/Linux, which you can find available for download on Vertrauen and the docs (what little there are) online and example setup in docs/binkdex.html.

Q: How do I import newsgroups into my board.

A: Synchronet v3.10 comes with NewsLink, an external JavaScript module for importing and exporting messages via NNTP (Network News Transfer Protocol). See docs/newslink.txt for details.

Q: How do I keep external programs (doors) from popping up windows and stealing the keyboard/mouse focus (Windows)?

A: Check SBBCTRL:Telnet->Configure->Minimize Externals.

Q: How do I set additional nodes for telnet access (Windows)?

A: Create the additional nodes in SBBCTRL:BBS->Configure->Nodes if you haven't already done so (Version 3 uses the same configuration information from the first telnet node for all telnet nodes and the COM port/modem configuration information is ignored). Set SBBCTRL:Telnet->Config->Last Node to highest node number you want available for telnet logins.

Q: How do I add files to the file database?

A: The easiest way is to use the ADDFILES utility which can automatically detect and import embedded descriptions (e.g. FILE\_ID.DIZ). See the chapter in the sysop manual on "Adding files" and the ADDFILES utility chapter for details. No, a files.bbs list is not necessary for adding files. Use "ADDFILES \* /DIZ" to search all directories for a new files.bbs, import any embedded descriptions (e.g. FILE\_ID.DIZ), and delete the file.bbs after importing it. If you don't have a file list (e.g. files.bbs) to import just use "ADDFILES \* /SZN" to search all directories for new files and use embedded (FILE\_ID.DIZ) description files, if they exist.

Q: Long filename support doesn't seem to be fully functional.

A: Support for files with long filenames (greater than the MS-DOS 8.3 format) is a work in progress.

The file database format is being replaced in v4.00 and will store long filenames as well as many other improvements.

Q: How do I enable forced upload/download ratios.

A: By default, Synchronet utilizes a "credit" system, where a user typically earns credits by uploading files, posting messages, playing online games, etc. And downloading files requires a number of credits based on the size of the file.

If you prefer to use a less flexible upload/download "ratio" system, you can do so by using the UDR (upload/download byte ratio) or UDFR (upload/download file ratio) ARS keywords in the Download Requirements of your file areas. These ARS keywords, like the PCR (post/call ratio) keyword use a specified minimum "percentage". So if you wish to enforce a 1:3 upload/download ratio, then use "UDFR 33" for 33%, 3:4 ratio would be "UDFR 75" for 75%, 1:2 would be "UDFR 50" for 50%, etc.

See <http://synchro.net/docs/security.html> for details on using ARS.

Q: How do I allow users to send Internet e-mail from the BBS?

A: See [http://synchro.net/docs/message\\_section.html](http://synchro.net/docs/message_section.html) for details.

Q: How do I disallow connections from specific host names or IP addresses (or ranges).

A: Add the disallowed host names to your text/host.can file or IP addresses to your text/ip.can file. To disallow a range of IP addresses, use the ^ wildcard character (e.g. "192.168.10.^"). To disallow a partial string in a host name, use the ~ wildcard character (e.g. "mofoisp.com~"). in Synchronet v3.10+, you can use the familiar '\*' wildcard character (e.g. "\*.aol.com" and "192.168.\*").

Q: How do I allow anonymous logins to the Synchronet FTP Server?

A: You need to create a Guest account on your BBS. Create a new user (login as "New" or use the User Editor included with v3.00b) with a user name of "Guest". Give this user the 'G' exemption and the 'G', 'M', and 'F' restrictions and delete the password for the account. You may also want to give the Guest account additional exemptions (e.g. L, P, T, etc).

A: Or preferably, run the exec/makeguest.js script included with v3.11+.

Q: How can I get door X to work?

A: If the door wasn't specifically written to run on Synchronet (XSDK) or WWIV BBS Software (typically using CHAIN.TXT), then it must support FOSSIL communications (most popular doors support FOSSIL) and the door must be configured to use a FOSSIL driver (yes, you'll need to read the door's documentation files).

The Synchronet FOSSIL driver is automatically configured and loaded when needed and ignores the COM port information used by doors, so configuration details such as COM port number and baud rate are irrelevant (you can just leave them set to their default values if the door configuration program allows them to be changed, or use COM1/38400 if it makes you happy).

Q: How can I get this FOSSIL door to work?

A: Make sure you have "Intercept I/O Interrupts" set to "No" for all FOSSIL doors in SBBSCtrl:BBS->Configure->External Programs->Online Programs (this includes Synchronet Match Maker v2.x).

Make sure you have the appropriate startup directory, command line, and drop file options set in SCFG. See SBBSDOOR.TXT for specific instructions on setting up some popular door games (i.e. LORD, BRE, GlobalWar, TW2002).

Make sure you DO NOT have any FOSSIL drivers loaded (including X00, BNU, ADF, COM/IP, WinFOSSIL, or NetModem) as they will conflict with Synchronet's FOSSIL driver.

Make sure the door supports FOSSIL communications (read the door's docs) and that the door is configured to use a FOSSIL driver. Synchronet's FOSSIL driver is automatically configured and loaded when needed - there is nothing you need to do to install or load the Synchronet FOSSIL driver.

As always, see the documentation if you have any questions that aren't answered here: <http://synchro.net/docs>

/\* End of v3cfgfaq.txt \*/

# Synchronet Version 3 and TCP/IP

## Answers to Frequently Asked Questions

---

### Table of Contents:

- [What inbound ports do I need to open in my firewall?](#)
  - [How come my friends can't connect to my BBS at my IP address?](#)
  - [What is my public IP address?](#)
  - [Why can't I relay Internet e-mail through my BBS?](#)
  - [Why can't I send Internet e-mail from my BBS?](#)
  - [Why can't my BBS receive Internet e-mail or inter-BBS instant messages?](#)
  - [Why can't users connect to my FTP server?](#)
  - [Why do FTP clients lock-up or time-out when listing directories or downloading files from my FTP server?](#)
  - [Why do external programs that use socket I/O not work on my Windows BBS?](#)
  - [Why do some or all of my servers get bind errors when starting or recycling?](#)
  - [How many nodes/clients/users can I support with my Internet connection?](#)
- 

#### Question:

*What inbound ports do I need to open in my **firewall**?*

#### Answer:

Depends on which Synchronet servers and services you wish to make available to Internet clients and which ports you have configured those servers and services to listen on.

The default Synchronet installation enables servers and services on the following ports:

| Protocol | TCP  | UDP | Comments                                                                                           |
|----------|------|-----|----------------------------------------------------------------------------------------------------|
| Telnet   | 23   | -   | For Telnet logins (highly recommended)                                                             |
| SSH      | 22   | -   | For SecureShell logins (recommended)                                                               |
| RLogin   | 513  | -   | Optional for quick-login from RLogin clients (e.g. SyncTerm)                                       |
| SMTP     | 25   | -   | Necessary for receiving Internet e-mail and inter-BBS instant messages                             |
| POP3     | 110  | -   | Allows BBS users to check their e-mail using standard Internet mail clients (e.g. Outlook Express) |
| FTP      | 21   | -   | Allows access to the BBS file/download areas using a standard FTP client or web browser            |
| HTTP     | 80   | -   | Required for access to the BBS's web server                                                        |
| NNTP     | 119  | -   | Allows BBS users to read and post messages using standard news readers/clients                     |
| Gopher   | 70   | -   | Archaic protocol allows reading of messages and other BBS info                                     |
| IRC      | 6667 | -   | Allows Internet Relay Chat (IRC) clients to connect to your BBS                                    |
| Finger   | 79   | 79  | Allows remote querying of BBS user info, who's online, and other BBS info                          |
| SYSTAT   | 11   | 11  | Allows remote querying of who's online (aka Active Users)                                          |
| QOTD     | 17   | 17  | Allows remote querying of the current auto-message (aka Quote Of The Day)                          |
| MSP      | 18   | 18  | Allows incoming inter-BBS instant messages without SMTP connectivity                               |

Enabling connectivity to Synchronet through your firewall is **no different** than enabling connectivity to any other TCP/IP server. Follow your firewall documentation for forwarding or opening ports for TCP/IP servers located "behind" the firewall. Your firewall may have the option of placing the entire BBS computer in a "DMZ" (opening **all** its ports to the public Internet), but doing so is not normally recommended.

---

#### Question:

*Howcome my friends can't connect to my BBS at my **192.168.x.x**, **172.[16-31].x.x**, or **10.x.x.x** IP address?*

#### Answer:

The IP address ranges listed above are reserved for use in **private** networks and are **not publicly addressible** from the Internet. See this [document](#) for technical details.

You do **not** want to advertise this IP address to the public since it is useless to anyone outside of your own private/local area network (LAN). IP addresses in these ranges are typically assigned to your computer by your router/firewall device (using DHCP) to allow multiple computers on your **private** network to share the same **public** IP address using a mechanism known as *Network Address Translation* ([NAT](#)). Clients on the Internet must use the IP address of your router/firewall device's public/WAN port to connect to your BBS. This IP address **will not** begin with *192.168*, *172.[16-31]*, or *10*.

---

**Question:**

*What is my **public** IP address?*

**Answer:**

If you need to know your public IP address, you can usually query your router/firewall device using its configuration interface (typically via Telnet or HTTP to its private/LAN port) or access [any one](#) of [many](#) public web-sites that can tell you what your public IP address is. However, it is usually much better to advertise a **hostname** (e.g. *vert.synchro.net*) rather than a cryptic hard-to-remember IP address (e.g. 69.104.209.211).

If you use a [Dynamic DNS](#) service to get a hostname for your BBS, they can usually correctly determine your public IP address automatically, even if your IP address changes. So you don't **need** to necessarily know what it is.

---

**Question:**

*Why can't I **relay Internet e-mail** through my BBS?*

**Answer:**

Indications of this problem include error messages in your e-mail client similar to the following:

```
553 Relaying through this server requires authentication. Please authenticate before sending.
```

```
550 Relay not allowed.
```

Or error messages similar to the following in your Synchronet Mail Server window/log output:

```
0504 !SMTP ILLEGAL RELAY ATTEMPT from <yourname@yourhost.org> [192.168.0.2] to someuser@somehost.com
```

Where the *from* address is that of your mail sending host and the *to* address is that of an external mail recipient that **you** are attempting to send e-mail to.

It is common and normal to see "illegal relay attempts" from remote systems that have happened on your mail server and are probing it to determine if it is an "Open Relay". So called "Open Relays" (mail servers that allow any and every host to relay mail through them) are a major source of SPAM on the Internet and are highly discouraged.

By default, the Synchronet Mail Server **disallows** the *relaying* of SMTP e-mail messages received for an external recipient (not destined for a local BBS user account).

You **can** allow specific hosts or users to relay e-mail through your mail server by either:

1. Entering the sending host's IP address or hostname in your `ctrl/relay.cfg` file.  
This file may be edited with the SBBSCtrl:Mail->Edit->Allowed Relay List menu option.

or:

2. Use SMTP authentication:
  - a. Enable the mail server configuration option to allow authenticated users to relay mail.  
This can be done by adding `ALLOW_RELAY` to the `Options` key of the `[mail]` section of your `ctrl/sbbs.ini` file.  
Or, if using SBBSCtrl, checking the "Allow Authenticated Users to Relay Mail" checkbox on the SMTP tab of the Mail Server Configuration dialog.
  - b. Configure your e-mail client to use SMTP authentication to login to your mail server using your BBS user name (i.e. alias) and password.

The Synchronet Mail Server supports the following SMTP authentication schemes:

- PLAIN
- LOGIN
- CRAM-MD5

(Note: password case sensitivity can be an issue when using CRAM-MD5 authentication)

---

**Question:**

*Why can't I **send Internet e-mail** from my BBS?*

**Answer:**

You must have the Synchronet *SendMail thread* enabled in your Synchronet Mail Server configuration.

If you **do not** see the following message in your Synchronet Mail Server window/log output when the server is started or recycled:

```
0000 SendMail thread started
```

then you do not have the SendMail thread enabled and your system cannot deliver any Internet e-mail messages until it is enabled and recycled (delivery of any previously queued outbound messages will be attempted at that time).

If your Synchronet SendMail thread cannot deliver e-mail messages, it could be for any of the following reasons:

1. You have your mail server configured for "Direct Delivery", but have an improperly configured DNS server IP address.  
Example errors indicating this condition include:

```
0000 !SEND INVALID DNS server address
```

```
0000 !SEND ERROR -1 obtainingMX records for someserver.com from 192.168.1.1
```

The configured DNS server IP address should usually be set to that of your ISP's primary DNS server.

Note: Synchronet v3.13b can automatically detect and use your DNS server's correct IP address during run-time. This feature is enabled by configuring the DNS server IP address in the mail server configuration to *blank* or `<auto>`.

You'll know this feature is active when you see log lines similar to the following:

```
0000 SEND using auto-detected DNS server address: 206.13.29.12
```

2. Your firewall, Internet Service Provider, or Anti-Virus software is blocking, intercepting, or filtering outbound connections to TCP port 25 (many consumer-class ISPs do this).

Example errors indicating this condition include:

```
0700 !SEND ERROR 60 connecting to SMTP server: smtp.somedomain.com
0023 !SEND ERROR 110 connecting to SMTP server: mx.somesite.org
```

You can verify if this is the case by attempting to Telnet to a known public SMTP server (e.g. *vert.synchro.net*) on TCP port 25.

You should see a mail server version banner similar to the following:

```
220 bbs.synchro.net Synchronet SMTP Server 1.362-Win32 Ready
```

If you cannot connect or do not see a mail server version banner, then *something* is filtering or blocking your outbound connections to TCP port 25.

If your ISP is blocking port 25, they will normally make an exception for their own mail servers (e.g. *mail.yourisp.com* or *smtp.yourisp.com*). If this is the case (and your ISP's mail server allows the *from* address of your e-mail message to contain any hostname or IP address of your choosing), then you need to configure your mail server to use your ISP's mail server as its relay server. **Do not** use your own mail server's hostname or IP address as the relay server (this will cause an undesirable message "loop").

If your ISP's mail server **only** allows e-mail to be sent from *somename@yourisp.com* then you need to contact your ISP about how you can send e-mail from a different domain using their mail server. Perhaps they only allow this feature when using SMTP authentication?

3. You have your mail server configured to use an external "Relay Server", but have an improperly configured relay server hostname or IP address.

Example errors indicating this condition include:

```
0000 !ERROR resolving hostname: badhostname.com
0680 !SEND ERROR 60 connecting to SMTP server: 192.168.1.1
```

4. You have your mail server configured to use an external "Relay Server", but the specified relay server requires SMTP authentication in order to allow relaying of mail.

Example errors indicating this condition include:

```
0000 !Delivery attempt #1 FAILED (somehost.org replied with: "550 Relay not allowed." instead of the expected
reply: "250 ...")
0000 !Delivery attempt #1 FAILED (somehost.org replied with: "553 Authentication required." instead of the
expected reply: "250 ...")
```

Synchronet v3.12+ supports the *Plain*, *Login*, and *CRAM-MD5* methods of SMTP authentication when relaying mail through an external relay server. To enable SMTP authentication when relaying, add one of the `RELAY_AUTH` flags to the `Options` value in the `[Mail]` section of your `ctrl/sbbs.ini` file. Or, if running SBBSCtrl-Win32, enable one of the authentication radio buttons on the "Relay" tab of the Mail Server Configuration dialog.

5. You have a message in your outbound e-mail queue that is flagged as *'in transit'*. If you're running only *one* instance of the Synchronet Mail Server, this is not a normal condition and the affected message **will not be sent** until the *'in transit'* flag is cleared.

Example log message indicating this condition:

```
0000 SEND Message #999 from Some User to someone@somesite.com - in transit
```

This condition can occur if the Synchronet SendMail thread is terminated unexpectedly while in the process of attempting the delivery an outbound e-mail message. The *'in transit'* flag is used to protect multiple instances of the SendMail thread from attempting to deliver the same e-mail message simultaneously.

If you only have one instance of the Synchronet SendMail thread active (the usual scenario), you can eliminate this problem by adding `SEND_INTRANSIT` to the `Options` value in the `[Mail]` section of your `ctrl/sbbs.ini` file. Or you can remove the *'in transit'* flags from all existing e-mail messages by running the `fixsmb` utility on your `data/mail` database or by running the `exec/notransit.js` script.

In general, you need to check your Synchronet Mail Server window/log output for details about why Internet e-mail delivery attempts may be failing.

---

### Question:

*Why can't my BBS receive Internet e-mail or inter-BBS instant messages?*

### Answer:

You must have the Synchronet SMTP (mail) server running and listening for incoming connections on TCP port 25 (the standard SMTP



port). You (or a friend) can test this basic connectivity by attempting to Telnet to port 25 (instead of port 23) at your BBS's hostname or public IP address from a remote location on the Internet. The remote Telnet client should see a successful connection and a text message similar to the following:

```
220 bbs.synchro.net Synchronet SMTP Server 1.362-Win32 Ready
```

You should also see evidence of the successful SMTP connection to the server in your Synchronet Mail Server window/log output. If you do not, then it's likely that your firewall or Internet Service Provider is blocking incoming connections to TCP port 25. Before concluding this is the case, verify that the remote Telnet client can connect to other SMTP servers first (e.g. *vert.synchro.net*, TCP port 25). If it cannot, then this remote client probably has restrictions on which (if any) connections he can make to TCP port 25. Try using a different, less restricted, remote Internet connection for your test.

If your firewall or Internet Service Provider is blocking incoming connections to TCP port 25 (many consumer-class ISPs do), then you won't be able to receive Internet e-mail on your BBS. Fixing your firewall configuration is rather simple, but changing ISPs is often not. One possible work-around is having a mail proxy (3rd party server) receive the e-mail for you and forward it to a non-standard, non-filtered/blocked SMTP port. Many Dynamic DNS services offer this [service](#) for a fee. Or a fellow sysop may be able and willing to perform this service for you as a favor.

---

#### Question:

*Why can't users connect to my FTP server?*

#### Answer:

You must have the Synchronet FTP server running and listening for incoming connections on TCP port 21 (the standard FTP port). See the previous answer about methods of testing this basic connectivity using a remote Telnet client.

If your FTP server window/log indicates an accepted FTP connection, then it's not a connectivity problem and probably a login failure.

FTP sessions require a **login**. If you have not created a *Guest* account for your BBS, then the FTP server will not allow *Anonymous* logins (most web browsers, for example, will attempt to login anonymously by default). If this is the problem, then either create a *Guest* account (preferably using the `exec/makeguest.js` module) or tell your FTP users that they must login with a valid BBS user account in order to use the FTP server.

---

#### Question:

*Why do FTP clients lock-up or time-out when listing directories or downloading files from my FTP server?*

#### Answer:

Your BBS computer is probably behind a *Network Address Translator* ([NAT](#)). NAT functionality is typically built into router/firewall devices. If your NAT device supports active and passive FTP servers "behind" the NAT, then you should have no problems. Unfortunately, this is not always the case: some cheaper consumer-level firewalls do not handle FTP server connections correctly or they do not support FTP servers listening on non-standard ports. Sometimes **passive** (PASV) transfers work fine, but **active** (PORT) transfers do not, or vice versa.

This [document](#) contains the technical details about how and why and the recommended solutions.

Note: Most web browsers (e.g. *Microsoft Internet Explorer*) use **passive** FTP transfer mode by default.

Note: Some FTP clients (e.g. the Windows command-line FTP client) **only** support **active** mode transfers.

Enabling the logging of FTP data channel activity can help diagnose these kinds of problems. This can be done by adding the `DEBUG_DATA` option to the `Options` value in the `[FTP]` section of your `ctrl/sbbs.ini` file or by checking the **Data Channel Activity** checkbox in the *Log* tab of the *FTP Server Configuration* dialog in the *Synchronet Control Panel* for Win32.

If you're having problems with **passive** transfers and you're seeing `!UNSUPPORTED COMMAND from username: 'P@SW'` in your FTP server log/window output, you're probably using an **SMC Barricade** router (see this [document](#) for details). Upgrade to Synchronet v3.13a (FTP Server Revision 1.296) or later to work-around this problem with this device.

If you're having problems with **passive** (PASV) transfers through your NAT/firewall device and you're running Synchronet v3.13a (FTP Server Revision 1.296) or later:

- If the remote client is attempting to connect to your [private IP address](#) (your NAT device isn't *translating* the PASV response from the FTP server) and you have a **static** public IP address, you can work-around this limitation of your NAT device by using the `PasvIpAddress` value in the `[FTP]` section of your `ctrl/sbbs.ini` file to specify your [public IP address](#).

This problem can be identified (on the client) by finding a comma-separated [private IP address](#) in the PASV response received from the FTP server (in response to a directory or file transfer request from the client).

Example:

```
# ftp yourbbs.synchro.net
Connected to yourbbs.synchro.net (70.19.142.182).
220 Please enter your user name.
Name (yourbbs.synchro.net:root): anonymous
```

```
331 User name okay, give your full e-mail address as password.
Password:
230 Guest logged in.
ftp> passive
Passive mode on.
ftp> dir
227 Entering Passive Mode (192,168,56,2,4,0) <--- IP address reported as 192.168.56.2 instead of 70.19.41.182
```

Use an FTP client that supports passive mode **and** can display all the responses received-from the FTP server to help identify this particular problem. The FTP client must be running on a system *outside* your private network, so you may need a friend to assist you with this.

If you have a dynamically-assigned IP address (via DHCP), then your IP address may change at some point, so setting the `PasvIpAddress` to a specific IP address may not be a long term solution for your FTP Server. In Synchronet v3.14a and later, you can enable the new *Lookup Passive IP* feature by checking the *Lookup* checkbox on the *Passive* tab of the *FTP Server Configuration Dialog* in SBBSCtrl-Win32, or by adding `LOOKUP_PASV_IP` to the `Options` value in the `[FTP]` section of your `ctrl/sbbs.ini` file. This option instructs the Synchronet FTP Server to perform a DNS hostname lookup on your BBS's public hostname and use the resulting IP address (which **should** be your BBS's public IP address) in passive responses.

- If your firewall cannot dynamically open/forward FTP PASV data ports for incoming passive FTP data connections, you can specify a limited *range* of TCP port numbers to use for passive transfers by modifying the `PasvPortLow` and `PasvPortHigh` values in the `[FTP]` section of your `ctrl/sbbs.ini` file. You will of course need to configure your firewall device to open/forward these ports to your FTP server.

---

#### Question:

*Why do external programs that use socket I/O (e.g. Synchronet Blackjack, Synchronet BBS List, DoorMUD, SEXYZ) not work on my Windows BBS?*

#### Answer:

Some "security" software (e.g. firewall and anti-virus programs) will interfere with the inheritance of socket descriptors between processes. One such program is the **ZoneAlarm Security Suite**. Another is **NOD32**. I don't know if this is an intentional security "feature" or a **design flaw**. If you have this (or similar) software installed, it may need to be disabled or **completely un-installed** for socket inheritance to work again.

---

#### Question:

*Why do some or all of my servers get **bind errors** when starting or recycling?*

#### Answer:

If you're getting bind errors when first starting up one or more Synchronet servers, similar to the following:

```
0420 !ERROR 48 binding FTP Server socket to port 21
```

this usually means you have another TCP/IP server on your system that is already bound to (and is presumably already listening for incoming connections on) this port. This could be a pre-existing instance of Synchronet or any other Telnet/Web/Mail/FTP servers that you may have installed on your system. You can use utilities such as [netstat](#) (for Windows or Unix) or [TCPView](#) (for Windows) to verify what programs (if any) have the TCP or UDP port in question already bound. If these utilities do not report any program is bound to (and listening) on this port, you can try Telnetting to the port in question and see if anything answers. If you're unable to connect to the port with a Telnet client and Synchronet cannot bind the port, your TCP/IP stack probably needs to be reset, so a system reboot may be in order.

If you're running a **Unix**-like operating system (*not* Windows) and get bind errors only when recycling servers, this is most likely because a TCP session is stuck in a TCP *TIMEWAIT* state (you can use [netstat](#) to verify this). The session will eventually time-out and close properly on its own, allowing the port to be re-bound at that time. To work-around this problem, you can either increase the `BindRetryCount` **and/or** `BindRetryDelay` values in your `ctrl/sbbs.ini` file, or you can add the following line to your `ctrl/sockopts.cfg` file:

```
REUSEADDR 1
```

Or, if running Synchronet v3.13b or later, your `ctrl/sockopts.ini` file:

```
REUSEADDR=1
```

---

#### Question:

*How many nodes/clients/users can I support with my Internet connection?*

#### Answer:

Depends on what those clients will be doing while connected. Here are some facts to consider:

1. **A BBS node doesn't consume any bandwidth when not *in use*.**
2. **An active TCP session doesn't consume any appreciable bandwidth when *idle*.**

### 3. **Most Internet connections are *asymmetrical* in nature (as in ADSL).**

This means your *upstream* channel usually has less bandwidth than your *downstream* channel.

When TCP/IP clients (users of your BBS's servers) *download* content from your servers (this includes viewing menus, reading messages, and playing door games on your BBS), they are primarily using your *upstream* channel.

So if you have a 1.5Mbps/128Kbps DSL connection, your *downstream* is 1.5Mbps while your *upstream* is only 128Kbps. If you have a "56K" dial-up connection, for example, your *downstream* is probably in the 43-53Kbps range while your *upstream* bandwidth cannot be any more than 33.6Kbps (or 48Kbps for V.92 modems).

If you are lucky enough to have an **SDSL** or other type of *symmetrical* Internet connection, then both your upstream and downstream channels are of the same bandwidth.

### 4. **Most BBS traffic is *bursty*.**

With the exception of large file transfers, most BBS traffic is sent and received in small bursts. For example, the BBS user's TCP session is idle while the user is viewing menus, reading messages, pausing between keystrokes, etc. Many clients sending and receiving data in small intermittent bursts can be active simultaneously without any perceptible impact on one another.

### 5. **Not all clients will be capable of saturating your upstream channel.**

If you have a 256Kbps upstream channel, for example, you could support four or five simultaneous "56K" clients all downloading files, and all getting 100% utilization of their respective downstream channels.

\$Id: sbbscon.txt,v 1.31 2006/05/16 19:14:17 deuce Exp \$

## 1. Introduction

-----

The Synchronet Console is the console-mode sysop interface to Synchronet v3. On Win32 operating systems, this interface (sbbs.exe) is an alternative to the graphical Synchronet Control Panel (sbbsctrl.exe). On Unix, this interface (sbbs) is currently the only option.

On Win32 operating systems, the exact same underlying server libraries (DLLs) are used to service the clients, so from the users perspective there should be no difference in performance or functionality. The console mode interface does use fewer system resources than the graphical control panel, so for some Win32 sysops that wish to conserve system resources, this may be the preferred interface. But, to date, this interface has been used almost exclusively by Unix sysops.

## 2. BBS Configuration

-----

The system-wide configuration files (ctrl/\*.cnf) are edited using the Synchronet configuration utility (SCFG). The executable filename is exec/scfg.exe on Win32 and exec/scfg on Unix.

SCFG needs to be told the location of your Synchronet ctrl directory, where it expects to find the Synchronet configuration (.cnf) files. This can be done by passing the path to your ctrl directory on the SCFG command line. Example:

```
scfg /sbbs/ctrl
```

or:

```
scfg /usr/local/sbbs/ctrl
```

or by setting the SBBSCtrl environment variable before running SCFG. Example:

Unix (bash): export SBBSCtrl=/sbbs/ctrl

Unix (sh): SBBSCtrl=/sbbs/ctrl && export SBBSCtrl

Unix (csh): setenv SBBSCtrl /sbbs/ctrl

Win32: set SBBSCtrl=/sbbs/ctrl

Generally speaking, if you are running Linux, your shell is bash. You can verify your shell by typing ``echo \$SHELL'' at a command line.

Many of the Synchronet utilities require this environment variable, so it is suggested you initialize this environment variable in your system startup and/or login scripts. See section 3 for details.

The SCFG application supports multiple forms of user interface. The default interface is currently X11 if available or curses/conio (fullscreen colored text) if not. To use a different interface, use the -iD (stdio), or -iA (ANSI output) command-line options. There are other command-line options to control the character set used, escape key delay, monochrome/color mode, etc. Type "scfg --help" to list all the available command-line options.

There are other system-configuration files that are simple text files to be edited using any ASCII text editor: ctrl/\*.cfg and text/\*.can. Each of these files should contain a description of its purpose and usage.

## 3. Initializing the SBBSCtrl Environment Variable

-----

For Windows NT-based operating systems, goto the Windows Control Panel->System->Advanced->Environment Variables->System Variables->New...

Variable: SBBSCtrl

Variable Value: c:\sbbs\ctrl

(replace c:\sbbs\ctrl with the full path to your ctrl directory)

For Windows 9x-based operating systems, edit your C:\AUTOEXEC.BAT file and add the line:

```
SET SBBSCtrl=c:\sbbs\ctrl
```

(replace c:\sbbs\ctrl with the full path to your ctrl directory)

For Unix bash/sh:

In the home directory of the user the BBS will be running as, edit the file named either .profile or .bash\_profile depending on if you're running bash or sh and add the line:

```
SBBSCtrl=/sbbs/ctrl && export SBBSCtrl
```

(replace /sbbs/ctrl with the full path to your ctrl directory)

For Unix csh/tcsh:

Again, in the home directory of the user the BBS will be running as, in the

file named .tcshrc (for tcsh) or .cshrc (for csh) add the line:

```
setenv SBBSCTRL /sbbs/ctrl
(replace /sbbs/ctrl with the full path to your ctrl directory)
```

#### 4. Server Configuration

-----  
The server/host-specific configuration options are set with sbbs command-line options (run "sbbs help" for a list) or by editing the Synchronet Initialization file (e.g. ctrl/sbbs.ini).

##### 4.1 Initialization File

-----  
The Synchronet initialization file is a plain text file in Windows ini format. Lines beginning with a semicolon (;) character are considered comments and are ignored. Configuration values are grouped into sections. Sections are defined by a "[section\_name]" line. The configuration sections are:

|            |                                                  |
|------------|--------------------------------------------------|
| [Global]   | - Settings applied to all servers and services   |
| [BBS]      | - Settings applied to the Telnet/RLogin server   |
| [Mail]     | - Settings applied to the SMTP/POP3 mail server  |
| [FTP]      | - Settings applied to the FTP server             |
| [Web]      | - Settings applied to the Web server             |
| [Services] | - Settings applied to Synchronet services        |
| [UNIX]     | - Settings applied to operations under Unix only |

Within each section is a list of values (one per line) in the form:  
name=value

It is recommended that sysops new to Synchronet, leave the default values intact unless instructed to do otherwise.

The default initialization file is ctrl/sbbs.ini. A different initialization filename may be used by specifying the path and filename on the sbbs command-line. Example:

```
sbbs /sbbs/ctrl/mybbs.ini
```

If the path and filename of the initialization file is not passed on the command-line, sbbs will use the SBBSCTRL environment variable to determine the location of your Synchronet ctrl directory, where it expects to find either <HOSTNAME>.ini or sbbs.ini.

If you are using the Synchronet Control Panel (for Win32) and want to use your current settings for the Synchronet Console or NT Services, you can export your settings to your ctrl/sbbs.ini file using the File->Export Settings menu option.

#### 5. For Unix Sysops

-----  
You will particularly want to pay attention to the [UNIX] section of your Initialization file.

If you do not want to run Synchronet (and all external programs/doors) as root, you will have to set the User and Group values. Also, the file permissions/ownership should be such that the other user would have read and write access to them. The best way to accomplish this is a command like:

```
chown -R sbbsuser:sbbsgroup /sbbs
```

If you want Synchronet to fork and run in the background as a daemon, logging via syslog rather than the local console, set Daemonize=True in this section.

To configure syslog and the LogFacility, a good default to use is LogFacility=3 Then, in /etc/syslog.conf you will need to add the line:  
local3.\* /var/log/synchronet.log

Depending on how your vendor set up syslog.conf initially, you may also want to exclude local3.\* from other log files (Notably /var/log/messages) how to do this varies with your syslogd implementation, but for BSD based ones (Which BSD and many Linux distros use) you would add ";local3.none" to the end of the first field in the /var/log/messages line.

You will have to create this file manually initially by running:  
touch /var/log/synchronet.log

Then send a HUP to syslogd like so:  
killall -HUP syslogd

You will want to investigate how your system rotates logs and set it up to rotate synchronet.log also.

Further use of the LogFacility setting is beyond the scope of this document. Read your syslog.conf manpage for more information about this. In particular, do NOT use the 'S' setting unless you are familiar with advanced syslogd configuration. The S setting will use different facilities for each feature of Synchronet as appropriate. Specifically, S will use:

```
LOG_AUTH
LOG_DAEMON
LOG_FTP (If available)
LOG_MAIL
LOG_CRON
```

## 5.1 Terminal Capabilities

As you may have noticed by now, most telnet clients designed for use with ANSI BBSes do not display fullscreen Unix programs correctly. Included with Synchronet is a pair of terminal capability definition files that enable you to run native fullscreen Unix programs and have the output displayed correctly in a standard ANSI-BBS terminal. These files are termcap and terminfo, located in your Synchronet install directory. Your system will use one or the other, and it won't hurt to install both. You will need to be logged in as root to install the files.

Installing the terminfo file:

- 1) Get the Synchronet ANSI-BBS terminfo file from:  
[http://cvs.synchro.net/cgi-bin/viewcvs.cgi/\\*checkout\\*/install/terminfo](http://cvs.synchro.net/cgi-bin/viewcvs.cgi/*checkout*/install/terminfo)
- 2) Enter the command ``tic terminfo''

Installing the termcap file:

- 1) Get the Synchronet ANSI-BBS termcap file from:  
[http://cvs.synchro.net/cgi-bin/viewcvs.cgi/\\*checkout\\*/install/termcap](http://cvs.synchro.net/cgi-bin/viewcvs.cgi/*checkout*/install/termcap)
- 2) Enter the command ``cat termcap >> /etc/termcap''
- 3) \*\*\* FreeBSD Only \*\*\* run the command:  
``cap\_mkdb -f /usr/share/misc/termcap /etc/termcap''

Once the terminal capability files are installed, edit the ExternalTermANSI value in the [BBS] section of your .ini file to read:  
ExternalTermANSI=ansi-bbs

Note: Once again, many Linux distros do not have a termcap. This is fine. You do NOT need to install the termcap-compat package. If termcap isn't installed, it means nothing uses it. Only if there is a termcap do you need to add the ansi-bbs termcap definition.

## 6. Running Synchronet

If you've initialized the SBBSCtrl environment variable and edited your BBS and server configuration, you are now ready to run Synchronet. You can do this by simply running exec/sbbs (off of the installation directory).

### 6.1 Running Synchronet Automatically During Boot-up (Unix)

If you want Synchronet to start automatically whenever your system boots, you will need to set that up using the system rc scripts. A few examples are:

Linux:

- 1) Get the Synchronet service run script (init file) from:  
[http://cvs.synchro.net/cgi-bin/viewcvs.cgi/\\*checkout\\*/install/init.d/sbbs](http://cvs.synchro.net/cgi-bin/viewcvs.cgi/*checkout*/install/init.d/sbbs)
- 2) Copy the run script (sbbs) into your /etc/init.d directory (if you don't have this directory, then your Linux distribution isn't supported by this file).
- 3) Add the Synchronet system service:  
# chkconfig --add sbbs
- 4) Start the Synchronet system service:  
# /etc/init.d/sbbs start

FreeBSD:

- 1) Get the Synchronet service run script (init file) from:  
[http://cvs.synchro.net/cgi-bin/viewcvs.cgi/\\*checkout\\*/install/rc.d/sbbs](http://cvs.synchro.net/cgi-bin/viewcvs.cgi/*checkout*/install/rc.d/sbbs)
- 2) Copy the run script (sbbs) into your /etc/rc.d directory  
(If using FreeBSD 4.x, install the rc\_subr port and copy the run script to /usr/local/etc/rc.d/sbbs.sh instead)



- 3) Set up the sbbs settings:  
In one of /etc/rc.conf, /etc/rc.conf.local, or /etc/rc.conf.d/sbbs add the line:  
sbbs\_enable=YES       # Required to run Synchronet
- 4) In one of the files from step three, add appropriate lines from the following (Defaults are shown here):  
sbbs\_flags=""               # Additional command-line flags  
sbbs\_pidfile="/var/run/sbbs.pid"   # Path of pid from your .ini  
sbbs\_dir="/sbbs/"               # Root sbbs path  
                              # The rest of the sbbs\_\*dir derive  
                              # from this be default  
  
sbbs\_ctrldir="\${sbbs\_dir}/ctrl/"  
sbbs\_execdir="\${sbbs\_dir}/exec/"  
sbbs\_program="\${sbbs\_execdir}/sbbs"   # Synchronet binary  
sbbs\_procname="\${sbbs\_program}"   # Process name as seen by ps(1)  
sbbs\_shell="/bin/sh"           # SHELL variable  
sbbs\_user="root"               # User to START sbbs as. If this is  
                              # not root, you cannot bind low ports  
  
sbbs\_group="wheel"           # Group to start sbbs as
- 5) Start the Synchronet system service (FreeBSD 5.x+):  
# /etc/rc.d/sbbs start  
(FreeBSD 4.x and lower):  
# /usr/local/etc/rc.d/sbbs.sh start

A note on SysOp paging:

For most systems, the BBS must either have write access to the sound card via /dev/dsp, or run as root to page the SysOp. FreeBSD supports an alternative method which requires one of the following:

On FreeBSD 4.x:

A custom kernel with the following option in the config  
pseudo-device speaker

On FreeBSD 5.x and higher:

One of:

- 1) A custom kernel with teh following option in the config  
device speaker
- 2) The speaker module loaded by either:
  - Running ``kldload speaker''
  - The line ``speaker\_load=YES''' in /boot/loader.conf

/dev/speaker should be read/writeable by the user the BBS runs as.

/\* End of sbbscon.txt \*/

Synchronet NT Services  
=====

\$Id: sbbsNTsvcs.txt,v 1.5 2005/09/11 00:07:40 rswindell Exp \$

If you're running Synchronet on a Microsoft Windows NT based operating system (e.g. Windows NT, Windows 2000, Windows XP, or Windows 2003), you can run Synchronet as a set of NT services using the executable `exec\sbbbsNTsvcs.exe`.

This single executable includes five (5) NT services that handle all the functionality of Synchronet v3.12-Win32:

| Name               | Description                                        |
|--------------------|----------------------------------------------------|
| -----              | -----                                              |
| SynchronetBBS      | Synchronet Telnet/RLogin Server                    |
| SynchronetFTP      | Synchronet FTP Server                              |
| SynchronetWeb      | Synchronet Web Server                              |
| SynchronetMail     | Synchronet SMTP/POP3 Mail Server                   |
| SynchronetServices | Synchronet Services (e.g. NNTP, IRC, Finger, etc.) |

You may install, disable, or remove each NT service independantly, or all at once. To install all the Synchronet NT services at once, run:

```
sbbsNTsvcs install
```

The configuration of the Synchronet servers and services is handled in your Synchronet initialization file (e.g. `ctrl/sbbs.ini`). The Synchronet NT services know the location of your Synchronet CTRL directory from the `SBBSCTRL` environment variable. See `sbbscon.txt` for more details about the `SBBSCTRL` environment variable and the initialization file (`sbbs.ini`).

Synchronet v3.12 and later automatically stores the relevant settings in your Synchronet initialization file (no exporting is necessary).

Once installed, the Synchronet NT services may be started from the NT Services dialog (under the Windows Control Panel, Administrative Tools, Services), by running the Synchronet Control Panel, or if the system is rebooted, the services will start automatically if enabled and configured for AutoStart.

To remove/un-install the Synchronet NT services, run:

```
sbbsNTsvcs remove
```

You may also enable and disable the Synchronet NT services with the `sbbsNTsvcs` command-line options. Run "`sbbsNTsvcs -?`" from a command prompt to see the available command-line options. Example:

Available Commands:

|         |                                |
|---------|--------------------------------|
| list    | to list services               |
| install | to install services            |
| remove  | to remove serivces             |
| disable | to disable services            |
| enable  | to re-enable disabled services |
| start   | to start services              |
| stop    | to stop services               |
| recycle | to recycle services            |
| mute    | to mute (sounds of) services   |
| unmute  | to unmute (sounds of) services |

To recycle the Synchronet Web Server service, run:

```
sbbsNTsvcs recycle web
```

Synchronet Control Panel  
-----

You should shutdown the Synchronet Control Panel (GUI) before installing the Synchronet NT services. The Synchronet Control Panel can be used to start, stop, configure, recycle, and monitor the NT services once installed, but it is not required for operation. If the Synchronet Control Panel is not running, the BBS servers will still function, even if no user is logged into Windows.

You cannot "spy" on a BBS node running as an NT service and the "Clients" window and the optional "Tray icon" in the Synchronet Control Panel will not reflect active clients connected to Synchronet NT services. There are plans to integrate these features with the Synchronet NT services in the future.

NT Event Log  
-----

Log messages (determined by the `LogMask` in your `sbbs.ini` file) will be logged as NT application event log messages (viewable in your NT Event Viewer).

You may wish to configure your NT event log to overwrite old messages

as needed:

1. Open the Event Viewer (e.g. from Control Panel->Administrative Tools)
2. Right click Application Log, Select Properties
3. Set the maximum log size to something like 8192 (8MB) or bigger
4. Select "Overwrite events as needed" or "older x days"

#### Debug Output

-----

If you wish to use a Windows debug output viewer (such as DebugView from <http://www.sysinternals.com/ntw2k/freeware/debugview.shtml>) to view the log output of one or more of the Synchronet NT Services, set the "Start parameters" of one or more of the services (under the Windows Control Panel) to "-debug".

/\* End of file \*/

\$Id: webservr.txt,v 1.17 2007/11/30 09:05:08 deuce Exp \$

## 1. Introduction

-----

The Synchronet Web Server is a mostly HTTP 1.1 compliant web server capable of handling basic web servicing tasks. It has most of the basic features of a general-purpose web server one would come to expect.

It also, through Server-Side JavaScript (SSJS) allows dynamic pages to be created which can access BBS data directly.

## 2. Web Server Configuration

-----

Most of the web server configuration is in your startup INI file (usually ctrl/sbbs.ini). The [Web] section contains the following unique settings:

### 2.1. Initialization file [Web] section keys

-----

RootDirectory=../web/root

This is the root directory of your web server... a request to  
http://yourbbs.synchro.net/index.html will be served out of this dir.

ErrorDirectory=error

The directory relative to RootDirectory where the various error  
message files are located. The error message files are named by the  
numeric HTTP error code they will represent and may be either .html  
or .ssjs files (.ssjs files take precedence over .html files for the  
same error).

IndexFileNames=index.html,index.ssjs

A comma-separated list of filenames in order of preference to serve as  
the default document in a directory. Many Sysops change this to:

IndexFileNames=index.html,index.htm,index.ssjs

Do not remove the index.ssjs unless you are not using the stock  
web pages at all.

Authorization=Basic,Default

A comma-separated list of authentication mechanisms in order of preference.  
The standards say that Basic must come first, but no browser currently  
appears to use Digest if Basic is listed first. Supported values are Basic  
and Digest. Digest is more secure as the users password is not sent in the  
clear over the wire.

CGIDirectory=cgi-bin

A directory relative to RootDirectory where any files found will be  
considered CGI-executable. Be careful what files you put in this  
directory.

CGIExtensions=.cgi

A comma-separated list of file extensions/suffixes. Files with these  
extensions will be considered CGI-executable and the web server will  
attempt to execute them as such.

DefaultCGIContent=text/plain

If the CGI program does not generate a content-type header, this value  
will be used for the MIME content-type specified in the HTTP response.

JavaScriptExtension=.ssjs

Files with this extension will be considered SSJS files. On all  
systems, this will be attempted to run with the JavaScript interpreter.

MaxInactivity=120

If a client holds a connection open for this many seconds without a  
request, the web server will shut down the connection.

MaxCgiInactivity=120

If a CGI script runs for more than this many seconds without any  
output, it will be terminated and the connection will be shut down.

HttpLogFile=../data/logs/http-

The prefix of log files if HTTP\_LOGGING is enabled (See next item) to  
store Common Logfile Format logs in. The current virtual host  
(if enabled, see next item), date, and .log are appended to this. ie:  
http-2005-03-12.log

Options=NO\_HOST\_LOOKUP

The | separated list of options to enable. In addition to the standard  
options, the web server also supports the following:

DEBUG\_RX

Log all received data to the console log, as well as various  
extra bits related to receiving data.

DEBUG\_TX

Log all transmitted data except the reply body itself, as well  
as various extra bits of information related to transmitted  
data.

VIRTUAL\_HOSTS

Supports name-based virtual hosts. If your system has multiple  
host names, you can have each host name return unique content  
depending on which hostname is used. ie: if  
freebsd.synchro.net and nix.synchro.net both resolved to your

system, you could have FreeBSD-specific pages on one, and general \*nix stuff on the other. A virtual host is added by simply putting the desired content into a sub-directory of RootDirectory with the desired hostname ie: web/root/freebsd.synchro.net/ if the browser doesn't send the request host name (very old browsers, or some automated tools) they will be served out of document root. It is therefore a good idea to put links to your various virtual hosts in an index.html page in RootDirectory something like this:

```
<html>
  <head>
    <title>Old Browser</title>
  </head>
  <body>
    Your browser is either too old to support
    name-based virtual hosts, or you have visited a
    virtual hosts that is not yet configured. The
    following are hosted here:<br>
    <a href="freebsd.synchro.net">freebsd.synchro.net
    </a><br>
    <a href="nix.synchro.net">nix.synchro.net</a><br>
  </body>
</html>
```

NO\_CGI

Disable CGI script execution.

HTTP\_LOGGING

Enable logging to a Common Logfile Format log as described in the HttpLogFile section. Usefull for running log analysis programs (like Webalizer: <http://www.mrunix.net/webalizer/>)

NO\_JAVASCRIPT

Disable SSJS execution.

## 2.2. Other web-related configuration files

In addition to the [Web] keys in the initialization file, the web server also uses some other configuration files:

ctrl/mime\_types.ini

Contains the file extension to mime-type mapping. Each line is in the format "extension = mime-type" ie: "html = text/html" The extensions are case-insensitive and do not include the '.'.

ctrl/webicons.ini

Contains the URLs to the icons used by the default 404.ssjs script for each file type/extension. Format is "extension = URL". Example: "html=/icons/layout.gif". Two "magical" extensions exist: DIRECTORY which is used for directories and DefaultIcon which is used for extensions which don't exist in the list.

ctrl/web\_handler.ini

Contains 2 sections, [CGI] and [JavaScript], where a list of file extensions and their associated content-creation handlers are specified. The [CGI] section is for natively-executed CGI handlers (e.g. "pl = perl" indicates "perl" will be used to handle ".pl" files). The [JavaScript] section is used for JS-executed content handlers (e.g. "xjs = xjs\_handler.js").

ctrl/cgi\_env.ini

Contains a list of system environment variables to pass to CGI processes. Each variable can have an optionally specified default value, over-ridden value, and prepended or appended text.

## 2.3. webctrl.ini per-directory configuration file

Each directory may have a webctrl.ini file which overrides certain settings for the directory it's in and all child directorys. Configuration keys may be set either globally, or in a per-filename group. Using the \* and ? wildcards as the group name such as [\*.html]. The following keys may be used in these files:

AccessRequirements:

Specifies an ARS string which all users must match to be able to access files in this directory. Will force an HTTP login.

Realm:

Sets the realm that is displayed to the user for the HTTP login. Default is the BBS name.

DigestRealm:

Sets the realm that is displayed to the user for the HTTP login when Digest authentication is being used. Default is the Realm value.

Authorization:

A comma-separated list of authentication mechanisms in order of preference. The standards say that Basic must come first, but no browser currently appears to use Digest if Basic is listed first. Supported values are Basic and Digest. Digest is more secure as the users password is never sent over the wire.

ErrorDirectory:

Specifies a different directory to check for error pages. If the error page is not found, will still check the global error directory.

CGIDirectory:

Specify an alternate CGI directory to check for CGI files.

PathInfoIndex:

Specifies that the index files can be ran for unlocated pages in the current directory. This effectively works like a custom 404 page.

For example, to require a login, but allow \*any\* user to access files in a directory, but only a sysop to access \*.log files, the following could be used:

AccessRequirements=level 0

[\*.log]

AccessRequirements=level90

### 3.0 JavaScript Web Server Objects

In addition to the standard JavaScript objects, the web server provides the following:

#### 3.1 http\_request object

The http\_request object contains information from the client that was included during this request. This objects properties are as follows:

|                |                                                                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| path_info      | - Contains extra path information that was included with the request AFTER the URI which identified this script. For example, if the request was for <code>http://www.synchro.net/script.ssjs/test/this</code> then path_info would contain the string <code>"/test/this"</code>          |
| method         | - Contains the HTTP method used to run the script. As of this writing, the available methods are "HEAD", "GET", "POST", and "OPTIONS".                                                                                                                                                    |
| virtual_path   | - The virtual path that this URI was reached by. This is the portion of the URI from the end of the host to the end of the filename.                                                                                                                                                      |
| query          | - This object contains the values of any form data which was submitted with the request. This is an associative "array" of name/value pairs. THE VALUES ARE ARRAYS OF STRINGS. The reason for this is that it is legal and often usefull to have multiple form fields with the same name. |
| query_string   | - If a query string was included, this is the raw, unparsed query string.                                                                                                                                                                                                                 |
| post_data      | - As with query_string but for data which was POSTed.                                                                                                                                                                                                                                     |
| header         | - An associative array of header name/value pairs.                                                                                                                                                                                                                                        |
| cookie         | - Much like the query object, this object contains key/value pairs of set cookies. Once again, this is an array of strings since multiple values for the same key can be set for cookies.                                                                                                 |
| real_path      | - The real OSs complete path to this script.                                                                                                                                                                                                                                              |
| ars            | - The ARS string which applies to this request.                                                                                                                                                                                                                                           |
| request_string | - The raw request string sent by the client.                                                                                                                                                                                                                                              |
| host           | - The value of the host header for this request.                                                                                                                                                                                                                                          |
| vhost          | - The virtual host serving this request.                                                                                                                                                                                                                                                  |
| http_ver       | - The HTTP version used for this request as a string.                                                                                                                                                                                                                                     |
| remote_ip      | - The IP address of the client.                                                                                                                                                                                                                                                           |
| remote_host    | - If the web server does host lookups (disabled by default), this is the remote hostname.                                                                                                                                                                                                 |

#### 3.2 http\_reply object

The http\_reply object is used to pass information about the reply back to the SynchroNet web server. The properties are as follows:

|        |                                                                                                                                                                                                                                                                                   |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| status | - HTTP status string. The default is generally "200 Ok"                                                                                                                                                                                                                           |
| header | - An associative array of headers to include with the reply. The only pre-defined one is "Content-Type" which defaults to "text/html".                                                                                                                                            |
| fast   | - This optional property can be set to "true" to make write()s go directly to the client for HTTP/1.0 connections. This prevents keep-alives from working but generally appears faster to the client. Since HTTP/1.1 requests use chunked mode, this isn't required for HTTP/1.1. |

#### 3.3 Extra global methods

The web server also adds new global methods. These are:

```
set_cookie(string key,  
           string value  
           [, time_t expires  
           [, string domain  
           [, string path
```



```
[, bool secure ]]]))
    Requests that the specified cookie be set.
```

#### 4.0 The SSJS Template System

-----

The default web pages use a SSJS Template engine which also allows for Theme support.

[Note: With this latest implementation of SSJS, @@ codes no longer can be nested.]

#### 4.1 The SSJS Template Scheme

-----

Each page consists of four parts:

The Header (../web/templates/default/header.inc)

This file contains the basic requirements for the HTML page. The opening HTML, doctype, title, CSS file link, etc. The header file includes the open body, System Name, and User greeting plus the initial page layout table start. The rest of the layout is continued in the next files.

Top Navigation (../web/templates/default/topnav.inc & ../web/lib/topnav\_html.ssjs)

The topnav.inc file contains the basic design of the "breadcrumbs" The links are dynamically generated by topnav\_html.ssjs so both files need to be addressed when modifying or creating themes. In the case of the default layout, topnav.inc has a left and right graphic and a middle section that the dynamic content goes. The background image is handled by CSS. You can change this to anything you like.

The topnav\_html.ssjs file may seem daunting at first, but it is pretty straight forward. It has a series of if statements that check the current page location and sets up the breadcrumbs based on what you want it to say.

For example:

You want to add a Links page called links.html in the main directory. You would add a check for the path to links.html as:

```
if(http_request.virtual_path==" /links.html")
    template.topnav.push({html: '<span class="tlink">
        Some Links</span>'});
```

Left Side Navigation (../web/templates/default/leftnav.inc & ../web/lib/leftnav\_nodelist.ssjs & ../web/lib/leftnav\_html.ssjs)

This starts the main table layout in the default layout and also provides two other things -- the main navigation links and a brief nodelisting that displays when users are online via telnet. The links are dynamically created as in the Top Navigation example above with the exception of the two static links.

Main Content (various files)

This is where the layout of the main content is created. It is best to look at the various files in ../web/templates/default & ../web/templates/default/msgs to see how the code is displayed for the various functions of the Web side of Synchronet. Some details on what each of the special codes contained in those files do will follow.

Footer (../web/templates/default/footer.inc)

This file contains the closing HTML and whatever bottom information you would like. In the case of the default layout, the Web Server/Synchronet versions and the XHTML 1.0 logo. Links to privacy statements or anything else can be placed here and they will be displayed at the bottom of each page.

#### 4.2 SSJS Theme Support

-----

Theme Layouts can be added to Synchronet by creating them and placing the \*.inc files in their own directory under ../web/templates/

Themes are activated by editing the ../web/templates/html\_themes.ssjs file. This file contains:

```
/* Set default theme name */
var DefaultTheme="Default";

/* Edit this bit to add/remove/modify theme descriptions and dirs */
Themes["Default"]=new Object;
Themes["Default"].desc="Default Synchronet Theme";
Themes["Default"].dir="default";
Themes["Default"].css="/synchronet.css";
```

Themes are added by editing below the Default Theme such as:

```
Themes["CoolTheme"]=new Object;
Themes["CoolTheme"].desc="My Cool Theme";
Themes["CoolTheme"].dir="cooltheme";
Themes["CoolTheme"].css="/cooltheme.css";
```

To change the Default Theme, change:

```
var DefaultTheme="Default";
```

to:

```
var DefaultTheme="CoolTheme";
```

#### 4.3 Special Codes Used in the SSJS Template System

-----

By looking at the message related files located in templates/default/msgs, it can be seen that some special codes are used to display dynamically created content. It is very important to maintain the information EXACTLY as seen in each file or else the messaging system will fail. While how it is displayed can be changed, the correct information will only be displayed by following the format in the \*.inc files.

For example the groups.inc:

```
<!-- Main Content -->

    <td class="main" valign="top"><br />

<table class="grouplist" border="0" cellpadding="2" cellspacing="2">
<tr>
<th class="grouplist">Message Group</th><th class="grouplist">Subs</th>
</tr>
<<REPEAT groups>>
<tr>
    <td class="grouplist">
        <a class="grouplist" href="subs.ssjs?msg_grp=^^groups:name^^">
            %%groups:description%%</a></td>
    <td class="grouplist" align="right">
        @@JS:msg_area.grp_list[RepeatObj.index].sub_list.length@@</td>
</tr>
<<END REPEAT groups>>
</table>
<br />

<!-- end Main Content -->
```

While the table layout can be changed or even eliminated, the information within the <<REPEAT groups>> and <<END REPEAT groups>> must remain intact. To remove the table yet keep the correct information, the resulting groups.inc would be changed to (while maintaining the main table layout in this case) to:

```
<!-- Main Content -->

    <td class="main" valign="top"><br />

<<REPEAT groups>>
    <a class="grouplist" href="subs.ssjs?msg_grp=^^groups:name^^">
        %%groups:description%%</a>
    @@JS:msg_area.grp_list[RepeatObj.index].sub_list.length@@<br />
<<END REPEAT groups>>

<br />

<!-- end Main Content -->
```

This principle applies to all the .inc files in msgs respectively.

#### 4.4 The SSJS Template Library

-----

%%name%% is replaced with the HTML encoded value of template.name

i.e.; Spaces are replaced with: this&nbsp;is&nbsp;html

^^name^^ is replaced with the URI encoded value of template.name

i.e.; Spaces are replaced with: this%20is%20URI

@@name@@ is replaced with the value if template.name

No changes or encoding is performed.

@@name:sname@@ is replaced with the value of template.name.sname  
(^^ and %% are also supported)

@@JS:js\_expression@@ is replaced with the return value of js\_expression  
(^^ and %% are also supported)

```
<<REPEAT name>>
    @@name:sname@@
<<END REPEAT name>>
```

Iterates over the array/object template.name and replaces name:sname with  
the value of template.name.sname.  
(^^ and %% are also supported)

#### 4.5 SSJS Message Configuration

Configuration settings for the SSJS Messaging system is located in the  
../web/lib/msgconfig.ssjs file:

```
max_messages=20;
max_pages=30;
next_msg_html="Next Message";
prev_msg_html="Previous Message";
next_page_html="NEXT";
prev_page_html="PREV";
showall_subs_enable_html="Show all subs";
showall_subs_disable_html="Show subs in new scan only";
show_messages_all_html="Show all messages";
show_messages_yours_html="Show messages to you only";
show_messages_your_unread_html="Show unread messages to you only";
show_messages_spacer_html="&nbsp;<b>|</b>&nbsp;";
anon_only_message="Message will be posted anonymously";
anon_allowed_message='<input type="checkbox" name="anonymous" value="Yes" /> \
Post message anonymously';
anon_reply_message='<input type="checkbox" name="anonymous" value="Yes" checked /> \
Post message anonymously';
private_only_message="Message will be marked private";
private_allowed_message='<input type="checkbox" name="private" value="Yes" /> \
Mark message as private';
private_reply_message='<input type="checkbox" name="private" value="Yes" checked /> \
Mark message as private';
```

Each of these are configurable. NOTE: Lines ending in "\" indicate the line  
below is part of the line above. The "\" is not part of the configuration as  
it too be removed.

See the actual file for the defaults currently in use.

#### 4.6 Embedded Javascript

The \*.inc files can (and do in the default layout) have embedded JavaScript  
which is parsed by the JavaScript engine. Care should be taken as a large  
number of embedded JavaScript in the \*.inc files slow down overall processing  
of pages.

Anything contained within @@JS: @@ is processed by the Server-side JavaScript  
engine.

For example, it can check if the user is Guest or an actual user with this line:

```
@@JS:if(user.number==0 || user.security.restrictions&UFLAG_G) \
'<html code for Guest>'; else '<html code for registered user>';@@
```

What this does is display links specific for Registered Users only to them  
and not Guest. There are many things that can be done with @@JS: @@ code.  
Note, it also can be used to display HTML based on location as in the  
node listing stuff. In this case, it checks for whether or not a user is  
online, or if the user is anywhere but the Who's Online page before displaying  
the Left side node listing.

IMPORTANT! Anything contained within @@JS: @@ MUST be on one line or there  
will be errors in parsing.

#### 4.7 global\_defs.ssjs

This version of the Web Layout now includes a new file called  
global\_defs.ssjs. It is located in the ../web/lib directory. This file can  
be used for creating global definitions that span all pages of a site.  
For example:

```
template.user_alias=user.alias;
```

Now @@user\_alias@@ can be in any \*.inc template files and it will display the user's alias.

Care should be excersied when using this file as loading it up with hundreds of predefined definitions may slow down overall page rendering as the file is loaded on every page. It would be better to just put a few popular definitions that are truly global rather many definitions. It would be ineffcient to have thirty of forty message definitions being loaded when a user is looking at the statistics page.

## 5.0 XJS files

-----  
XJS files, handled by exec/xjs\_handler.js are what many people consider to be an easier method of generating SSJS files. XJS files are HTML files with JS commands embedded in them using special tags much like PHP. XJS files are translated on-the-fly to .ssjs files using the same name with .ssjs appended. For example, a file named test.xjs will, when ran, generate a test.xjs.ssjs file.

## 5.1 XJS syntax

-----  
In an XJS file, everything not within a special xjs tag is send to the remote host unmodified, and everything inside the xjs tag is interpreted as JS statements to be executed at that point in the file. The xjs tag begins with either "<?xjs" or "<?" and ends with ">". A simple example would be:

```
<html><head><title><?xjs write(system.name) ?></title></head>
<body>
Your SysOp "<?xjs write(system.operator) ?>" welcomes you to
<?xjs write(system.name) ?>
</body>
</html>
```

This would send the following web page to the remote system:

```
<html><head><title>My Brand New BBS</title></head>
<body>
Your SysOp "Sysop" welcomes you to
My Brand New BBS
</body>
</html>
```

Looping constructs are permitted, however, not using brackets can result in unexpected effects. Because of this, it is reccomended to \*always\* use brackets with looping and flow control items.

The following example displays the numbers from one to 10.

```
<html><head><title>Counter</title></head>
<body>
<?xjs
var i;
for(i=1; i<=10; i++) {
?>
<?xjs write(i) ?><br>
<?xjs } ?>
</body>
</html>
```

## 5.2 XJS-specific global methods and properties

-----  
The following JS commands are available to XJS files only.

xjs\_load(filename)

Runs the specified xjs file at the current position. Local variables are NOT visible to xjs\_load()ed pages. The filename is assumed to be relative to the including file (or absolute.)

cwd

Contains the path that the current xjs script was loaded from and which parameters to xjs\_load() are assumed to be relative to. If you change the value of cwd, it will change the location where xjs\_load() will check for files.

```
/* End of file */
```

## Synchronet NewsLink Installation/Configuration

=====

\$Id: newslink.txt,v 1.6 2012/07/18 14:52:35 deuce Exp \$

### Background

-----

NewsLink is an external module for Synchronet BBS Software v3.1+. It is used to link one or more Synchronet sub-boards (aka message areas) with one or more USENET (or other NNTP network) newsgroups on one or more News Servers.

News Servers use the Network News Transfer Protocol (NNTP) to transfer messages to and from a News Client (aka newsreader). Popular News Clients include Microsoft Outlook Express, Netscape Communicator, Mozilla, Pine, Xnews, and many others. NewsLink emulates a News Client based on the NNTP standard. News Servers normally listen for incoming connections on TCP port 119 and usually require an authenticated user login.

NewsLink is written in 100% JavaScript and the source code (if you're interested) is contained in the files newslink.js and newsutil.js in the Synchronet EXEC directory.

### Step-by-Step Installation/Configuration

-----

If you have installed Synchronet v3.10 or v3.11 from scratch (not an upgrade from an earlier version), then some of the following steps may have already been completed for you.

1. Go into SCFG->External Programs->Timed Events and created a timed event for NewsLink (if one doesn't already exist). Example:

```
[NEWSLINK Timed Event]
1: Internal Code           NEWSLINK
2: Start-up Directory
3: Command Line           ?newslink
4: Enabled                Yes
5: Execution Node          1
6: Execution Days of Month Any
7: Execution Days of Week All
8: Execution Frequency     48 times a day
9: Requires Exclusive Execution No
10: Force Users Off-line For Event No
11: Native (32-bit) Executable <doesn't matter>
12: Use Shell to Execute   <doesn't matter>
13: Background Execution   <doesn't matter>
14: Always Run After Init/Re-init No
```

You may change the Execution Frequency to suit your needs.

2. Go into SCFG->Message Areas and create a new Message Group for your USENET message areas (assuming you're linking to a USENET News Server). Example:

```
[USENET Group]
1: Long Name              USENET Newsgroups
2: Short Name             USENET
3: Internal Code Prefix   USENET_
4: Access Requirements
```

Note: At this point you may want to login to your New Server with a regular News Client (aka newsreader), retrieve a list of available newsgroups and make a list of the newsgroups you would like linked into your BBS's message areas.

3. Now go into SCFG->Message Areas->USENET->Sub-boards and create some sub-boards to link to newsgroups on the News Server. Example:

```
[alt.bbs.allsysop Sub-board]
1: Long Name              alt.bbs.allsysop
2: Short Name             alt.bbs.allsysop
3: QWK Name              allsysop
4: Internal Code          USENET_ALLSYSOP
5: Newsgroup Name         alt.bbs.allsysop
6: Access Requirements
7: Reading Requirements
8: Posting Requirements   REST NOT G
9: Operator Requirements
10: Moderated Posting User
11: Maximum Messages      5000
12: Purge by Age          Disabled
13: Duplicate Checking    Enabled (10000 message CRCs)
```

4. For each NewsLinked sub-board, set Network Options->Internet (UUCP/NNTP) to "Yes".
5. Edit your ctrl/newslink.cfg file:
  - A. Change the "SERVER" line to specify the host name or IP address of the News Server you are linking with. If the News Server is listening on a non-standard TCP port (e.g. other than 119), you will need to add a "PORT" line specifying the TCP port number of the server.
  - B. If the News Server requires an authenticated user login, change the "USER" and "PASS" lines to specify the username and password you use to login to the above News Server. If the "USER" line is missing or commented-out (with a semicolon), then no user authentication will be attempted (sometimes called "anonymous login").
  - C. For each sub-board you configured in step 3, add an "AREA" line, specifying each sub-board's "Internal Code" (from SCFG->Message Areas). Example: AREA USENET\_ALLSYSOP
  - D. By default, NewsLink will only import articles (aka messages) posted *after* the first scan of the newsgroups (i.e. it will not import *any* articles during the first run). If you want NewsLink to import *all* existing articles for *all* linked newsgroups (potentially many *thousands* of messages, depending on the News Server), add the line "FLAGS I" (case insensitive).  
WARNING: Using this feature may cause the initial NewsLink execution to take a long time to download all the existing articles (potentially many thousands) that may be stored on the News Server.
  - E. Remove or comment-out (with a semicolon) the "DISABLED" line.
  - F. If you wish to use TLS/SSL, uncomment (remove the semicolon) the "TLS" line.
6. Now you may force the NEWSLINK timed event to run. In SBBSCTRL (Win32 GUI), you can do this by clicking BBS->Force Timed Event->NEWSLINK, or for the console or NT services versions of SynchroNet, you can touch or create the file data/newslink.now to force the event to run.
7. Watch your events log for the NewsLink execution progress and details. Remember, if the 'I' flag was not specified (globally or per-newsgroup) in your ctrl/newslink.cfg, then no articles (aka messages) will be imported during the first run and only newly-posted articles will be imported on subsequent runs.

#### Area Flags

Each "AREA" line in the NewsLink configuration file can specify one or more "flag characters" to enable special behaviors. These same flag characters may be specified globally (for all linked areas) using the "FLAGS" line.

The supported flags characters are:

```
x  do not add tearlines & taglines to exported messages
n  do not add "From Newsgroup" text to imported messages
t  do not add tearline to imported messages
a  convert extended-ASCII chars to ASCII on imported messages
r  remove "Newsgroups:" header field from imported messages
b  decode single-part (uuencoded or yenc-encoded) binary attachments
i  import all articles on first run (not just new articles)
s  no subject filtering (do not check against text/subject.can)
m  moderate imported messages (a subop will need to validate every message)
```

To specify one or more flag characters for a specific area, you must use the following "AREA" line syntax:

```
AREA <sub-code> <newsgroup> <flags> [attachment_dir]
```

where:

<sub-code> is the sub-board's complete internal code (including the group's code prefix, if there is one).

<newsgroup> is the name of the newsgroup as it is represented on the News Server. If this field is missing or is less than 2 characters in length, the newsgroup name from SCFG is used.

<flags> is the optional list of flag characters.

[attachment\_dir] is the optional destination directory for decoded binary file attachments (only used with the 'b' flag)

Examples:



```
AREA USENET_ALLSYSOP alt.bbs.allsysop nib /sysop_attachments
AREA USENET_ALLSYSOP - asr
```

## Multiple Servers/Config Files

ctrl/newslink.cfg is just the default NewsLink configuration file name. If you specify the path to a different file on the "?newslink" command-line, NewsLink will use that configuration file instead. This allows you to have multiple NewsLink timed events configured with different configurations for different News Servers.

## Command-line Options

Other supported NewsLink command-line options:

```
-d      enable debug output
-ri     reset import pointers (import all)
-ui     update import pointers (import none)
-re     reset export pointers (export all)
-ue     update export pointers (export none)
-ne     no e-mail addresses
-nm     no mangling of e-mail addresses (disable anti-SPAM measures)
-um     un-mangle e-mail addresses when importing
-ix [n] import a fixed number (n) of articles/messages (default is 500)
```

## Controlling SPAM

Most SPAM messages on public NNTP networks (e.g. USENET) are directed at multiple newsgroups. For this reason, the newslink.cfg file supports a "max\_newsgroups\_per\_article" value which defaults to 5.

If you find NewsLink is filtering valid messages with an error in the log like:  
!Too many newsgroups (7)  
try increasing this value.

If you find NewsLink is allowing SPAM through that is addressed to multiple newsgroups, try reducing this value. Set max\_newsgroups\_per\_article to 0 to disable this feature.

NewsLink also uses the text/subject.can filter file to filter (ignore) articles with specific text in the subject and the ctrl/twitlist.cfg file to ignore messages from specific users or domains.

## Yielding

NewsLink will yield time-slices based on the "lines\_per\_yield" (default: 5) and "yield\_length" (default: 1) values. If you find NewsLink is consuming a large portion of your CPU cycles, try decreasing the "lines\_per\_yield" value or increasing the "yield\_length" value.

## Stopping NewsLink

NewsLink can be stopped by "touching" (creating or updating) the file:  
data/newslink.stop

## JSexec

With Synchronet v3.11, it is also possible to run NewsLink externally from Synchronet (or as a background timed event) by using JSexec.

/\* End of file \*/

## Synchronet ListServer Installation/Configuration

=====

\$Id: listserver.txt,v 1.1 2004/10/23 03:47:15 rswindell Exp \$

### Overview

-----

The Synchronet ListServer (exec/listserver.js) allows Internet e-mail clients to subscribe and (optionally) contribute messages to your local message bases using nothing but a standard Internet mail client (e.g. Outlook/OE, Eudora, Pine, etc.).

### Requirements

-----

The Synchronet ListServer requires Synchronet v3.12 or later.

### How it works

-----

The Synchronet ListServer works much like traditional Internet mailing list servers, such as "Majordomo" and "Listserv".

The user of an Internet e-mail client "subscribes" to one of your mailing lists by sending an e-mail message to the address of your ListServer (typically "listserver@yourbbs.com"). In the body of the message (or optionally, the subject), the line "subscribe listname" must exist, where "listname" is the name of one of your mailing lists (as configured in your ctrl/listserver.ini file, described later).

Once subscribed, the user will automatically receive, in their mailbox, any messages posted in the message area (sub-board) corresponding to the mailing list. A user may unsubscribe from the mailing list by the same means but using the command "unsubscribe listname".

The ListServer will respond to other "control messages" like these. A control message of "help" will return a list of valid commands to the requesting client's mailbox.

### Submissions

-----

Writable mailing lists allow the user to submit messages to the list by sending e-mail messages addressed to "listname@yourbbs.com". Submissions are automatically posted to the message area (sub-board) associated with the mailing list and forwarded to all mailing list subscribers.

### Configuration

-----

1. If you haven't already, create the message areas (sub-boards) in SCFG that you want to make available as mailing lists. Be sure to make a note of the configured "internal code" associated with each sub-board you are going to associate with a list.
2. Edit your ctrl/listserver.ini file, creating a separate section for each mailing list and specifying the "internal code" of the associated sub-board using the "sub" key. See the stock ctrl/listserver.ini for an example.

If you wish to allow control commands in the subject of messages addressed to the ListServer, set "SubjectCommand = true" in the "Global Settings" portion of the file.

Set "disabled = false" in the "Global Settings" portion of the file to enable the ListServer.

3. Edit your ctrl/mailproc.ini file, adding (if you don't have it already) the following section:

```
[listserver.js]
to = listserver, listserv
passthru = false
```

Include in the "to" value the names of any writable (not readonly) lists from your listserver.ini file. Be sure to separate the names with commas.

4. Add the command-line of "?listserver" as a timed-event in SCFG->External Programs->Timed Events. This is necessary for the distribution of messages posted to sub-boards associated with mailing lists.
5. Recycle your mail server, if it didn't automatically recycle after saving your changes in SCFG in step 4. This is required to re-load your modified ctrl/mailproc.ini file into the mail server.

## Maintenance

-----

The list of subscribers is kept in the file data/subs/code.list.sub, where "code" is the sub-board's internal code. There is currently no mechanism for removing subscribers with bad e-mail addresses, but there will be in the future.

/\* End of file \*/

## Synchronet ListGate Installation/Configuration

=====

\$Id: listgate.txt,v 1.1 2005/01/03 05:23:33 rswindell Exp \$

### Overview

-----

The Synchronet ListGate (exec/listgate.js) allows a sysop to gate one or more sub-boards (message areas) with existing, remotely hosted, Internet mailing lists.

### Requirements

-----

The Synchronet ListGate requires Synchronet v3.12 or later.

### How it works

-----

The Synchronet ListGate module (listgate.js) actually only handles the "exporting" of new messages posted to the local sub-board to the mailing list (sent as SMTP e-mail messages). The "importing" of messages from the mailing list is handled by a special feature of the Synchronet SMTP server which allows messages received for specific addresses to be posted to local sub-boards.

### Configuration

-----

1. If you haven't already, create the message area (sub-board) in SCFG that you want to gate with an existing mailing list. Be sure to make a note of the configured "internal code".

If you set the access or posting requirements, you'll need to append "OR USER=0" to the requirements string(s) to allow posts from unauthenticated SMTP clients.

2. Edit your ctrl/listgate.ini file, creating a separate section for each sub-board that will be gated. Example (if "mysub" is the internal code for the gated sub-board and "mybbs.com" is the BBS's public host or domain name for receiving e-mail):

```
[mysub]
to = listname@listserver.com
from = listname@mybbs.com
```

If the line "disabled=true" is included in the file, remove it to enable the listgate.

3. Edit your ctrl/alias.cfg file, adding a line to accept submissions from the mailing list as posts to the locally gated sub-board:

```
listname sub:mysub
```

This will redirect any SMTP e-mail messages received for "listname@mybbs.com" as posts to the sub-board "mysub".

4. Configure your account on the list server (if you can) to not "echo back" any submissions received from you. This will prevent the dreaded "dupe loop" you would otherwise encounter.

The listgate module will automatically not export any messages that were posted via SMTP (preventing any "dupe loop" from the BBS to the list).

5. Setup a timed event (in SCFG->External Programs->Timed Events) to run the command-line "?listgate.js" periodically. This will export any new messages in the sub-board(s) configured in listgate.ini as e-mail submissions to the mailing list.

### Maintenance

-----

The export pointers are kept in the file data/subs/code.listgate.ptr, where "code" is the sub-board's internal code.

You can execute "?listgate.js -u" to update the export pointers \*without\* actually exporting any existing messages to the list, or you may execute "?listgate -r" to reset the export pointers, so that \*all\* the existing messages in the sub-board will be exported to the mailing list (even if they were previously exported).

/\* End of file \*/

**\*\* ircd.js : The Synchronet IRCd Service \*\***  
by: Randy Sommerfeld <cyan@rrx.ca>

|     |                                                              |
|-----|--------------------------------------------------------------|
| 1   | .... Introduction                                            |
| 2   | .... About this document                                     |
| 2.1 | .... What this document hopes to accomplish                  |
| 2.2 | .... What this document won't accomplish                     |
| 2.3 | .... Assumptions we make about you                           |
| 3   | .... Installation                                            |
| 3.1 | .... Getting the IRCd up and running                         |
| 3.2 | .... Linking to the Synchronet IRC Network (irc.synchro.net) |
| 3.3 | .... Using JExec to run the IRCd                             |
| 4   | .... About the Synchronet IRC Network (irc.synchro.net)      |
| 5   | .... Technical information                                   |
| 5.1 | .... Limits of the Synchronet IRCd                           |
| 5.2 | .... Compliance with RFC's, and established protocols        |
| 5.3 | .... Compatibility with other IRCd's                         |
| 6   | .... The Future                                              |
| 7   | .... Frequently Asked Questions                              |
| 7.1 | .... Installation Questions                                  |
| 7.2 | .... Technical Questions                                     |

#### ----- 1.0 -- Introduction -----

The Synchronet IRCd (IRC daemon) service is the newest Synchronet service written in 100% Javascript. It's currently the largest (and perhaps the most complex) service available for Synchronet. The IRCd service aims to eventually be a 'communications bridge' of sorts which will allow multiple BBS's to link their "multinode chat" areas together in a network so that users may talk to one another from the comfort of their home BBS. While accomplishing this goal, the Synchronet IRCd intends to also be a fully-functional IRC daemon that rivals the bigger, legacy UNIX IRC daemons in terms of features. This way, everyone who wishes to chat on a common network will be able to use the local BBS, or a standard IRC client (if they wish.)

#### ----- 2.0 -- About this document -----

##### [2.1] - What this document hopes to accomplish

First and foremost, this document is intended to be an absolute reference regarding the Synchronet IRCd. This document should be the first place to look if you're having a problem of some kind, regardless of what that problem may be. It intends to be an installation guide which will aide you through setting up your own IRCd. Or, if you're simply curious about the IRCd from a theoretical standpoint (it's not every day that you get to see an IRC daemon as functional as this implemented inside of a scripting language), this document will do its best to explain certain implementation decisions to you.

##### [2.2] - What this document won't accomplish

This document is not a replacement for your brain. If you're having a problem of some kind, and it's not covered by this document, PLEASE take a few hours to think about the problem and solve it. Use a systematic approach to the problem; "If I do X, does Y still happen?" Remember, installation and maintenance of the Synchronet IRCd isn't life or death, it's a hobby, and like all good hobbies, you're supposed to learn something from it. So, take a few days before you even consider asking for support -- take a deep breath, go jog around the block, turn off the computer -- and then come back later. On the other hand, if you've found a unique or interesting solution to your problem, and it hasn't been covered here, then by all means tell us about it!

We don't intend to teach you the basics of IRC here. While you can certainly still get the Synchronet IRCd up and running with a minimal amount of effort and IRC knowledge, your experience will be more enjoyable if you have at the very least a working knowledge of IRC. To gauge your knowledge of IRC, answer the following questions either with a 'true' or 'false':

- \* People chat in 'rooms' on IRC.
- \* 'IRCop' is short for 'IRC cop', the police officers of IRC.
- \* A 'ping' is something submarines use to gauge distance under water.
- \* CTCP and DCC stand for Client Tunneling Carrier Protocol and Direct Channel Cutting, respectively.

If you answered 'true' to any of the above questions, or if you don't understand some of the terminology used in the questions, then it's best that you go to <http://www.irchelp.org> and read through their very helpful documents about IRC. This way, you'll gain a better understanding about what IRC really is, what it's here for, the terminology, and a little history, too. At the very least, this is so that when someone asks you to edit your 'M:Line' to solve or diagnose a problem, you'll at least know what they're talking about. We don't cover any IRC basics in this document.

Don't be surprised if you're ignored or simply referred to a URL without further explanation if you ask for help and refer to an IRC \*channel\* as a room, call \*IRC operators\* 'cops', don't understand why a 10 second 'ping' time is bad, or not know how to DCC SEND a file (such as your ircd.conf) to be inspected. That includes not having your NAT or proxy set correctly to properly masquerade your IP address on a DCC CTCP.

## [2.3] - Assumptions we make about you

In short, this document assumes that you know:

- \* How to use your computer and operating system effectively.
- \* The basics of IRC and its terminology.
- \* How to use, configure, and make basic modifications to Synchronet.
- \* The basics of the Internet (or at least the ability to visualize a routed, distributed network)

## ===== 3.0 -- Installation =====

### [3.1] - Getting the IRCd up and running

(1) Make sure you are running a modern version of the Synchronet BBS software. The Synchronet IRCd is not intended to run on versions 3.10 or earlier. While care has been taken to ensure backwards compatibility with 3.10, it should only be run on that version for testing purposes. Upgrading may include retrieving the latest version of Synchronet from the Synchronet CVS (if you're familiar with building from CVS), from the Synchronet web site (<http://www.synchro.net>), or elsewhere. Synchronet 3.11 is the minimum required version to run version 1.0 (or higher) of the Synchronet IRCd.

Older versions of Synchronet also come packaged with older Javascript libraries. The Synchronet IRCd makes use of some of the newer Javascript library features (such as the ability to treat strings as an array of characters), therefore it's critical to have a modern JS library. If your Javascript library is from 2002 or earlier (you can check this in the 'system information' part of your BBS main menu), then it's too old. Time to upgrade.

(2) Check to make sure that all of the components of the Synchronet IRCd are resting in their correct directories. While you're at it, check the file sizes to make sure they're relatively normal (i.e. not 0 bytes long.) Make sure that all of the files are readable by the user you're intending to run the IRCd as. Here's a chart to help:

| File | Synchronet | Dir. | Description |
|------|------------|------|-------------|
|------|------------|------|-------------|

|                 |      |  |                                      |
|-----------------|------|--|--------------------------------------|
| irclib.js       | exec |  | Javascript IRC library functions     |
| ircd.js         | exec |  | IRCd core                            |
| ircd_channel.js | exec |  | Channel functions/objects            |
| ircd_server.js  | exec |  | Functions/objects for servers        |
| ircd_unreg.js   | exec |  | Handling of unregistered connections |
| ircd_user.js    | exec |  | Functions/objects for users          |
| ircd.conf       | ctrl |  | Primary configuration file           |
| ircmotd.txt     | text |  | Message users see upon connect       |

(3) Edit your relevant services configuration file (in the Synchronet 'ctrl' directory.) For Synchronet 3.11, this is most likely "services.ini", however, backwards compatibility has been retained with the old style "services.cfg" just in case you want to bring yours over from an older version of Synchronet. You may run the Synchronet IRCd on any port you wish, however, standard convention dictates that it be run on port 6667 (which is the standard port that most IRC clients will try to connect to, first.) It's important that the IRCd be defined as a STATIC service (i.e. one that runs in the background continuously and does not require a connection to start the service), and that it uses the LOOP option (so that if you wish to restart only the IRCd, or if it crashes, it comes back up immediately.)

The combined options for 'services.cfg' represented in hex is '806', so that a completed services.cfg line should look like this:

| ;protocol | port | max_clients | options(hex) | command-line |
|-----------|------|-------------|--------------|--------------|
| IRC       | 6667 | 0           | 806          | ircd.js      |

An example IRC section from services.ini looks like this, and should already be present in services.ini for Synchronet 3.11 and above. The IRCd is enabled by default in Synchronet 3.11, so, it's likely that it's already operating:

```
[IRC]
Port=6667
Options=STATIC | LOOP
Command=ircd.js
```

The 'maximum clients' value used inside of the services configuration is \*ignored\* by Synchronet Services since that value is managed by the IRCd itself. The maximum number of IRCclients can be changed on a Y:Line in



your ircd.conf, and is set to 100 by default (more on the ircd.conf later.)

(4) Take a look at your ircd.conf, and familiarize yourself with the options and terminology. Although the ircd.conf has been specifically made so that it works right out of the box, you may wish to tweak some options. The configuration file is self-documenting, so you should carefully read about each of the configuration options. At the very least, this will familiarize you with what a 'Z:Line' is, and where a 'K:Line' is, or how to enable passwords on sensitive IRC commands. If you intend to link to the Synchronet IRC Network, pay careful attention to where the C:Line and N:Line pair is configured. We'll be dealing with those later.

If you're already familiar with other IRC daemon configuration files, the Synchronet ircd.conf has been carefully engineered to be compatible with the Bahamut ircd.conf. Thus, if you've already got a working Bahamut configuration, putting that ircd.conf in place of the stock ircd.conf should give you an IRCd configured exactly the way it was on Bahamut. Configuration files from other IRC daemons have *not* been tested, so your mileage may vary with those (although the configuration should be very similar.) Certainly if you find an ircd.conf from another daemon that works, let us know about it!

(5) Restart your BBS, or recycle your services so that your new IRC service is brought online with your BBS. Watch your BBS console carefully, as you should see something similar to the following messages:

```
srcv 0007 IRC SynchronetIRCD-1.1b(1.102) started.  
srcv 0007 IRC Reading Config: /sbbs/ctrl/ircd.conf
```

You may have to scroll up to see the message. Any errors should be self-explanatory (and usually involve not being able to read the configuration file.) If you get an error about not being able to bind to a socket, or that a socket is already in use, then you already have something running on the port you defined in your services configuration. Could it be another IRC server running? Try disabling any other IRC servers or proxies and restart the BBS. If you recently restarted Synchronet with users connected to an already operating IRCd, then it's likely that some of your sockets are in a 'TIME\_WAIT' state. Wait a minute or two for the condition to clear up, then try again. Repeat this process of elimination until your IRCd starts successfully.

(6) Test your new IRCd by connecting to it with an IRC client. At the very least, using 'telnet' to connect to the IRCd port should give you something similar to the following line:

```
:rrx.synchro.net NOTICE * :*** SynchronetIRCD-1.1b(1.102) (RoadRunner X) Ready.
```

Which is the standard Synchronet IRCd banner, informing you that the IRCd is accepting new connections correctly.

At this point, you should have a working, fully-functional IRC server available for users to use. You may want to point Synchronet's internal IRC client to your local IRC server (just 'localhost' or '127.0.0.1'), so that your BBS users will be able to make use of your new IRCd. By default, an O:Line has been added so that anyone connecting either from the BBS itself, or your internal network will be able to make use of the /OPER command to become an IRC Operator. To become an IRC Operator, first /WHOIS yourself to make sure that your hostname is your local BBS's hostname (i.e. mybbs.synchro.net) If so, execute '/OPER Sysop <pass>', where <pass> is your local BBS system password. If your hostname isn't the local BBS's hostname, go ahead and edit ircd.conf (in the Synchronet 'ctrl' directory) and add an O:Line for your hostname as directed by the instructions inside of the configuration file.

### [3.2] - Linking to the Synchronet IRC Network (irc.synchro.net)

Connecting your IRC server to the Synchronet IRC Network has certain advantages. The best advantage is that you'll be allowing any users who use your IRC server (i.e. your BBS users) to also talk to other users on the network. Since the network is largely BBS-oriented, there are a wide variety of channels available for users to join and discuss various topics. Furthermore, you'll be offering your system to the pool of IRC servers available for the public to use. Just like how DOVE-Net is a network of Synchronet systems that pass message group messages to one another, the Synchronet IRC Network is a network of Synchronet systems for the purposes of chatting in realtime.

In later versions of the IRCd or the network, you'll be able to offer your BBS's files over IRC, integrate your BBS chat area with IRC, or even receive alerts about new messages or BBS email while on IRC.

Linking with the Synchronet IRC Network takes about as much time (if not less) as it does to establish a DOVE-Net node. Therefore, the procedures have been kept very similar. The Synchronet IRC Network is the only IRC network in the world that allows you to link without going through a tiresome application process, or other such bureaucracy. In fact, no network admin need be present at all for you to link to the Synchronet IRC network.

Just as with establishing a DOVE-Net node, the following steps must be taken to ensure a smooth link to the Synchronet IRC Network:

(1) Ensure that you have a DOVE-Net node established. Although you aren't required to be a member of DOVE-Net to be a member of the Synchronet IRC Network, you need to at least go through the same automatic registration process to obtain and configure your QWK-ID. Instructions about obtaining your QWK-ID can be found here: <http://www.synchro.net/docs/dove-net.txt> Only steps 1 and 2 need be followed, but heed the document's warning: "Remember the password you used to create this account, you'll need it later." You do not need to create a new QWK-ID for the IRCd if you already have an existing one for DOVE-Net.

(2) Setup the "dyndns.js" module with your appropriate QWK-id information so that the hostname "mybbs.synchro.net" will point towards your correct IP address. This is required so that users who try to reach your IRC server will be able to resolve the hostname used on the IRC network. That way, if anyone wishes to connect to your server/BBS specifically, they'll be able to use "mybbs.synchro.net" (i.e. if your server happens to be faster, closer, or offers interesting BBS features.) The dyndns.js module comes with Synchronet 3.11.

To enable the dyndns module, add a timed event inside of SCFG under "External Programs->Timed Events" to run the command "?dyndns <pass>" every now and then. Replace <pass> with your exact QWK password. Running the module once per day is usually acceptable, although you may want to run it more often if your IP address is prone to changing rapidly.

After the timed event has been configured, force the event to be ran. This can be done at the BBS itself (by using ";EXEC ?dyndns <pass>" from the BBS main menu.) Then, after about 3 or 5 minutes, attempt to ping your new hostname (qwk-id.synchro.net, where 'qwk-id' is your qwk-id.) It is very important that the IP address you're connecting to the Synchronet IRC Network from and the IP address that your new hostname (in the form of qwk-id.synchro.net) resolves to are exactly the same, otherwise you won't be able to link. This is to prevent anyone from arbitrarily linking their server to the network by utilizing your qwk-id.

(3) Edit your ircd.conf and include a C/N line pair for connecting to 'vert.synchro.net'. These should be commented out in the stock ircd.conf, and will look like this:

```
#C:vert.synchro.net:QWK_PASSWORD:*.synchro.net:6667:30
#N:vert.synchro.net:*.synchro.net::30
```

Remove the '#' from each line, and replace 'QWK\_PASSWORD' with the password you were assigned (or selected) when registering for a QWK-ID. The ircd.conf contains a description of what each of the lines (and fields) mean. It is very important that you leave the asterisks as they are, especially on the N:Line. This is because the server you're connecting to may be randomly assigned, and the server will never echo your QWK password back to you, so it chooses to echo a '\*' back instead. An asterisk in the N:Line also forbids any servers from connecting \*to\* you, which is important, since you'll only be doing outbound connects with this C/N pair.

(4) Restart your BBS (or, if you know how to become an IRC operator, simply use the /REHASH command), and you should see a message similar to the following in your Synchronet console:

```
srvc 0008 IRC Routing: Auto-connecting to rrx.synchro.net
srvc 0008 IRC Routing: Connected! Sending info...
srvc 0008 IRC 0018 Accepted new connection: 154.5.119.21 port 6667
srvc 0008 IRC Routing: Link with rrx.synchro.net established, states: TS
```

If you see any messages in regards to "Server not configured" or "Connection reset by peer", it's highly likely that you've mistyped your QWK password into the C:Line in your ircd.conf. Double-check to make sure that the password is correct, and that you haven't otherwise malformed the C/N line pair. In particular, make sure all the asterisks (as per the default) are where they should be.

Otherwise, if you have received those messages, then you're connected! You should be able to join the typical busy Synchronet IRC channels, #bbs and #synchro.net, and be able to chat with people across the network. You can find network administrators in #opers if you have any questions or concerns.

### [3.3] - Using JSEXEC to run the IRCd

There are times where you may wish to run the IRCd service separately from Synchronet so that whenever your BBS goes up or down, the IRCd isn't affected. A special program, included with Synchronet, is called "JSEXEC" and is intended for use in this way. By using JSEXEC, your IRCd will remain operational regardless of what your BBS is doing, while still integrating with all of the regular Synchronet features. In fact, in most respects, running

the IRCd via JSEXEC is the preferred method.

To run your IRCd with JSEXEC, make sure that you've followed all the installation instructions above. In particular, take a look at your M:Line on your ircd.conf and ensure that the last argument is the port you wish to run the IRCd on (typically 6667.) If you're currently running the IRCd through Synchronet, shutdown your BBS and comment out (or remove) the sections in your services configuration files (services.ini or services.cfg) so that the service is not restarted when you bring the BBS back up.

Just like when you're running the IRCd from within Synchronet, you need to tell JSEXEC that the service you're running is to be 'looped,' which is done with the -l option. Thus, a typical JSEXEC execution will look like this:

```
jsexec -l ircd
```

The above command is typed from within the Synchronet 'exec' directory. All console commands and errors are logged to the terminal that JSEXEC was started from. You should see the standard IRCd startup messages, which means that the IRCd is now operational through JSEXEC. Connecting to the IRCd should now work as per normal.

----- 4.0 -- About the Synchronet IRC Network (irc.synchro.net) -----

The Synchronet IRC Network is currently a small network with a BBS focus. Like all new IRC networks, we hope that with the help of other BBS sysops around the globe, the Synchronet IRC Network will grow to become a thriving community sporting a wide variety of topics. Currently, the network has a very relaxed authoritative structure -- perhaps one of the most relaxed among all IRC networks. Even so, certain 'common sense' rules still apply.

Servers linked to the Synchronet IRC Network are automatically put into the DNS round-robin for 'irc.synchro.net', which means that you can expect to see connections from other clients who choose to use that address to connect to IRC. You should expect to see your server listed in the round-robin within about 30 minutes, although it typically takes less.

----- 5.0 -- Technical Information -----

#### [5.1] - Limits of the Synchronet IRCd

Although the Synchronet IRCd Service is written in Javascript, an interpreted scripting language, it has been written to scale relatively well. Thanks to the DALnet Bahamut testing team, the IRCd has successfully held over 1,000 clients without any noticeable slowdown. The old limit of 100 users has been eliminated since version 1.1 of the IRCd.

If you notice any slowdown or scaling problems, please let us know.

#### [5.2] - Compliance with RFC's, and established protocols

The Synchronet IRCd has always aimed to be compliant with RFC1459, which was the first published IRC specification. However, it has chosen to deviate from the RFC where appropriate. This might be because of errors inside the RFC itself (i.e. +p channels being listed as "\*" instead of "Prv",) for the purpose of added functionality (i.e. handling of the PASS message for dynamic QWK connections,) or for security (not displaying some sensitive STATS output to users who are not IRC operators.) Compliance with the newer IRC RFC papers (inclusive of RFC's 2810 through 2813) is mostly correct, however deviates wherever Bahamut-specific extensions conflict.

The DALnet-style Bahamut extensions to the server-to-server protocol involve improving performance between server links by reducing the amount of traffic that needs to go across any link. Furthermore, extra arguments are added to common commands (NICK, MODE, TOPIC, etc) in order to better establish the authenticity of the message. In particular, timestamps ("TS") have been added to many commands in order to resolve conflicts between messages. Although the Bahamut extensions are largely undocumented, the author chose to use these extensions as a base for extending the IRC protocol (as described in RFC1459) for the purpose of providing modern features.

The Synchronet IRCd diverges from common IRC practice and Bahamut IRC protocol in the following fashion:

\* The Synchronet IRCd does NOT make use of the "ident" protocol, which is popular among larger IRC networks. This exclusion was decided on because it provides very little in the way of authoritative information. Instead, a user has been considered to be "identified" (by the lack of a tilde in the username portion of the user "user@host" mask) when they have correctly identified to a local BBS account. Identifying to the BBS account is done by sending a PASS message in the registration stage. Checks against a local BBS account are done against the username, and then the nickname respectively. Thus, any IRC servers not running the Synchronet IRCd MUST NOT accept ident, as it could seriously compromise aBBS-style authorization structure.

\* The PASS message has been extended to allow for the passing and checking of QWK passwords in the case of dynamic connections:

```
PASS <password> :<qwk-id> QWK
```

No destination is specified within the message, as the routing is handled by static configuration directives (in the form of flags on the N:Line) which show a single path for the message to take. This is to ensure that the password cannot be sent over an arbitrary connection, improving the security of the message. The reply to a query looks like this:

```
PASS <result> :<qwk-id> QWK <origin>
```

Where <result> is "OK" if the password check succeeded, and anything else (typically "VOID") on failure. The origin is specified so that the message may be routed back to the correct server. Thus, a PASS message without an origin is a check, and a PASS message with an origin is a reply.

\* Leaf servers are considered to be 'untrusted' servers by default, due to the highly dynamic nature of a Synchronet-based IRC network. This is to prevent bogus messages from being injected into the network, false representation of authority, or otherwise harmful activity. Since untrusted servers are allowed to connect to the network, leaf servers are restricted in the following way beyond the standard behavior:

- All timestamps received from a leaf are ignored and are instead replaced by the current time. Thus, nickname collisions cannot be forced, and TS blasting is prevented.
- User mode +o (oper) is ignored. However, local operators still retain authority over their local server.
- The KILL and SQUIT messages are ignored and reversed if the target is connected to a server beyond the scope of the leaf.
- Services authorization modes (+z, +r, +q and friends) are ignored.
- Authenticity of mode change messages (channel ops, voice, bans, etc) are strictly checked and reversed if there's a mismatch. Mode hacking is thus prevented.
- All channel modes are bounced on behalf of the leaf by the hub upon a resync.
- Private/Secret channels are not revealed to the leaf unless a user on the leaf explicitly joins the channel.

#### [5.3] - Compatibility with other IRCd's

The Synchronet IRCd has only been tested to be link compatible with:

- \* Bahamut 1.4.35, 1.4.36 <http://bahamut.dal.net>
- \* Andy Church's IRC Services 5.0 <http://www.ircservices.za.net>

The IRCd should be compatible with any other daemon that supports the DALnet-style Bahamut extensions. If you successfully link another IRCd (including a services package, or other pseudo-server,) then please feel free to let us know about it in #synchronet. Patches may be accepted to allow the IRCd to be link compatible with other protocols at the sole discretion of the author.

#### ===== 6.0 -- The Future =====

Although the original intention of the IRCd was to allow users to interact between one another from the BBS multi-node chat area, that has yet to occur. Eventually, users will be able to talk to one another from various BBS's and not even be aware that they're using IRC as the transport protocol for their chat sessions. For the time being, one can use the Synchronet IRC client (irc.js) to connect to their local IRC server.

Further compatibility with the later Bahamut daemons is planned, including the server-to-server "RESYNCH" command, user-accessible "WATCH", and "SILENCE." Also, more umodes will be supported, in addition to the possibility of gaining some of the Bahamut channel modes (i.e. +c) Exception modes (+e, etc) and exception lines (to circumvent K:Lines) may be implemented.

Some sort of mechanism will be implemented to allow individual BBS's to share their message and file areas over IRC. This means that you'll be able to DCC send/receive files from a BBS, QWK packets, messages, and that sort of thing.

I'm sure DigitalMan has a ton of cool ideas, too ;)

#### ===== 7.0 -- Frequently Asked Questions =====

##### [7.1] - Installation Questions

Q: After setting up my IRCd, and trying to connect, it gives me an error stating: "You're not authorized to use this server." or "Your host isn't among the privileged." (Numeric 463)

A: Most likely your Javascript libraries are too old. Upgrade to Synchronet 3.11, which contains sufficiently modern Javascript libraries. If you're certain that you're running new libraries, double-check your I:Lines and Y:Lines inside the ircd.conf. Javascript libraries before the year 2003 are considered to be too old; you can check your Javascript library version from the 'system information' portion of your BBS main menu.

Q: I try to connect and it tells me "Password Incorrect" or "Denied."  
(Numeric 464)

A: You've required a password to be passed to the IRC server via the PASS command upon registration. Check your I:Lines.

Q: I keep on getting "Error setting up socket for listening" or "Error binding socket to TCP port."

A: It's highly likely that you already have something running on the port that you've defined the IRCd to use (typically 6667.) If you have another IRCd running, shut it down or redefine the port that the Synchronet IRCd uses to listen for connections. If you get this error after having recently shut down Synchronet, one or more sockets may be stuck in 'TIME\_WAIT' or similar, in which case you should wait until they expire (as viewable with "netstat".)

Q: My server keeps on trying to auto-connect to other servers, and I don't want this! What do I do?

A: The IRCd will attempt to auto-connect if there's a port defined in the C:Line for that server. Furthermore, a connect frequency must be defined in a Y:Line for the applicable IRC class for an auto-connect to be attempted. Remove the port from the C:Line, or turn the auto-connect frequency down to 0 in the Y:Line.

Q: I installed the IRCd, but where are IRC Services? (ChanServ, NickServ, etc.)

A: IRC Services are not a part of the Synchronet IRCd. Instead, we recommend that you make use of an external services package such as Andy Church's IRC Services (<http://www.ircservices.za.net>) Furthermore, any services package which is compatible with the Bahamut inter-server protocol should work. There *may* be an IRC services package written in Javascript for Synchronet later.

Q: I've linked to the Synchronet IRC Network, and now I'm getting all sorts of weird 'Routing' messages.

A: These messages are sent whenever an 'important' event occurs on the IRC network. This includes whenever a server links or delinks from the network, which occurs with quite a bit of regularity. These messages are informational only, and simply indicate that your IRCd (and the network at large) are operating correctly.

Q: I've successfully installed my IRCd, now how do I become an IRC Operator?

A: Use the /OPER command, which has a syntax of '/OPER <nick> <pass>'. By default, the Synchronet IRCd configuration file is already pre-configured so that you may OPER to a nickname of 'Sysop' with the BBS system password so long as you're connecting from the BBS machine itself, or a system on your local network. This means you would execute the command like this: '/OPER Sysop <syspass>'. Also check out the O:Line section in ircd.conf.

Q: My question isn't answered in this document, where can I go?

A: First, make sure you've read this document *\*in its entirety\**  
Second, read section 2.2 again.  
Third, if you're still having a problem, feel free to join #synchronet on irc.synchro.net, the author typically uses the nick of 'Cyan' Support is not given via email or otherwise.

## [7.2] - Technical Questions

Q: Why were the Bahamut protocol extensions used instead of EFnet, Undernet, Unreal, or otherwise?

A: The Bahamut extensions were simply the extensions that the author was most familiar with. Bahamut is a widely-deployed IRC daemon in use by many networks (DALnet in particular,) so it's had a strong test cycle. Furthermore, all modern IRC extensions are relatively similar to each other, but differ in name. For example, Bahamut's SJOIN is similar to EFnet's NJOIN.

Q: Will you be adding support for <X> protocol, or for <X> IRCd?

A: Highly unlikely. However, feel free to add in your own support and send me a diff. Depending on the scope of the changes, I may choose to include them, or perhaps offer it externally as a patch. When coding in support for different IRC protocol extensions, try to make use of modularity so that it's possible to link together servers of different protocols, with the IRCd as the bridge. Patches that follow this 'modular' approach are more likely to be accepted.

Q: Can I make modifications to the IRCd?

A: By all means, go for it! If you think that you've made a particularly useful or clever hack, please feel free to send your changes to ircd@rrx.ca (in unified diff format, preferably) along with a description of what you've changed.

Q: I'm an IRC guru, and I'd like to talk to the author about the IRCd, where do I go?

A: The author can typically be found in #synchronet as 'Cyan' on irc.synchro.net, and welcomes all discussion about IRC protocol, theory (especially as it relates to IRC3 proposals,) or general banter among long-standing IRC users.

=====- EOF -=====



Synchronet Dynamic Domain Name System (DynDNS) Service  
=====

\$Id: dyndns.txt,v 1.2 2007/08/05 02:26:38 rswindell Exp \$

## Introduction

-----  
If your BBS computer or router has a dynamically-assigned IP address (e.g. via DHCP or PPP), you need a dynamic hostname so that your BBS users can always logon to your BBS without knowing your current IP address (since it can change at any time).

Even if you have a static (never changing) IP address, it is much more friendly to users to provide them with a hostname than an IP address with which to logon to your BBS. Humans find it much easier to remember (and type) "vert.synchro.net" than "69.104.209.210".

There are many dynamic DNS services available on the Internet (most are free). Here are some popular DynDNS services (not associated with Synchronet):

<http://www.dtdns.com/>  
<http://www.no-ip.com/>  
<http://www.dyndns.org/>  
<http://www.zoneedit.com/>  
<http://www.d2g.com/>  
<http://www.bbs.us/dns/>

Some of these services can even host full domain names (e.g. yourbbs.com) with dynamic IP updates and even provide advanced services such as mail-relaying and port-forwarding (sometimes for a fee). All of these services require some 3rd party software to run in the background when your system boots and automatically detect changes to your IP address (not strictly required if you have a static IP address).

## yourbbs.synchro.net

-----  
If you'd like a "synchro.net" hostname, we offer a free, very easy-to-use dynamic hostname service exclusively for Synchronet sysops. Your synchro.net hostname will always point to your current IP address, as long as your BBS is running. Using this service requires Synchronet v3.10 or later (for Win32 or Unix).

## Directions:

1. Download the following file into your Synchronet "exec" directory:  
<http://cvs.synchro.net/cgi-bin/cvsweb.cgi/~checkout~/exec/dyndns.js>
2. If you haven't already, create a QWK networking account on Vertrauen (telnet://vert.synchro.net), using your BBS's QWK-ID for the username (see <http://synchro.net/docs/dove-net.txt> for more details).

Note: You do not have to actually join DOVE-Net to get a synchro.net hostname, if you do not wish.

3. Setup a timed event (in SCFG->External Programs->Timed Events) to run the command-line "?dyndns yourpass" periodically, where yourpass is the password you used when you created your QWK networking account on Vertrauen. "DYNDNS" is the suggested internal code to use for this event.

If you have a frequently changing IP address (and Synchronet v3.11 or later) set "Always Run After Init/Re-init" to "Yes". With this option set to "Yes", frequent timed updates should not be necessary.

If you have a static IP address, one run (update) per month should be sufficient.

Make sure the "Enabled" is set to "Yes".

4. After saving your changes in SCFG, and verifying that the BBS/Telnet Server has recycled, check the event log window or console output to verify that the "DYNDNS" event has executed. You can force the event to execute with the (Win32) SBBSCtrl:BBS->Force Timed Event menu option, or by "touching" (creating or modifying) the file "dyndns.now" in your Synchronet data directory.
5. Attempt to ping or connect to services at your-id.synchro.net, where your-id is your BBS's QWK-ID. It may take a few minutes for a new hostname to become active or for an existing hostname to resolve to a newly changed IP address.

You're ready to go with your new synchro.net hostname!

## Options

-----

-mx [hostname]

DNS MX (mail exchange) resource records are used to designate a specific hostname as the "mail exchange" (receiving SMTP server) for a specific hostname. If there is no MX record, then the hostname itself is assumed to perform the role of "mail exchange". Some receiving SMTP servers now require that the *sending* SMTP server correspond to the MX record of the "from" hostname. If you want to specify an MX record for your hostname, you can do so by adding "-mx" (optionally followed by a hostname) to your "?dyndns" command-line (after the password). If no hostname is specified, then an MX record pointing back to your hostname will be created. Example:

```
"?dyndns password -mx mail.someother.org"
```

-hn <base\_hostname>

If you want to update a hostname that is *different* than your BBS's QWK-ID, then you can specify that hostname (QWK-ID) on the "?dyndns" command-line (after the password) by using the "-hn" option followed by the hostname to use (not including ".synchro.net"). There must be a corresponding QWKnet account on Vertrauen for the specified hostname and password.

[ip\_address]

If for some reason you want to specify a *different* IP address than the one that your BBS will be using to connect with Vertrauen, you can include that IP address on the "?dyndns" command-line (after the password). This is an advanced feature that you should not need to use under normal conditions, even if your BBS is behind a firewall/router using network address translation (NAT) - the DynDNS server running on Vertrauen uses your correct *public* IP address by default.

```
/* End of file */
```

~~~~~

Apr 15 2000

Rob Swindell

~~~~~

SyncEdit is an external message editor originally designed for use with Synchronet Multinode BBS Software. It was later extended to support other BBS packages as well.

~~~~~

Robert D. Bouman is the author of SyncEdit. Unfortunately he passed away a few years ago. I, Rob Swindell, am the author of Synchronet BBS Software and while I was given permission to include an unregistered demo version of SyncEdit in Synchronet distributions, I am not the author of SyncEdit and have never had any official connection with the product. I've decided to create this help file to answer all the common questions I get regarding SyncEdit (the most common being: "How do I register YOUR product, SyncEdit?").

~~~~~

Since it is no longer possible to purchase SyncEdit registrations, I have been including my personal registration key (SYNCEDIT.KEY, registered to "Digital Dynamics") with Synchronet distributions since v2.30b. Feel free to use this registration key guilt-free, as that's what I think Robert would have wanted. If you're using v2.20 or later, you'll need to convert this key to the SYNCEDIT.CFG file format by using the KEY2CFG.EXE program (included with the SyncEdit archive: SYEDTxxx.ZIP).

~~~~~

There is a Y2K bug in all known versions of SyncEdit where the year 2000 is displayed as 100, causing the status bar to wrap and making the program unusable. I've released a group of binary-edited SyncEdit executables (for all located versions) that works around this problem: the year is still displayed as 100, but the status bar no-longer wraps. The file that includes these patched files is SYEDTY2K.ZIP and can be downloaded via Telnet/Zmodem or FTP from [vert.synchro.net](http://vert.synchro.net).

~~~~~

First, it is very important that you know WHAT VERSION of SyncEdit you're trying to get working. You can tell by looking at the file size of your SYNCEDIT.EXE in your Synchronet EXEC directory and comparing to the following list of known/located released versions:

|        |       |
|--------|-------|
| 166222 | 1.5.3 |
| 166570 | 2.0.0 |
| 162594 | 2.0.5 |
| 164490 | 2.20A |
| 170900 | 2.50  |
| 192126 | 3.00  |
| 191906 | 3.05  |

If the file size of your SYNCEDIT.EXE does not match any of the above, check your SyncEdit documentation for the version number or logon to Vertrauen ([vert.synchro.net](http://vert.synchro.net)) to obtain one of the above versions (and the Y2K patches).

Different versions of SyncEdit require different configurations in SCFG (SBBCTRL->BBS->Configure)->External Programs->External Editors->SYNCEDIT:

[illegible]

```

Name          SyncEdit          °
° Internal Code          SYNCEDIT          °
° Local Command Line      %!syncedit %! %1 %f          °
° Remote Command Line      %!syncedit %! %1 %f          °
° Access Requirements      ANSI          °
° Intercept I/O Interrupts  Yes <- Versions 1.0 - 1.5.0 only! °
° Quoted Text              None          °
° QuickBBS Style (MSGTMP)  No          °
° Expand Line Feeds to CRLF  No          °
° BBS Drop File Type        Synchronet      XTRN.DAT          °

```

v1.5.3 - v2.0.5



binkd Docs are available at <http://www.doe.carleton.ca/~nsoveiko/fido/binkd/man/binkd.man.html>  
Ok, let's just run through them then... I'll use the following:

*bP* for your binkd path (Mine is /usr/bbs/sbbs/fido)  
*sP* for the sbbs path  
*fZ* for the Fidonet Zone (1 for north america)  
*fA* Your complete fidonet address (ie: 1:140/17)  
*SysN* For your BBS name (ie FreeBSD Synchronet)  
*SL* For your BBS location (ie Yorkton, Sk)  
*SN* For the sysop name (ie Stephen Hurd)  
*HfA* Your hubs address (ie 1:140/1)  
*HiA* Your hubs internet address  
*PASS* The password for the hub

Note, IONs are listed as 300bps in the nodeinfo.  
----- Start of binkd.conf -----

```
domain fidonet bP/outbound fZ
domain fido alias-for fidonet
domain fidonet.org alias-for fidonet
address fA@fidonet
sysname "SysN"
location "SL"
sysop "SN"
nodeinfo 300,TCP,BINKP
connect-timeout 60
try 10
hold 600
send-if-pwd
log bP/binkd.log
loglevel 4
percents
printq
backresolv
inbound bP/inbsecure
inbound-nonsecure bP/inbound
temp-inbound bP/incomplete
minfree 2048
minfree-nonsecure 2048
kill-dup-partial-files
kill-old-partial-files 86400
kill-old-bsy 43200
flag /usr/bbs/sbbs/data/fidoin.now *.pkt *.PKT
flag /usr/bbs/sbbs/data/fidoin.now *.su? *.mo? *.tu? *.we? *.th? *.fr? *.sa? *.SU? *.MO? *.TU? *.WE? *.TH? *.FR? *.SA?
prescan
node HfA@fidonet -md HiA PASS
```

----- End of binkd.conf -----

Then you need to set up sbbsecho.  
Here's my sbbsecgo.cfg file with the same replacements as above:  
----- Start of sbbsecho.cfg -----

```
NOTIFY 1
SECURE ECHOMAIL
FUZZY_ZONE
FLO_MAILER
KILL_EMPTY
LOG 0FFFFFFF
INBOUND bP/inbound/
SECURE_INBOUND bP/inbsecure/
OUTBOUND bP/outbound/
ARCSIZE 0
PKTSIZE 0
PKTPWD ALL PASS
PACKER ZIP 0 504B
    PACK /usr/local/bin/zip %f %s
    UNPACK /usr/local/bin/unzip -j %f -d %s
END
USEPACKER ZIP ALL
```

```
AREAFIX      ALL  PASS
ROUTE_TO    HfA  ALL
```

----- End of sbbsecho.cfg -----

That's all find and dandy if Fido is the only FTN network you're on... but what if you're a member of more than one.. For example, xleague (<http://www.x-bit.org>) \*and\* FidoNet? Well, things get a teensy bit more complicated. We'll need a couple more replacements.

```
xA
    Your complete x-league address (ie: 777:777/52)
HxA
    Your x-league hubs address (ie: 777:777/0)
HxiA
    Your x-league hubs internet address (ie: x-bit.org)
xPASS
    The password for the x-league hub
```

Moreover, we'll have to explicitly route all FidoNet stuff by zone to your fidonet hub. The DEFAULT\_ZONE does not change per network with sbbsecho... it is what's set up as the zone for your first FTN address. I've **bolded** all the lines that are different from the previous example.

----- Start of binkd.conf -----

```
domain fidonet bP/outbound fZ
domain fido alias-for fidonet
domain fidonet.org alias-for fidonet
domain xleague bP/outbound fZ
address fA@fidonet
address xA@xleague
sysname "SysN"
location "SL"
sysop "SN"
nodeinfo 300,TCP,BINKP
connect-timeout 60
try 10
hold 600
send-if-pwd
log bP/binkd.log
loglevel 4
percents
printq
backresolv
inbound bP/inbsecure
inbound-nonsecure bP/inbound
temp-inbound bP/incomplete
minfree 2048
minfree-nonsecure 2048
kill-dup-partial-files
kill-old-partial-files 86400
kill-old-bsy 43200
flag /usr/bbs/sbbs/data/fidoin.now *.pkt *.PKT
flag /usr/bbs/sbbs/data/fidoin.now *.su? *.mo? *.tu? *.we? *.th? *.fr? *.sa? *.SU? *.MO? *.TU? *.WE? *.TH? *.FR? *.SA?
prescan
node HfA@fidonet -md HiA PASS
node HxA@xleague -md HxiA xPASS
```

----- End of binkd.conf -----

And the corresponding sbbsecho.cnf looks like this:

----- Start of sbbsecho.cfg -----

```
NOTIFY 1
SECURE_ECHOMAIL
FUZZY_ZONE
FLO_MAILER
KILL_EMPTY
LOG 0FFFFFFF
INBOUND bP/inbound/
SECURE_INBOUND bP/inbsecure/
OUTBOUND bP/outbound/
ARCSIZE 0
PKTSIZE 0
PKTPWD 1:ALL PASS
PKTPWD 2:ALL PASS
PKTPWD 3:ALL PASS
PKTPWD 4:ALL PASS
PKTPWD 5:ALL PASS
PKTPWD 6:ALL PASS
PKTPWD 777:ALL xPASS
PACKER ZIP 0 504B
    PACK /usr/local/bin/zip %f %s
    UNPACK /usr/local/bin/unzip -j %f -d %s
END
```



USEPACKER ZIP 1:ALL  
USEPACKER ZIP 2:ALL  
USEPACKER ZIP 3:ALL  
USEPACKER ZIP 4:ALL  
USEPACKER ZIP 5:ALL  
USEPACKER ZIP 6:ALL  
USEPACKER ZIP 777:ALL  
AREAFIX 1:ALL *PASS*  
AREAFIX 2:ALL *PASS*  
AREAFIX 3:ALL *PASS*  
AREAFIX 4:ALL *PASS*  
AREAFIX 5:ALL *PASS*  
AREAFIX 6:ALL *PASS*  
AREAFIX 777:ALL *xPASS*  
**ROUTE\_TO *HfA* 1:ALL**  
**ROUTE\_TO *HfA* 2:ALL**  
**ROUTE\_TO *HfA* 3:ALL**  
**ROUTE\_TO *HfA* 4:ALL**  
**ROUTE\_TO *HfA* 5:ALL**  
**ROUTE\_TO *HfA* 6:ALL**  
**ROUTE\_TO *HxA* 777:ALL**  
  
----- End of sbbsecho.cfg -----

## Synchronet User Documentation

### Table of Contents

[How to use the User's Manual](#)

[Quick Key Sequence Guides](#)

[Nomenclature](#)

[Logging onto the BBS](#)

[Logging off the BBS](#)

[Displaying Menus](#)

[Control Commands](#)

[Setting Up Your User Account](#)

[Account Defaults Configuration](#)

[Electronic Mail Functions](#)

[Electronic Mail Menu](#)

[Send Local E-mail](#)

[Send Internet Email / NetMail](#)

[Read Mail](#)

[Read Sent Mail](#)

[Send Sysop Feedback](#)

[Message Base Functions](#)

[Group and Subboard Selection](#)

[Reading Messages](#)

[Read Messages](#)

[New Message Scan](#)

[Browse](#)

[Continuous New Scan](#)

[Posting Messages](#)

[Posting a message](#)

[Private message vs E-mail](#)

[Auto Message](#)

[Synchronet Message Editor](#)

[Slash Commands](#)

[Control \(Ctrl\) Commands](#)

[Text Attributes \(Ctrl-Ax where x is attribute\)](#)

[QWK Mail Packet Menu](#)

[Configuration Commands](#)

[Configuration Menu](#)

[Convert Credits to Minutes](#)

[Unfiltered Input Switch](#)

[New Message Scan List Configuration](#)

[New Message Scan Date/Time Pointers](#)

[Reinitialize New Scan Date/Time Pointers](#)

[Online User Interaction](#)

[Users Online <Ctrl-U>](#)

[List Node Activity {/L}](#)

[Send Private Message to Node <Ctrl-P>](#)

[Chatting Features](#)

[Chat Menu](#)

[Paging disable/enable](#)

[Activity Alerts disable/enable](#)

[Page Sysop to Chat](#)

[Talk with the Guru](#)

[Join/Initiate Multinode Chat](#)

[Private Node to Node Chat](#)

[Chat Command Prompt \(/ from within chat\)](#)

[Page User to Chat \(multinode chat only\)](#)

[Change Channel \(multinode chat only\)](#)

[Text File Section](#)

[External Services](#)

[User Listings](#)

- [Information/Statistics](#)
  - [Information Command Menu](#)
  - [System Information](#)
  - [Version Information](#)
  - [Subboard Information](#)
  - [Your Account Statistics](#)

- [File Transfer Section](#)
  - [Library and Directory Selection](#)
  - [Wildcards](#)
  - [Listing Files](#)
  - [Extended File Information Listing](#)
  - [New File Scan](#)
  - [Search for Filename](#)
  - [Find Text in Description](#)
  - [Quick Keys](#)
  - [File Transfer Area Information](#)
    - [File Transfer Information Menu](#)
    - [BBS File Transfer Policy](#)
    - [Current Directory Information](#)
    - [Users With Access to Directory](#)
    - [Your File Transfer Account Statistics](#)
  - [File Transfer Area Configuration](#)
    - [File Transfer Area Configuration Menu](#)
    - [Batch Download Quick File Flagging](#)
    - [Change New File Scan Date/Time](#)
  - [Downloading Files](#)
    - [Download File or Files](#)
    - [Download User to User File](#)
    - [Download Options](#)
  - [Uploading Files](#)
    - [Upload File or Files](#)
    - [Upload User to User File](#)
    - [Upload File to Sysop](#)
    - [Upload Procedures](#)
    - [Remove / Edit File](#)
  - [Batch File Transfers](#)
    - [Batch/Bidirectional Commands](#)
  - [File Extraction/ Temp Directory](#)
    - [Temp Directory Commands](#)
    - [Extract from archive in Temp Directory](#)
    - [Add files to Temp file](#)
    - [Download Temp file](#)
  - [Creating and downloading file lists](#)

[Back to Table of Contents](#)

# How to use the User's Manual

**Quick Key Sequence Guides**

-----

Each command title will be followed by a quick key sequence guide. The command sequences are the exact keystrokes to enter from the Main or File Transfer prompt to execute that command without pausing to look at menus. Key sequences to be entered from the Main prompt are enclosed in brackets. For example the Send E-Mail command title will be listed as:

Send E-Mail [ES]

You would type E then S from the Main Prompt (noted by the brackets) to Send E-Mail. Key sequences to be entered from the File Transfer prompt are enclosed in parenthesis. For example the Find File command title would be listed as:

Find File (F)

For this command you would type F from the File Transfer prompt (noted by the parenthesis). If a command may be entered from either the Main or File Transfer prompt it will be enclosed in curly brackets. For example the Join Multinode Chat command title would be listed as:

Join/Initiate Multinode Chat {CJ}

For this command you would type C then J from either the Main or File transfer prompt.

[Back to Table of Contents](#)

**Nomenclature**  
-----

Throughout the documentation Carriage Return (or the ENTER key) is referred to as <CR>.

There are two primary prompts on the Synchronet BBS, the Main prompt from the Main section, and the File Transfer prompt, from the File Transfer section. These can be identified by diagrams in the Message Base Functions chapter and the File Transfer Section chapter. When referred to in the documentation the Main and File Transfer prompts are italicized.

Most key strokes referred to in the documentation are highlighted in bold>.

[Back to Table of Contents](#)

# Logging onto the BBS

After dialing and connecting to the BBS:

At NN: prompt enter your username or user number.

At the PW: prompt enter your unique user password.

[Back to Table of Contents](#)

# Logging off the BBS

To logoff the BBS enter O from either the Main prompt or the File Transfer prompt. Enter /O for a quick logoff.

[Back to Table of Contents](#)

# Displaying Menus

Hitting ? from almost any prompt will display a menu of available commands. From inside chat or the Synchronet editor /? will display a menu of the available commands.

[Back to Table of Contents](#)

# Control Commands

Control commands are commands that are available from anywhere on the BBS. Control commands are primarily used to control the scrolling of text at anytime on the system or to abort a command in process such as a file search. Two online user to user commands are also available to list the users currently online or send messages at anytime while on the BBS. Control commands are entered by holding down the Ctrl key and hitting the control letter. For example, to enter the control command Ctrl-S hold down the Ctrl key and strike the S key. The following control commands are available at all times while on the BBS:

<sm heading>Scrolling Text/ Messages/ System Commands  
Ctrl-S Pause scrolling text/messages  
Ctrl-Q Continue scrolling text/messages  
Ctrl-C Abort scrolling text/messages or a BBS command

<sm heading>Online User Information/Interaction  
Ctrl-P Send a private one line message to another user  
Ctrl-U List users online

<sm heading>Other Information  
Ctrl-K List control commands  
Ctrl-T Time online info

[Back to Table of Contents](#)

# Setting Up Your User Account

**Account Defaults Configuration [D]**  
-----

To display and or change your user account defaults hit D from the Main prompt. Your account defaults will be displayed.

A) ANSI Terminal : Yes, Color  
E) External Editor : None  
L) Screen Length : Auto Detect (24)  
K) Command Keys : Synchronet

X) Expert Menu Mode : On  
P) Screen Pause : Off  
S) Spinning Cursor : Off  
C) Clear Screen Between Messages : Off  
N) Prompt for New Message / File Scan : Off  
F) Auto New File Scan\* : Off  
R) Remember Current Sub-board : Off  
B) Batch Download File Flagging : Off  
W) Change Password (if allowed)  
Q) Quit (and accept settings)

The A) option toggles ANSI terminal support. The ANSI definition of terminal escape sequences used for text attributes and animation is supported by most communications programs and dumb terminals. If your terminal supports ANSI, this option should be set to "Yes". You can also tell the system if you have a color or monochrome display with this option.

The E) option selects an external editor as your default message editor. External editors may have more functionality or a command structure that you may prefer over the internal editor. You may only select external editors from a list that the sysop has configured for the system.

The L) option sets how many lines (or rows) your display has. If your terminal or communications program has a status line, you will want this set to the number of lines your terminal can display excluding the status line. If this option is set to "Auto Detect", the number of screen lines your terminal supports will be automatically detected upon each logon.

The K) option selects an optional command key set. You may select an alternate command key set resembling another BBS type or any command set the sysop has predefined. You may also define your own command set. If you select an alternate command set you will also receive matching menus. However, if you define your own set you will not receive matching menus.

The X) option toggles Expert menu mode on and off. When Expert menu mode is turned off menus automatically appear at each new command prompt. When Expert menu mode is turned on menus only appear as requested with the '?' command. Expert menu mode off is generally referred to as Novice mode.

The P) option toggles screen pausing. When this option is on, every screen full of text displayed to you will pause with a "Hit a key" prompt. When this option is off, the user must use the Ctrl-S and Ctrl-Q commands to pause and unpause scrolling text.

The S) option toggles the spinning cursor on and off

The C) option toggles screen clearing before each message (posts or E-mail) is displayed.

The N) option toggles whether or not the BBS will automatically prompt you for a new message and file scan upon logging on.

The F) option toggles automatic new file scanning after a New Message Scan All (NA or /N) from the Main Section.

The R) option is used to toggle whether or not the BBS remembers your current subboard each time you logoff.

The B) option is used to toggle the Batch Download File Flagging prompt that is displayed after each screen full of files are listed to you. This function is handy for quickly adding files to the batch download queue without having to type the name of each file. Batch Download File Flagging can be temporarily toggled with the &B command from the Transfer Section.

The W) option is not available on all systems. If the sysop has configured the system so that users can edit their passwords, this option will appear on the menu and allow you to change your current password.

[Back to Table of Contents](#)

## Electronic Mail Functions

### Electronic Mail Menu [E]

-----  
Selecting E from the Main prompt brings you to the E-mail Prompt. From here you can read, send, and kill E-mail (mail to and from users on your BBS) and NetMail (mail to and from users on other BBSs). Your E-mail options are:

S Send E-mail  
N Send NetMail  
R Read E-Mail\NetMail  
K Read\Kill mail you have sent  
F Send feedback to the sysop  
Q Quit to the Mail Menu

[Back to Table of Contents](#)

#### **Send E-mail [ES]**

-----

Select Send E-Mail to send a mail message to another user's mailbox. You will be prompted for the user's name or user number. If you don't know the correct spelling for the target user and don't know the user number, you can enter in just part of the user's name and the system will help you find the user's full name. Enter the message title then your message. All Synchronet editor commands are available to you (see Synchronet Message Editor) in the message unless you have selected another (external) editor as your default editor. /S will save and send your message when you are finished.

[Back to Table of Contents](#)

#### **Send NetMail [EN]**

-----

Select Send NetMail to send a personal message to user on another BBS across a Fidonet style network. This feature works exactly like the Send E-Mail command except that you will be prompted for the user's name and Fidonet address. The syntax is as follows:

Username @Fidonet Address

Ex: John Doe @1:138:110

[Back to Table of Contents](#)

#### **Read Mail [ER]**

-----

After selecting Read Mail you will receive a list of the mail messages in your mailbox. An \* will be displayed next to any unread mail messages. If you have only one message in your mailbox, that message will be displayed rather than the list. Either event will be followed by the Read Mail prompt. At the Read Mail prompt you have the following commands available:

# read a mail message (where # is the number of the message).  
<CR> read the next mail message.  
- read the previous mail message.  
L list all mail message titles and authors.  
R re-read the current mail message.  
D delete the current mail message.  
A automatically reply in mail to the author of the current mail message.  
This will prompt for mail message removal.  
F forward the current mail message to another user. This will automatically remove the mail message from your mailbox.  
Q quit the Read Mail prompt and return to the Main prompt.  
? display the Read Mail menu

[Back to Table of Contents](#)

#### **Read Sent Mail [EK]**

-----

After selecting Read Sent Mail you will receive a list of active mail messages you have sent and to whom you have sent them. An \* next to a title denotes an unread mail message. If you have only one active mail message that message will be displayed rather than the list. Either event will be followed by the Read Sent Mail prompt. At the Read Sent Mail prompt the following commands are available:

# read a sent mail message (where # is the number of the message).  
<CR> read the next sent mail message.  
- read the previous sent mail message.  
L list all sent mail message titles and recipients.  
R re-read the current sent mail message.  
D delete the current sent mail message.  
Q quit the Read Sent Mail prompt and return to the Main prompt.

[Back to Table of Contents](#)

#### **Send Sysop Feedback [EF]**

-----

Send Sysop Feedback sends a private message to the Sysop. Again, all Synchronet editor commands are available to you (see Synchronet Message Editor) in the message unless you have selected another (external) editor as your default editor. /S will save and send your feedback when you are finished.

[Back to Table of Contents](#)

## Message Base Functions

The Synchronet message base is a two level system. It is organized in groups and subboards. A group is a generalized collection of related subboards. Subboards contain specific interest messages. Groups tend to be categories such as Public, Computers, 18 and Over, etc. Subboard categories are more specific such as: Public/Politics, Public/Game



Hints, Computers/Macintosh, Computers/DOS, Fidonet/For Sale, Fidonet/Trekkies, etc.

At the Main prompt you can always see your current group and subboard. The Main prompt and the File Transfer prompt can be easily differentiated by the bracket style. The Main prompt uses square brackets [ ] and the File Transfer prompt uses parenthesis ( ). The first name and number is your current group. The second name and number is your current subboard. The clock displays your time left online unless you have a time exemption in which case it will display your time spent online. ? will display the Main Menu.

[Back to Table of Contents](#)

## Group and Subboard Selection

There are two ways to select groups and subboards. You can jump to the group and subboard #'s directly or you can jump through the groups and subboards sequentially until you reach the desired selection. If you wish to jump to the group and subboard #'s directly you may want to view a list of groups or subboards first. From the Main prompt enter:

```
* to list all available subboards in your current group
/* to list all available groups

# to jump to a new subboard (where # is the subboard number)
/# to jump to a new group (where # is the group number)

} to jump forward to the next subboard
{ to jump backward to the previous subboard
] to jump forward to the next group
[ to jump backward to the previous group.
```

[Back to Table of Contents](#)

## Reading Messages

To read messages in a subboard you can either go to the Read Messages prompt or you can scan for new messages. There are three different new scan commands available. All new scan commands can scan either your current subboard, all subboards in your current group, or all subboards in all groups. Remember that the control commands are available at all times while on the BBS to control text scrolling. Ctrl-S to pause, Ctrl-Q to continue, and Ctrl-C to abort.

[Back to Table of Contents](#)

### Read Messages [R]

-----  
Read Messages, activated by hitting R at the Main prompt, is used to read messages sequentially forward or backward in your current subboard (displayed at the prompt). While reading messages you can also reply to and post messages. The following are the Read Message commands:

```
<CR> Display next message
- Display previous message
L Lists message titles and authors.
T Lists message titles and authors of next ten messages
  (advances current message forward ten messages)
R Re-read current message
F Searches for specified text in all messages.
B Bypasses remaining messages in current subboard and moves to next
  subboard in new scan list if applicable.
I Displays information about current subboard
Y Lists posts addressed to you
C Displays remaining messages in current subboard continuously
  without pausing for a Message Scanning prompt.
A Reply publicly regarding current message.
M Reply privately in mail regarding current message.
Z Reply privately in message area regarding last message.
D Deletes only the last message in current subboard if you are the
  message's author.
P Post a message in current subboard (see Synchronet Editor commands)
W Post a private message in current subboard.
Q Quits to Main prompt.
```

[Back to Table of Contents](#)

### New Message Scan [N]

-----  
After selecting New Message Scan you will be prompted to scan for new messages in your current subboard, all subboards in your current group, or all subboards in all groups. Browse and Continuous New Scan will also prompt for these selections.

After each displayed message you are provided a Read Message prompt and may execute any Read Message commands (See Read Messages). When <CR> is hit after

the last message is displayed or the subboard is bypassed (B) the new scan will continue with the next subboard in the new scan list. "Group" and "All" new scans will skip over subboards that do not contain any new messages.

Quick Keys:

- New Message Scan Subboard [NS]
- New Message Scan Group [NG]
- New Message Scan All [NA]

[Back to Table of Contents](#)

#### **Browse [B]**

-----

Browse works exactly like New Scan except on subboards that do not contain any new messages. Unlike New Scan Group, which will skip over a subboard that contains no new messages, Browse will display the last message read in a subboard that contains no new messages and provide you with a Read Message prompt (see Read Messages) allowing you to still read and post messages in that subboard. When <CR> is hit after the last message is displayed or the subboard is bypassed, by hitting B, the browse will continue with the next subboard in the new scan list.

Quick Keys:

- Browse Subboard [BS]
- Browse Group [BG]
- Browse All [BA]

[Back to Table of Contents](#)

#### **Continuous New Scan [Z]**

-----

Continuous New Scan displays all new messages continuously without providing a Message Scanning prompt.

Quick Keys:

- Continuous New Scan Subboard [ZS]
- Continuous New Scan Group [ZG]
- Continuous New Scan All [ZA]

[Back to Table of Contents](#)

## **Posting Messages**

#### **Posting a message [P]**

-----

To post (write) a message in your current subboard hit P from the Main or the Message Scanning prompt. You will also be prompted to post when you pass the last message in your current subboard when reading, new scanning, or browsing. After selecting Post Message you will be prompted whether to make the post private or public. When posting a message all Synchronet Editor commands are available to you (see Synchronet Message Editor) unless you have selected an external editor as your default editor. In which case you should consult the documentation for the external editor you have selected.

[Back to Table of Contents](#)

#### **Private message vs E-mail**

-----

Private Messages cannot be delivered directly into another user's mailbox as can E-mail or NetMail so when ever possible use E-mail and NetMail to send confidential messages to other users. However, QWK style BBS networks do not support NetMail. If you wish to send a confidential message to a user on another BBS through a QWK style network you should use a private message on a subboard. You can tell what types of networks are being utilized on a subboard by hitting I from the read message prompt or [IS] from the main prompt. NetMail is the most convenient and direct way to send confidential messages to users on other BBSs if the BBSs are connected to a Fidonet style network. (see also E-mail and NetMail)

[Back to Table of Contents](#)

#### **Auto Message [A]**

-----

The auto message is displayed to every user when the user logs on. You may change the Auto Message from the Main prompt by hitting A. Remember that as soon as another user changes the Auto Message your message will be deleted.

[Back to Table of Contents](#)

## **Synchronet Message Editor**

When using the Synchronet Message Editor to post, send mail, or send feedback you have a wealth of powerful message editing functions available. There are two types of editor commands available: Slash (/) commands and Control (Ctrl) commands.

Slash commands are preceded by the forward slash key (/) and must be entered from the beginning of any new line. Control (Ctrl) commands may be entered from any cursor position on any line.

There are two help menus available when in the editor. /? displays all standard message editing commands and /ATTR displays all available text attributes which can set by the Ctrl-A command. Text attributes set with the Ctrl-A command do not become visible until the line is word wrapped or a carriage return <CR> is entered.

[Back to Table of Contents](#)

**Slash Commands (/)**

-----  
/? Message editor help menu  
/S Save message  
/ABT Abort message  
/CLR Clear message  
/L List message  
/Lx List message from line number x  
/E Edit last line  
/Ex Edit line number x  
/D Delete last line  
/Dx Delete line number x  
/I Insert line before last  
/Ix Insert line before line number x  
/T Edit message title  
/ATTR Text attribute code list

[Back to Table of Contents](#)

**Control (Ctrl) Commands**

-----  
Ctrl-C Cancel current line  
Ctrl-V Center current line  
Ctrl-R Redraw current line  
Ctrl-X Delete current line  
Ctrl-Y \* Delete to end of line  
Ctrl-W Delete word to left  
Ctrl-D \* Delete word to right  
Ctrl-B \* Move cursor to beginning of line  
Ctrl-E \* Move cursor to end of line  
Ctrl-N \* Move cursor to next word  
Ctrl-\ \* Move cursor to previous word  
Ctrl-F \* Move cursor position forward one space  
Ctrl-Bkspc \* Move cursor position back one space  
Ctrl-Minus \* Toggle insert/overwrite mode  
Ctrl-Ax \* Toggle new text attribute where x is attribute

\* Requires ANSI

[Back to Table of Contents](#)

**Text Attributes (Ctrl-Ax where x is attribute)**

-----  
  
K Black foreground (zero) 0 Black background  
R Red foreground 1 Red background  
G Green foreground 2 Green background  
Y Yellow foreground 3 Yellow background  
B Blue foreground 4 Blue background  
M Magenta foreground 5 Magenta background  
C Cyan foreground 6 Cyan background  
W White foreground 7 White background  
  
H High intensity  
I Blinking  
N Normal (reset attributes)  
P Insert pause into message  
L Insert form feed into message

[Back to Table of Contents](#)

**QWK Mail Packet Menu [Q]**

Hitting Q from the Main prompt accesses the QWK Mail Packet Menu. QWK mail packets are used in conjunction with an offline mail reader. When you download a QWK mail packet all of your new messages in all of the subboards flagged in your newscan list and a list of new uploaded files will be compiled into a QWK file readable by most QWK supported offline mail readers. After logging off the BBS you can read the messages, E-mail, and new files list with your offline reader. You can reply to and post messages from your offline reader and upload the reply packet (.REP) to the BBS and your messages will be posted for you.

(see QWK Offline Reader Diagram) There are many offline readers available as shareware and commercial products. The QWK Mail Packet Menu offers the following commands:

D Download QWK message packet  
U Upload REP reply packet  
B Bidirectional simultaneous QWK & REP transfer  
Q Quit to Main prompt

When D or B is selected to download a QWK packet you will be prompted to select a what type of Ctrl-A filtering you wish to employ. Synchronet uses Ctrl-A codes in messages to signify color and other text attributes.

E Expand the Ctrl-A codes to ANSI, which is recommended if you wish to display ANSI color and have an ANSI compatible offline reader.  
L Leave the Ctrl-A codes in. In which case you would need an offline reader capable of converting Ctrl-A codes to ANSI.  
<CR> Strip out the Ctrl-A codes, removing the color and other text attributes from the messages.

If you are not sure which method to use the safest bet is to just hit <CR> to strip out the Ctrl-A codes.

Your QWK packet will not contain any messages you have already read since you logged on. If you wish to reset your new message scan pointers back to their state when you logged on use the Reinitialize New Message Scan Pointers [&I] command before downloading your QWK packet. After downloading a QWK packet your New Message Scan Pointers will always be set to the last messages stored in the packet. If you wish to new scan messages while still online after downloading a QWK packet you may wish to use the [&I] or [&P] command before attempting a new message scan. (see Reinitialize New Message Scan Pointers and New Message Scan Date/Time Pointers in next section)

[Back to Table of Contents](#)

## Configuration Commands

### Configuration Menu [&]

-----

Hitting & from the Main prompt access the Configuration prompt. Your choices from the Configuration prompt are:

C Convert Credits to Minutes  
R Unfiltered Input Switch  
N New Message Scan Configuration  
P New Message Scan Pointers  
I Reinitialize Message Pointers  
Q Quit to Main Menu

[Back to Table of Contents](#)

### Convert Credits to Minutes [&C]

-----

To convert online credits to online minutes hit C from the Configuration prompt. You cannot convert minutes back to credits.

[Back to Table of Contents](#)

### Unfiltered Input Switch [&R]

-----

The Unfiltered Input Switch, R from the Configuration prompt, is used to allow text files created offline with control codes, ANSI graphics and colors, and page formatting to be uploaded into a message or a mail message. To toggle unfiltered input ON hit R from the Configuration prompt. The next message you post or mail message you send will allow you to ASCII upload a text file from your computer by executing your terminal program's upload command (often PGUP) and selecting ASCII upload. Save your message by hitting Ctrl-Z. After you save the message the Unfiltered Input Switch will be toggled OFF.

[Back to Table of Contents](#)

### New Message Scan List Configuration [&N]

-----

New Message Scan List Configuration, N from the Configuration prompt, allows you to set which subboards you wish to be scanned in a new scan, browse, or express new scan. When you select New Scan Configuration you will be prompted with a list of groups. Select the group number you wish to view. A list of subboards in the selected group will be displayed. An asterisk \* next to a subboard designates the subboard is toggled ON for new scans. To toggle a subboard OFF select the subboard number. To toggle a subboard back ON, select the subboard number again. Hit Q to quit back to the group selection. Select another group or Q to quit back to the Main prompt. Your selections in the New Message Scan List Configuration are saved and remain in effect until the next time you change the settings.

[Back to Table of Contents](#)

**New Message Scan Date/Time Pointers [&P]**

-----

The New Message Scan Date/Time Pointers, P from the Configuration prompt, allows you to set back the date and time a new scan will begin to look for new messages in a specified subboard. After hitting P from the Configuration prompt select the group number you wish to change or A to change all groups. If an individual group was selected a list of subboards in the selected group will be displayed. Select the subboard you wish to change or hit A to select all subboards in that group. Enter in the new date and time. When finished hit Q to quit back to the group selection. Select another group or Q to quit back to the Main prompt.

The pointer is automatically set to the date and time of the most recently posted message you have read in a subboard each time you read messages.

[Back to Table of Contents](#)

**Reinitialize New Scan Date/Time Pointers [&I]**

-----

I from the Configuration prompt resets all your New Message Scan Date/Time Pointers back to the their original values for your current logon. This command is helpful if you have already executed a New Message Scan and then wish to download a QWK message packet containing new messages you have already read online or at any time you wish to reset your new message pointers back to their original state for your current logon.

[Back to Table of Contents](#)

## Online User Interaction

**Users Online <Ctrl-U>**

-----

To list users currently online enter Ctrl-U at any time from any section of the BBS. Users Online lists the user names, which nodes they are logged on to, their connect rate, and what each user's current action is on the BBS.

[Back to Table of Contents](#)

**List Node Activity {/L}**

-----

List Node Activity, /L from the Main prompt, the File Transfer prompt, or from inside a chat session, lists the current activity of each node on the system. List Node Activity lists all the information of Users Online (Ctrl-U) but also lists inactive nodes and their current state.

[Back to Table of Contents](#)

**Send Private Message to Node <Ctrl-P>**

-----

Ctrl-P from anywhere in the BBS will allow you to send a private one line message to another user on another node or all users on the system. After entering Ctrl-P you will see a list of users currently online and their node numbers, enter the node number or A for all users. Then enter your one line message followed by <CR>. Your message will be sent to the user on the node specified or all users on the system.

[Back to Table of Contents](#)

## Chatting Features

**Chat Menu {C}**

-----

The Chat Menu, C from the Main or File Transfer prompt, allows you to join or initiate multiuser chat, page the sysop to chat, talk with the system's all knowing Guru, toggle your paging switch ON and OFF, and toggle the activity alert switch ON and OFF. Commands available from the Chat prompt are:

- D Disable and enable paging
- A Disable and enable activity alerts
- J Join or initiate a Multinode chat session
- P Private node to node chat initiate/join
- T Talk with the system Guru
- C Page the sysop to chat
- Q Quit to Main prompt

- Ctrl-U List users online
- Ctrl-P Send private message to another user

[Back to Table of Contents](#)

**Paging disable/enable {CD}**

-----

Paging disable/enable lets you allow other users to or disallow other users



from paging you to chat. The default is Paging enabled. To disable Paging hit D from the Chat prompt. The system will show you your new node status. A (P) after your node status designates Paging is disabled. If there is no (P) after your node status then Paging is enabled (default). If you need to check your's or another user's Paging status use Ctrl-U to list users online and their current status.

[Back to Table of Contents](#)

**Activity Alerts disable/enable {CA}**

-----  
Activity alerts are messages displayed to you regarding other users activities such as logons and logoffs. The default is for Activity Alerts enabled. If you wish to disable Activity Alerts so they are not displayed to your console hit A from the Chat prompt. To reenable Activity Alerts hit A again from the Chat prompt. The system will show you your new node status. An (A) after your node status designates Activity Alerts are disabled. If there is no (A) after your node status then Activity Alerts are enabled (default). If you need to check your's or another user's Activity Alerts status use Ctrl-U.

[Back to Table of Contents](#)

**Page Sysop to Chat {CC}**

-----  
To page the sysop to chat hit C from the Chat prompt. This will alert the sysop that you wish to chat with him/her if the sysop is available.

[Back to Table of Contents](#)

**Talk with the Guru {CT}**

-----  
The system Guru is a programmable artificial intelligence engine capable of simulating and understanding human conversation. The guru may be programmed to assist users in BBS usage or purely for entertainment. To enter a conversation with the Guru hit T from the Chat prompt.

[Back to Table of Contents](#)

**Join/Initiate Multinode Chat {CJ}**

-----  
Multinode chatting is joined by hitting J from the Chat prompt. You are immediately placed in channel 1 of a 99 channel chat system. From this point any text you type will be sent out on the channel when you press <CR>. Entering a forward slash (/) will enter you into the Chat Command mode (see Chat Command prompt). Remember CTRL-U (list users online) and CTRL-P (send private message to another node) are available from anywhere on the BBS.

[Back to Table of Contents](#)

**Private Node to Node Chat {CP}**

-----  
To initiate or join someone in a two way private node to node chat hit P from the Chat prompt. If you are initiating a private chat you will be prompted for the username you wish to chat with. It will page that user to join you. If you have been paged to join someone in a private chat hit P from the Chat Menu and you will be placed in private chat with the user who paged you. Once in a private chat session hit / to acces the Chat Command mode (see Chat Command prompt). All Chat commands are available from within Private Node to Node Chat except /P and /#. An additional command, /E, is available to toggle local echo off and on so you can invoke split screen chat with your terminal program. For example, with Telix, split screen chat is ALT-Y.

[Back to Table of Contents](#)

**Chat Command Prompt (/ from within chat)**

-----

? Display Chat Command Menu  
P Page another user to chat  
# Change channel (where # is the number of the channel)  
L List node activity  
E Toggle local echo off/on (private node to node chat only)  
Q Quit chat session

Ctrl-U List users online  
Ctrl-P Send private message to another user online

[Back to Table of Contents](#)

**Page User to Chat (multinode chat only)**

-----  
Enter P from the chat command prompt to page another user online to chat. The targeted user will receive a brief message requesting they join you in multinode chat.



[Back to Table of Contents](#)

**Change Channel (multinode chat only)**

-----

You can see what channel you and others are in by listing users, Ctrl-U. If you wish to change to a different channel enter the channel number at the chat command prompt. If there are no other users in the channel you will be prompted to password protect the channel. By password protecting a channel you can create a "private" chat session. Anyone entering a password protected channel will be prompted to enter the password upon joining the channel. If you password protect a channel be sure to send the password to the users you wish to join the channel. Only the first user in the channel has the option of password protecting the channel. The password protection is lifted after the user who protected the channel leaves. The number of channels available to you is determined by the sysop.

[Back to Table of Contents](#)

## Text File Section

The BBS may contain text files available for you to read online. These files may be anything from BBS rules to special interest information to ANSI art. To enter the Text File Section hit G from the Main prompt. The system will display a list of text file areas. These are groups of related text files. Select an area and a list of files contained in that text file area will be displayed or hit Q to quit back to the Main prompt. Each file will be numbered. Enter a number at the prompt to view a file or Q to quit back to the text file area list. Remember that the Ctrl commands are available at all times while on the BBS. Ctrl-S to pause, Ctrl-Q to continue, and Ctrl-C to abort.

[Back to Table of Contents](#)

## External Services

External services (commonly referred to as Doors) are programs the sysop has added to the system and made available to the users. Programs may be games, database applications, online shopping, date/match making, and virtually thousands of other possible services. Hitting X from the Main prompt will enter you into the External Services menu where you may select from the listed services available. Each service should contain its own online instructions and help. If a service requires a charge for access the rate will be listed next to the title.

[Back to Table of Contents](#)

## User Lists

**User Lists [U]**

-----

To display a list of user accounts on the BBS, users who have access to your current subboard, or a logon list for the day hit U from the Main prompt. You will be prompted for which list you would like to view.

Quick Keys:  
User Account List [UU]  
User Account List of Subboard [US]  
Logon List for the Day [UL]

[Back to Table of Contents](#)

## Information/Statistics

**Information Command Menu [I]**

-----

Hitting I from the Main Menu will access the Information Command prompt. Information on the BBS, your user account, and the current subboard can be accessed through the Information Commands.

I System information  
V Version information on Synchronet  
S Information on your current subboard  
Y Your user account information and statistics

[Back to Table of Contents](#)

**System Information [II]**

-----

To view information regarding the BBS hit I Information Commands prompt.

[Back to Table of Contents](#)

**Version Information [IV]**  
-----

To view information regarding the version of Synchronet your node is running hit V from the Information Commands prompt.

[Back to Table of Contents](#)

**Subboard Information [IS]**  
-----

To view information regarding your current subboard hit S from the Information Commands prompt.

[Back to Table of Contents](#)

**Your Account Statistics [IY]**  
-----

To view your account statistics hit Y from the Information Commands prompt. Statistics include credits, minutes, uploads, downloads, E-mail, posts, logons, time online, and account expiration date.

[Back to Table of Contents](#)

**File Transfer Section**

The Synchronet File Transfer section is entered from the Main prompt by hitting T. To return to the Main prompt from the File Transfer prompt hit Q. The File Transfer section is organized in the same manner as the Message area; a bi-level hierarchal system with groups and sub groups. In the case of the File Transfer section there are libraries and directories. Libraries are groups of related directories. Libraries might be organized such as Shareware, Graphics, etc. The Shareware library, for example, may contain directories such as Windows, Business, Utilities, etc. The File Transfer prompt identifies your current library and directory. The first name and number is your current Library. The second name and number is your current directory. ? will display the Main Menu.

[Back to Table of Contents](#)

**Library and Directory Selection**

There are two ways to select libraries and directories. You can jump to the library and directory #'s directly or you can jump through the libraries and directories sequentially until you reach the desired selection. If you wish to jump to the library and directory #'s directly you may want to view a list of libraries and directories first. From the File Transfer prompt enter:

- \* to list all available directories in your current library
- /\* to list all available libraries
- # to jump to a new directory (where # is the directory number)
- /# to jump to a new library (where # is the library number)
- } to jump forward to the next directory
- { to jump backward to the previous directory
- ] to jump forward to the next library
- [ to jump backward to the previous library

[Back to Table of Contents](#)

**Wildcards**

Many commands in the File Transfer section accept wildcards. Wildcards allow you to search for or list a specific file or files based on part of a filename. Wildcard syntax is entered by typing part of the filename in conjunction with wildcard characters. \* will wild the remainder of a filename and/or the remainder of an extension and ? will wild specific character positions. No extension after the filename will wild the entire extension. <CR> for the entire filename will wild the entire filename and select/list all files. When wildcards are accepted the prompt will appear as follows Filespec [ \*.\* ]. Some examples of wildcard uses are:

- Filespec [ \*.\* ]: FONT?.ZIP = FONT1.ZIP or FONT2.ZIP not FONTMAN.ZIP
- Filespec [ \*.\* ]: FONT\*.ZIP = FONTMAN.ZIP or FONT1.ZIP not LANMAN.ZIP
- Filespec [ \*.\* ]: FONTMAN.Z\* = FONTMAN.ZIP or FONTMAN.ZOO not FONTMAN.png
- Filespec [ \*.\* ]: <CR> = all files
- Filespec [ \*.\* ]: FONTMAN = FONTMAN.ZIP or FONTMAN.png

<heading>

[Back to Table of Contents](#)

## Listing Files (L)

L from the File Transfer prompt lists files in your current directory. Wildcards may be used.

[Back to Table of Contents](#)

## Extended File Information Listing (E)

Hitting E from the File Transfer prompt allows you to list a file or files with extended information. This displays the filename, uploader name, time to download, credit cost, times downloaded, date/time uploaded, file date/time, and last date/time downloaded. If an extended description is included it will also be displayed. Wildcards may be used.

[Back to Table of Contents](#)

## New File Scan (N)

N from the File Transfer prompt lists all new files uploaded since your last logon. The last logon date can be changed with the Change New File Scan Date/Time command (&P). After selecting New File Scan you will be prompted to scan your current directory, all directories in your current library, or all directories in all libraries.

Quick Keys:  
New File Scan Current Directory (ND)  
New File Scan Current Library (NL)  
New File Scan All Libraries (NA)

[Back to Table of Contents](#)

## Search for Filename (S)

S from the File Transfer prompt allows you to search for a filename in your current directory, all directories in your current library, or all directories in all libraries. Wildcards may be used.

Quick Keys:  
Search for Filename Current Directory (SD)  
Search for Filename Current Library (SL)  
Search for Filename All Libraries (SA)

[Back to Table of Contents](#)

## Find Text in Description (F)

F from the File Transfer prompt allows you to list files by searching for a string of text in the file descriptions. Partial words may be used. Wildcards, however, may not be used. You will be prompted to search your current directory, all directories in your current library, or all directories in all libraries. You also will be prompted to expand the search to include extended file descriptions if desired.

Quick Keys:  
-----  
Find Text in Description Current Directory (FD)  
-----  
Find Text in Description Current Library (FL)  
-----  
Find Text in Description All Libraries (FA)  
-----

[Back to Table of Contents](#)

## File Transfer Area Information

### File Transfer Information Menu (I)

To access the File Transfer Information prompt hit I from the File Transfer prompt. The following choices are available from the prompt:

T File transfer policies  
D Current directory information  
U User account list with access to current directory  
Y Your file transfer statistics  
Q Quit back to Main menu

[Back to Table of Contents](#)

**BBS File Transfer Policy (IT)**

-----  
T from the File Transfer Information prompt displays the BBS's file transfer policies, rates, upload credit system etc.

[Back to Table of Contents](#)

**Current Directory Information (ID)**  
-----

D from the File Transfer Information prompt displays information on your current directory.

[Back to Table of Contents](#)

**Users With Access to Directory (IU)**  
-----

U from the File Transfer Information prompt lists all users with access to your current directory. The list can be sorted alphabetically or numerically by user number.

[Back to Table of Contents](#)

**Your File Transfer Account Statistics (IY)**  
-----

Y from the File Transfer prompt displays your File Transfer account statistics including credits, downloads, and uploads.

[Back to Table of Contents](#)

## File Transfer Area Configuration

**File Transfer Area Configuration Menu (&)**  
-----

Hit & from the File Transfer prompt to access the File Transfer Area Configuration prompt. From this prompt you have the following choices:

- B Batch Download File Flagging toggle ON/OFF
- P New File Scan pointers
- Q Quit to Main Menu

[Back to Table of Contents](#)

**Batch Download Quick File Flagging (&B)**  
-----

B from the File Transfer Information prompt toggles Batch Download Quick File Flagging ON and OFF. This feature allows you to add files to your batch download queue quickly and easily by simply with file list commands L, S, F, and N. With Batch File Flagging turned on file listings will list one page of files at a time displaying a letter next to each file and a prompt at the end of the list. Enter the letters of any files you wish to add to your batch download queue then hit <CR> to continue the listings. Batch Download Quick File Flagging can be set ON as your default in User Defaults (see User Defaults). Remember CTRL-C (abort listing) is available from anywhere on the BBS.

[Back to Table of Contents](#)

**Change New File Scan Date/Time (&P)**  
-----

The new file scan date and time is set each time you logoff. If you wish to new scan files from a different date than the last date you logged on, hit P from the File Transfer Information prompt. After entering the new date and time new file scans (ND, NL, and NA) will scan from the new date and time you have set. The new file scan date and time will be reset when you logoff.

[Back to Table of Contents](#)

## Downloading Files

**Download File or Files (D)**  
-----

To download a file (or files) or add files to the Batch Download queue hit D from the File Transfer prompt. You will be prompted to enter the filename. You may use wildcards.

Filespec [\*.]\*: filename.ext or wildcards

All file matches in you current directory will be listed one by one with download options after each file (see Download Options). If no matches are found in your current directory the search will automatically expand to all directories in all libraries. If a wildcard was entered the next file in the search will be displayed with download options (see Download Options) after each file is downloaded, entered into the batch/bidirectional queue or skipped.

[Back to Table of Contents](#)

**Download User to User File (/D)**

-----

To download personal user to user file transfer sent to you by another user hit /D from the File Transfer prompt. All user to user files sent to you will be listed one by one with download options (see Download Options) after each file. User to user file transfers may not be allowed on all systems.

[Back to Table of Contents](#)

**Download Options**

-----

Download options vary depending on which and how many protocols the sysop has installed on the BBS. The most common protocols are Xmodem, Ymodem, Zmodem, and Ymodem-G. The download options are:

protocol Download file with selected protocol (usually X, Y, Z, or G).

B Add file to Batch/Bidirectional Download queue (see Batch File Transfers).

Q Quit to File Transfer prompt and abort search.

<CR> Display next file and download options. Will quit to File Transfer prompt on last file.

[Back to Table of Contents](#)

**Uploading Files**

**Upload File or Files (U)**

-----

To upload a file to the current directory or add a file or files to the Batch Upload Queue (see Batch File Transfers) hit U from the File Transfer prompt. You will be prompted for the filename. Now see Upload Procedures.

[Back to Table of Contents](#)

**Upload User to User File (/U)**

-----

To upload a file to another user or users hit /U from the File Transfer prompt. You will be prompted to enter the filename. After confirming the filename you will be prompted to select the destination user or users. Enter the user's name, user number, or partial name. After the user has been selected you may select additional users or <CR> when finished. User to user file transfers may not be allowed on all systems. Now see Upload Procedures.

[Back to Table of Contents](#)

**Upload File to Sysop (Z)**

-----

To upload a file to the sysop's file directory hit Z from the File Transfer prompt. You will be prompted for the filename. User to sysop file uploads may not be available on all systems. Now see Upload Procedures.

[Back to Table of Contents](#)

**Upload Procedures:**

-----

You now may be asked if the file is part of a multiple file set. If the file is not a part of a set (1 of 3 for example) then hit <CR> or N. You may now be asked to rate the file based on its contents. Next you will be prompted to enter the one line file description as you wish it to appear in the file listing. Then you may be asked to enter an extended description. If you do not wish to enter an extended description hit <CR> or N. When entering an extended description all Synchronet editor commands are available to you (see Synchronet Message Editor) in the description unless you have selected another (external) editor as your default editor. /S will save your description when you are finished.

After the extended description you will be prompted with upload options. The upload options are:

protocol Upload file with selected protocol (usually X, Y, Z, or G).

B Add file to Batch Upload queue (except user to user).

Q Quit to File Transfer prompt and abort search.

[Back to Table of Contents](#)

**Remove / Edit File (R)**

-----

To remove any file or edit the description or extended description of any file you have uploaded on to the BBS hit R from the File Transfer prompt.

[Back to Table of Contents](#)

**Batch File Transfers**



You can make uploading and or downloading multiple files much easier by using batch file transfers. In batch file transfers you have two queues (like holding tanks) in which to store files to be transferred. The upload queue holds files to be uploaded and the download queue holds files to be downloaded. To place files into the queues use the standard upload and download commands from the File Transfer prompt but select B for batch instead of a protocol when prompted with upload or download options. The upload and download commands that accept batch transfers are D, U, /D, and Z. The maximum number of files you may store in your batch queues is determined by the sysop.

An additional option available with batch file transfers is bidirectional file transfers. Bidirectional file transfers allow you to upload and download files simultaneously in the same amount of time it takes to upload or download in one direction. Using bidirectional file transfers you can optimize your time online by virtually doubling your file transfer speed if you are uploading and downloading. To use bidirectional file transfers your communications software must either support bidirectional protocols directly or allow you to add them as external protocols. Currently the two most popular bidirectional protocols are Bimodem and HS/Link. All modems should support bidirectional file transfers without any problem except for US Robotics HST modems in high speed modes. These modems are designed to send high-speed in one direction and 450bps in the other simultaneously. You will not benefit from bidirectional file transfers with this modem at high-speed. If you have a US Robotics Dual Standard modem make sure you are calling in the v.32 or v.32bis mode to utilize bidirectional file transfers.

[Back to Table of Contents](#)

**Batch/Bidirectional Commands (B)**

-----  
To use Batch/Bidirectional commands hit B from the File Transfer prompt. If you have no files in either your upload or download queues you will not be able to use Batch/Bidirectional commands until you add at least one file to one of your queues. Only protocols supporting batch uploads and downloads will be available from these commands. Xmodem for example will not be an option. The Batch/Bidirectional commands are:

- B Start a bidirectional file transfer
- U Upload files from upload queue
- D Download files from download queue
- L List files in Upload and Download queues
- R Remove a file or files from Upload or Download queues
- C Clear (remove all) files from Upload or Download queues
- Q Quit to File Transfer prompt

[Back to Table of Contents](#)

**File Extraction / Temp Directory**

If implemented on the BBS, file extraction allows you to extract files from an archived file (ZIP, ARC, LZH, etc.) and download only the file or files you need from that archive. This can be extremely helpful if you need only one 50k file from a 1MB ZIP file online.

You can even extract files from nested archives. For example if an archive called FONTMAN.ZIP contained two archives within itself, BLUE.ZIP and RED.ZIP and BLUE.ZIP contained two files called FILE1.FON and FILE2.FON you would be able to download just the file FILE1.FON if that were the only file you needed.

All file extraction is done in your Temp Directory. This is an area you may extract files to and download from. You may work on one original archive at a time in the Temp Directory. The Temp Directory may, however, contain many files extracted from the original archive.

The Temp Directory is also used to create and download a new file scan list or a complete list of all files on the BBS.

[Back to Table of Contents](#)

**Temp Directory Commands (T)**

-----  
Hit T from the File Transfer prompt to access the Temp Directory prompt. From this prompt the following commands are available:

- E Extract files from an archive into the Temp Directory
- N Create a New Files list
- F Create a list of all files on the BBS
- A Add or create a Temp File
- D Download a Temp File
- I Information on files
- V View the contents of any archive located in Temp Directory
- L List all files in the Temp Directory
- X Extract from an archive located in Temp Directory
- R Remove file or files from Temp Directory
- Q Quit to File Transfer prompt



### 1) Extract from archive in Temp Directory

-----  
To extract files from an archive into the Temp Directory hit E from the Temp Directory prompt. Enter the filename you wish to extract from. You may use a wildcard.

[Back to Table of Contents](#)

### 2) Add files to Temp file

-----  
To download from the Temp Directory you must first create a Temp File containing any files you wish to download from the Temp Directory. To create a Temp File or add to a Temp File select A from the Temp Directory prompt. You can view the contents on any archives located in the Temp Directory by selecting V from the Temp Directory prompt. If you wish to extract files from a nested archive located in the Temp Directory select X.

[Back to Table of Contents](#)

### 3) Download Temp file

-----  
When your Temp File contains all the files you wish to download select D from the Temp Directory prompt to download the Temp File.

[Back to Table of Contents](#)

## Creating and downloading file lists

Entering N from the Temp Directory prompt will create a file listing of all new file uploads called NEWFILES.TXT, and place the file in the Temp Directory. To download the file hit A to create a Temp File then D to download the Temp File. The list is a standard ASCII text file and can be read from any standard text editor or wordprocessor. Remember that creating this file in the Temp Directory will delete any files that are already in the directory.

To create and download a complete file list of all files on the BBS hit F from the Temp Directory prompt. This will create a file called FILELIST.TXT. Then hit A to create a Temp File then D to download the Temp File. Again, creating the FILELIST.TXT will erase all existing files in the Temp Directory.

[Back to Table of Contents](#)

---

## Copyright © 2000 by [Rob Swindell](#)

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .

## Synchronet Message Base Specification Version 1.21

### Table of Contents

[Introduction](#)

[Implementation Levels](#)

[Definitions](#)

[Acronyms](#)

[Data Types](#)

[File Formats](#)

[Index](#) (\*.SID)

[Header](#) (\*.SHD)

[Header Allocation](#) (\*.SHA)

[Data](#) (\*.SDT)

[Data Allocation](#) (\*.SDA)

[CRC History](#) (\*.SCH)

[Header Field Types](#)

[Data Field Types](#)

[Message Attributes](#)

[Translation Types](#)

[Agent Types](#)

[Network Types](#)

[Media Types](#)

[Message Storage Pseudo Code](#)

[Message Retrieval Pseudo Code](#)

[SMBUTIL](#)

[CHKSMB](#)

[FIXSMB](#)

[SMBLIB \(C library\)](#)

[Data Types and Constants](#) (SMBDEFS.H)

[Global Variables](#) (SMBVARS.C)

[Function Prototypes](#) (SMBLIB.H)

[Library Functions](#) (SMBLIB.C)

[Miscellaneous](#) (CRC\*. \* and LZH.\*)

[SMBLIB Storage Example](#)

[SMBLIB Retrieval Example](#)

[SMBLIB Performance Issues](#)

[Bibliography](#)

## Introduction

### *Q. What is SMB?*

A. SMB (Synchronet Message Base) is a technical specification for the storage format of electronic mail messages. These e-mail messages may all be contained in one database, or, more commonly, separated into categorized databases. These message databases (or message bases) are also referred to as "sub-boards", "forums", "conferences", and "SIGs". The messages may be directed to an individual person, sent to a group of individuals, or sent to everyone who can read messages in that message base. Messages may be created and read solely at one physical location, or imported from and exported to a message network that may span continents. Message bases that are connected to a message network are often called "echoes".

### *Q. Why SMB?*

A. The Synchronet Message Base is designed to store high volumes of messages while maintaining optimum search, retrieval, and creation performance. These messages are not limited to mere text. In addition to text, SMB defines the storage of digitized sound, MIDI, graphics, fonts, animation, as well as other multimedia data and triggers for localized multimedia. SMB thrives on a multi-user environment where messages are being created, read, modified, and deleted by multiple tasks simultaneously. With the large message networks of today being the rule, rather than the exception, and high volumes of messages being imported on a daily, sometimes hourly basis, creation and deletion speed is of the utmost importance. This is where SMB really shines. Being extensible enough to handle message formats from networks of today and tomorrow, and fast enough to import more messages that humanly readable, the SMB format will more than meet your message storage needs.

### *Q. Why a specification?*

- A. Message bases are often accessed and modified by a number of different programs. Often these programs are developed by individuals or companies other than the original designer of the message base format. This specification is an attempt to aid developers in creating programs that access or modify a message base stored in the SMB format.

***Q. Who can use this specification?***

- A. Anyone that has interest in the Synchronet Message Base format at either an educational or professional level. Specifically, software developers interested or currently involved in the development of message readers, editors, echomail (toss/scan) programs, message transfer agents (MTAs), network gateways, and bulletin board systems. Much of the information in this specification is intended for those with preexisting programming knowledge, so those with little or no programming experience may find it hard to comprehend.

***Q. What does the SMB specification include?***

- A. The text you are reading is part of the SMB specification: a single text document that defines the storage format of each of the six files of an SMB format message base and how they are related to each other.

Included with this specification is C source code to be used as an example to programmers of how to access an SMB format message base and public domain library functions (SMBLIB) that can be compiled and linked into programs that access an SMB format message base developed by third parties. An SMB utility program (SMBUTIL) is also included with C source code as an example of how to use the SMBLIB functions.

***Q. Where did the SMB specification come from?***

- A. Digital Dynamics (southern California based software development company) released "Synchronet Multinode BBS Software Version 1a" in June of 1992 as one of the first BBS packages to be designed from the ground-up to operate in a multi-node environment with incredible speed and reliability, with a large suite of multi-node specific features and design innovations.

The original message base format was designed with localized messaging and low volume message networks in mind. By January of 1993, it was clear that high volume message networks (FidoNet, RelayNet, Usenet, etc.) were the preference of most BBS users and a new message base format was required to allow for high volume message storage, improved storage, retrieval, and maintenance performance, as well as lower storage space requirements.

Rather than introduce another new message format, Digital Dynamics sought to implement an existing public specification for a format that would meet current and future message storage needs. More than a few specifications were seriously considered at one time or another, but after careful examination, design flaws and lack of extensibility eliminated them from the long term plans of Digital Dynamics and Synchronet BBS Software. Thus began the design of the "Synchronet Message Base" (SMB) format.

At the request of many message related program developers, Digital Dynamics created and released the SMB specification before the release of "Synchronet Version 2.00" to allow lead-time on developing support programs for the new format.

Digital Dynamics strongly encourages developers of message related programs (including software that directly competes with Synchronet or other Digital Dynamics products) to implement support for SMB. Though this is a public specification and Digital Dynamics encourages developer suggestions, it will remain under the sole control of Digital Dynamics unless specifically stated otherwise in a future revision of this specification.

Digital Dynamics requests that any organizations that wish to adopt or ratify this specification, in part or whole, notify Digital Dynamics through any of the contact methods listed at the beginning of this document.

***Q. How does SMB store messages?***

- A. Each message base is stored in a set of binary files. This set consists of between three and six files depending the storage method used. The base filename (maximum of eight characters under DOS) is the same for all six files of the same message base and unique among the filenames of other message bases in the same directory. The six files each have a different three character extension. The first character of the extension is always the letter 'S' (for SMB), while the second and third characters define the contents of the file.

Two of the six files associated with each message base are not re-creatable and therefore are the most important when considering data integrity. These two files are the data file (with a .SDT extension) and header file (.SHD extension). Both of these files use 256 byte blocks and have associated block allocation tables (stored in .SDA and .SHA respectively) so that deleted message blocks may be used by new messages without creating odd sized unused 'holes' in the files. The block allocation table files (.SDA

and .SHA) can be recreated with the information stored in the header (.SHD) file. When using Hyper Allocation storage method, the allocation files (.SDA and .SHA) are not used.

For fast indexing, there is a small fixed length index file (with a .SID extension). This file allows for the immediate location of message header records based on sender's name or user number, recipient's name or user number, subject, message number, or message attributes. This file can be recreated with the data stored in the header (.SHD) file.

The last file is an optional CRC history (.SCH) file. It contains 32-bit CRCs of a configurable number of messages imported or created locally. This is to help eliminate duplicate messages created by user or program error. The CRC history file can be recreated with the combination of information stored in the data (.SDT) and header (.SHD) files.

***Q. How fast do messages import into an SMB message base?***

- A. This is a very important question for systems for that import large volumes of messages. Of course, the answer depends on the storage format which you are importing from, the average length of messages, the design of the program which is performing the import process, as well as the hardware and system software being used. What's important is that SMB will allow the fastest import process possible with any given combination of the above factors.

Since system storage capacity is rarely infinite, neither is the number of messages which can be stored in a message base. System operators must define the maximum number of messages to be stored in a message base, the maximum age of the messages in that message base, or a combination of both. When using the Self-packing storage method (defined later in this document), the smaller the number of messages stored in a message base, the faster the import process. The SMB format is flexible enough to support multiple levels of import performance based on optimizations for storage space or speed. Most system operators will almost invariably choose speed over space, but which choices are available is determined by the importing program. This specification defines three storage methods, from slowest to fastest: Self-packing, Fast Allocation, and Hyper Allocation. Other options defined in this specification may affect storage performance, including duplicate message checking and message compression/encryption.

***Q. How much storage is required for an SMB message base?***

- A. The biggest factor in determining storage requirements for a message base is the maximum number of messages to be stored in the base (defined by the system operator) and the average size of each message. The minimum required storage for a message base is 32 bytes plus 532 bytes per message (plus four bytes per message if duplicate message checking is used and three bytes per message if Self-packing or Fast Allocation storage methods are used).

The SMB format was originally designed to be "self-packing", meaning purged (deleted) message header and data blocks will be used automatically by new messages. Relying solely on self-packing, an SMB format message base will never "shrink" in size. This is not to say that it will continually "grow" in size, but that without specific packing procedures, deleted message blocks may remain unused for extended periods of time, meanwhile using some amount of storage space that could be recovered using specific packing procedures. The Fast Allocation and Hyper Allocation storage methods do not use deleted message blocks for new messages so specific packing procedures must be used if any messages are deleted and that storage space is to ever be recovered.

Limiting the maximum age of messages in an SMB message base is another way to control the storage requirements. While maximum message age definition is optional, the definition of the maximum number of messages is not.

***Q. How many messages can be stored per SMB message base?***

- A. Without considering storage limitations or message data lengths greater than 256, the theoretical maximum number of messages that can be stored in a single SMB message base is 16.7 million. Considering the variable length nature of message and header data, it is suggested that the system operator allow no more than 1 million messages per base.

To determine an estimated maximum number of messages for a message base using the average message data length as a factor, use the following formula:

4.2 billion divided by the average message length rounded up to be evenly divisible by 256.

If the average message data length is 1500 bytes, the estimated maximum number of messages would be 2,734,375 (4.2 billion divided by 1536).

Implementations of this format may be further limited by available system memory.

[Back to Table of Contents](#)

## **Implementation Levels**

The SMB format can be implemented to varying degrees between programs without creating compatibility issues. Rather than have developers specifically state which features they have and have not implemented, we have defined seven levels of implementation (represented by Roman numerals I through VII). For a program or software package to meet an implementation level, it must have all of the features listed for that level and all of those for each level below it. The minimum suggested implementation is level I. The SMBUTIL program included with this specification is an example of a level I implementation with features from some of the higher implementation levels.

### **Level I**

The minimum suggested level of implementation. Messages contain merely ASCII text displayable on an ANSI terminal. Messages can be added to the message base and if the maximum number of messages is exceeded, messages are removed or marked for deletion.

### **Level II**

The addition of file attachments, multiple index/header entries per message (multiple destinations), multiple text bodies for the separation of message text and tag/origin lines (for example), forwarding, threading, and specific FidoNet kludge header field support makes this level of implementation more realistic for bulletin board system and EchoMail software implementation.

Synchronet Multinode BBS Software v2.00 has a level II implementation of this specification.

### **Level III**

This implementation adds support for translation strings defined later in this document for data compression, encryption, escaping, and encoding. This level is still limited to basic ASCII text and ANSI escape sequence entry and retrieval.

Synchronet Multinode BBS Software v2.10 has a level III implementation of this specification.

### **Level IV**

The storage and retrieval of embedded and attached images is added in this level of implementation. Supported images are limited to single binary or text data blocks that can be displayed or transferred to the user (automatically, or by request) if their display and translation protocols define specific support for the image type.

### **Level V**

This level of implementation adds support for embedded and attached sound data. This includes digitized sound and MIDI data. Supported sounds are limited to single binary or text data blocks that can be played or transferred to the user (automatically or by request) if their presentation and translation protocols define specific support for the sound type.

### **Level VI**

Localized sound and image data can be triggered by messages stored and retrieved in an implementation of this level.

### **Level VII**

Complete multimedia support is reached in this implementation level with support for embedded and attached animation, sound, and video data.

[Back to Table of Contents](#)

## **Definitions**

### **Control Characters**

When specifying control characters (ASCII 1 through 31), the caret symbol "^" or the abbreviation "ctrl-" followed by a character will be used to indicate the value. ^A is equivalent to ASCII 1, ^B ASCII 2, etc. The case of the control character is not significant (i.e. ^z and ^Z are equivalent). The control character ^@ (ASCII 0) will be specified as NULL or 0.

### **Hexadecimal**

Base sixteen numbering system which includes the digits 0-9 and A-F. Hexadecimal numbers are represented in this document with a prefix of "0x" or

"\x" or a suffix of "h". Hexadecimal letter digits are not case sensitive (i.e. the number 0xff is the same as 0xFF).

**File dump**

When example file dumps are displayed, the format is similar to that of the output from the DOS DEBUG program. With the exception of the ASCII characters, all numbers are in hexadecimal.

| Offset | Byte values                                     | ASCII characters |
|--------|-------------------------------------------------|------------------|
| 000000 | 53 4D 42 1A 10 01 20 00 F4 01 00 00 F4 01 00 00 | SMB... ..        |
| 000010 | 20 00 00 00 D0 07 00 00 D0 07 00 00 00 00 00 00 | .....            |

**Bit values**

Bit (or flag) values are represented in C notation as (1<<x) where x is the bit number. (i.e. bit number 7 (1<<7) is the same as 0x80).

**Word storage**

All words (16-bit) and double words (32-bit) are stored in Intel 80x86 (little endian) format with bytes stored from low to high (reverse of the Motorola 680x0 word storage format).

A 16-bit word with the value 1234h is stored as 34h 12h.

**Translation strings**

Translation strings (xlat variables) are arrays of words (16-bit) in the order of the original storage translation. The last translation type is followed by a 16-bit zero (defined later as XLAT\_NONE). If there are no translations, then the first and only element of the array is XLAT\_NONE.

If multiple translations are used, the translation order must be reversed upon retrieval to obtain the proper data.

**Local e-mail**

When referring to the local e-mail message base of a Synchronet BBS, we are referring specifically to the message base with the name "MAIL" stored in the "DATA" directory (e.g. \SBBS\DATA\MAIL).

Messages stored in this message base are different in the following respects:

- The SMB\_EMAIL status header attribute is set ON
- Hyper Allocation storage method is not supported
- The "To" and "From" fields of the message indexes do NOT contain CRCs

[Back to Table of Contents](#)

**Acronyms:**

|        |                                                                 |
|--------|-----------------------------------------------------------------|
| ANSI   | American National Standards Institute                           |
| ASCII  | American Standard Code for Information Interchange              |
| BBS    | Bulletin Board System                                           |
| C      | The C programming language as defined by ANSI X3.159-1989       |
| CR     | Carriage Return character (ASCII 13)                            |
| CRC    | Cyclic Redundancy Check                                         |
| CRC-16 | Standard 16-bit CRC using 1021h polynomial (seed 0)             |
| CRC-32 | Standard 32-bit CRC using EDB88320h polynomial (seed -1)        |
| CRLF   | Carriage Return character followed by a Line Feed character     |
| FSC    | FidoNet Standards Committee (FTS proposal)                      |
| FTN    | FidoNet Technology Network                                      |
| FTS    | FidoNet Technical Standard                                      |
| LF     | Line Feed character (ASCII 10)                                  |
| QWK    | Compressed message packet format for message reading/networking |
| RFC    | Request for Comments                                            |
| SMB    | Synchronet Message Base                                         |
| UT     | Universal Time (formerly called "Greenwich Mean Time")          |

[Back to Table of Contents](#)

**Data types**

|       |                                                                                               |
|-------|-----------------------------------------------------------------------------------------------|
| uchar | Unsigned 8-bit value (0 through 255).<br>C example:<br><br>#define uchar unsigned char        |
| short | Signed 16-bit value (-32768 through 32767).<br>"short" is a C keyword indicating "short int". |



ushort Unsigned 16-bit value (0 through 65535).  
C example:

```
#define ushort unsigned short
```

ulong Unsigned 32-bit value (0 through 4294967295).  
C example:

```
#define ulong unsigned long
```

time\_t Unsigned 32-bit value.  
Seconds since 00:00 Jan 01 1970 (Unix format).  
Used for all time/date storage in SMB as part of the when\_t data type. This time format will support dates through the year 2105.  
time\_t is defined by ANSI C as a long (signed) which can limit its date support to the year 2038 depending on the library routines used.

ASCII String (aka character array) of 8-bit ASCII characters. Characters with the bit 7 set (80h through FFh) represent the IBM PC extended ASCII character set. When data or header fields of this type are stored in the header, a NULL terminator may or may not be present.  
C example:

```
uchar str[80];
```

ASCIIZ ASCII string with (non-optional) NULL terminator.  
C example:

```
uchar str[81];
```

nulstr ASCII string immediately terminated by NULL.  
C example:

```
uchar *nulstr="";
```

undef Data buffer with undefined contents.  
C example:

```
uchar buf[BUF_LEN];
```

when\_t Date/Time stamp including time-zone adjustment information.  
C example:

```
typedef struct {
    time_t  time;    // Time stamp (in local time)
    short   zone;    // Zone constant or Minutes (+/-) from UT
} when_t;
```

time:

A time value of 0 is invalid and indicates an un-initialized time stamp.

Time stamps are always stored in universal time. i.e. Regardless of what the local time zone is, Jan 1st 1994 00:00 will always be stored as 2D24BD00h.

zone:

If the zone is in the range -720 to +720, it represents the number of minutes east or west of UT. Values in this range should only be used for time zones not otherwise represented here.

If the zone is greater than 720 or less than -720, then the following bits have special meaning:

```
(1<<12)      // Non-US time zone      (east of UT)
(1<<13)      // Non-US time zone      (west of UT)
(1<<14)      // U.S. time zone
(1<<15)      // Daylight savings
```

The lower 12 bits (0 through 11) contain the number of minutes east or west of UT (not accounting for daylight savings).

If the time zone is one specified in the U.S. Uniform Time Act, the following values represent the zone:

```
AST 0x40F0    // Atlantic      (-04:00)
EST 0x412C    // Eastern       (-05:00)
CST 0x4168    // Central       (-06:00)
MST 0x41A4    // Mountain      (-07:00)
PST 0x41E0    // Pacific       (-08:00)
YST 0x421C    // Yukon         (-09:00)
```

```
HST 0x4258      // Hawaii/Alaska      (-10:00)
BST 0x4294      // Bering              (-11:00)
```

With bit 15 set, the following values represent the same zone with the presence of daylight savings:

```
ADT 0xC0F0      // Atlantic          (-03:00)
EDT 0xC12C      // Eastern            (-04:00)
CDT 0xC168      // Central            (-05:00)
MDT 0xC1A4      // Mountain          (-06:00)
PDT 0xC1E0      // Pacific            (-07:00)
YDT 0xC21C      // Yukon              (-08:00)
HDT 0xC258      // Hawaii/Alaska      (-09:00)
BDT 0xC294      // Bering            (-10:00)
```

The following non-standard time zone specifications may also be used:

```
MID 0x2294      // Midway              (-11:00)
VAN 0x21E0      // Vancouver          (-08:00)
EDM 0x21A4      // Edmonton           (-07:00)
WIN 0x2168      // Winnipeg           (-06:00)
BOG 0x212C      // Bogota              (-05:00)
CAR 0x20F0      // Caracas             (-04:00)
RIO 0x20B4      // Rio de Janeiro      (-03:00)
FER 0x2078      // Fernando de Noronha (-02:00)
AZO 0x203C      // Azores              (-01:00)
LON 0x1000      // London              (+00:00)
BER 0x103C      // Berlin              (+01:00)
ATH 0x1078      // Athens              (+02:00)
MOS 0x10B4      // Moscow              (+03:00)
DUB 0x10F0      // Dubai               (+04:00)
KAB 0x110E      // Kabul               (+04:30)
KAR 0x112C      // Karachi             (+05:00)
BOM 0x114A      // Bombay              (+05:30)
KAT 0x1159      // Kathmandu           (+05:45)
DHA 0x1168      // Dhaka               (+06:00)
BAN 0x11A4      // Bangkok             (+07:00)
HON 0x11E0      // Hong Kong           (+08:00)
TOK 0x121C      // Tokyo               (+09:00)
SYD 0x1258      // Sydney              (+10:00)
NOU 0x1294      // Noumea              (+11:00)
WEL 0x12D0      // Wellington          (+12:00)
```

`fidoaddr_t` FidoNet address stored as four ushorts that represent the zone, network, node, and point (in that order).  
C example:

```
typedef struct {
    ushort zone,
           net,
           node,
           point;
} fidoaddr_t;
```

`typestr_t` ASCIIZ string with ushort type prefix.  
C example:

```
typedef struct {
    ushort type;    // Specifier for type of 'str'
    uchar  str[];   // ASCIIZ filename or other string data
} typestr_t;
```

`mattach_t` File attachment information with type prefix, translation string, and filename.  
C example:

```
typedef struct {
    ushort type;    // Attachment type
    ushort xlat[];  // Translations of data in attachment
    uchar  str[];   // ASCIIZ filename
} mattach_t;
```

`vattach_t` Video file attachment information with type, compression, translation string, and filename.  
C example:

```
typedef struct {
    ushort type;    // Attachment type
    ushort comp;    // Compression method
    ushort xlat[];  // Translations of data in attachment
```

```
        uchar    str[]; // ASCIIIZ filename

        } vattach_t;

mtext_t      Message text with translation string prefix.
              C example:

              typedef struct {

                ushort  xlat[]; // Translations of text
                uchar    text[]; // Actual text data

                } mtext_t;

ftext_t      Formatted message text with translation string prefix and
              format type.
              C example:

              typedef struct {

                ushort  type;    // See Image Types for valid types
                ushort  xlat[]; // Translations of data
                uchar    data[]; // Actual formatted text data

                } ftext_t;

membed_t     Embedded data with type prefix, translation string, and ASCIIIZ
              description.
              C example:

              typedef struct {

                ushort  type;    // Specifier for type of 'dat'
                ushort  xlat[]; // Translations of embedded data
                uchar    name[]; // ASCIIIZ char description of embedded data
                uchar    dat[];  // Binary data

                } membed_t;

vembed_t     Embedded video data with type, compression method, translation
              string, and ASCIIIZ description.
              C example:

              typedef struct {

                ushort  type;    // Specifier for type of 'dat'
                ushort  comp;    // Compression method
                ushort  xlat[]; // Translations of embedded data
                uchar    name[]; // ASCIIIZ char description of embedded data
                uchar    dat[];  // Binary data

                } vembed_t;
```

[Back to Table of Contents](#)

## File Formats

### Index File (\*.SID)

The index file for each message base contains one record per message in the base. Each record is fixed length using the following format:

#### Index Record:

*C example:*

```
typedef struct {

    ushort  to;        // 16-bit CRC of recipient name (lower case) or user number
    ushort  from;      // 16-bit CRC of sender name (lower case) or user number
    ushort  subj;      // 16-bit CRC of title/subject (lower case)
    ushort  attr;      // attributes (MSG_PRIVATE, MSG_READ, etc. flags)
    ulong   offset;    // byte offset of message header in header file
    ulong   number;    // message serial number (1 based)
    time_t  time;      // import date/time stamp (Unix format)

} idxrec_t;
```

**Example file dump (16 messages starting with message number 15):**

```
000000 36 4F 13 07 2A 77 00 00 20 00 00 00 0F 00 00 00
000010 BE 62 76 2C 36 4F 46 0A 7F B2 00 00 20 01 00 00
000020 10 00 00 00 C7 29 78 2C 36 4F 70 6F 46 FF 00 00
000030 20 02 00 00 11 00 00 00 AD D3 7A 2C 70 6F 13 07
000040 46 FF 00 00 20 03 00 00 12 00 00 00 D6 F8 7F 2C
000050 36 4F E1 EA E7 E9 00 00 20 04 00 00 13 00 00 00
```

```
000060 1E 7B 85 2C 37 0D 2E DF 4D 79 00 00 20 05 00 00
000070 14 00 00 00 5C E1 A1 2C 90 54 2D 5A 86 62 00 00
000080 20 06 00 00 15 00 00 00 39 2E A2 2C 70 6F 1A 8B
000090 46 FF 00 00 20 07 00 00 16 00 00 00 D0 7B A8 2C
0000A0 2E DF 1A 8B 4D 79 00 00 20 08 00 00 17 00 00 00
0000B0 FF 7B A8 2C B4 D9 35 7C 23 B1 00 00 20 09 00 00
0000C0 18 00 00 00 CE D4 BA 2C 36 4F BC D8 B2 E7 00 00
0000D0 20 0A 00 00 19 00 00 00 14 5F C3 2C BA A8 4E B0
0000E0 67 76 00 00 20 0B 00 00 1A 00 00 00 6F 89 C3 2C
0000F0 36 4F 0C 01 19 9C 00 00 20 0C 00 00 1B 00 00 00
000100 F8 30 C6 2C 36 4F FA 48 0E 55 00 00 20 0D 00 00
000110 1C 00 00 00 6A 94 D3 2C 36 4F F1 CE CF A2 00 00
000120 20 0E 00 00 1D 00 00 00 53 DB D5 2C 8D A6 21 CE
000130 F7 AB 00 00 20 0F 00 00 1E 00 00 00 31 29 DC 2C
```

**Field descriptions:**

**To:**

The 'To' field is the CRC-16 of the name of the intended recipient agent of this message or the intended recipient's user number. If the CRC is stored, the text must be converted to lower case (A-Z changed to a-z) before the CRC is calculated. If the message is forwarded to another agent, the original or new index record must be changed to contain the CRC-16 of the new recipient name or user number. This field must always contain the recipient user number for local e-mail on a Synchronet BBS. Outbound netmail stored in the Synchronet local e-mail message base will contain 0 in this field.

**From:**

This field, similar to the 'To' field, contains the CRC-16 of the name of the sending agent of this message or the sender's user number. If the CRC is stored, the text must be converted to lower case (A-Z changed to a-z) before the CRC is calculated. If the message is forwarded to another agent, the original or new index record must be changed to contain the CRC-16 of the new sender name or user number. If the message was imported into the local e-mail message base on a Synchronet BBS via netmail, this field will contain 0.

**Subj:**

The 'Subj' field contains the CRC-16 of the message's subject. The subject must be converted to lower case (A-Z changed to a-z) and all preceding "re: "'s and "re:"'s removed before calculating the CRC-16.

**Attr:**

This ushort is a bit field of the specific attributes for this message. It is a clone of the 'attr' element of the msghdr\_t structure.

**Offset:**

This ulong is the offset (in bytes) in the header file for this message's header record.

**Number:**

This ulong is the serial number of this message. Valid values are 1 through 0xffffffff. No two index records in the same message base may have the same message number. All index records must have sequential, but not necessarily consecutive, message numbers.

**Time:**

This field is the date/time stamp the message was imported to or posted in the message base. It is a clone of the 'when\_imported.time' element of the msghdr\_t structure.

**[Back to Table of Contents](#)**

**Header File (\*.SHD)**

Each SMB header file is made up of two distinct sections: base header records and message header records (usually the bulk of the file).

**Base Header Records:**

Base header records are blocks of data that apply to the entire message base and are of variable length. This specification defines only one base header record, the "Status info" (smbstatus\_t) record. This status info record must be the first base header record in the file (offset 0) and must be modified if additional base header records are added.

Additional header records allow other developers to store configuration and status information particular to their application needs. It also allows for future header record definitions as part of this specification without causing backward compatibility issues.

Each base header record contains a fixed length portion (smbhdr\_t) and an optional variable length portion.

Whenever a base header record is read or updated (written), it must first be successfully locked and subsequently unlocked (using the file system record locking facilities).

The first base header record (Status Info) is used as a semaphore when writing to the message index (.SID) file and, when using the Hyper Allocation storage method, writing to the message data (.SDT) file. This record must be successfully locked before writing and subsequently unlocked. This is to insure that multiple applications simultaneously writing to the same message base does not result in corrupted data.

#### **Message Header Records:**

Following the last base header record is the first message header record. Each header record is stored in one or more 256 byte blocks. There must be exactly one active message header record for every index record in the index file. (Note: This does not include deleted message headers that have not been overwritten by a new message header).

Each message header record contains a fixed length portion (msghdr\_t), a list of zero or more fixed length data fields (dfield\_t), and a list of three or more variable length header fields (hfield\_t).

The value of the data stored in the zero or more unused bytes of the last header record block have an undefined value, though whenever possible developers should initialize to binary zero for human readability.

Whenever a message header record is read or updated (written), it must first be successfully locked and subsequently unlocked.

#### **Base Header Record (Fixed Portion):**

*C example:*

```
typedef struct {
    uchar   id[4];           // text or binary unique hdr ID
    ushort  version;         // version number (initially 100h for 1.00)
    ushort  length;          // length including this struct

    } smbhdr_t;
```

#### **Base Header Record Field Descriptions:**

##### **Id:**

This is a four byte unique ID identifying the type of the base header record. The bytes may contain any value, but printable ASCII characters are preferred. The only ID defined in this specification is "SMB^Z" used by the Status Info base header record.

##### **Version:**

This is a version number of the base header record type. Base header records of different versions may have different formats or contain different information. This is to aid the application in determining if the record is pertinent and if so, to what degree. The Status Info base header record uses this version field to define the version of the format for the entire message base (currently 0x121 for version 1.21).

##### **Length:**

This is entire length in bytes of this header record (including both fixed and variable portions).

#### **Base Header #1 (Status info) Record (Variable Portion):**

*C example:*

```
typedef struct {
    ulong   last_msg;         // last message number posted or imported
    ulong   total_msgs;       // total messages currently in message base
    ulong   header_offset;    // byte offset to first header record
    ulong   max_crcs;         // Maximum number of CRCs to keep in history
    ulong   max_msgs;         // Maximum number of messages to keep in base
    ushort  max_age;          // Maximum age of messages (days) to keep in base
    ushort  attr;             // Attribute bits

    } smbstatus_t;
```

#### **Base Header #1 (Status Info) Record (Variable Portion) Field Descriptions:**

##### **Last\_msg:**

This is the serial number of the last message imported or posted into this message base. The index, header, and data records for this message may possibly not exist (due to deletion). This field is used for determining the message number to give to a new message being imported or posted into this message base. This field must be updated for every message added to the message base.

##### **Total\_msgs:**

This is the total number of active messages currently in the message base. This number should match the number of records in the index (.SID) file

and active header records in the header (.SHD) file. This field must be updated whenever a message is added to or removed from the message base.

**Header\_offset:**

This is the byte offset to the first message header record. It is useful for skipping all the base header records and going directly to the first message header record.

**Max\_crcs:**

This is the maximum number of message CRCs to store in the CRC history (.SCH) file for duplicate message checking. If this field contains 0, then duplicate message checking is disabled.

**Max\_msgs:**

This is the preferred maximum number of messages to keep in this message base as specified by the system operator. It is used by maintenance programs that trim the message base down by removing old messages. This field should be ignored by applications importing or posting messages allowing them to exceed this maximum at will.

**Max\_age:**

This field is the maximum age (in days) of messages to keep in the message base. It is used by maintenance programs to purge out-dated messages from the message base.

**Attr:**

This is a bit field containing specific attributes (or flags) that may define the way messages are stored or retrieved from the this message base. The following attributes are defined:

**SMB\_EMAIL (1<<0)**

Indicates the message base is specifically for messages to or from local users. When this bit is set, the idxrec.to and idxrec.from fields will contain the user numbers (or 0 for non-user destination/source) instead of the CRC-16 of the agent name.

**SMB\_HYPERALLOC (1<<1)**

Indicates the message base uses the Hyper Allocation storage method. This bit should not be cleared by an application without first deleting all the messages in the message base. This is due to the fact the Hyper Allocation is not downward compatible with the Self-packing and Fast Allocation storage methods.

When used with Synchronet BBS software, a message base must NOT have both of the above attributes set. The only message base that should have the SMB\_EMAIL attribute set is the DATA\MAIL message base.

**Base Header #1 (Status info) Record Contents:**

```
smbhdr.id="SMB\x1a";           // SMB^Z
smbhdr.version=0x121;          // v1.21
smbhdr.length=sizeof(smbhdr_t)+sizeof(smbstatus_t); smbstatus_t status;
```

**Additional Base Headers:**

Additional headers from developers must have initial 8 bytes in smbhdr\_t format, length must include size of smbhdr\_t, and header\_offset of smbstatus\_t must be changed to include the size of the additional header(s).

**Example file dump (base header portion only):**

```
000000 53 4D 42 1A 20 01 20 00 F4 01 00 00 F4 01 00 00 SMB.....
000010 20 00 00 00 D0 07 00 00 D0 07 00 00 00 00 00 00 .....
```

**Message Header Record (Fixed portion):**

*C example:*

```
typedef struct {
    uchar   id[4];           // SHD^Z (same for all types and versions)
    ushort  type;            // Message type (this is the definition of type 0)
    ushort  version;         // Version of type (initially 100h for 1.00)
    ushort  length;          // Total length of fixed portion + all fields
    ushort  attr;            // Attributes (bit field) (duplicated in SID)
    ulong   auxattr;         // Auxiliary attributes (bit field)
    ulong   netattr;         // Network attributes (bit field)
    when_t  when_written;    // Date/Time message was originally created
    when_t  when_imported;   // Date/Time message was imported (locally)
    ulong   number;          // Message number (unique, not necessarily seq.)
    ulong   thread_orig;     // Original message number in thread
    ulong   thread_next;     // Next message in thread
    ulong   thread_first;    // Number of first reply to this message
    uchar   reserved[16];    // 16 reserved bytes for future use
    ulong   offset;          // Offset for buffer into data file (0 or mod 256)
```



```

    ushort  total_dfields;  // Total number of data fields

    } msghdr_t;

typedef struct {

    ushort  type;           // See "Data Field Types" values
    ulong   offset;         // Offset into buffer
    ulong   length;         // Length of data field in buffer

    } dfield_t;

typedef struct {

    ushort  type;           // See "Header Field Types" for values
    ushort  length;         // Length of buffer
    uchar   dat[length];

    } hfield_t;
```

**Example file dump (one header record, both fixed and variable length portions):**

```

000020 53 48 44 1A 00 00 20 01 F5 00 00 00 00 00 00 SHD... ..
000030 00 00 00 00 00 46 DB F7 2C 00 00 7D D7 29 2D 00 00 .....,...}x)-..
000040 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000060 00 00 00 00 00 02 00 00 00 00 00 00 00 4A 01 00 00 .....J...
000070 02 00 4A 01 00 00 53 00 00 00 00 00 13 00 4D 61 ..J...S.....Ma
000080 72 69 61 6E 6E 65 20 4D 6F 6E 74 67 6F 6D 65 72 rianne Montgomer
000090 79 30 00 0C 00 43 61 72 6F 6C 20 47 61 69 73 65 y0...Carol Gaise
0000A0 72 60 00 07 00 46 61 72 6E 68 61 6D A4 00 14 00 r`...Farnham....
0000B0 31 3A 31 33 38 2F 31 30 32 2E 30 20 32 63 66 38 1:138/102.0 2cf8
0000C0 30 35 37 36 A5 00 14 00 31 3A 33 34 33 2F 31 30 0576....1:343/10
0000D0 30 2E 30 20 32 63 66 33 62 39 30 61 A3 00 23 00 0.0 2cf3b90a..#.
0000E0 31 33 38 2F 31 30 32 20 31 20 32 37 30 2F 31 30 138/102 1 270/10
0000F0 31 20 32 30 39 2F 32 30 39 20 31 30 33 2F 30 20 1 209/209 103/0
000100 33 35 35 02 00 02 00 02 00 03 00 08 00 01 00 8A 355.....
000110 00 66 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .f.....
```

**Contents of example header:**

```

id                SHD^Z
type              0000h
version           0120h
length            245
attr              0000h
auxattr           00000000h
netattr           00000000h
when_written      Sat Nov 27 17:57:10 1993
when_imported     Tue Jan 04 15:54:21 1994
number            1
thread_orig       0
thread_next       0
thread_first      0
reserved[16]
offset            0
total_dfields     2

dfield[0].type     00h
dfield[0].offset   0
dfield[0].length   330
dfield[1].type     02h
dfield[1].offset   330
dfield[1].length   83

hfield[0].type     00h
hfield[0].length   19
hfield[0]_dat      Marianne Montgomery
hfield[1].type     30h
hfield[1].length   12
hfield[1]_dat      Carol Gaiser
hfield[2].type     60h
hfield[2].length   7
hfield[2]_dat      Farnham
hfield[3].type     A4h
hfield[3].length   20
hfield[3]_dat      1:138/102.0 2cf80576
hfield[4].type     A5h
hfield[4].length   20
hfield[4]_dat      1:343/100.0 2cf3b90a
hfield[5].type     A3h
hfield[5].length   35
hfield[5]_dat      138/102 1 270/101 209/209 103/0 355
hfield[6].type     02h
hfield[6].length   2
hfield[6]_dat      02 00
hfield[7].type     03h
```

```
hfield[7].length      8
hfield[7]_dat         01 00 8A 00 66 00 00 00
```

### **Fixed Portion Field descriptions:**

#### **Id:**

This field (regardless of the header type or version) must always contain the the string "SHD^Z". This is to aid in the restoration of a corrupted header file and give a visual indication of the beginning of a new header record when viewing dumps of the header file.

#### **Type:**

This is the message header type. Only one type is currently defined by this specification (type 0). Any and all future header types will have the first 4 fields (10 bytes) in the same format of type 0. This allows other types (with different lengths) to be skipped because the 4th field (length) will always be in the same position.

#### **Version:**

This is the version of this header type. This specification defines version 1.21 of message header type 0 (stored as 121h).

#### **Length:**

This is the total length of this message header record (including both fixed and variable length portions, but NOT including unused block space).

#### **Attr:**

This is a bit field (16-bit) containing basic message attributes (flags) for this message. An exact duplicate of this field is stored in the index file as well. They must always match.

#### **Auxattr:**

This is a bit field (32-bit) containing the auxiliary attributes (flags) for this message. The attributes stored in this variable are more specific in nature and less critical than those in the Attr field.

#### **Netattr:**

This is a bit field (32-bit) containing the network attributes (flags) for this message. The attributes stored in this variable are related solely to message networking.

#### **When\_written:**

This is the date and time when the message was originally created.

#### **When\_imported:**

This is the date and time when the message was posted on or imported into the local message system.

#### **Number:**

This is the message's unique serial number (from 1 to FFFFFFFFh). This field is duplicated in the index file. They must always match.

#### **Thread\_orig:**

If this message is a reply, then this field contains the number of the original message that was replied to. If this message was not a reply, this field will contain the value 0.

#### **Thread\_next:**

If this message is a reply, and there are later replies to that message (the message number contained in the Thread\_orig field), then this field will contain the number of the next reply in the chain. If this message is the only reply to the original message, this field will contain the value 0.

#### **Thread\_first:**

If there are any replies to this message (after it has been posted), this field will contain the number of the first reply to this message. If there are no replies to this message, this field will contain the value 0.

#### **Reserved:**

Unused bytes, reserved for future definition in the message header type 0 specification.

#### **Offset:**

The byte offset into the data file, specifying the start of the buffer for all data associated with this message. This value must be either 0 or modulus 256. When retrieving the actual data portion of data fields, the physical offset into the file will be the offset of the message data buffer (this field) plus the offset of the individual data field (msghdr\_t.offset+dfield\_t.offset).

#### **Total\_dfields:**

This field contains the total number of data fields associated with this message. The value of this field must match the actual number of data fields stored in the header (dfield\_t data types following the fixed portion of the message header).

### **Variable Portion Field descriptions:**

See the Header Field Type and Data Field Type sections for the descriptions

of the values contained in these fields.

[Back to Table of Contents](#)

## **Message Header Block Allocation (\*.SHA)**

If this message base uses the Hyper Allocation storage method (the SMB\_HYPERALLOC bit is set in the smbstatus\_t.attr field), then this file is not created or used.

This file contains no header or signature data. Each byte (uchar) in the file specifies the allocation state of the corresponding 256 byte block in the header (\*.SHD) file. A value of 0 indicates a free header block, and a value of 1 indicates an allocated block. Other non-zero values are undefined.

This file must always be opened DENY ALL (non-shareable).

[Back to Table of Contents](#)

## **Message Data (\*.SDT)**

This file contains no header or signature data. It contains the text and other embedded data for the messages in a single message base. The data for each message always begins on a 256 byte block boundary. The data in the unused portion of a data block is undefined, but should be initialized to NULL whenever possible.

This file must always be opened DENY NONE (shareable).

Data fields of type TEXT\_BODY and TEXT\_TAIL must have all trailing white space and control characters removed (i.e. the last character of the data record must be in the range 21h to FFh). The only exception to this rule, is if the TEXT\_BODY is terminated with multiple contiguous CRLFs, only the last CRLF should be removed. A CRLF should always be appended to the text data when it is displayed.

When reading from this file, it is a good idea to make sure the message header for the data being read is currently locked (though no single message header should be locked for extended durations of time). This will insure that no other application will write to this portion of the file while it's being read (read from disk, not displayed).

When using the Hyper Allocation storage method, the Status Info message base header must be successfully locked before writing to this file and subsequently unlocked.

[Back to Table of Contents](#)

## **Message Data Block Allocation (\*.SDA)**

If this message base uses the Hyper Allocation storage method (the SMB\_HYPERALLOC bit is set in the smbstatus\_t.attr field), then this file is not created or used.

This file contains no header or signature data. Each word (ushort) in the file specifies the allocation state of the corresponding 256 byte block in the data (\*.SDT) file. A value of 0 indicates a free block, and a non-zero value indicates the number of message header records associated with this message data (most often 1). Each block can be used by up to 65,535 header records.

This file must always be opened DENY ALL (non-shareable).

[Back to Table of Contents](#)

## **CRC history for duplicate message checking (\*.SCH)**

This file is optional and contains no header or signature data. Each long word (ulong) in the file contains a CRC-32 of previously posted/imported messages. These CRCs can be used to check a candidate message for posting/import to be sure the message isn't a duplicate created by human or program error. The maximum number of CRCs to store is defined in the first message base header record (smbstatus\_t.max\_crcs).

The CRC is calculated on the first TEXT\_BODY data field before any translations are applied (e.g. encoding, compression, encryption).

This file must always be opened DENY ALL (non-shareable).

[Back to Table of Contents](#)

## **Header Field Types:**

These are the defined valid values for hfield\_t.type:

Name : SENDER  
Value : 00h  
Data : ASCII  
Multiple : Yes, order significant  
Required : Yes  
Summary : Name of agent that sent this message

If blank (0 length or nulstr), assumed "Anonymous". If multiple SENDER fields exist, then the message has been forwarded and the order of the fields in the record must match the forwarding order (chronologically). When forwarding a message, the original SENDER field should be left intact and new SENDER, FORWARDED, and RECIPIENT fields added to the end of the record.

Name : SENDERAGENT  
Value : 01h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : AGENT\_PERSON or previous SENDERAGENT if exists  
Summary : Type of agent that sent this message

If multiple SENDER fields exist, then the message has been forwarded. If any of the forwarding agents is of a type other than AGENT\_PERSON, then this field must follow that SENDER field to specify the agent type.

Name : SENDERNETTYPE  
Value : 02h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : NET\_NONE or previous SENDERNETTYPE if exists  
Summary : Type of network message was sent from

If multiple SENDERNETADDR fields are included, a SENDERNETTYPE field should be included before each to determine what data type the address is stored in.

Name : SENDERNETADDR  
Value : 03h  
Data : undef  
Multiple : Yes, order significant  
Required : No  
Default : Previous SENDERNETADDR if exists  
Summary : Network address for agent that sent this message

The SENDERNETTYPE field indicates the data type of this field. If the SENDERNETTYPE is of type NET\_INTERNET, the local-part of the Internet address is optional. If the local-part separator character ('@') is omitted, the SENDER field is assumed to be the local-part of the address.

Name : SENDEREXT  
Value : 04h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : Previous SENDEREXT if exists  
Summary : Extension of sending agent

This field is useful for storing the sending agent's extension, when the agent's extension binds more tightly than the agent's name.

For example, Synchronet Multinode BBS Software stores local e-mail with the sending and receiving agent's user numbers stored as their respective extensions. This is done so that if a user name changes for some reason, messages will not "disappear" from the user's mail box.

If the SMB\_EMAIL status header attribute is set, then the "From" field in the index must contain the binary value of this field rather than the CRC-16 of the SENDER (name) field.

Name : SENDERPOS  
Value : 05h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : Previous SENDERPOS if exists  
Summary : Position of sending agent

Primarily for documentary purposes, this field contains the position of the sending agent (i.e. President, Sysop, C.E.O., MIS Director, etc).

It can also be useful for getting a message or reply to the intended recipient when the agent name is not located or is unknown, but the position of the agent is known and specified.

Name : SENDERORG  
Value : 06h

Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : Previous SENDERORG if exists  
Summary : Organization name of sending agent

Primarily for documentary purposes, this field contains the organization to which the sending agent belongs (i.e. Microsoft, Joe's BBS, SoCal User's Group, etc).

Name : AUTHOR  
Value : 10h  
Data : ASCII  
Multiple : Yes  
Required : No  
Default : First SENDER  
Summary : Name of agent that created this message

This field can only be added by the process that originally creates the message. It should not be included if same as first SENDER field. If multiple AUTHOR fields exist, then the message was created by multiple agents and is considered valid. The order of multiple AUTHOR fields in the record is not significant.

Name : AUTHORAGENT  
Value : 11h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : SENDERAGENT or previous AUTHORAGENT if exists  
Summary : Type of agent that created this message

This field can only be added by the process that originally creates the message. It should not be included if same as first SENDERAGENT field. If multiple AUTHOR fields exist, then the message was created by multiple agents and if the agent type for any of the authors is other than AGENT\_PERSON, an AUTHORAGENT field must follow to specify the agent type.

Name : AUTHORNETTYPE  
Value : 12h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : SENDERNETTYPE or previous AUTHORNETTYPE if exists  
Summary : Type of network this author is member of

Name : AUTHORNETADDR  
Value : 13h  
Data : undef  
Multiple : Yes, order significant  
Required : No  
Default : SENDERNETADDR or previous AUTHORNETADDR if exists  
Summary : Network address of this author

Name : AUTHOREXT  
Value : 14h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : SENDEREXT or previous AUTHOREXT if exists  
Summary : Extension of this author

Name : AUTHORPOS  
Value : 15h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : SENDERPOS or previous AUTHORPOS if exists  
Summary : Position of this author

Name : AUTHORORG  
Value : 16h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : SENDERORG or previous AUTHORORG if exists  
Summary : Organization this author belongs to

Name : REPLYTO  
Value : 20h  
Data : ASCII  
Multiple : Yes, but only last is valid  
Required : No  
Default : SENDER  
Summary : Name of agent that replies should go to

Name : REPLYTOAGENT  
Value : 21h  
Data : ushort  
Multiple : Yes, but only last is valid  
Required : No  
Default : SENDERAGENT  
Summary : Type of agent that replies should go to

Name : REPLYTONETTYPE  
Value : 22h  
Data : ushort  
Multiple : Yes, but only last is valid  
Required : No  
Default : SENDERNETTYPE  
Summary : Type of network that replies should go to

Name : REPLYTONETADDR  
Value : 23h  
Data : undef  
Multiple : Yes, but only last is valid  
Required : No  
Default : SENDERNETADDR  
Summary : Network address that replies should go to

Name : REPLYTOEXT  
Value : 24h  
Data : ASCII  
Multiple : Yes, but only last is valid  
Required : No  
Default : SENDEREXT  
Summary : Extension of agent that replies should go to

Name : REPLYTOPOS  
Value : 25h  
Data : ASCII  
Multiple : Yes, but only last is valid  
Required : No  
Default : SENDERPOS  
Summary : Position of agent that replies should go to

Name : REPLYTOORG  
Value : 26h  
Data : ASCII  
Multiple : Yes, but only last is valid  
Required : No  
Default : SENDERORG  
Summary : Organization of agent that replies should go to

Name : RECIPIENT  
Value : 30h  
Data : ASCII  
Multiple : Yes, order significant  
Required : Yes  
Default : "All"  
Summary : Name of agent to receive this message

If multiple RECIPIENT fields exist, the message has been forwarded and for each additional RECIPIENT field (after the initial RECIPIENT), there should be a FORWARDED field. The order of the RECIPIENT fields in the record must match the order in which the message was sent and forwarded (chronologically).

Name : RECIPIENTAGENT  
Value : 31h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : AGENT\_PERSON or previous RECIPIENTAGENT if exists  
Summary : Type of agent to receive this message

If multiple RECIPIENT fields exist, the message has been forwarded. If any of the recipient agents are of a type other than AGENT\_PERSON, this field must follow the RECIPIENT field to specify the agent type.

Name : RECIPIENTNETTYPE  
Value : 32h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : NET\_NONE or previous RECIPIENTNETTYPE if exists  
Summary : Type of network to receive this message

Name : RECIPIENTNETADDR  
Value : 33h  
Data : undef  
Multiple : Yes, order significant  
Required : No



Default : Previous RECIPIENTNETADDR if exists  
Summary : Address of network to receive this message

Name : RECIPIENTEXT  
Value : 34h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : Previous RECIPIENTTEXT if exists  
Summary : Extension of agent to receive this message

If SMB\_EMAIL status header attribute is set, then the "To" field in the index must contain the binary value of this field rather than the CRC-16 of the RECIPIENT (name) field. This is the case specifically with the local e-mail message base on a Synchronet BBS.

Name : RECIPIENTPOS  
Value : 35h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : Previous RECIPIENTPOS if exists  
Summary : Position of agent to receive this message

Name : RECIPIENTORG  
Value : 36h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : Previous RECIPIENTORG if exists  
Summary : Type of agent to receive this message

Name : FORWARDTO  
Value : 40h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Summary : Name of agent this message is to be forwarded to

Name : FORWARDTOAGENT  
Value : 41h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTAGENT or previous FORWARDTOAGENT if exists  
Summary : Type of agent this message is to be forwarded to

Name : FORWARDTONETTYPE  
Value : 42h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTNETTYPE or previous FORWARDTONETTYPE if exists  
Summary : Type of network this message is to be forwarded to

Name : FORWARDTONETADDR  
Value : 43h  
Data : undef  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTNETADDR or previous FORWARDTONETADDR if exists  
Summary : Network address this message is to be forwarded to

Name : FORWARDTOEXT  
Value : 44h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTTEXT or previous FORWARDTOEXT if exists  
Summary : Extension of agent this message is to be forwarded to

Name : FORWARDTOPOS  
Value : 45h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTPOS or previous FORWARDTOPOS if exists  
Summary : Position of agent this message is to be forwarded to

Name : FORWARDTOORG  
Value : 46h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTORG or previous FORWARDTOORG if exists  
Summary : Organization of agent this message is to be forwarded to

Name : FORWARDED  
Value : 48h  
Data : when\_t  
Multiple : Yes, order significant  
Required : Yes, if forwarded  
Summary : Date/Time this message was forwarded to another agent

Name : RECEIVEDBY  
Value : 50h  
Data : ASCII  
Multiple : Yes, order significant  
Required : Yes, if receiving agent is other than RECIPIENT  
Summary : Name of agent that received this message

Name : RECEIVEDBYAGENT  
Value : 51h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTAGENT or previous RECEIVEDBYAGENT if exists  
Summary : Type of agent that received this message

Name : RECEIVEDBYNETTYPE  
Value : 52h  
Data : ushort  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTNETTYPE or previous RECEIVEDBYNETTYPE if exists  
Summary : Type of network that received this message

Name : RECEIVEDBYNETADDR  
Value : 53h  
Data : undef  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTNETADDR or previous RECEIVEDBYNETADDR if exists  
Summary : Network address that received this message

Name : RECEIVEDBYEXT  
Value : 54h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTEXT or previous RECEIVEDBYEXT if exists  
Summary : Extension of agent that received this message

Name : RECEIVEDBYPOS  
Value : 55h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTPOS or previous RECEIVEDBYPOS if exists  
Summary : Position of agent that received this message

Name : RECEIVEDBYORG  
Value : 56h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Default : RECIPIENTORG or previous RECEIVEDBYORG if exists  
Summary : Organization of agent that received this message

Name : RECEIVED  
Value : 58h  
Data : when\_t  
Multiple : Yes, order significant  
Required : Yes, if received  
Default : NULL  
Summary : Date/Time this message was received

Name : SUBJECT  
Value : 60h  
Data : ASCII  
Multiple : No  
Required : Yes, but may be blank (0 length or nulstr)  
Summary : Subject/title of message

Name : SUMMARY  
Value : 61h  
Data : ASCII  
Multiple : No  
Required : No  
Summary : Summary of message contents, created by AUTHOR

Name : COMMENT  
Value : 62h

Data : ASCII  
Multiple : Yes  
Required : No  
Summary : Comment about this message, created by SENDER

This field is useful for adding notes to a message when forwarding to a new recipient.

Name : CARBONCOPY  
Value : 63h  
Data : ASCII  
Multiple : Yes  
Required : No  
Summary : List of agents this message was also sent to

This field is optional and only for the use of notifying the recipient of who else received the message.

Name : GROUP  
Value : 64h  
Data : ASCII  
Multiple : Yes  
Required : No  
Summary : Name of group of users to receive message on recipient system

This field is used when sending to a group name across a network, where the group can be expanded into multiple header records for each agent on the destination system.

Name : EXPIRATION  
Value : 65h  
Data : when\_t  
Multiple : No  
Required : No  
Summary : Date/Time that this message will expire

Name : PRIORITY  
Value : 66h  
Data : ulong  
Multiple : No  
Required : No  
Default : 0  
Summary : Message priority (0 is lowest, FFFFFFFFh is highest)

Name : FILEATTACH  
Value : 70h  
Data : ASCII  
Multiple : Yes  
Required : No  
Summary : Name/file specification of attached file(s)

Name of attached file(s). Wildcards allowed. MSG\_FILEATTACH attribute must be set. If the MSG\_FILEATTACH attribute is set but this field is not included, the SUBJECT field is assumed to be the filename(s).

Name : DESTFILE  
Value : 71h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Summary : Destination name for attached file(s)

Wildcards allowed. FILEATTACH field must also be included.

Name : FILEATTACHLIST  
Value : 72h  
Data : ASCII  
Multiple : Yes  
Required : No  
Summary : Name of ASCII list of attached filenames

Wildcards not allowed in ASCII list filename. Wildcards allowed in ASCII list. MSG\_FILEATTACH attribute must be set.

Name : DESTFILELIST  
Value : 73h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Summary : Name of ASCII list of destination filenames

Wildcards not allowed in ASCII list filename. Wildcards allowed in ASCII list.

Name : FILEREQUEST  
Value : 74h  
Data : ASCII  
Multiple : Yes

Required : No  
Summary : Name of requested file

Wildcards allowed. MSG\_FILEREQUEST attribute must be set

Name : FILEPASSWORD  
Value : 75h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Summary : Password for FILEREQUEST

Name : FILEREQUESTLIST  
Value : 76h  
Data : ASCII  
Multiple : Yes  
Required : No  
Summary : Name of ASCII list of filenames to request

Wildcards allowed.

Name : FILEPASSWORDLIST  
Value : 77h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Summary : Name of ASCII list of passwords for FILEREQUESTLIST

Name : IMAGEATTACH  
Value : 80h  
Data : `mattach_t`  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of attached image file for display

MSG\_FILEATTACH attribute must be set. See Image Types for valid `mattach_t.type` values.

Name : ANIMATTACH  
Value : 81h  
Data : `mattach_t`  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of attached graphical animation file for display

MSG\_FILEATTACH attribute must be set. See Animation Types for valid `mattach_t.type` values.

Name : FONTATTACH  
Value : 82h  
Data : `mattach_t`  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of attached font definition file

MSG\_FILEATTACH attribute must be set. See Font Types for valid `mattach_t.type` values.

Name : SOUNDATTACH  
Value : 83h  
Data : `mattach_t`  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of attached sound file for playback

MSG\_FILEATTACH attribute must be set. See Sound Types for valid `mattach_t.type` values.

Name : PRESENTATTACH  
Value : 84h  
Data : `mattach_t`  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of attached presentation definition file

MSG\_FILEATTACH attribute must be set. See Present Types for valid `mattach_t.type` values.

Name : VIDEOATTACH  
Value : 85h  
Data : `vattach_t`  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of attached interleaved video/sound file

MSG\_FILEATTACH attribute must be set. See Video Types for valid `vattach_t.type` values and Video Compression Types for valid `vattach_t.comp` values.

Name : APPDATAATTACH  
Value : 86h  
Data : mattach\_t  
Multiple : Yes, order significant  
Required : No  
Summary : Name of attached application data file for process/display

MSG\_FILEATTACH attribute must be set. See Application Data Types for valid mattach\_t.type values.

Name : IMAGETRIGGER  
Value : 90h  
Data : typestr\_t  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of image file to trigger for display

See Image Types for valid typestr\_t.type values.

Name : ANIMTRIGGER  
Value : 91h  
Data : typestr\_t  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of animation file to trigger for display

See Animation Types for valid typestr\_t.type values.

Name : FONTRIGGER  
Value : 92h  
Data : typestr\_t  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of font definition file to trigger

See Font Types for valid typestr\_t.type values.

Name : SOUNDTRIGGER  
Value : 93h  
Data : typestr\_t  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of sound file to trigger for playback

See Sound Types for valid typestr\_t.type values.

Name : PRESENTTRIGGER  
Value : 94h  
Data : typestr\_t  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of presentation definition file to trigger

See Present Types for valid typestr\_t.type values.

Name : VIDEOTRIGGER  
Value : 95h  
Data : typestr\_t  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of interleaved video/sound file to trigger

See Video Types for valid typestr\_t.type values.

Name : APPDATATRIGGER  
Value : 96h  
Data : typestr\_t  
Multiple : Yes, order significant  
Required : No  
Summary : Type and filename of application data file to trigger

See Application Data Types for valid typestr\_t.type values.

Name : FIDOCTRL  
Value : A0h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Format : keyword ":" [" "] appdata  
Summary : FTS/FSC-compliant control information line

Any FidoNet FTS/FSC-compliant control information ("kludge") line that does not have an equivalent representation here. All data not unique to the actual control line, including leading and trailing white space, Ctrl-A (01h) character and terminating CR must be omitted. Defined in FTS-0001.

Name : FIDOAREA

Value : A1h  
Data : ASCII  
Multiple : No  
Required : No  
Summary : FTN EchoMail conference name.

Defined in FTS-0004.

Name : FIDOSEENBY  
Value : A2h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Format : net"/"node [" "[net"/"]node] [...]  
Summary : Used to store two-dimensional (net/node) SEEN-BY information

Often used in FTN EchoMail environments. Only the actual SEEN-BY data is stored and SEEN-BY: is stripped along with any leading and trailing white space characters. Defined in FTS-0004.

Name : FIDOPATH  
Value : A3h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Format : net"/"node [" "[net"/"]node] [...]  
Summary : Used to store two-dimensional (net/node)

Defined in FTS-0004. ^aPATH: is stripped along with any leading and trailing white space characters.

Name : FIDOMSGID  
Value : A4h  
Data : ASCII  
Multiple : No  
Required : No  
Format : origaddr " " serialno  
Summary : MSGID field as specified in FTS-0009.

Name : FIDOREPLYID  
Value : A5h  
Data : ASCII  
Multiple : No  
Required : No  
Format : origaddr " " serialno  
Summary : REPLY field as specified in FTS-0009.

Name : FIDOPID  
Value : A6h  
Data : ASCII  
Multiple : No  
Required : No  
Format : pID " " version [" "serialno]  
Summary : Identification string of program that created this message

Defined FSC-0046. "^aPID:" and any white space is not included.

Name : FIDOFLAGS  
Value : A7h  
Data : ASCII  
Multiple : Yes  
Required : No  
Summary : Used to store the FTN FLAGS kludge information

Note that all FLAG options that have binary representation in the message header must be removed from the FLAGS string prior to storing it. Only the actual flags option string is stored and ^aFLAGS is stripped along with any leading and trailing white space characters. Defined in FSC-0053.

Name : RFC822HEADER  
Value : B0h  
Data : ASCII  
Multiple : Yes, order significant  
Required : No  
Format : field-name ":" [field-body] [CRLF]  
Summary : Undefined RFC-822 header field

Internet Message storage format, that does not have an equivalent representation here. Folded header fields are allowed. Terminating CRLF may be omitted.

Name : RFC822MSGID  
Value : B1h  
Data : ASCII  
Multiple : No  
Required : No  
Format : "<" addr-spec ">"



Summary : Message-ID field as specified in RFC-822.

Name : RFC822REPLYID  
Value : B2h  
Data : ASCII  
Multiple : No  
Required : No  
Format : "<" addr-spec ">"  
Summary : In-Reply-To field as specified in RFC-822.

Name : UNKNOWN  
Value : F0h  
Data : undef  
Multiple : Yes  
Required : No  
Summary : Undefined header field of undefined type

This field is useful for retaining binary header fields (that do not have an equivalent representation here) between message storage formats.

Name : UNKNOWNASCII  
Value : F1h  
Data : ASCII  
Multiple : Yes  
Required : No  
Summary : Undefined header field of type ASCII

This field is useful for retaining ASCII header fields (that do not have an equivalent representation here) between message storage formats.

Name : UNUSED  
Value : FFh  
Data : undef  
Multiple : Yes  
Required : No  
Summary : Unused (deleted) header field

The data contained in this header field is of an unknown type and should not be processed.

Note:  
----

Specifically, not defined are the values F000h through FFFFh. These values are to be used for user or system defined header fields. Digital Dynamics requests that any developers or organizations that wish to have additional header fields added to this specification notify Digital Dynamics through any of the contact methods listed at the beginning of this document.

[Back to Table of Contents](#)

## Data Field Types:

These are the defined valid values for dfield\_t.type:

| Val Name<br>--- ---- | Data<br>---- | Description<br>-----                                                                                                                                                                                                                                          |
|----------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 00h TEXT_BODY        | mtext_t      | Displayable text (body of message).<br>Included in duplicate message checking.<br>All terminating white space and control characters are to be truncated from data (except when multiple contiguous CRLFs terminate the text, only the last CRLF is removed). |
| 01h TEXT_SOUL        | mtext_t      | Non-displayed text.<br>Not normally displayed. Not necessarily displayable.<br>Included in duplicate message checking.                                                                                                                                        |
| 02h TEXT_TAIL        | mtext_t      | Displayable text (tag/tear/origin lines, etc).<br>Not included in duplicate message checking.<br>All terminating white space and control characters are to be truncated from data.                                                                            |
| 03h TEXT_WING        | mtext_t      | Non-displayed text.<br>Not normally displayed. Not necessarily displayable.<br>Not included in duplicate message checking.                                                                                                                                    |
| 10h FTEXT_BODY       | ftext_t      | Formatted equivalent of TEXT_BODY to be displayed in place of TEXT_BODY if format is supported. See Image Types for valid values of ftext_t.type.                                                                                                             |
| 12h FTEXT_TAIL       | ftext_t      | Formatted equivalent of TEXT_TAIL to be                                                                                                                                                                                                                       |

|                  |          |                                                                                                                                                                            |
|------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  |          | displayed in place of TEXT_TAIL if format is supported. See Image Types for valid values of ftext_t.type.                                                                  |
| 20h IMAGEEMBED   | membed_t | Type and data of embedded raster image file for display.<br>See Image Types for valid membed.type values.                                                                  |
| 21h ANIMEMBED    | membed_t | Type and data of embedded graphical animation file for display.<br>See Animation Types for valid membed.type values.                                                       |
| 22h FONTEMBED    | membed_t | Type and data of embedded font definition file. See Font Types for valid membed_t.type values.                                                                             |
| 23h SOUNDEMBED   | membed_t | Type and data of embedded sound file for playback.<br>See Sound Types for valid membed_t.type values.                                                                      |
| 24h PRESENTEMBED | membed_t | Type and data of embedded presentation definition file.<br>See Present Types for valid membed_t.type values.                                                               |
| 25h VIDEOEMBED   | vembed_t | Type and data of embedded video/sound file for playback.<br>See Video Types for valid vembed_t.type values.<br>See Video Compression Types for valid vembed_t.comp values. |
| 26h APPDATAEMBED | membed_t | Type and data of embedded application data file for process/display.<br>See Application Data Types for valid membed_t.type values.                                         |
| FFh UNUSED       | undef    | Space allocated for future update/expansion                                                                                                                                |

Specifically, not defined are the values F000h through FFFFh. These values are to be used for user or system defined data fields. Digital Dynamics requests that any developers or organizations that wish to have additional data fields added to this specification notify Digital Dynamics through any of the contact methods listed at the beginning of this document.

[Back to Table of Contents](#)

## Message Attributes:

These are the bit values for idxrec\_t.attr and msghdr\_t.attr:

|               |        |                                                   |
|---------------|--------|---------------------------------------------------|
| MSG_PRIVATE   | (1<<0) | // Private                                        |
| MSG_READ      | (1<<1) | // Read by addressee                              |
| MSG_PERMANENT | (1<<2) | // Permanent                                      |
| MSG_LOCKED    | (1<<3) | // Msg is locked, no editing possible             |
| MSG_DELETE    | (1<<4) | // Msg is marked for deletion                     |
| MSG_ANONYMOUS | (1<<5) | // Anonymous author                               |
| MSG_KILLREAD  | (1<<6) | // Delete message after it has been read          |
| MSG_MODERATED | (1<<7) | // This message must be validated before export   |
| MSG_VALIDATED | (1<<8) | // This message has been validated by a moderator |

### Auxiliary Attributes:

These are the bit values for msghdr\_t.auxattr:

|                 |        |                                     |
|-----------------|--------|-------------------------------------|
| MSG_FILEREQUEST | (1<<0) | // File request                     |
| MSG_FILEATTACH  | (1<<1) | // File(s) attached to Msg          |
| MSG_TRUNCFILE   | (1<<2) | // Truncate file(s) when sent       |
| MSG_KILLFILE    | (1<<3) | // Delete file(s) when sent         |
| MSG_RECEIPTREQ  | (1<<4) | // Return receipt requested         |
| MSG_CONFIRMREQ  | (1<<5) | // Confirmation receipt requested   |
| MSG_NODISP      | (1<<6) | // Msg may not be displayed to user |

### Network Attributes:

These are the bit values for msghdr\_t.netattr:

|                 |        |                        |
|-----------------|--------|------------------------|
| MSG_LOCAL       | (1<<0) | // Msg created locally |
| MSG_INTRANSIT   | (1<<1) | // Msg is in-transit   |
| MSG_SENT        | (1<<2) | // Sent to remote      |
| MSG_KILLSSENT   | (1<<3) | // Kill when sent      |
| MSG_ARCHIVESENT | (1<<4) | // Archive when sent   |
| MSG_HOLD        | (1<<5) | // Hold for pick-up    |

|               |         |                                       |
|---------------|---------|---------------------------------------|
| MSG_CRASH     | (1<<6)  | // Crash                              |
| MSG_IMMEDIATE | (1<<7)  | // Send Msg now, ignore restrictions  |
| MSG_DIRECT    | (1<<8)  | // Send directly to destination       |
| MSG_GATE      | (1<<9)  | // Send via gateway                   |
| MSG_ORPHAN    | (1<<10) | // Unknown destination                |
| MSG_FPU       | (1<<11) | // Force pickup                       |
| MSG_TYPELOCAL | (1<<12) | // Msg is for local use only          |
| MSG_TYPEECHO  | (1<<13) | // Msg is for conference distribution |
| MSG_TYPENET   | (1<<14) | // Msg is direct network mail         |

[Back to Table of Contents](#)

## Translation Types:

Definition for values of \*.xlat[x]:

|              |   |                                                                                                                           |
|--------------|---|---------------------------------------------------------------------------------------------------------------------------|
| XLAT_NONE    | 0 | // No translation/End of translation list                                                                                 |
| XLAT_LF2CRLF | 1 | // Expand sole LF to CRLF                                                                                                 |
| XLAT_ESCAPED | 2 | // 7-bit ASCII escaping for ctrl and 8-bit data                                                                           |
| XLAT_HUFFMAN | 3 | // Static and adaptive Huffman coding compression                                                                         |
| XLAT_LZW     | 4 | // LZW (Lempel-Ziv-Welch) encoding for compression<br>// Terry Welch, IEEE Computer Vol 17, No 6<br>// June 1984, pp 8-19 |
| XLAT_LZC     | 5 | // LZC (modified LZW) encoding for compression<br>// Unix compress program                                                |
| XLAT_RLE     | 6 | // Run length encoding compression                                                                                        |
| XLAT_IMPLODE | 7 | // Implode compression (PKZIP v1.x)                                                                                       |
| XLAT_SHRINK  | 8 | // Shrink compression (PKZIP v1.x)                                                                                        |
| XLAT_LZH     | 9 | // LZH dynamic Huffman coding<br>// Haruyasu Yoshizaki, LHarc<br>// November, 1988                                        |

[Back to Table of Contents](#)

## Agent Types:

|               |   |                                              |
|---------------|---|----------------------------------------------|
| AGENT_PERSON  | 0 | // To or from person                         |
| AGENT_PROCESS | 1 | // Unknown process, identified by agent name |

Agent types E000h through EFFFh are reserved for Synchronet process types (defined specifically by Digital Dynamics).

**Note:**

Specifically not defined are agent types F000h through FFFFh. These values are to be used for user or system defined agent types. Digital Dynamics requests that any developers or organizations that wish to have additional agent types added to this specification notify Digital Dynamics through any of the contact methods listed at the beginning of this document.

[Back to Table of Contents](#)

## Network Types:

|              |   |                       |                |
|--------------|---|-----------------------|----------------|
|              |   | // Net Type           | Address Format |
|              |   | // -----              |                |
| NET_NONE     | 0 | // Locally created    | none           |
| NET_UNKNOWN  | 1 | // Unknown            | undef          |
| NET_FIDO     | 2 | // FTN network        | fidoaddr_t     |
| NET_POSTLINK | 3 | // PostLink network   | none           |
| NET_QWK      | 4 | // QWK based network  | ASCII          |
| NET_INTERNET | 5 | // The Internet       | ASCII          |
| NET_WWIV     | 6 | // WWIV based network | ulong          |
| NET_MHS      | 7 | // MHS network        | ASCII          |

[Back to Table of Contents](#)

## Media Types:

**Image Types:**

|               |      |                                                   |
|---------------|------|---------------------------------------------------|
| IMAGE_UNKNOWN | 0x00 | // Use image signature header to determine format |
| IMAGE_ASC     | 0x01 | // ASCII text/IBM extended ASCII graphics         |
| IMAGE_ANS     | 0x02 | // ANSI X3.64 terminal escape sequences           |
| IMAGE_AVT     | 0x03 | // AVATAR terminal escape sequences               |
| IMAGE_LVI     | 0x04 | // LVI terminal escape sequences                  |
| IMAGE_GIF     | 0x05 | // Compuserve Graphics Interchange Format (GIF)   |
| IMAGE_TIF     | 0x06 | // Tagged Image Format (AKA TIFF)                 |
| IMAGE_JPG     | 0x07 | // Joint Photographers Electronics Group (JPEG)   |
| IMAGE_T16     | 0x08 | // TrueVision 16-bit bitmap (TGA)                 |

|           |      |                                              |
|-----------|------|----------------------------------------------|
| IMAGE_T24 | 0x09 | // TrueVision 24-bit bitmap (TGA)            |
| IMAGE_T32 | 0x0a | // TrueVision 32-bit bitmpa (TGA)            |
| IMAGE_PCX | 0x0b | // ZSoft PaintBrush graphics                 |
| IMAGE_BMP | 0x0c | // Windows bitmap                            |
| IMAGE_RLE | 0x0d | // Windows bitmap (compressed)               |
| IMAGE_DIB | 0x0e | // Display independant bitmap                |
| IMAGE_PCD | 0x0f | // Kodak PhotoCD                             |
| IMAGE_G3F | 0x10 | // Group 3 FAX                               |
| IMAGE_EPS | 0x11 | // Encapsulated PostScript                   |
| IMAGE_RTF | 0x12 | // Rich text format                          |
| IMAGE_RIP | 0x13 | // Remote Imaging Protocol Script (RIPscrip) |
| IMAGE_NAP | 0x14 | // NAPLPS                                    |
| IMAGE_CDR | 0x15 | // Corel Draw!                               |
| IMAGE_CGM | 0x16 | // Computer graphics metafile                |
| IMAGE_WMF | 0x17 | // Windows metafile                          |
| IMAGE_DFX | 0x18 | // Autodesk AutoCAD                          |
| IMAGE_IFF | 0x19 | // Amiga Interchange File Format             |

**Animation Types:**

|              |   |                                                  |
|--------------|---|--------------------------------------------------|
| ANIM_UNKNOWN | 0 | // Use file signature header to determine format |
| ANIM_FLI     | 1 | // Autodesk animator                             |
| ANIM_FLC     | 2 | // Autodesk                                      |
| ANIM_GL      | 3 | // Grasprt                                       |
| ANIM_IFF     | 4 | // Amiga Interchange File Format                 |

**Video Types:**

|               |   |                                                  |
|---------------|---|--------------------------------------------------|
| VIDEO_UNKNOWN | 0 | // Use file signature header to determine format |
| VIDEO_QTIME   | 1 | // Apple Quick-time                              |
| VIDEO_FQTIME  | 2 | // Apple Flattened Quick-time                    |
| VIDEO_AVI     | 3 | // Windows Auto/Video Interleave                 |
| VIDEO_ULT     | 4 | // OS/2 Ultimotion                               |

**Video Compression Types:**

|               |    |                                                 |
|---------------|----|-------------------------------------------------|
| VCOMP_UNKNOWN | 0  | // Use file signature header to determine codec |
| VCOMP_RLE     | 1  | // Apple animation                              |
| VCOMP_SMC     | 2  | // Apple graphics                               |
| VCOMP_RPZA    | 3  | // Apple video                                  |
| VCOMP_KLIC    | 4  | // Captain crunch                               |
| VCOMP_CVID    | 5  | // CinePak                                      |
| VCOMP_RT21    | 6  | // Intel indeo R2                               |
| VCOMP_IV31    | 7  | // Intel indeo R3                               |
| VCOMP_YVU9    | 8  | // Intel YVU9                                   |
| VCOMP_JPEG    | 9  | // JPEG                                         |
| VCOMP_MRLE    | 10 | // Microsoft RLE                                |
| VCOMP_MSVC    | 11 | // Microsoft video 1                            |

**Font Types:**

|              |   |                                                  |
|--------------|---|--------------------------------------------------|
| FONT_UNKNOWN | 0 | // Use file signature header to determine format |
| FONT_TTF     | 1 | // Windows TrueType                              |
| FONT_PFB     | 2 | // PostScript Type 1 Font Binary                 |
| FONT_PFM     | 3 | // PostScript Type 1 Font Metric                 |
| FONT_AMIGA   | 4 | // Amiga Bitmapped                               |
| FONT_AGFA    | 5 | // CompuGraphic Fonts                            |

**Sound Types:**

|               |   |                                                  |
|---------------|---|--------------------------------------------------|
| SOUND_UNKNOWN | 0 | // Use file signature header to determine format |
| SOUND_MOD     | 1 | // MOD format                                    |
| SOUND_VOC     | 2 | // Sound Blaster VOC format                      |
| SOUND_WAV     | 3 | // Windows 3.1 WAV RIFF format                   |
| SOUND_MID     | 4 | // MIDI format                                   |
| SOUND_GMID    | 5 | // General MIDI format (standardized patches)    |
| SOUND_SMP     | 6 | // Turtle Beach SampleVision format              |
| SOUND_SF      | 7 | // IRCAM format                                  |
| SOUND_AU      | 8 | // Sun Microsystems AU format                    |
| SOUND_IFF     | 9 | // Amiga Interchange File Format                 |

**Application Data Types:**

|                     |    |                                                  |
|---------------------|----|--------------------------------------------------|
| APPDATA_UNKNOWN     | 0  | // Use file signature header to determine format |
| APPDATA_WORDPERFECT | 1  | // WordPerfect Document                          |
| APPDATA_WKS         | 2  | // Lotus 123 Worksheet (?)                       |
| APPDATA_WK1         | 3  | // Lotus 123 Worksheet rev 1                     |
| APPDATA_WK2         | 4  | // Lotus 123 Worksheet rev 2                     |
| APPDATA_WK3         | 5  | // Lotus 123 Worksheet rev 3                     |
| APPDATA_DBF         | 6  | // dBase III data file                           |
| APPDATA_PDX         | 7  | // Paradox data file                             |
| APPDATA_EXCEL       | 8  | // Excel data file                               |
| APPDATA_QUATRO      | 9  | // Borland Quattro Pro file                      |
| APPDATA_WORD        | 10 | // Microsoft Word                                |

# Message Storage Pseudo Code

The following is a "C like" pseudo code listing example of adding a message to an SMB message base. SMBLIB contains C functions to do most of the following operations. We are supplying this pseudo code as a general definition of the order of required operations in writing to the message base. Many details have been left out to simplify the code and to demonstrate only the basic principles.

```
shd = open ( MSGBASE.SHD , READ/WRITE/DENY_NONE )
sdt = open ( MSGBASE.SDT , READ/WRITE/DENY_NONE )
sid = open ( MSGBASE.SDT , READ/WRITE/DENY_NONE )

lock ( shd , smbhdr )
read ( shd , smbstatus )

if ( smbstatus.attr & SMB_HYPERALLOC )
    msg.hdr.offset = filelength ( sdt )

else {
    number_of_blocks = length_of_message_data / SDT_BLOCK_LEN
    if ( length_of_message_data % SDT_BLOCK_LEN ) /* unevenly divisible */
        number_of_blocks = number_of_blocks + 1

    sda = open ( MSGBASE.SDA , READ/WRITE/DENY_ALL )

    if ( fast_allocation_mode )
        seek ( sda , END_OF_FILE )

    else {
        seek ( sda , BEGINNING_OF_FILE )
        while ( not end_of_file ( sda ) ) {
            read ( sda , allocated , number_of_blocks * 2 )
            if ( allocated = 0 ) {
                seek_backwards ( sda , number_of_blocks * 2 )
                break
            }
        }
    }

    msg.hdr.offset = ( current_position ( sda ) / 2 ) * SDT_BLOCK_LEN

    allocated = 1

    write ( sda , allocated , number_of_blocks * 2 )

    close ( sda )
}

seek ( sdt , msg.hdr.offset )

write ( sdt , message_data )

if ( smbstatus.attr & SMB_HYPERALLOC )
    msg.idx.offset = filelength ( shd )

else {
    number_of_blocks = length_of_message_header / SHD_BLOCK_LEN
    if ( length_of_message_header % SHD_BLOCK_LEN ) /* unevenly divisible */
        number_of_blocks = number_of_blocks + 1

    sha = open ( MSGBASE.SHA , READ/WRITE/DENY_ALL )

    if ( fast_allocation_mode )
        seek ( sha , END_OF_FILE )

    else {
        seek ( sha , BEGINNING_OF_FILE )
        while ( not end_of_file ( sha ) ) {
            read ( sha , allocated , number_of_blocks )
            if ( allocated = 0 ) {
                seek_backwards ( sha , number_of_blocks )
                break
            }
        }
    }

    msg.idx.offset = ( current_position ( sha ) * SHD_BLOCK_LEN )
    msg.idx.offset = msg.idx.offset + smbstatus.header_offset

    allocated = 1

    write ( sha , allocated , number_of_blocks )

    close ( sha )
}
```

```

seek ( shd , msg.idx.offset )

msg.hdr.number = smbstatus.last_msg+1

write ( shd , msg.hdr )

smbstatus.total_msgs = smbstatus.total_msgs + 1
smbstatus.last_msg = msg.hdr.number

write ( shd , smbstatus )

write ( sid , msg.idx )

unlock ( shd , smbstatus )

```

[Back to Table of Contents](#)

## Message Retrieval Pseudo Code

```

shd = open ( MSGBASE.SHD , READ/WRITE/DENY_NONE )
sdt = open ( MSGBASE.SDT , READ/WRITE/DENY_NONE )
sid = open ( MSGBASE.SDT , READ/WRITE/DENY_NONE )

read ( sid , msg.idx )

seek ( shd , msg.idx.offset )

lock ( shd , msg.hdr )

read ( shd , msg.hdr )

seek ( sdt , msg.hdr.offset )

read ( sdt , msg.hdr.data_length )

unlock ( shd , msg.hdr )

```

[Back to Table of Contents](#)

## SMBUTIL

SMBUTIL is a utility that can perform various functions on an SMB message base. The primary purpose of SMBUTIL is as an example to C programmers of how to use the SMLIB functions to access and modify an SMB message base. The complete C source code for SMBUTIL is included and functions from it can be used or modified by developers at their own discretion. The following files make up SMBUTIL:

|             |                                                       |
|-------------|-------------------------------------------------------|
| SMBUTIL.EXE | Compiled and linked for 16-bit DOS (ready to run)     |
| SMBUTIL.C   | C functions                                           |
| SMBUTIL.H   | C definitions and variable prototypes                 |
| SMBUTIL.WAT | Makefile for Watcom C/C++ (type wmake -f smbutil.wat) |
| SMBUTIL.BOR | Makefile for Borland C/C++ (type make -f smbutil.bor) |

The usage syntax is as follows:

```
SMBUTIL [/opts] cmd smb_filespec.shd
```

where cmd is one or more of the following:

```

l[n] = list msgs starting at number n
r[n] = read msgs starting at number n
v[n] = view msg headers starting at number n
k[n] = kill (delete) n msgs
i<f> = import from text file f
s     = display msg base status
c     = change msg base status
m     = maintain msg base - delete old msgs and msgs over max
p[k] = pack msg base (k specifies minimum packable Kbytes)

```

where opts is one or more of the following:

```

a     = always (force) packing
z<n> = set time zone (n=min +/- from UT or 'EST','EDT','CST',etc)

```

and smb\_filespec is the base filename or file specification (wildcards) for the message base. If wildcards are used, the ".SHD" extension must be specified.

An example command line:

```
SMBUTIL MP C:\SBBS\DATA\SUBS\*.SHD
```

would maintain and pack all the message bases found in the C:\SBBS\DATA\SUBS directory.

[Back to Table of Contents](#)



# **CHKSMB**

CHKSMB is a utility that performs a comprehensive analysis of a message base to find any possible errors and calculate the number of packable bytes. It does not "fix" a message base if any errors are found, it only reports the specific errors (and exits with a non-zero error level). If any errors are reported, packing the message base with SMBUTIL may rebuild the damaged files. If that doesn't work, then use FIXSMB as a last resort.

C source code for CHKSMB is also included as an example to programmers of how to use SMLIB functions.

The usage syntax is as follows:

```
CHKSMB [/opts] smb_filespec.shd
```

where opts is one or more of the following:

```
q   = quiet mode (no beeps)
s   = stop after an erred message base (for use with wildcards)
p   = pause after an erred message base (wait for key press)
t   = don't check for unsupported translation strings (faster)
e   = display extended information on corrupted messages
```

An example command line:

```
CHKSMB /QP C:\SBBS\DATA\SUBS\*.SHD
```

would check all the message bases in the C:\SBBS\DATA\SUBS directory, without beeping on errors, and pausing after an erred message base.

[Back to Table of Contents](#)

# **FIXSMB**

FIXSMB is a utility that will rebuild the index and allocation files for a message base. Since the message headers are not necessarily stored sequentially, the order of the messages in the index may be changed when the index is rebuilt. Messages are also re-numbered, so only use this program if the index is corrupted and the messages are extremely important.

C source code for FIXSMB is also included as an example to programmers of how to use SMLIB functions.

The usage syntax is as follows:

```
FIXSMB [/M] smb_file
```

An example command line:

```
FIXSMB \SBBS\DATA\MAIL
```

Only use the "/M" command line switch if fixing an older Synchronet e-mail message base (created with SBBS v2.1 or earlier). Once the SMB\_EMAIL status attr is set ("SMBUTIL S" will report a status attr of 1), the "/M" is not required.

[Back to Table of Contents](#)

# **SMLIB**

SMLIB is a library of C functions for accessing and storing messages in an SMB format message base. It can eliminate much of the development time for developers that wish to use the library in whole or in part, or use the functions as examples for their own message base function library. The library consists of the following files:

|           |                                                           |
|-----------|-----------------------------------------------------------|
| SMBDEFS.H | Constant definitions, macros, and data types              |
| SMLIB.H   | Library constants and function prototypes                 |
| SMLIB.C   | Function definitions                                      |
| SMBVARS.C | Global variable definitions (doubles as declaration file) |

For developers to use this library with their program, they must include the "SMLIB.H" header file at the top of each C file that uses any of the library functions, global variables, data types, macros, and constants. This can be done by simply adding the following line to each .C file:

```
#include "smlib.h"
```

If SMLIB.H is included, there is no need to include SMBDEFS.H or SMBVARS.C.

To link the library functions and variables with a main program, the files SMBVARS.OBJ and SMLIB.OBJ must be linked with the main program .OBJ files.

If the operating system is DOS, be sure that all .OBJ files are compiled for the same memory model.

Example MAKEFILES for compiling and linking SMBUTIL with Borland C/C++ (SMBUTIL.BOR) and Watcom C/C++ (SMBUTIL.WAT) are included.

[Back to Table of Contents](#)

## **SMBDEFS.H**

The SMBDEFS.H file contains important constant definitions and data types (also defined in this document). If ever this document and SMBDEFS.H are inconsistent with each other, then SMBDEFS.H is to be considered correct and this document in error. If such a discrepancy is found, please notify Digital Dynamics so it can be corrected in a future revision of the specification.

Most notable of the data types is a structure called `smbmsg_t` (not defined in this document). It contains the fixed and variable portions of a message's header record as well as convenience pointers to the sender's name (`smbmsg_t.to`), recipient's name (`smbmsg_t.from`), network addresses, and more. If multiple SENDER header fields are included (for example), then `smbmsg_t.to` will point to the last SENDER header field in the header record. Convenience pointers for other data items work in the same fashion if multiple header fields of the same type exist in the header record.

Variables of the `smbmsg_t` data type (and pointers to variables of `smbmsg_t` type) are used as arguments to many of the SMLIB functions.

[Back to Table of Contents](#)

## **SMBVARS.C**

The SMBVARS.C file contains definitions of the global variables used by the SMLIB functions. It is a fairly small file since there are a small number of global variables (by design). This file is used for both definitions and declarations, so no "extern" declarations need to be made in developers source code as long as SMBVARS.C or (preferably) SMLIB.H is included in the source code.

[Back to Table of Contents](#)

## **SMLIB.H**

The SMLIB.H file contains prototypes of all the functions in the SMLIB.C file. It is necessary to include this file in C source code if any of the SMLIB functions are used. The following C source line will include this file:

```
#include "smlib.h"
```

and should be placed near the top of all C source files that use SMLIB functions, variables, constants, or data types.

Function prototypes are necessary for compilers to know the correct calling syntax of a function and detect incorrect usage. Prototypes are also useful as a quick reference for programmers as to the correct calling syntax of a specific function.

[Back to Table of Contents](#)

## **SMLIB.C**

The SMLIB.C file contains the actual SMLIB library functions. This source file is not a stand alone program, but instead must be compiled and linked with a main source file to create the executable program.

The functions in this file are organized in a logical order, but their order is actually irrelevant to the compiling, linking, and execution of the resulting program.

A comment block precedes each function, explaining what the function does, how the passed parameters are used, and what the return code (if any) indicates. A more detailed explanation of each function is included here:

**int smb\_open(int retry\_time)**

The `smb_open()` function must be called before the message base is accessed (read from or written to). The parameter, `retry_time`, is the maximum number of seconds to wait while retrying to lock the message base header. If `retry_time` is 0, then the message base header is not locked or read (this is called "Fast Open" and should only be used when speed is more important than checking for compatibility and validity upon opening). The global variable `smb_file` must be initialized with the path and base filename of the message base. This function returns 0 on success, 1 if the .SDT file could not be

opened, 2 if the .SHD file could not be opened, and 3 if the .SID file could not be opened. If the message base header could not be locked, this function returns -1. If the message base ID is incorrect, it returns -2. And if the message base is of an incompatible version, it returns -3.

The `errno` global variable (standard of most C libraries) will most likely contain the error code for open failure.

#### **int smb\_open\_da(int retry\_time)**

The `smb_open_da()` function is used to open the data block allocation file for writing messages to a message base. The parameter, `retry_time`, is the maximum number of seconds to wait while retrying to open the file. This function returns 0 on success. -1 is returned if an open error other than "Access Denied" is returned from the operating system, and the global variable `errno` will contain the error code. -2 is returned if the `retry_time` has been reached, and -3 is returned if the file descriptor could not be converted to a stream by the `fdopen()` function.

`fclose(sda_fp)` should be called immediately after all necessary file access has been completed.

This function is not used with the Hyper Allocation storage method.

#### **int smb\_open\_ha(int retry\_time)**

The `smb_open_ha()` function is used to open the header block allocation file for writing messages to a message base. The parameter, `retry_time`, is the maximum number of seconds to wait while retrying to open the file. This function returns 0 on success. -1 is returned if an open error other than "Access Denied" is returned from the operating system, and the global variable `errno` will contain the error code. -2 is returned if the `retry_time` has been reached, and -3 is returned if the file descriptor could not be converted to a stream by the `fdopen()` function.

`fclose(sha_fp)` should be called immediately after all necessary file access has been completed.

This function is not used with the Hyper Allocation storage method.

#### **int smb\_create(ulong max\_crcs, ulong max\_msgs, ushort max\_age, ushort attr, int retry\_time)**

The `smb_create()` function is used to create a new message base or reset an existing message base. The parameters `max_crcs`, `max_msgs`, `max_age`, and `attr` are used to set the initial status of the message base status header. The parameter, `retry_time` is the maximum number of seconds to wait while retrying to lock the message base header. This functions returns 0 on success or 1 if the message base header could not be locked.

#### **int smb\_trunchdr(int retry\_time)**

The `smb_trunchdr()` function is used to truncate the header file when packing the message base and writing the new header information back to the header file. The parameter, `retry_time` is the maximum number of seconds to wait while retrying to truncate the header file. Returns 0 on success, -1 if error was other than "Access Denied", or -2 if `retry_time` reached.

#### **int smb\_locksmbhdr(int retry\_time)**

The `smb_locksmbhdr()` function is used to lock the first message base (status) header. The parameter, `retry_time` is the number of seconds to wait while retrying to lock the header. The `smb_unlocksmbhdr()` function should always be used to unlock the header after accessing the message base header (usually with `smb_getstatus()` and/or `smb_putstatus()`). Returns 0 if successful, -1 if unsuccessful.

#### **int smb\_unlocksmbhdr()**

The `smb_unlocksmbhdr()` function is used to unlock a previously locked message base header (using `smb_lockmsghdr()`). Returns 0 on success, non-zero on failure.

#### **int smb\_getstatus(smbstatus\_t \*hdr)**

The `smb_getstatus()` function is used to read the status message base header into the `hdr` structure. Returns 0 on success, 1 on failure.

#### **int smb\_putstatus(smbstatus\_t hdr)**

The `smb_putstatus()` function is used to write the status information to the first message base header. The parameter `hdr`, contains the status information to be written. Returns 0 on success, 1 on failure.

#### **int smb\_getmsgidx(smbmsg\_t \*msg)**

The `smb_getmsgidx()` function is used to get the byte offset for a specific message header in the message header file based on the message base index.

If msg->hdr.number is non-zero when this function is called, then the index will be searched for this message number. If the message number is found in the index, the msg->idx.offset is set to the byte offset of the message header record in the header file and msg->offset is set to the record offset of the index record in the index file, and the function returns 0. If the message number is not found in the index, the function returns 1.

If msg->hdr.number is zero, msg->idx.offset and msg->idx.number are obtained from the index record at record offset msg->offset. If msg->offset is an invalid record offset when this function is called, the function returns 1. Otherwise, the function returns 0.

**int smb\_getlastidx(idxrec\_t \*idx)**

Reads the last index record of the currently open message base into the idxrec\_t structure pointed to by idx. Returns 0 if successful, -1 if the index is empty or unopened, or -2 if the record can't be read.

**int smb\_getmsgghdrln(smbmsg\_t msg)**

The smb\_getmsgghdrln() function is used to calculate the total length of message header msg including both fixed and variable length portions. This function returns the length of the header record in bytes.

**long smb\_getmsgdatlen(smbmsg\_t msg)**

The smb\_getmsgdatlen() function is used to calculate the total length of the data for message msg. This function returns the length of all data fields combined.

**int smb\_lockmsgghdr(smbmsg\_t msg, int retry\_time)**

The smb\_lockmsgghdr() function is used to lock the header record for message msg. The parameter retry\_time is the maximum number of seconds to wait while retrying to lock the header. Returns 0 on success, -1 on failure. The function smb\_unlockmsgghdr() should immediately be called after accessing the message header (usually with smb\_getmsgghdr() or smb\_putmsgghdr()).

**int smb\_getmsgghdr(smbmsg\_t \*msg)**

The function smb\_getmsgghdr() is used to read the header record for message msg. msg->idx.offset must be initialized to the byte offset of the header record in the header file before this function is called. The function smb\_freemsgmem() must be called to free the memory allocated by this function for the header and data fields. This function returns 0 on success, -1 if the fixed portion of the message header record could not be read, -2 if the message header ID was incorrect, -3 if memory could not be allocated, -4 if a data field could not be read, -5 if the fixed length portion of a header field could not be read, -6 if the variable length portion of a header field could not be read, -7 if one or more of the mandatory header fields (SENDER, RECIPIENT, or SUBJECT) are missing, -8 if total\_dfields extends beyond the end of the header record, or -9 if incompatible header version.

Several convenience pointers in the msg structure are initialized by this function to point to the last occurrence of the SENDER (msg->from), RECIPIENT (msg->to), SUBJECT (msg->subj), etc.

**int smb\_unlockmsgghdr(smbmsg\_t msg)**

The smb\_unlockmsgghdr() function is used to unlock a previously locked message header (with smb\_lockmsgghdr()). This function returns 0 on success, non-zero on failure.

**int smb\_addcrc(ulong max\_crcs, ulong crc, int retry\_time)**

The smb\_addcrc() function is used to add a CRC-32 to the CRC history file for a message base, automatically checking for duplicates. The parameter max\_crcs should be the max\_crcs defined in the status header of the message base. The parameter crc, is the CRC-32 of the TEXT\_BODY and TEXT\_SOUL data fields for the message. The parameter retry\_time is the maximum number of seconds to wait when retrying to open the CRC history file.

This function returns -1 if there was an open error, -2 if the retry\_time was reached, -3 if there was a memory allocation error, 1 if the CRC already exists in the CRC history file (indicating a duplicate message), or 0 on success (and no duplicate).

**int smb\_hfield(smbmsg\_t \*msg, ushort type, ushort length, void \*data)**

The smb\_hfield() function is used to add a header field to the structure msg. The parameters type, length, and data, must be specified according to the header field values listed in this specification. This function returns 0 on success, non-zero on memory allocation error. The function smb\_freemsgmem() must be called to free the memory allocated by this function.

**int smb\_dfield(smbmsg\_t \*msg, ushort type, ulong length)**

The `smb_dfield()` function is used to add a data field to the structure `msg`. The parameters `type` and `length` must be specified according to the data field values listed in this specification. This function returns 0 on success, non-zero on memory allocation error. The function `smb_freemsgmem()` must be called to free the memory allocated by this function.

**`int smb_addmsgghdr(smbmsg_t *msg, smbstatus_t *status, int storage, int retry_time)`**

The `smb_addmsgghdr()` function is used to add a new message header to the message header file and update the index file. The `msg` and `status` structures are updated to reflect the new total messages, last message number, etc. The `storage` parameter is used to indicate the storage method to use (either `SMB_SELFPACK`, `SMB_FASTALLOC`, or `SMB_HYPERALLOC`). If the storage type is `SMB_SELFPACK`, the header block allocation file will be searched for unused block(s) to store this header. If the storage type is `SMB_FASTALLOC` or `SMB_HYPERALLOC`, the header is stored at the end of the header file. Returns 0 on success, non-zero on failure. The parameter `retry_time` is the maximum number of seconds to wait while retrying to lock and open files.

**`int smb_putmsg(smbmsg_t msg)`**

The `smb_putmsg()` function calls both the `smb_putmsgghdr()` and `smb_putmsgidx()` functions to write the header and index elements of a message to the appropriate files. Returns 0 on success, non-zero on failure.

**`int smb_putmsgidx(smbmsg_t msg)`**

The `smb_putmsgidx()` function is used to store a message index in the message index file. The message index can be for a new message or an existing message. Returns 0 on success, non-zero on failure.

**`int smb_putmsgghdr(smbmsg_t msg)`**

The `smb_putmsgghdr()` function is used to store a message header in the message header file. The message header can be for a new message or an existing message. Returns 0 on success, non-zero on failure.

**`void smb_freemsgmem(smbmsg_t msg)`**

Frees allocated memory for the header and data fields in the `msg` structure. This function must be called to free the memory allocated by the functions `smb_hfield()`, `smb_dfield()`, and `smb_getmsgghdr()`.

**`long smb_hdrblocks(ulong length)`**

The `smb_hdrblocks()` function is used to calculate the number of blocks required to store a message header of length size (in bytes). This function returns the number of blocks required.

**`long smb_datblocks(ulong length)`**

The `smb_datblocks()` function is used to calculate the number of blocks required to store message data of length size (in byte). This function returns the number of blocks required.

**`long smb_allochdr(ulong length)`**

The `smb_allochdr()` function is used to search for free blocks to store a message header of length bytes and mark the free blocks as allocated in the header allocation file. This function returns the byte offset to the header record or a negative number on error. The function `smb_open_ha()` should be called prior to calling this function and `fclose(smb_fp)` should be called after. The function is called from `smb_addmsgghdr()`, so you probably have no need to call this function directly.

**`long smb_fallochdr(ulong length)`**

The `smb_fallochdr()` function works exactly the same as the `smb_allochdr()` function except it is much faster because the header allocation file is not searched for free blocks. The function is called from `smb_addmsgghdr()`, so you probably have no need to call this function directly.

**`long smb_hallochdr(ulong header_offset)`**

This `smb_hallochdr()` functions works exactly the same as the `smb_fallochdr()` function except the `status.header_offset` is passed as the argument and the header allocation (.SHA) file is not updated so `smb_open_ha()` need not be called. The function is called from `smb_addmsgghdr()`, so you probably have no need to call this function directly.

**`long smb_allocdat(ulong length, ushort headers)`**

The `smb_allocdat()` function is used to search for free blocks to store length amount of data for a message. The parameter `headers`, indicates the number of message headers that are associated with this data. Normally, the `headers` parameter will be 1, unless this message is part of a mass mailing. The offset to the allocated data blocks is returned, or a negative value on error. The

function `smb_open_da()` should be called prior to calling this function and `fclose(sda_fp)` should be called after.

**long `smb_fallocdat(ulong length, ushort headers)`**

The `smb_fallocdat()` function works exactly the same as the `smb_allocdat()` function except it is much faster because the data allocation file is not searched for free blocks.

**long `smb_hallocdat()`**

The `smb_hallocdat()` function works exactly the same as the `smb_hallocdat()` function except no argument is passed and the data allocation file (.SDA) is not updated so `smb_open_da()` need not be called.

**int `smb_incdat(ulong offset, ulong length, ushort headers)`**

The `smb_incdat()` function is used to increment the header counter in the data allocation file for the data starting at the byte offset and length size in bytes. The parameter `headers`, indicates the number of headers to add to the current allocation value in the data allocation file. Returns 0 on success, non-zero on failure.

**int `smb_freemsg(smbmsg_t msg, smbstatus_t status)`**

The `smb_freemsg()` function is used to free the disk space allocated for the header and data fields of the message `msg`. Returns 0 on success, non-zero on failure. The parameter, `status`, must be the current status from the message base header for this message base.

**int `smb_freemsgdat(ulong offset, ulong length, ushort headers)`**

The `smb_freemsgdat()` function is used to decrement the data block allocation records in the data allocation file associated with the data in the data file by the value of the `headers` parameter (normally 1). The parameter `offset` indicates the byte offset to the beginning of the message data in the data file and the parameter `length` is the total length of the message data. Returns 0 on success, non-zero on failure.

**int `smb_freemsghdr(ulong offset, ulong length)`**

The `smb_freemsghdr()` function is used to set the header block allocation records in the header allocation file to 0 (indicated non-allocated block). The parameter `offset` indicates the byte offset to the beginning of the header record being freed and the parameter `length` indicates the total length of the header record. Returns 0 on success, non-zero on failure.

**int `smb_stack(int op)`**

The `smb_stack()` function is used to save and restore message base information so that multiple message bases can be open simultaneously. The stack can save up to 4 message bases (allowing 5 simultaneously open message bases). The stack is a "last in, first out" storage area for open message bases. If the `op` parameter is `SMB_STACK_PUSH`, `smb_stack()` will save (push) the current message base onto the stack. Calling `smb_stack(SMB_STACK_POP)` will restore (pop) the most recently pushed message base off the stack. Calling `smb_stack(SMB_STACK_XCHNG)` will exchange the most recently pushed message base and the current message base (replacing the top of the stack with the current message base).

**void `smb_close()`**

Closes the header, data, and index files for the currently open message base.

[Back to Table of Contents](#)

## **Miscellaneous SMLIB Files**

**CRC32.H**                    **C header file for CRC-32 calculations**

This file contains a static 32-bit CRC table (`crc32tbl[]`) and a macro (`ucrc32`) that uses this table to calculate 32-bit CRCs one byte at a time.

Example:

```
    ulong crc=0xffffffff;

for(i=0;i<length;i++)
    crc=ucrc32(buf[i],crc);
crc=~crc;
```

**CRC16.C**                    **C functions for 16-bit CRC calculations**

This file contains a function (`ucrc16`), to calculate 16-bit CRCs one byte at a time and a function (`crcl6`) that uses the `ucrc16()` function to calculate the 16-bit CRC of an ASCIIZ character string.

Example:

```
    ushort crc;  
  
crc=crc16("Text");
```

## **LZH.H                      Function prototypes for LZH.C**

This file contains function prototypes for the two most important functions in LZH.C, lzh\_encode() and lzh\_decode().

Example:

```
    uchar str[256],lzh[512];  
    long length;  
  
strcpy(str,"This is a string of text");  
length=lzh_encode(str,strlen(str),lzh);  
lzh_decode(lzh,length,str);
```

## **LZH.C                      C functions for LZH encoding (compression/decompression)**

This file contains the functions for encoding and decoding LZH compressed data. If the macro LZH\_DYNAMIC\_BUF is defined when this file is compiled, temporary buffers will be dynamically allocated as opposed to static. This may be slower than the static buffer method, but frees the allocated memory after encoding or decoding. If free memory for your application is an issue, then define this macro when compiling this file.

Example (Borland C):

```
bcc -c -DLZH_DYNAMIC_BUF lzh
```

Example (Watcom C):

```
wcc -dLZH_DYNAMIC_BUF lzh
```

[Back to Table of Contents](#)

# **SMBLIB Storage Example**

```
#include "smblib.h"  
#include "crc16.c"  
  
int main(void)  
{  
    char    str[256]                // General purpose string  
        ,*msg_text="Hello, world!" // Message text  
        ,nul_buf[SDT_BLOCK_LEN]={0} // NULL initialized buffer  
        ;  
    int     i                      // General purpose integer  
        ,storage=SMB_SELFPACK      // Default storage method  
        ,retry=10                  // Retry for opening/locking files  
        ;  
    ushort  max_age=0              // Default maximum age of messages  
        ,xlat=XLAT_NONE            // Translation string  
        ,tzone=PST                 // Time zone  
        ,copies=1                  // Number of copies of this msg  
        ;  
    ulong   max_msgs=500           // Default max number of msgs  
        ,max_crcs=0               // Default max crcs  
        ,length                   // Length of msg text  
        ,offset                    // Offset to msg text in data file  
        ;  
    smbmsg_t  msg;                 // Message structure  
    smbstatus_t status;            // Message base status record  
  
    strcpy(smb_file,"MSGBASE");    // We'll use "MSGBASE" for the name  
    if((i=smb_open(retry))!=0) {    // Can't open!?!  
        printf("smb_open returned %d\n",i);  
        return(1); }  
  
    if(!filelength(fileno(shd_fp))) // Message base not created yet  
        smb_create(max_crcs        // Create with default settings  
            ,max_msgs  
            ,max_age  
            ,storage==SMB_HYPERALLOC  
                ? SMB_HYPERALLOC : 0 // SMB_EMAIL if this was e-mail  
            ,retry  
        );  
  
    if((i=smb_locksmbhdr(retry))!=0) { // Can't lock status base header  
        printf("smb_locksmbhdr returned %d\n",i);  
        smb_close();  
        return(1); }  
}
```



```

if((i=smb_getstatus(&status))!=0) {           // Can't read status base header
    smb_unlocksmbhdr();
    smb_close();
    printf("smb_getstatus returned %d\n",i);
    return(1); }

if(status.attr&SMB_HYPERALLOC)
    storage=SMB_HYPERALLOC;
else
    storage=SMB_SELFPACK;

length=strlen(msg_text);                      // Get length of message
length+=sizeof(xlat);                        // Add length of xlat string

if(storage==SMB_HYPERALLOC)                  // Allocate space for message text
    offset=smb_hallocdat();
else {
    if((i=smb_open_da(retry))!=0) {
        smb_unlocksmbhdr();
        printf("smb_open_da returned %d\n",i);
        smb_close();
        return(1); }
    if(storage==SMB_FASTALLOC)
        offset=smb_fallocdat(length,copies);
    else
        offset=smb_allocdat(length,copies);
    fclose(sda_fp); }

fseek(sdt_fp,offset,SEEK_SET);                // Seek to beginning of data block
fwrite(&xlat,sizeof(xlat),1,sdt_fp);          // Write xlat string
fwrite(msg_text,strlen(msg_text),1,sdt_fp);  // Write message text
fwrite(nul_buf,SDT_BLOCK_LEN-length         // Write NULLs out to end of block
,1,sdt_fp);
fflush(sdt_fp);                             // Flush output buffer
smb_unlocksmbhdr();                          // Unlock status base header

memset(&msg,0,sizeof(smbmsg_t));             // Initialize header to NULL
memcpy(msg.hdr.id,"SHD\x1a",4);              // Always set to SHD^Z
msg.hdr.version=SMB_VERSION;
msg.hdr.when_written.time=time(NULL);
msg.hdr.when_written.zone=tzone;
msg.hdr.when_imported.time=time(NULL);
msg.hdr.when_imported.zone=tzone;
msg.hdr.offset=offset;

strcpy(str,"All");                           // Send message to "All"
if((i=smb_hfield(&msg,RECIPIENT,strlen(str),str))!=0) {
    printf("smb_hfield returned %d\n",i);
    smb_freemsgdat(offset,length,copies);
    smb_close();
    return(1); }

strlwr(str);                                // If this were e-mail, idx.to
msg.idx.to=crcl6(str);                      // would be the "to" user number

strcpy(str,"Sysop");                         // Send message from "Sysop"
if((i=smb_hfield(&msg,SENDER,strlen(str),str))!=0) {
    printf("smb_hfield returned %d\n",i);
    smb_freemsgdat(offset,length,copies);
    smb_freemsgmem(msg);
    smb_close();
    return(1); }

strlwr(str);                                // If this were e-mail, idx.from
msg.idx.from=crcl6(str);                    // would be the "from" user number

strcpy(str,"This is a test");                // Set the message subject/title
if((i=smb_hfield(&msg,SUBJECT,strlen(str),str))!=0) {
    printf("smb_hfield returned %d\n",i);
    smb_freemsgdat(offset,length,copies);
    smb_freemsgmem(msg);
    smb_close();
    return(1); }

strlwr(str);
msg.idx.subj=crcl6(str);

if((i=smb_dfield(&msg,TEXT_BODY,length))!=0) {
    printf("smb_dfield returned %d\n",i);
    smb_freemsgdat(offset,length,copies);
    smb_freemsgmem(msg);
    smb_close();
    return(1); }

if((i=smb_addmsgghdr(&msg,&status,storage,retry))!=0) {
    printf("smb_addmsgghdr returned %d\n",i);
    smb_freemsgdat(offset,length,copies);
    smb_freemsgmem(msg);
    smb_close();

```

```

        return(1); }

smb_freemsgmem(msg); // Unnecessary if exiting main()
smb_close(); // Unnecessary if exiting main()
return(0);
}

```

[Back to Table of Contents](#)

## SMBLIB Retrieval Example

```

#include "smblib.h"

int main(void)
{
    char        ch; // General purpose character
    int         i,  // General purpose integer
              retry=10; // Retry for opening/locking files
    ushort     xlat; // Translation string
    ulong      l;   // General purpose long integer
    smbmsg_t   msg; // Message structure

    strcpy(smb_file,"MSGBASE"); // We'll use "MSGBASE" for the name
    if((i=smb_open(retry))!=0) { // Can't open!?!
        printf("smb_open returned %d\n",i);
        return(1); }

    if(!filelength(fileno(shd_fp))) { // Message base not created yet
        printf("Empty\n");
        smb_close();
        return(0); }

    for(msg.offset=0;!ferror(sid_fp);msg.offset++) {

        fseek(sid_fp,msg.offset*sizeof(idxrec_t),SEEK_SET);
        if(!fread(&msg.idx,1,sizeof(idxrec_t),sid_fp))
            break;

        if((i=smb_lockmsgshdr(msg,retry))!=0) {
            printf("smb_lockmsgshdr returned %d\n",i);
            break; }
        if((i=smb_getmsgshdr(&msg))!=0) {
            smb_unlockmsgshdr(msg);
            printf("smb_getmsgshdr returned %d\n",i);
            break; }
        if((i=smb_unlockmsgshdr(msg))!=0) {
            smb_freemsgmem(msg);
            printf("smb_unlockmsgshdr returned %d\n",i);
            break; }

        printf("Subj  : %s\n",msg.subj);
        printf("To    : %s\n",msg.to);
        printf("From  : %s\n",msg.from);
        printf("Date  : %s\n",ctime((time_t *)&msg.hdr.when_written.time));

        for(i=0;i<msg.hdr.total_dfields;i++)
            switch(msg.dfield[i].type) {
                case TEXT_BODY: // Only show BODY and TAIL data fields
                case TEXT_TAIL:
                    fseek(sdt_fp,msg.hdr.offset+msg.dfield[i].offset
                        ,SEEK_SET);
                    fread(&xlat,sizeof(xlat),1,sdt_fp);
                    if(xlat!=XLAT_NONE) // No translations supported
                        continue;
                    for(l=sizeof(xlat);l<msg.dfield[i].length;l++) {
                        ch=fgetc(sdt_fp);
                        if(ch)
                            putchar(ch); }
                    printf("\n");
                    break; }
            printf("\n");

        smb_freemsgmem(msg); } // Free memory allocated by smb_getmsgshdr()

    smb_close();
    return(0);
}

```

[Back to Table of Contents](#)

## SMBLIB Performance Issues

Since importing messages is the usually the most time consuming task likely undertaken by an SMB application, it is also the most susceptible to design

issues that effect performance.

### Opening and Closing

When importing multiple messages for a single message base, it appears logical to open the message base, import all the messages, then close it. This indeed is preferred over opening and closing the message base for each message.

When importing multiple messages for possibly non-consecutive message bases, developers may easily make the mistake of opening and closing the message base for each message. This is not necessary and can considerably hinder the import performance. The easiest solution is to only close the message base and open a new one if the next message to be imported is not for the same message base as the previously imported message. Example:

```
smb_file[0]=0;
for(i=0;i<total_messages_to_be_imported;i++) {
    if(strcmp(get_messagebase_for_this_message(i),smb_file)) {
        if(smb_file[0]) /* We've already opened one */
            smb_close();
        strcpy(smb_file,get_messagebase_for_this_message(i));
        smb_open(10); }
    /* Import this message */
}
if(smb_file[0])
    smb_close();
```

A more advanced method is to keep multiple message bases open at the same time. Due to the likely limitation of total file handles on the system, it is suggested to keep the number of simultaneously open message bases at or below 3. SMBLIB includes the function `smb_stack()` to easily "push" and "pop" message bases without closing them (push is the equivalent to "save" and pop is the equivalent to "restore"). The downside of this function is that you cannot access message bases on the stack without actually popping them off (in reverse of the order they were pushed). You can however "exchange" the current message base with the message base on the top of the stack (most recently pushed). To intelligently juggle more than two open message bases, the developer should create their own equivalent of the `smb_stack()` function so they can access the message bases on the stack without popping them off. An example of keeping a maximum of two message bases open using `smb_stack()`:

```
char last_messagebase[128],new_messagebase[128];

smb_file[0]=0;
last_messagebase[0]=0;
for(i=0;i<total_messages_to_be_imported;i++) {
    strcpy(new_messagebase,get_messagebase_for_this_message(i));
    if(strcmp(new_messagebase,smb_file)) { /* Not current message base */
        if(smb_file[0]) { /* We've already opened one */
            if(!strcmp(new_messagebase,last_messagebase)) { /* Same as last */
                strcpy(last_messagebase,smb_file);
                smb_stack(SMB_STACK_XCHNG); } /* Restore previous base */
            else {
                if(last_messagebase[0]) {
                    smb_stack(SMB_STACK_XCHNG);
                    smb_close();
                    strcpy(last_messagebase,new_messagebase); }
                else {
                    strcpy(last_messagebase,smb_file);
                    smb_stack(SMB_STACK_PUSH); } /* Save current base */
                strcpy(smb_file,new_messagebase);
                smb_open(10); } }
        else {
            strcpy(smb_file,new_messagebase);
            smb_open(10); } }
    /* Import this message */
}
if(smb_file[0])
    smb_close();
if(last_messagebase[0]) {
    smb_stack(SMB_STACK_POP);
    smb_close(); }
```

The second example would be of negligible performance gain over the first example (6 open operations versus 7) if the messages to import were in the following order:

```
msg[0] --> msgbase[0]          // 0 opened
msg[1] --> msgbase[1]          // 0 pushed 1 opened
msg[2] --> msgbase[1]
msg[3] --> msgbase[2]          // 1 closed 0 popped 0 closed 2 opened
msg[4] --> msgbase[0]          // 2 pushed 0 opened
msg[5] --> msgbase[2]          // 0 pushed 2 popped (exchanged)
msg[6] --> msgbase[3]          // 2 closed 0 popped 0 closed 3 opened
msg[7] --> msgbase[0]          // 3 pushed 0 opened
```

The second example would be of significant performance gain over the first example (4 open operations versus 8) if the messages to import were in the following order:

```
msg[0] --> msgbase[0]          // 0 opened
msg[1] --> msgbase[1]          // 0 pushed 1 opened
msg[2] --> msgbase[0]          // 1 pushed 0 popped (exchanged)
msg[3] --> msgbase[1]          // 0 pushed 1 popped (exchanged)
msg[4] --> msgbase[0]          // 1 pushed 0 popped (exchanged)
msg[5] --> msgbase[2]          // 0 pushed 1 popped (exchanged) 1 closed 2 opened
msg[6] --> msgbase[3]          // 2 pushed 0 popped (exchanged) 0 closed 3 opened
msg[7] --> msgbase[2]          // 3 pushed 2 popped (exchanged)
```

More advanced use of "stack-like" message base file handle storage can easily reduce the number of open operations, therefore increasing import performance under more adverse message base ordering conditions.

### **Compression**

If any message data compression features are offered by the application, it is important the the application not unnecessarily compress data that will not save any storage space. While this may seem an obvious statement, please review the following pseudo-code example:

```
if ( message_data_length < SDT_BLOCK_LEN )
    // Store uncompressed data
else {
    // Compress data
    if ( ( compressed_data_length / SDT_BLOCK_LEN )
        < ( message_data_length / SDT_BLOCK_LEN ) ) // Saves a block or more
        // Store compressed data
    else
        // Store uncompressed data
}
```

Since the SMB format stores message data in fixed length blocks, there is no point in storing a message in compressed format if it requires the same number of blocks as the uncompressed format (i.e. a message that is two blocks in length in uncompressed format and only a block and a half in length when compressed should not be stored in compressed format since it still requires two full blocks of storage). It is important to note that in the above example, the length of the data translation string was not taken into account in determining the number of required blocks. Also, the `smb_datblocks()` function is normally used in determining the number of required blocks to store a given data length and it is a little more involved than simply dividing the length of the data by `SDT_BLOCK_LEN`.

### **[Back to Table of Contents](#)**

## **Bibliography**

Title : The C Programming Language  
Publisher : Prentice Hall  
Author : Brian W. Kernighan and Dennis M. Ritchie

Document : ARPANET Request for Comments (RFC) #822  
Title : Standard for the Format of ARPA Internet text messages  
Publisher : SRI International  
Author : David H. Crocker, University of Delaware

Document : FTS-0001  
Publisher : FSC  
Author : Randy Bush, Pacific Systems Group

Document : FTS-0004  
Title : EchoMail Specification  
Publisher : FSC  
Author : Bob Hartman

Document : FTS-0009  
Title : A standard for unique message identifiers and reply chain linkage  
Publisher : FSC  
Author : Jim Nutt

Document : FSC-00046  
Title : A Product Identifier for FidoNet Message Handlers  
Publisher : FSC  
Author : Joaquim H. Homrighausen

Document : FSC-00053  
Title : Specifications for the ^aFLAGS field  
Publisher : FSC  
Author : Joaquim H. Homrighausen

### **[Back to Table of Contents](#)**

# Implementations

Product : Synchronet Multinode BBS Software  
Developer : Digital Dynamics  
Level : III  
Version : 2.20

Product : Synchronet/FidoNet Import/Export Utility (SBBSFIDO)  
Developer : Digital Dynamics  
Level : III  
Version : 2.23

Product : Synchronet UTI (Universal Text Interface) Driver  
Developer : Digital Dynamics  
Level : III  
Version : 2.23

Product : SBBSecho FidoNet Packet Tosser for Synchronet  
Developer : Digital Dynamics  
Level : III  
Version : 1.11

Product : NetXpress Internet UUCP for Synchronet  
Developer : Merlin Systems  
Level : II  
Version : 1.50

Product : InterEcho FidoNet Packet Tosser  
Developer : InterMail Sales Inc  
Level : II  
Version : 1.11

## [Back to Top](#)

---

**[Copyright](#) © 2000 by [Rob Swindell](#)**

[Synchronet BBS Software](#) (Synchronet) Version 3 is comprised of several documentation, library, executable, and [source code](#) files, all of which are covered by the [GNU General Public License](#) with the exception of the following portions covered by the [GNU Lesser General Public License](#): SMLIB and XSDK.

Synchronet Version 2 (for DOS and OS/2) and its source code was released to the [Public Domain](#) by [Digital Dynamics](#) in 1997 and remains Public Domain software today. Synchronet Version 3 is not Public Domain software.

[Rob Swindell](#)  
PO Box 501  
Yorba Linda, CA 92885  
<http://www.synchro.net>

For the complete Copyright Information please read the [Copyright Documentation](#) .