

ATARI MUTINEER

BBS DOCUMENTATION

Version 1.3.183

(C)2026 Rick Collette (megolith)

Operation, programming, and troubleshooting reference.

CONTENTS

1. Notices And License
2. Atari Mutineer Documentation
3. Atari Mutineer Sysops Guide
4. Altirra 850 Modem Reference
5. Atari Mutineer Core Programmers Guide
6. Runtime Architecture
7. REMMS Programmers Guide
8. Atari Mutineer Doors Programmers Guide
9. Data And File Formats
10. Troubleshooting

NOTICES AND LICENSE

docs/mutineer/NOTICES.md

Notices And License

Atari Mutineer Project

(C)2026 Rick Collette (megalith)

Atari Mutineer BBS, REMMS, and the reference MMS sources are distributed under the repository license. The authoritative license text is the root LICENSE file.

The documentation, BASIC sources, REMMS sources, reference MMS sources, packaging scripts, and release artifacts should preserve this project notice unless a later license file explicitly changes it.

ATARI MUTINEER DOCUMENTATION

docs/mutineer/README.md

Atari Mutineer Documentation

The current runtime is REMMS 1.7 with session contract 9, resident pager and disk-backed text APIs, non-printable console controls, raw XMODEM transfer support, and full ASCII/ATASCII acceptance coverage.

These documents describe the current REMMS-based runtime. Generated build outputs and historical debugging notes are not documentation sources.

RDETACH closes REMMS-owned R: state and restores the exact original E: and R: HATABS entries while leaving K: untouched. Offline tools and the shutdown path therefore return to usable Atari BASIC; RUNBBS reinstalls REMMS when the board is started again.

Operators And Sysops

- Notices And License
- SysOps Quickstart
- Sysops Guide
- Altirra 850 Modem Reference
- Data And File Formats
- Troubleshooting

Developers

- Runtime Architecture
- Core Programmers Guide
- REMMS Programmers Guide
- Doors Programmers Guide
- Door Revamp
- REMMS/MOE Parity Check

Development Scope

The programmer guides describe how to add modules, doors, data files, menu assets, and REMMS-aware caller I/O. They intentionally do not include instructions for rebuilding Mutineer from source.

The packed runtime registry is D1:>DATA>APPREG.DAT.

ATARI MUTINEER SYSOPS GUIDE

docs/mutineer/MUTINEER_BBS_SYSOPS_GUIDE.md

Atari Mutineer Sysops Guide

This guide covers installing, initializing, and operating an Atari Mutineer BBS from an already-downloaded ATR disk image.

For file formats and current emulator settings, also see
Data And File Formats , Altirra 850 Modem ,
and Troubleshooting .

Install The ATR

Start with the Atari Mutineer ATR you downloaded:

```
atari-mutineer-v<VERSION>-16mb.atr
```

The 16MB release auto-boots and includes the BBS, documentation, working space, and all bundled doors in their configured D1: paths.

Keep a clean copy of the downloaded ATR before first boot. The BBS writes live user, message, file, and caller data to its disk, so your working copy becomes your live board image.

Altirra

1. Open Altirra.
2. Mount the Atari Mutineer ATR as drive D1:.
3. Boot the Atari with the disk mounted.
4. Let the auto-start sequence run.

Do not mount a Mutineer release ATR as D1: when you only want to inspect or repair it. Release ATRs auto-boot from D1:. For maintenance, boot another DOS or utility disk as D1: and mount the Mutineer ATR as a secondary disk.

Real Hardware

On real Atari hardware, use RespeQt, AspeQt, or another SIO disk utility to mount the ATR as D1: or copy the ATR contents onto a physical floppy, hard disk, or other Atari-compatible storage device.

For a direct SIO mount:

1. Attach your SIO2PC/SIO2USB interface.
2. Open RespeQt, AspeQt, or your preferred SIO utility.
3. Assign the Atari Mutineer ATR to drive D1:.
4. Boot the Atari from D1:.

For a physical floppy or hard-disk installation:

1. Use an Atari disk-copy utility or host-side ATR utility that can copy files from the ATR to the target disk.
2. Preserve the ATR's root files and directory structure.
3. Make sure the target boots the same DOS environment as the ATR.
4. Boot the Atari from the target as D1:.

Do not copy only the BASIC programs. The BBS also needs its DOS startup files, REMMS resident, configuration, menus, user database, message base, file base, and door catalog files.

Boot Flow

The auto-start ATR uses:

```
AUTOEXEC.BAT -> RHND850.COM -> REMMS.COM -> BASIC /I <<D1:AUTOGO  
AUTOGO -> RUN "D1:>CORE>RUNBBS" -> APPMGR -> WFC
```

WFC waits for a carrier, prompts the caller to login or register, then loads CORE with the active caller handoff.

The local wait-for-call screen shows live time/date, call state, REMMS error bytes, and activity. Console keys on that screen are operator controls:

- START: local sysop main menu through CORE, with access level 9.
- SELECT: offline BBS maintenance through INIT.
- OPTION: shut down the BBS program.
- HELP: toggle local snooping for the current call.
- Shift+HELP: toggle persistent sysop availability.
- Ctrl+HELP: enter or leave sysop chat.

These controls do not have printable aliases, so caller and local input may use S, L, !, and every other printable character normally.

WFC detects ASCII versus ATASCII from the first return received after HIT, then resolves the matching assets through PATHREG.DAT. Callers can use the A command after login to toggle the text mode.

The BASIC prompt routines handle backspace/delete according to the selected terminal mode. REMMS handles carrier state, byte routing, and newline mirroring.

For maintenance without auto-starting the board, boot another DOS or utility disk as D1: and mount the Mutineer ATR as D2: or another secondary drive. Then use SDX copy and directory commands against the secondary drive.

First User

The first registered user is user number 1 and access level 9. Later self-registered users are access level 1.

Registration is handled by LOGIN. It asks for:

- Full name.
- Handle.
- Email address.
- Password.
- Password confirmation.

Before the account is created, LOGIN displays a confirmation screen with the entered full name, handle, email, and masked password. The caller can create the account, edit the entered fields, or cancel registration.

The P command on the main menu opens profile editing. Callers can edit their own full name, email, and password. Level 9 sysops can also load another user and change handle, status, or access level. LOGIN rejects duplicate handles and rewrites the user record and index together.

Passwords are masked with * at the remote terminal.

Access Levels And Time

ACL.CFG controls session and daily limits:

```
# Level, sestime, daytime
0,0,0
1,30,90
2,45,120
3,60,180
4,90,300
5,120,300
6,300,600
7,300,600
8,300,600
9,999,999
```

Level 0 is banned. Level 9 is the sysop level. WFC uses the RTIME8-compatible clock to apply daily/session limits.

Do not run complete initialization against a live board unless you have a backup. It recreates live board data.

Offline Initialization

Run INIT offline from the Atari console. It can recreate:

- CONFIG and CALLERS.DAT.
- USERS.DAT, USERS.IDX, and NEXTUSER.DAT.
- ACL.CFG.
- Default message base.
- Default file area.
- Default DOORS.CFG.

Use complete initialization only when you intend to reset the board data.

Message Bases

Message bases use MSG?.MET, MSG?.DAT, and MSG?.IDX.

Caller commands are inside the message module:

- Q: quick scan.
- S: summary.
- R: read messages.
- E: enter message.
- K: kill message.
- A/L: select or list bases.
- X: return to the main menu.

Run MSGMGR offline for sysop maintenance:

- Create/update bases.
- Post/read/list messages.
- Delete/undelete messages.
- Verify indexes and pointers.
- Rebuild/compact active messages.

File Areas

File areas use FILE?.MET, FILE?.DAT, and FILE?.IDX.

Caller commands are inside the file module:

- B, F, A, N: browse/list areas and files.
- D: download by file number or name.
- U: upload into quarantine for operator review.
- I/H/? : file-area information.
- X: return to the main menu.

Every caller upload is a held record under the area quarantine path. It is not listed or downloadable until approved. Run FILEMGR offline as a sysop or co-sysop (access level 8+) for maintenance:

- Create/update areas.
- Add file records.
- Review held uploads and approve them into the public upload path.
- Reject or restore records.
- Verify physical file paths.
- Rebuild/compact active records.

Doors

Door catalog source rows live in DOORS.CFG:

```
# #, Door Name, filename, path, level
1, Blackjack, BLKJACK, D1:>DOORS>BLKJACK>, 2
2, Dumjin, DUMJIN, D1:>DOORS>DUMJIN>, 2
3, Arena, ARENA, D1:>DOORS>ARENA>, 2
```

Door paths point into the bundled D1:>DOORS> tree. The BBS does not parse DOORS.CFG while callers are online; package/offline tooling compiles it into D1:>DATA>DOORCAT.DAT and matching door menus. If a caller selects a door whose disk/path is absent, the BBS reports DOOR UNAVAILABLE. and returns to the door menu.

For example, Dumjin is installed at:

```
D1:>DOORS>DUMJIN>
```

After changing door rows, rebuild the runtime catalog before taking calls. Door installation is an offline sysop activity.

The 16MB Mutineer ATR includes all bundled doors. Blackjack and Dumjin are REMMS doors, and Arena's caller module is REMMS-compatible as well. Arena's two overlays are local tournament setup tools.

Text Assets

Menus and static pages are included on the ATR. To customize them on a running board, edit the live copies on your working disk image or installed Atari disk, then back up that disk before making more changes.

Keep ASCII and ATASCII copies consistent when customizing a live board. ASCII and native ATASCII asset roots are configured in PATHREG.DAT; the packaged defaults are D1:>ASCII> and D1:>ATASCII>.

Caller Command Surface

Main menu commands in CORE:

- ?: display main menu.
- B: bulletin menu.
- H: help text.
- W: welcome text.
- X: toggle expert prompt mode.
- M: enter message area.
- T: display text file 1.
- E: leave feedback.
- Y: sysop chat path. When D1:>DATA>SYSOP.CFG is 0, callers see

SYSOP NOT AVAILABLE. and stay in CORE. When enabled, the local console rings three times at five-second intervals. Press the Atari START console key during that window to answer and enter line-mode SYSCHAT; the caller can cancel with Ctrl-X. Otherwise the caller is told the SysOp is unavailable but has been notified.

- P: edit the current caller profile.
- Q: quick logoff; immediately disconnects.
- G: confirmed goodbye; optionally saves feedback, displays OBBS, then disconnects.
- L: toggle line-feed mode.
- A: toggle ASCII/ATASCII mode.
- C: caller list.
- D: door menu.
- F: file area.
- I: info text.

Message, file, and door subcommands are documented in Data And File Formats .

Testing A Board

For an emulator-hosted board, connect to the modem port exposed by Altirra. An interactive telnet client uses the named Altirra-850-interactive-telnet.ini profile with negotiation enabled. In PuTTY use Telnet and set local echo to Auto or Force off, never Force on. The default address is:

```
telnet 127.0.0.1 6507
```

Useful checks:

- Confirm HIT appears.
- Register or login.
- Confirm Loading CORE ... appears during module changes.
- Test ?, M, F, D, Y, Q, and G.
- Hang up and reconnect without restarting Altirra.

Backups

Before resetting data or replacing an ATR, keep a copy of:

- User files.
- Caller history.
- Message base files.
- File area metadata.
- Door catalog.
- Customized text assets and config files.

A newly downloaded ATR contains baseline data files. Preserve live board data separately before replacing the deployed image or reinstalling onto new media.

ALTIRRA 850 MODEM REFERENCE

docs/mutineer/ALTIRRA_850_MODEM.md

Altirra 850 Modem Reference

Mutineer expects this runtime path:

Atari BASIC -> REMMS E:/K: wrappers or raw API -> R:
-> RHND850.COM -> emulated Atari 850 -> Altirra modem -> host TCP

Use Altirra 4.40.200.17000. The supported executable path is
D:\ALTIRRA\Altirra64.exe.

System Profile

Area Setting
--- ---
System XL/XE Computer
Video NTSC
Memory 64K XL/XE memory
CPU 6502C, normal clock, illegal instructions allowed
BASIC Enabled
Boot Fast boot enabled
Keyboard Present, raw mode off, full raw mode off, function keys off
Pause Pause when inactive off

Disk and cartridge setup:

Area Setting
--- ---
D1: Mutineer release ATR
D2:-D8: Optional utility, backup, or data disks
Cartridge SpartaDOS X cartridge
Disk SIO patch Enabled
Disk burst transfers Off

Device Setup

Device option Setting
--- ---
R-Time 8 Enabled
Atari 850 Enabled
Atari 850 emulation level Full, not minimal/HLE
Atari 850 extended baud Enabled
CIO R: patch Off
CIO H: patch Off
CIO P: patch On
CIO T: patch Off
Device CIO burst transfers On
Device SIO burst transfers Off
Device SIO patch Off
Accelerate with SIO patch Off
Accelerate with PBI patch On

The 850 must use Full emulation. REMMS uses the 850 status/sense path to distinguish active carrier, lost carrier, timeout, and fatal serial errors. Minimal or HLE 850 emulation is not a supported BBS runtime.

Modem Setup

Configure the modem as a child device under the Atari 850.

Modem option Supported value
--- ---
Mode Inbound/listen
Address 127.0.0.1
Port 6507
IPv6 Off

```
| Network simulation | None |
| Unthrottled | Off |
| Connect rate | 9600 |
| Check rate | Off |
| Outbound dialing | Off |
| Telnet negotiation | Off for raw socket clients, on for PuTTY/Telnet clients |
| Telnet LF translation | Off |
```

Connect callers to:

127.0.0.1:6507

For PuTTY, select Telnet, connect to 127.0.0.1 port 6507, set Terminal > Local echo to Auto or Force off, and leave local line editing off. Do not use local echo Force on because password fields are already masked by Mutineer.

Portable Profile Values

The known-good interactive profile is tools/profiles/Altirra-850-interactive-telnet.ini . It fixes the listener at 127.0.0.1:6507, enables Altirra Telnet negotiation, disables LF rewriting, and keeps outbound connections disabled. In short, the interactive profile uses telnet=true and telnetlf=false.

The key device records are:

```
"850" = "{\"baudex\":true,\"emulevel\":2}"
"modem" = "{\"ipv6\":false,\"port\":6507,\"netmode\":\"none\", \"unthrottled\":false,\"connect_rate\":9600,\"check_rate\":false,\"telnet\":true,\"outbound\":false,\"telnetlf\":false}"
```

For raw-socket callers, use the same modem record with "telnet":false.

The relevant profile flags are:

```
"Hardware mode" = 1
"Memory mode" = 4
"CPU: Allow illegal instructions" = 1
"BASIC enabled" = 1
"Kernel: Fast boot enabled" = 1
"Disk: SIO patch enabled" = 1
"Disk: Burst transfers enabled" = 0
"Devices: CIO R: patch enabled" = 0
"Keyboard: Raw mode" = 0
"Keyboard: Full raw mode" = 0
"Keyboard: Enable function keys" = 0
```

Baud And R: Configuration

REMMS opens R: on reserved IOCB 5, applies XIO 38/34/36, and enters concurrent mode with XIO 40. The current connected-data baud selector is 14, matching the default 9600-baud Altirra modem profile. The wait-for-call command selector is 15.

Do not enable Altirra's CIO R: patch. The packaged RHND850.COM handler and full 850 emulation are part of the supported serial path.

Operating Notes

- Boot the release ATR as D1:.
- Use the full ATR when you want bundled doors installed on the same image.
- Close stale Altirra instances before starting another BBS run so the listener port is not shared.
- Keep Telnet LF translation off. Mutineer and REMMS own CR/LF handling.

- Keep terminal local echo off or automatic. Mutineer echoes accepted input.

ATARI MUTINEER CORE PROGRAMMERS GUIDE

docs/mutineer/MUTINEER_BBS_CORE_PROGRAMMERS_GUIDE.md

Atari Mutineer Core Programmers Guide

Atari Mutineer is a set of standalone Atari BASIC programs loaded with RUN. REMMS owns communications and session state so the current program can be discarded without losing caller identity or connection state.

See Runtime Architecture for the end-to-end flow,
REMMS Programmers Guide for the resident API, and
Data And File Formats for disk layouts.

Runtime Modules

```
RUNBBS -> APPMGR -> WFC -> APPMGR -> LOGIN -> APPMGR -> CORE
CORE -> APPMGR -> DOORS / MSGAREA / FILEBASE / SYSCHAT / XFERSEL
```

Module	Responsibility
---	---
RUNBBS	initialize REMMS and request WFC
APPMGR	resolve stable key, validate access, prepare route, and RUN
WFC	answer calls, terminal detection, local dashboard
LOGIN	login, registration, registration confirmation, and profile editing
CORE	main caller command loop and subsystem dispatch
MSGAREA	multi-base scan/read/enter/kill
FILEBASE	multi-area list/download/upload/info
DOORS	catalog lookup, access check, and external door launch
SYSOP	sysop availability/config actions
SYSCHAT	line-mode caller/sysop chat
XFERSEL	transfer protocol selection and native protocol dispatch
INIT	offline board initialization
FILEMGR	offline file-area maintenance
MSGMGR	offline message-base maintenance

LOGIN is the supported login, registration, and user profile implementation.

Common BASIC Routines

Reusable routines live under source/text/inc/. A module includes one at a physical line-number base:

```
2005 INCLUDE "console_open_tail.inc"
```

Include files use strictly increasing relative lines beginning at 0. The preprocessor adds the directive base to every included line and to branch targets that name another line in the same include. Targets not declared by the include remain absolute, so calls to module-owned routines keep their normal physical numbers. Includes cannot include other includes.

Keep module declarations, menu policy, error destinations, and domain logic in the owning .bas file. Use a shared include only when its variable and error contracts match every consumer. Keep shared routines small and explicit so a module's line ownership remains easy to inspect.

Application Registry

The packed runtime registry is D1:>DATA>APPREG.DAT.

Rows contain:

handle, key, return, type, flags, access, route, term, name, path

Current stable keys:

Key	Name	Default	return
\$0001	WFC	\$0000	
\$0008	LOGIN	\$0001	
\$0010	CORE	\$0001	
\$0020	DOORS	\$0010	
\$0030	MESGAREA	\$0010	
\$0040	FILEBASE	\$0010	
\$0045	SYSCHAT	\$0010	
\$0050	SYSOP	\$0010	
\$0060	XFERSEL	\$0040	

Modules dispatch by writing the requested key to session +90/+91, the optional return key to +92/+93, saving owned session state, and running D1:>CORE>APPMGR.

When adding a module, reserve a stable key, decide the return key, define the minimum access level, and decide whether APPMGR should enter it as a caller module or a local-only tool. Caller modules inherit the current terminal mode, line-feed mode, user identity, access level, time limit, and online state from the REMMS session block.

Online Module Contract

Every supported online module:

1. Discovers API, public state, and session through \$03E8.
2. Validates session magic 77 and supported version 1..3.
3. Restores identity, modes, access, user number, session deadline, and any module-specific byte.
4. Calls USR(ROPEN) for a remote session.
5. Sets mode host and route both; local sessions set both to local.
6. Opens caller-facing E: and K: channels as needed.
7. Calls USR(RSTAT) at input boundaries and returns to WFC if session +87 is no longer online.
8. Saves owned fields before another RUN.

Do not assume BASIC variables or IOCBs survive RUN. Do not parse RING, CONNECT, or NO CARRIER in BASIC; that belongs to REMMS.

Module Design Checklist

Use this checklist for a new caller-facing module:

- Read RAPI, RPUB, and SS from the discovery record.
- Validate PEEK(SS)=77 and PEEK(SS+1)=6.
- Restore only the session fields the module owns.
- If PEEK(SS+87)=1, call USR(ROPEN,0) and then USR(RMODE,1):USR(RROUTE,2).
- If PEEK(SS+87)=8, treat it as a local sysop session and use local route.
- Open caller-facing E: and K: channels after route/mode is set.
- Use a timed prompt in the same shape as the other modules:
[HH:MM:SS Command >, [HH:MM:SS Msg Command >, or
[HH:MM:SS File Command >.
- Before disk reads or writes during a remote call, suspend the modem owner with USR(RCLOSE,1), perform bounded disk work, close disk IOCBs, and resume with USR(ROPEN,0).
- Call USR(RSTAT) or USR(REVENT) at blocking input boundaries.
- On carrier loss or fatal REMMS status, call USR(RFAIL,reason) and run WFC.
- Before dispatching another module, store caller identity, access, time, modes,

and module-specific handoff bytes back to SS.

Avoid global side effects. A module should not clear all 192 session bytes, change the answer-rings policy, close R: with hangup semantics, or alter terminal mode unless that is the explicit command being handled.

Prompts And Menus

Menus are ordinary text files and are reloaded whenever a parent command loop regains control. Prompts are generated by the module so they can include time-left and the right command noun. Do not embed or cache menu text in BASIC.

Main prompt:

```
[ HH:MM:SS Command >
```

Message prompt:

```
[ HH:MM:SS Msg Command >
```

File prompt:

```
[ HH:MM:SS File Command >
```

Door prompt:

```
[ HH:MM:SS Door Command >
```

HH:MM:SS is the caller's remaining session time. Local sysop sessions may use the same prompt shape with unrestricted or configured local time.

Use the resident pager for long caller-facing displays. Use direct output only for short status lines, prompts, and command responses.

Session Fields Used By BASIC

Offset	Meaning
---	---
+0/+1	magic/version
+2..+42	handle length/text
+43..+83	secondary identity length/text
+84	ATASCII mode
+85	line-feed mode
+86	expert/transfer mode
+87	1 remote online, 8 local sysop handoff
+90..+93	requested/return stable keys
+96	transient module or transfer mode
+97	current message/file/door handoff byte
+102	current runtime app handle
+122	delayed remote byte
+123	access level
+124/+125	user number high/low
+126/+127	session end minute high/low
+155/+156	exit reason and pending-stat flag
+159/+160	answer-after rings and ring count
+161/+162	next/return runtime handles
+163..+191	transfer pathname length/text

Many modules clear all 192 bytes before constructing a new handoff. When adding a field, update remms/src/constants.inc, every store/restore routine that owns it, the contract checker, and the REMMS documentation together.

REMMS Module Skeleton

```
1000 IF PEEK(1000)<>82 OR PEEK(1001)<>69 OR PEEK(1002)<>77 THEN RUN "D1:>CORE>WFC"
1010 RAPI=PEEK(1004)+256*PEEK(1005):RPUB=PEEK(1006)+256*PEEK(1007):SS=PEEK(1010)+256*PEEK(1011)
```

```

1020 RMODE=RAPI+3:RROUTE=RAPI+6:ROPEN=RAPI+9:RSTAT=RAPI+15:RGET=RAPI+18:RPUT=RAPI+21:RTGET=RAPI+27
1030 IF PEEK(SS)<>77 OR PEEK(SS+1)<1 OR PEEK(SS+1)>3 THEN RUN "D1:>CORE>WFC"
1040 IF PEEK(SS+87)=1 THEN R=USR(ROPEN):IF R<>0 THEN RUN "D1:>CORE>WFC"
1050 IF PEEK(SS+87)=1 THEN R=USR(RMODE,1):R=USR(RROUTE,2)
1060 OPEN #3,12,0,"E":OPEN #4,4,0,"K:"
1070 R=USR(RSTAT):IF PEEK(SS+87)<>1 THEN RUN "D1:>CORE>WFC"

```

For text input, call RTGET; retry on \$0100 (256) and hand carrier/fatal/timeout results to RFAIL.

Use raw RGET/RPUT only for terminal detection and binary protocols. Text editors use canonical ATASCII 7E for destructive backspace and 7F for a one-space tab, and ignore unsupported extended editing controls.

Return Paths

Use APPMGR for normal module switches. Write the requested app key to SS+90/+91, write the desired return key to SS+92/+93, save module state, and run D1:>CORE>APPMGR.

Use direct RUN "D1:>CORE>WFC" only after explicit logoff, local shutdown, carrier loss, fatal REMMS failure, or an unrecoverable application failure. Use direct RUN "D1:>CORE>CORE" only for local-only maintenance flows that have already restored their session state.

Disk Ownership Pattern

Remote modules should follow this shape around disk work:

```

2600 IF ONLINE<>1 THEN RETURN
2610 R=USR(RCLOSE,1):RETURN
2620 IF ONLINE<>1 THEN RETURN
2630 R=USR(ROPEN,0):R=USR(RMODE,1):R=USR(RROUTE,2):RETURN

```

Open disk files only between the suspend and resume calls. For long displays, read a chunk into a string, close the disk file, resume R:, emit the buffered text, and repeat. Do not wait for caller input while a disk IOCB is open.

CORE Command Surface

CORE currently dispatches:

- A: ASCII/ATASCII toggle.
- B: bulletins.
- C: caller list.
- D: doors.
- E: feedback.
- F: filebase.
- G: confirmed goodbye.
- H or ?: help/function list.
- I: information.
- L: line-feed toggle.
- M: message area.
- Q: quick logoff.
- T: text files.
- W: welcome.
- X: expert prompt toggle.
- Y: sysop chat/fallback.

WFC remote entry offers login, new user, and goodbye. Its local controls are START for local sysop CORE, SELECT for offline tools, and OPTION for shutdown.

Persistence

- Users: USERS.DAT, USERS.IDX, NEXTUSER.DAT, ACL.CFG.
- Caller history: CALLERS.DAT.
- Feedback: FEEDBACK.DAT.
- Messages: MSGBASE.CFG and MESG?.MET/.DAT/.IDX.
- Files: FILE?.MET/.DAT/.IDX.
- Doors: offline DOORS.CFG, runtime DOORCAT.DAT, generated DOORMNU, and the configured external disk/path.

INIT, MSGMGR, and FILEMGR own offline creation, verification, rebuild, and compaction. Runtime modules should preserve existing record sizes and pointer semantics described in Data And File Formats .

Current Implementation Notes

- The active runtime registry is APPREG; APPMGR reads bounded packed records.
- REMMS wraps and chains E:, K:, and R:. Local E:/K: calls go directly to the saved handlers; IOCB 5 is reserved for R:.
- Page-six helpers load menu data in blocks and emit the local copy with one CIO buffer operation. Remote bytes still use the normal REMMS translation and carrier-failure path.
- CORE, MSGAREA, FILEBASE, DOORS, MSGMAIL, and MSGMOD reload their packaged text menu on every parent-loop return. DOORS uses the fixed-record door catalog instead of parsing CSV while callers wait.
- Terminal detection is performed by WFC from the first caller return before normal line input.
- Every bundled door uses the REMMS API and the canonical text-input contract.

RUNTIME ARCHITECTURE

docs/mutineer/RUNTIME_ARCHITECTURE.md

Runtime Architecture

Atari Mutineer uses standalone Atari BASIC programs connected by a REMMS-owned resident session block. A BASIC RUN discards variables and closes IOCBs; resident state and disk files carry the caller across module switches.

Boot Flow

```
SpartaDOS X
-> RHND850.COM
-> REMMS.COM
-> BASIC /I <<D1:AUTOGO
-> RUNBBS
-> APPMGR
-> WFC
```

REMMS installs at \$1E00, publishes discovery at \$03E8, initializes its 352-byte session block, and starts in local mode with its E: and K: wrappers installed. The wrappers directly chain the validated saved handlers for local operations; route or carrier changes do not rewrite HATABS.

Only RUNBBS, APPMGR, PATHREG.DAT, PATHREG.BAK, and the staged PATHREG.NEW update target have fixed bootstrap paths. Runtime menus, static assets, common config/data files, scheduler files, transfer spool files, and global network queue/config files are classified through the compact path registry. Application, door, message, and file locations remain owned by their specialized registries or metadata.

RUNBBS calls the API initializer, writes stable key \$0001 (WFC) into the session block, and runs APPMGR.

Wait For Call

WFC opens REMMS host mode and keeps output local while waiting:

```
RING -> answer-ring counter -> ATA -> CONNECT -> online=1 -> route both
```

The local dashboard consumes REVENT, which owns console, keyboard, modem, remote-input, carrier, and error polling:

- START: local level-9 sysop session in CORE.
- SELECT: offline initialization/maintenance.
- OPTION: shut down BASIC.

On a remote connection WFC detects ASCII versus ATASCII from the first return, performs login or registration, appends caller history, commits session state, and requests CORE through APPMGR.

Application Dispatch

```
RUNBBS -> APPMGR -> WFC -> APPMGR -> CORE
CORE -> APPMGR -> DOORS / MESGAREA / FILEBASE / SYSCHAT / SYSOP / XFERSEL
```

Modules write a requested stable key to session +90/+91 and an optional return key to +92/+93, then RUN "D1:>CORE>APPMGR". APPMGR resolves APPREG.DAT through PATHREG, validates the registered minimum access level,

updates runtime handles, and runs the registered path. Loader CLOSE records RESUME_PENDING, and a BASIC destination performs exactly one non-hangup ROPEN,RESUME. Native transfer workers are the exception: APPMGR loads and validates the COM file while suspended, then performs one guarded ROPEN,RESUME immediately before calling the worker because there is no intervening BASIC destination.

APPMGR bulk-loads and checksum-validates the packed, key-sorted registry once per invocation, then resolves destination and return keys in memory. The trusted FILEBASE/FILEWORK overlay pair and bundled doors may return directly after validating the REMMS session and updating current/next/return handles; invalid sessions fall back to APPMGR.

Change Area is canonical in resident FILEBASE because it has the same configured-ID and access checks as the former FILEWORK copy without two BASIC reloads. FILEWORK contains only operations that need the worker overlay.

Session Ownership

The canonical layout is in remms/src/constants.inc and documented in the REMMS Programmers Guide . Important fields are:

- identity strings at +2..+83;
- terminal modes at +84..+86;
- online state at +87;
- app keys at +90..+93;
- module/transfer state at +96/+97;
- current app at +102;
- access/user/time fields at +123..+127;
- answer-rings state at +159/+160;
- XFR1 transfer control block at +163..+337;
- native message MSG1 control block at +340..+351;
- unread count/cache validity and scheduler job state at +317..+322.

Online modules validate magic/version, restore their session fields, reopen REMMS host state, set route both, reopen caller-facing E:/K: channels, and check carrier at input boundaries. Local modules use mode/route local.

Caller I/O

- USR(REVENT) performs the foreground event-pump pass.
- USR(RGET) returns one exact raw remote byte, \$0100 no data, \$0101 carrier loss, \$0103 fatal I/O, or \$0104 timeout.
- USR(RTGET) returns one canonical text byte or the same structured result.
- USR(RPUT,byte) transmits one raw byte.
- Wrapped E: output is local, remote, or mirrored according to route.
- Wrapped E:/K: input uses remote bytes only while session +87 is online and shares the RTGET translation path; otherwise calls chain to the saved local handlers.
- Text modules use canonical ATASCII (9B EOL, 7E backspace, 7F tab).
- Native protocol binaries use the REMMS transfer service table so terminal newline translation does not corrupt binary blocks.

REMMS preserves the SIO DCB around CIO operations and keeps its validated E:/K: wrappers installed. Reinstall safely refreshes or chains a changed HATABS entry instead of switching handlers according to route.

Disk And Modem Ownership

REMMS reserves IOCB 5 for R:. Caller-facing BASIC modules never perform disk I/O while that physical modem owner is active: they request a non-hangup suspend with RCLOSE,1, read or write a bounded disk chunk, close the disk channel, and resume with ROPEN,0. Long caller lists, menus, message/file lists, and text downloads buffer data between those boundaries before sending it through the resident pager. Text uploads close the modem only for the brief append and never hold a disk file open while waiting for caller input.

External transfers use the same ownership boundary for disk blocks. FILEBASE builds an XFR1 request, XFERSEL selects the protocol, and APPMGR binary-loads the native protocol from D1:>PROTO>. Protocol binaries call REMMS native transfer services and never own R: directly.

Runtime Path Ownership

Direct D1: literals in BASIC must be one of:

- fixed bootstrap/app-dispatch paths;
- paths resolved through PATHREG;
- APPREG/DOORCAT registered targets;
- message-base paths derived from MSGBASE.CFG and MESG?.MET;
- file-area paths derived from FILE?.MET;
- native worker handoff, lock, journal, swap, or queue paths documented in the relevant data-format section.

tools/runtime_path_check.py enforces this classification for BASIC sources.

Permission Model

The runtime permission model is compact:

- app access comes from APPREG minimum level;
- door access comes from DOORCAT minimum level;
- message access comes from MESG?.MET read/post levels plus command gates;
- file access comes from FILE?.MET view/download/upload levels and protocol policy;
- profile/user-edit authority comes from LOGIN and USERS.DAT.

The deferred full per-level matrix design is not parsed by caller modules.

tools/permission_model_check.py enforces the compact model.

Carrier And Cleanup

REMMS 1.7 distinguishes confirmed carrier loss, negative modem CIO status, timeout, recoverable FIFO conditions, and BASIC/application failure. Fatal paths call RFAIL, preserve the first operation and CIO status, end text input, reconcile the physical owner, and record one exit/event transition. BASIC enters WFC only after the device is closed or explicitly faulted; deliberate APPMGR transitions retain the tracked-resume policy.

Before offline tools or shutdown, WFC calls RDETACH. This closes owned modem state and restores the exact saved E: and R: handlers while leaving K: alone. A later RUNBBS entry calls RINIT to reinstall REMMS.

Normal logoff closes REMMS, which performs the guarded modem hangup sequence, closes the IOCB, clears call parser state, and restores local operation.

Module Responsibilities

- RUNBBS: initialize REMMS and request WFC.

- APPMGR: registry lookup, access check, route preparation, and RUN.
- WFC: dashboard, answer/login/register, local controls, caller persistence.
- CORE: main commands, text, feedback, callers, dispatch, and logoff.
- MESGAREA: multi-base scan/read/enter/kill and message-service dispatch.
- MSGEDIT, MSGMAIL, MSGMOD: compose, private-mail, and moderation UIs; they stage requests but do not write live message storage.
- MSGSTORE.COM: transient lock-guarded writer and storage authority.
- MSGRCV.COM, MSGCMP.COM: transient recovery and compaction workers.
- SCHED: idle WFC job dispatcher for verification, networking, and conditional compaction.
- FILEBASE: multi-area list/download/upload/info and XFERSEL dispatch.
- DOORS: access-filtered catalog and external door launch.
- SYSOP: remote caller/sysop chat fallback.
- XFERSEL: transfer protocol selection and native protocol dispatch.
- INIT, FILEMGR, MSGMGR: initialization and fenced maintenance launchers.

The supported login/registration path remains in WFC.

REMMS PROGRAMMERS GUIDE

docs/mutineer/REMMS_PROGRAMMERS_GUIDE.md

REMMS Programmers Guide

REMMS is the supported resident modem, input, output-routing, and session layer for Atari Mutineer. It is not an MMS ABI clone. BASIC code discovers REMMS dynamically and calls the published jump table.

REMMS.COM loads at \$1E00, installs through INITAD, publishes a discovery record, and raises MEMLO so Atari BASIC modules can call it safely.

Discovery Record

The loader publishes this record at \$03E8:

Offset	Contents
---: ---	
+0..+2	ASCII signature REM
+3	not written by REMMS; owned by the 850 R: handler, see below
+4..+5	API jump-table address, little-endian
+6..+7	public-state address
+8..+9	resident-end address
+10..+11	session-block address
+12..+13	native transfer service table
+14..+15	session transfer block (SS+163)

Atari BASIC therefore reads the API pointer from decimal 1004/1005, the public block from 1006/1007, and the session block from 1010/1011.

Byte +3 (\$03EB) is the XL OS CARTCK location. REMMS does not own it and never writes it: the 850 R: handler, relocated just above REMMS at MEMLO, uses it as working storage and rewrites it thousands of times per call. The publish loop skips it. The remms-lifecycle case `discovery_after_connect` traces every write to the other discovery bytes under the Altirra debugger and watches REMMS's only store into the record (`sta REMMS_DISCOVERY,x`) for index 3. It fails if REMMS writes \$03EB, if anything but REMMS writes another discovery byte, or if the signature, pointers or `RPUB+5/+6` change after a connected call. The other discovery bytes are XL OS boot-time variables (`SUPERF`, `CKEY`, `CASSBT`, `DERRF`, `ACMVAR`) that nothing writes once DOS is running; the layout is inherited from the MMS memory map. Read the REMMS version from the public block (`RPUB+5` major, `RPUB+6` minor).

BASIC PEEK Reference

Discovery record at decimal 1000 (\$03E8):

Address	Meaning	Expected value
---: --- ---		
PEEK(1000)	signature byte 0	82 (R)
PEEK(1001)	signature byte 1	69 (E)
PEEK(1002)	signature byte 2	77 (M)
PEEK(1003)	not REMMS (850 R: handler scratch)	do not use; read <code>RPUB+5</code>
PEEK(1004)+256*PEEK(1005)	API jump table	assign to <code>RAPI</code>
PEEK(1006)+256*PEEK(1007)	public block	assign to <code>RPUB</code>
PEEK(1008)+256*PEEK(1009)	resident end	diagnostics only
PEEK(1010)+256*PEEK(1011)	session block	assign to <code>SS</code>

Public block offsets from `RPUB`:

Offset	PEEK	Meaning
---: --- ---		
+0..+4	signature	REMMS
+5	major	1

+6	minor	7
+7	mode	0 local, 1 host
+8	route	0 local, 1 remote, 2 both
+9	R: open flag	0 closed, nonzero open
+10	last CIO status	raw Atari CIO status byte
+11	last REMMS error	see error table below
+12	installed flag	nonzero after handlers installed
+13	last CONSOL value	raw console register sample
+16..+21	console/debug bytes	raw/current/last/poll/CH/ATACHR
+22/+23	event sequence	increments on recorded events
+24	physical state	see physical-state table
+25	connection state	see connection-state table
+26	last operation	see operation table
+27	last CIO status mirror	most recent device status
+28	transition	0 none, 1 run, 2 resume, 3 close, 4 suspend

Session block offsets from SS:

Offset	PEEK/POKE	Meaning
---:	---	---
+0	magic	77
+1	version	6
+2	name length	0..40
+3..+42	name bytes	caller handle
+43	address length	0..40
+44..+83	address bytes	secondary identity/email
+84	AMODE	0 ASCII translation, 1 native ATASCII
+85	LMODE	0 CR only, 1 CR/LF in ASCII output
+86	XMODE	expert/transfer flag used by BASIC
+87	online state	1 remote online, 8 local sysop handoff
+88	snoop state	0 remote-only output, 1 local and remote output
+90/+91	request key	low/high app key for APPMGR
+92/+93	return key	low/high app key for APPMGR
+96	transfer/module mode	module-owned scratch
+97	call byte	message/file/door handoff byte
+102	current app	runtime app handle
+103..+106	pending stats	messages/uploads/downloads/calls
+122	delayed byte	REMMS/BASIC handoff scratch
+123	access level	0..9
+124/+125	user number	high/low
+126/+127	end minute	high/low absolute day minute
+128..+138	pager config/state	enabled, rows, flags, counters
+139	physical state	mirror of resident physical owner
+140	connection state	local/wait/ring/answer/connect/lost/error
+141/+142	last event	kind/value
+143	CIO status	last status byte
+144	text mode	0..5, see text table
+145	text state	input field state/counter
+146	console gate	local console de-bounce
+147/+148	event sequence	low/high
+149	FIFO overflow	nonzero after overflow
+150	open policy	0 resume, 1 fresh WFC
+151	transition	run/resume/close/suspend telemetry
+152	last operation	open/status/get/put/etc.
+153	text choice	one-byte choice buffer
+154	WFC action	local console action latch
+155	exit reason	carrier/fatal/timeout/application
+156	stats pending	nonzero tells WFC to run WFCSTAT
+157	config profile	none/wait-call/connected
+158	ownership sequence	increments on physical owner changes
+159	answer rings	1..9
+160	ring count	current pre-answer count
+161/+162	next/return app handles	runtime handles
+163..+337	XFR1 transfer block	external transfer request/result
+340..+351	MSG1	native message operation/result control block
+317..+322	message/job cache	unread count, idle minutes, job state/result

BASIC USR Reference

REMMS 1.7 keeps every older API offset stable. Each entry is three bytes:

Offset	Common name	BASIC call	Arguments	Return
---:	---	---	---	---
+0	RINIT	USR(RAPI)	none	0 OK, error code on failure

+3	RMODE	USR(RAPI+3,mode)	0 local, 1 host 0 OK, 5 bad arg, other error
+6	RROUTE	USR(RAPI+6,route)	0 local, 1 remote, 2 both 0 OK, 5 bad arg, other error
+9	ROPEN	USR(RAPI+9,policy)	0 resume, 1 fresh WFC 0 OK, error code on failure
+12	RCLOSE	USR(RAPI+12,kind)	0 hangup close, 1 suspend for disk I/O 0 OK, error code on failure
+15	RSTAT	USR(RAPI+15)	none current connection state/error code
+18	RGET	USR(RAPI+18)	none byte 0..255 or \$010x structured result
+21	RPUT	USR(RAPI+21,byte)	raw byte 0..255 0 OK or \$010x structured result
+24	RCON	USR(RAPI+24)	none console action 0..6
+27	RTGET	USR(RAPI+27)	none canonical text byte or \$010x structured result
+30	PBEGIN	USR(RAPI+30)	none 0 OK or error code
+33	PPUT	USR(RAPI+33,byte)	canonical display byte 0 OK or error code
+36	PEND	USR(RAPI+36)	none pager result code
+39	RRESET	USR(RAPI+39)	none 0 OK, 5 if physical owner busy
+42	REVENT	USR(RAPI+42)	none event_kind*256+value
+45	TBEGIN	USR(RAPI+45,text_mode)	1 line, 2 masked, 3 choice, 4 translated, 5 editor 0 OK or 5 bad arg
+48	TEND	USR(RAPI+48)	none 0 OK
+51	RFAIL	USR(RAPI+51,reason)	4 carrier, 6 fatal, 7 timeout, 8 app 0 OK or \$0103 fatal cleanup
+54	RMSG	USR(RAPI+54)	none current connection state/error code
+57	RDETACH	USR(RAPI+57)	none 0 OK or close/restore error
+60	RCONCFG	USR(RAPI+60,snoop)	0 remote only, 1 local mirror 0 OK, 5 bad arg

RTEXT is the fixed page-six entry at \$0601 rather than a resident jump-table slot:

```
R=USR(1537,ADR(PATH$),LEN(PATH$),ADR(WORK$),FLAGS)
```

WORK\$ reserves 256 bytes. Its first 64 bytes hold the EOL-terminated path and the remaining 192 bytes are the streaming buffer. Flag bit 0 clears the display before output and bit 1 enables paging. Results are 0 success, 1 user abort, 2 invalid request, 3 open failure, 4 disk failure, and 5 output or carrier failure. The service preserves the current route, suspends the modem around each disk chunk, and resumes it before sending that chunk.

RDETACH ends resident text/pager activity, closes REMMS-owned R: state, restores the exact saved E: and R: HATABS pointers, leaves K: untouched, and returns routing to local. REMMS discovery and memory remain present; RINIT installs the handlers again.

Set the common names once after discovery:

```
1010 RINIT=RAPI:RMODE=RAPI+3:RROUTE=RAPI+6:ROPEN=RAPI+9:RCLOSE=RAPI+12
1020 RSTAT=RAPI+15:RGET=RAPI+18:RPUT=RAPI+21:RCON=RAPI+24:RTGET=RAPI+27
1030 PBEGIN=RAPI+30:PPUT=RAPI+33:PEND=RAPI+36:RRESET=RAPI+39
1040 REVENT=RAPI+42:TBEGIN=RAPI+45:TEND=RAPI+48:RFAIL=RAPI+51
```

Result And Enum Tables

Structured byte results returned by RGET, RTGET, and RPUT:

Decimal	Hex	Meaning
---	---	---
256	\$0100	no data available
257	\$0101	carrier lost
258	\$0102	recoverable I/O/FIFO condition
259	\$0103	fatal I/O
260	\$0104	timeout

REMMS error codes:

Code	Meaning
---	---
0	none
1	bad mode
2	bad route
3	missing handler
4	R: not open
5	bad argument
6	CIO failure
7	carrier lost
8	FIFO overflow
9	timeout

Event kinds returned by REVENT:

Kind	Meaning	Value byte
---	---	---
0	none	0
1	text	canonical text byte
2	console	1 START, 2 SELECT, 3 OPTION
3	call state	connection-state value
4	carrier loss	error/reason
5	recoverable	error/reason
6	fatal	error/reason
7	timeout	error/reason

Physical states:

Code	Meaning
---	---
0	closed
1	opening
2	open
3	closing
4	resume pending
5	fault

Connection states:

Code	Meaning
---	---
0	local
1	wait for call
2	ringing
3	answering
4	connected
5	lost
6	error

Text input modes for TBEGIN:

Mode	Meaning	Behavior
---	---	---
0	none	no text owner
1	line	editable line, empty return allowed
2	masked	editable password, echoes *
3	choice	one visible key followed by EOL
4	translated	translated byte stream without line editor
5	editor	canonical bytes without automatic echo; the application interprets cursor, insert, and delete keys

Editor mode leaves received editing controls out of the display stream. The application redraws accepted text explicitly, so ATASCII cursor controls and ANSI cursor sequences cannot be recorded as message-body bytes.

Operation codes stored in telemetry:

Code	Operation
---	---
0	none
1	OPEN
2	XIO 38
3	XIO 34
4	XIO 36
5	XIO 40
6	STATUS
7	GET
8	PUT
9	CLOSE
10	carrier STATUS/sense

RGET/RPUT return \$0100 no data, \$0101 carrier lost, \$0102 recoverable I/O, \$0103 fatal I/O, or \$0104 timeout. REVENT returns event_kind*256+value. RFAIL accepts reason 4 carrier, 6 fatal I/O, 7 timeout, or 8 application failure. It preserves a more specific resident reason, ends text input, and closes through the lifecycle owner.

Modes, Routes, And HATABS

The central invariant is: local mode never polls, reads, or writes R:.

- Local mode forces local output and restores the original E:/K: HATABS entries.
- Host-open begins in local-output mode while REMMS waits for a call.
- After CONNECT, REMMS marks session +87 online, selects route both, and installs its E:/K: wrappers.
- A remote-only or mirrored route is active only while session +87 is 1.
- Carrier loss or fatal modem I/O clears online/text state, closes R: through the physical owner, restores local mode/route, and records one pending event. Cleanup CLOSE failure leaves explicit PHYSICAL_FAULT.

The wrappers chain to the saved handlers directly. A local E: get first restores the saved E: entry in HATABS, because SDX's console handler resolves E: through HATABS again and would otherwise re-enter the wrapper. Every chained return reinstalls the wrappers. HATABS entries are found by device letter each time; the saved indexes are transient session bytes that a BASIC program switch clears. REMMS preserves the zero-page working range, DVSTAT, and the SIO DCB around the relevant CIO calls.

Call Lifecycle

Opening R: configures concurrency and resets the host parser.
USR(RSTAT) consumes modem result bytes in bounded batches and recognizes:

```
RING -> ATA -> CONNECT
```

Session +159 contains the answer-after-rings setting (1..9) and +160 contains the current ring count. REMMS sends ATA once when the configured count is reached. The parser stops on the complete CONNECT terminator and queues every already-available caller byte in order. RING, CONNECT, and pre-connect NO CARRIER parsing is disabled after connection. Connected carrier loss is sensed out of band through the 850 status/sense path; negative CIO status and timeout have distinct structured results and exit reasons.

Raw And Text Input

RGET (RAPI+18) and RPUT (RAPI+21) are byte-exact. WFC uses raw input for the first-return terminal probe. Native transfer protocols use the REMMS transfer service table published in discovery bytes +12..+15. Normal BBS and door input uses RTGET (RAPI+27). Remote E:/K: reads use the same translated path, so there is one text conversion implementation.

When session +84 is zero, the corrected MOE input table is:

```
| ASCII input | Canonical ATASCII |  
| ---: | ---: |  
| BEL 07 | bell FD |  
| BS 08 | backspace 7E |  
| TAB 09 | tab 7F |  
| LF 0A | space 20 |  
| VT 0B | 0B |  
| FF 0C | clear 7D |  
| CR 0D | EOL 9B |  
| 1C..1F | spaces |
```

Other bytes are unchanged. CRLF and CRNUL are returned as one EOL; a lone LF still maps to a space. Ctrl-C 03 and Ctrl-S 13 are discarded. When session +84 is nonzero, ATASCII bytes pass through unchanged except for those two

remote flow-control bytes.

The resident pager never uses printable letters as controls. At a pager prompt, Return advances one line, Space advances one page, Ctrl-Q shows the remainder, and Ctrl-C quits the listing. During output, Ctrl-S pauses, Ctrl-Q resumes, and Ctrl-C aborts. Printable S, C, and Q are preserved as ordinary typeahead for the next text input operation.

Standard line editors consume only the canonical values: 7E performs destructive backspace, 7F inserts one space, and unsupported clear, cursor, line insert/delete, and character insert/delete controls are ignored. Strict MOE semantics mean incoming ASCII DEL 7F is indistinguishable from ATASCII tab and therefore inserts one space; it is not a backspace alias.

Translated input is bounded by a 60-byte watermark in a 64-byte FIFO. A modem status snapshot of 60 bytes or more is not partially delivered: REMMS sends XOFF, clears the FIFO, rebuilds the concurrent stream, and returns recoverable event kind 5, value 8. Every BASIC field reader restarts from an empty field. This intentionally rejects an unpaced long paste rather than accepting a rotated or truncated line.

Text Output

Wrapped E: output is raw in ATASCII mode. In ASCII mode REMMS maps:

```
| ATASCII output | Remote bytes |
| ---: | --- |
| EOL 9B | CR, plus LF when session +85 is enabled |
| backspace 7E | BS 08 |
| tab 7F | space 20 |
| bell FD | BEL 07 |
| clear 7D | ANSI ESC[2J ESC[H |
```

The FD mapping intentionally fixes the duplicate-7D comparison in the MOE source. Binary output must bypass E: and use raw RPUT.

External transfer ownership

FILEBASE writes an XFR1 request into session +163..+337, XFERSEL selects a registered protocol, and APPMGR loads the native COM file from D1:>PROTO>. REMMS publishes the native transfer service table at discovery +12/+13 and the transfer block pointer at +14/+15. Protocol binaries use that table for raw bytes, binary-mode lifecycle, progress, cancellation, and result state.

Caller-facing BASIC must not overlap disk CIO with an active physical R: owner. Use RCLOSE,1, perform bounded disk I/O, close the disk channel, and then call ROPEN,0 before pager or prompt output. Buffer or chunk long displays; never leave a disk file open while waiting for caller input. RCLOSE,1 sends XOFF and keeps remote input paused across disk work. The next TBEGIN discards paused typeahead and sends XON only after the new field is ready. Callers must not substitute fixed sleeps for this handshake.

Console Commands

USR(RAPI+24) returns:

```
| Value | Physical input |
| ---: | --- |
| 0 | none |
| 1 | START or keyboard L |
| 2 | SELECT or keyboard S |
| 3 | OPTION or keyboard ! |
```

WFC maps these values to local sysop login, offline tools, and shutdown.

Console actions use only the physical console switches and XL/XE HELP latch:

Code	Key	Action
1	START	local sysop login
2	SELECT	offline tools
3	OPTION	shutdown to BASIC
4	HELP	toggle snooping for this call
5	Shift+HELP	toggle persistent sysop availability
6	Ctrl+HELP	enter or leave sysop chat

REMMS clears HELPPFG after consuming \$11, \$51, or \$91. It never reads ordinary CH or K: input for console control, and there are no printable aliases.

Session Block

REMMS owns a 352-byte block with magic 77 and session version 9.

Offset	Meaning
0..1	magic/version
2..42	caller handle length and text
43..83	secondary identity length and text
84..86	ATASCII, line-feed, and transfer/expert modes
87	online state: 1 remote, 8 local-console handoff
88	current snoop state
90..93	requested and return stable app keys
96	transfer/module mode
97	current message/file/door handoff byte
102	current application handle
123	access level
124..125	user number, high/low
126..127	session end minute, high/low
139..154	physical/connection/event/CIO/text/transition telemetry
155..156	exit reason and pending-stat flag
157..158	configuration profile and ownership sequence
159..160	answer-rings policy and current ring count
161..162	next/return runtime app handles
163..337	XFR1 transfer / FST1 file-store workspace
317..322	unread-count and scheduler fields while transfer is inactive
338..339	transfer mark-session token
340..351	MSG1 message service control block

BASIC programs must preserve the fields they do not own across RUN.

BASIC Startup Pattern

```
1000 IF PEEK(1000)<>82 OR PEEK(1001)<>69 OR PEEK(1002)<>77 THEN RUN "D1:>CORE>WFC"
1010 RAPI=PEEK(1004)+256*PEEK(1005)
1020 RPUB=PEEK(1006)+256*PEEK(1007):SS=PEEK(1010)+256*PEEK(1011)
1030 RMODE=RAPI+3:RROUTE=RAPI+6:ROPEN=RAPI+9
1040 RSTAT=RAPI+15:RGET=RAPI+18:RPUT=RAPI+21:RTGET=RAPI+27
1050 IF PEEK(SS)<>77 OR PEEK(SS+1)<>9 THEN RUN "D1:>CORE>RUNBBS"
1060 R=USR(ROPEN,0):IF R<>0 THEN R=USR(RAPI+51,6):RUN "D1:>CORE>WFC"
1070 R=USR(RMODE,1):R=USR(RROUTE,2)
```

Local modules instead call USR(RMODE,0) and USR(RROUTE,0).

ATARI MUTINEER DOORS PROGRAMMERS GUIDE

docs/mutineer/MUTINEER_BBS_DOORS_PROGRAMMERS_GUIDE.md

Atari Mutineer Doors Programmers Guide

Doors are external Atari BASIC programs launched by the DOORS module from a catalog entry. Supported release doors are bundled in the 16MB BBS ATR.

Door Disk Layout

```
D1:>DOORS><GAME>>
<GAME>.bas
optional overlays and data
DOOR.cfg
README.txt
TEXT/*.asc
TEXT/*.ata
```

The 16MB Mutineer ATR includes the bundled door directories and catalog rows. Development builds may still produce isolated door images, but they are not release artifacts.

Catalog

Door catalog source rows are:

```
id, door name, filename, path, access level
```

Example:

```
3, Arena, ARENA, D1:>DOORS>ARENA>, 2
```

The path must end in >. The package/offline compiler validates this source, generates D1:>DATA>DOORCAT.DAT, and generates the visible DOORMNU files. At caller time, DOORS loads DOORCAT.DAT, filters rows by caller access, and runs the precomputed target path. A trapped missing target prints DOOR UNAVAILABLE. and returns to the menu.

Door catalog edits are offline work. After changing DOORS.CFG or adding a new DOOR.CFG row, rebuild the runtime door catalog before putting the BBS back online. Caller-facing launches use only the generic Opening Door... notice.

REMMS Door Contract

A supported caller-visible door must:

- discover RAPI, RPUB, and RSESS through \$03E8;
- validate session magic/version;
- preserve caller identity, modes, access, user number, session deadline, and the current handoff byte at SS+97;
- call USR(ROPEN) and select host/both for a remote caller;
- use USR(RSTAT) at blocking input boundaries;
- use normal E:/K: or RTGET=RAPI+27 for terminal text;
- use RGET/RPUT only for deliberate raw or binary bytes;
- save its fields and clear POKE SS+98,0 before RUN "D1:>CORE>DOORS";
- return to WFC when session +87 is no longer 1.

Minimal entry:

```

1000 RAPI=PEEK(1004)+256*PEEK(1005):RPUB=PEEK(1006)+256*PEEK(1007):SS=PEEK(1010)+256*PEEK(1011)
1010 RMODE=RAPI+3:RROUTE=RAPI+6:ROPEN=RAPI+9:RSTAT=RAPI+15:RTGET=RAPI+27
1020 IF PEEK(SS)<>77 OR PEEK(SS+1)<1 OR PEEK(SS+1)>3 THEN RUN "D1:>CORE>WFC"
1030 R=USR(ROPEN):IF R<>0 THEN RUN "D1:>CORE>WFC"
1040 R=USR(RMODE,1):R=USR(RROUTE,2)
1050 OPEN #3,12,0,"E:":OPEN #4,4,0,"K:"
1060 PRINT #3;"HELLO FROM A REMMS DOOR"
1070 R=USR(RSTAT):IF PEEK(SS+87)<>1 THEN RUN "D1:>CORE>WFC"
1080 X=USR(RTGET):IF X=256 THEN 1070
1090 POKE SS+98,0:RUN "D1:>CORE>DOORS"

```

Local-only tools should use mode/route local and must not poll R:.

Door Development Checklist

- Keep the caller-facing program name short enough for the door catalog row.
- Store game data under the door's configured path, not in D1:>CORE>.
- Keep supported release door programs and data under D1:>DOORS>.
- Validate REMMS discovery and session version before touching caller I/O.
- Reopen REMMS with ROPEN,0 after entry from DOORS.
- Use mirrored route for active remote callers and local route for sysop-only setup tools.
- Keep long screens pager-friendly: 40 columns, explicit returns, and no terminal-specific cursor controls unless you provide ASCII and ATASCII paths.
- Use line-oriented text input for commands and forms; reserve raw RGET/RPUT for binary protocols.
- Before file writes, suspend the modem owner, write, close disk IOCBs, and resume before prompting again.
- Clear POKE SS+98,0 and return to D1:>CORE>DOORS after a normal game exit. DOORS reloads and displays DOORMNU on every return.
- Return to D1:>CORE>WFC after carrier loss, timeout, fatal REMMS failure, or a deliberate disconnect.

Caller-Facing Door Skeleton

```

1000 IF PEEK(1000)<>82 OR PEEK(1001)<>69 OR PEEK(1002)<>77 THEN RUN "D1:>CORE>WFC"
1010 RAPI=PEEK(1004)+256*PEEK(1005):RPUB=PEEK(1006)+256*PEEK(1007):SS=PEEK(1010)+256*PEEK(1011)
1020 RMODE=RAPI+3:RROUTE=RAPI+6:ROPEN=RAPI+9:RCLOSE=RAPI+12:RSTAT=RAPI+15
1030 RGET=RAPI+18:RPUT=RAPI+21:RTGET=RAPI+27:REVENT=RAPI+42:TBEGIN=RAPI+45:TEND=RAPI+48:RFAIL=RAPI+51
1040 IF PEEK(SS)<>77 OR PEEK(SS+1)<>7 THEN RUN "D1:>CORE>WFC"
1050 ONLINE=PEEK(SS+87):LOCAL=0:IF ONLINE=8 THEN LOCAL=1
1060 IF ONLINE=1 THEN R=USR(ROPEN,0):IF R<>0 THEN R=USR(RFAIL,6):RUN "D1:>CORE>WFC"
1070 IF ONLINE=1 THEN R=USR(RMODE,1):R=USR(RROUTE,2)
1080 IF LOCAL THEN R=USR(RMODE,0):R=USR(RROUTE,0)
1090 OPEN #3,12,0,"E:":OPEN #4,4,0,"K:"
1100 PRINT #3;"DOOR READY"
1110 REM GAME LOOP BEGINS HERE
8990 POKE SS+98,0:RUN "D1:>CORE>DOORS"

```

Use TBEGIN,1 for line input, TBEGIN,2 for masked input, and TBEGIN,3 for single-choice commands. A RTGET or REVENT result at or above 256 is a structured status, not a printable byte.

Door Prompts

Door prompts should match the rest of the BBS:

[HH:MM:SS Door Command >

Use the caller's remaining session time from SS+126/+127. If the door has a game-specific prompt, keep the leading time-left block and put the game noun

after it, for example:

```
[ HH:MM:SS Arena Command >>
```

Do not print fixed time/range placeholders. If a value is not real, omit it.

State And Files

Use a door-owned disk path for config, players, matches, scores, and display assets. Document every session scratch byte a door claims. SS+96 is used for door/transfer mode and SS+97 is shared by module handoffs, so save and restore them deliberately.

Do not clear the whole session block, change the online byte, or hang up the caller unless disconnect is an explicit door feature. Do not mix terminal-translated E: output with binary transfer payloads.

Suggested door files:

```
| File | Purpose |
| --- | --- |
| .BAS | caller-facing entry |
| .CFG | game configuration |
| PLY.DAT | player/user records |
| LED.DAT | leaderboard |
| TEXT/*.ASC | ASCII screens |
| TEXT/*.ATA | ATASCII screens |
| DOOR.CFG | catalog row for the sysop |
| INSTALL.TXT | 40-column install instructions |
```

For mutable files, prefer fixed-size records or line-oriented records with a version/header line. Rebuildable indexes should contain enough information to recover from a partially deleted or compacted data file.

Bundled Doors

```
| Game | Runtime status | Persistent support |
| --- | --- | --- |
| BLKJACK | REMMS caller/local door | BJLEAD.DAT leaderboard |
| DUMJIN | REMMS caller/local door | DUNGLEAD.DAT plus session scratch |
| ARENA | REMMS caller/local door | config, player, match, and champion data |
```

The maintained dungeon source, catalog filename, install path, and release artifact all use DUMJIN.

Tournament Arena

The bundled Arena caller program is a REMMS caller-facing door. It supports remote and local sessions, preserves the BBS session fields, routes input/output through the current REMMS API, returns normally to D1:>CORE>DOORS, and returns carrier loss to D1:>CORE>WFC.

Arena files:

```
| File | Role |
| --- | --- |
| ARENA.bas | caller-facing tournament module |
| ARENA1.ovl | open-registration setup |
| ARENA2.ovl | bracket creation |
| ARENA.CFG | tournament state |
| ARENAPLY.DAT | player records |
| ARENAMAT.DAT | match records |
| ARENALED.DAT | champion records |
| DOOR.cfg | catalog row |
```

The ARENA1.ovl and ARENA2.ovl programs are local sysop setup tools; they do not attach to the caller communications session.

DATA AND FILE FORMATS

docs/mutineer/DATA_FORMATS.md

Data And File Formats

This page documents files used by the packaged ATR and live board.

ATR Directory Layout

Normal Mutineer ATR:

- Root: AUTOEXEC.BAT, AUTOGO, RHND850.COM, REMMS.COM, README.
- D1:>CORE>: tokenized BASIC runtime modules and interactive overlays.
- D1:>TOOLS>: tokenized local/job utility modules.
- D1:>PROTO>: native transfer protocol binaries.
- D1:>DATA>: runtime databases and config.
- Configured ASCII root: ASCII menus/static files.
- Configured ATASCII root: ATASCII menus/static files.
- D1:>MAIL>: default message base files.

Configuration Files

D1:>DATA>PATHREG.DAT is the fixed bootstrap location for the binary path registry. Its eight-byte header contains PREG, version 1, record size 64, record count, and generation. Each 64-byte record contains a 16-bit stable ID, flags, path length, an eight-byte name, and a 52-byte uppercase SpartaDOS path.

Flags identify directories, required files, writable targets, and text assets.

Paths are limited to D1: through D8: and reject controls, wildcards, traversal, malformed syntax, duplicates, and truncation. INIT and host tooling validate the complete registry and atomically replace it through .NEW and .BAK. Runtime modules look up and cache only the records they use, validating the selected record before consuming its path. If both the live and backup registries are invalid, REMMS detaches and INIT stops at usable BASIC.

The registry supplies data, core, tools, protocols, ASCII, ATASCII, mail, network, and doors roots plus stable records for packaged menus, static text, common user/caller files, transfer handoff files, scheduler files, and global network queue/config files. APPREG.DAT, DOORCAT.DAT, message metadata, and file metadata remain the authorities for their dynamic targets.

Runtime path ownership classes:

- fixed bootstrap: RUNBBS, APPMGR, PATHREG.DAT, PATHREG.BAK, and PATHREG.NEW;
- path registry: common config/data/spool files with stable PATHREG IDs;
- app registry: registered BASIC/native application targets;
- door catalog: door target paths and minimum access levels;
- message metadata: MMSG?.MET/.DAT/.IDX plus base-local message worker files under the configured message root;
- file metadata: FILE?.MET/.DAT/.IDX, public upload paths, and quarantine paths;
- native worker contracts: lock, journal, swap, recovery, and compaction files owned by message/file native workers.

D1:>DATA>CONFIG is line-oriented text:

CSECT
CBYTE

CALLNO
MSECT
MBYTE
MSG
HMSG
IDLEMIN
ANSRINGS

WFC, CORE, and SYSOP read it to track caller history pointers and message counts and the inactivity logout timer in minutes. IDLEMIN=0 disables idle logout. ANSRINGS is the number of RING indications REMMS waits before answering, clamped to 1..9. The packaged default is 657, 48, 0, 559, 21, 1, 1, 15, 1.

D1:>DATA>MSGBASE.CFG stores the message-base root on line 1 and the active base IDs on line 2 (for example, 0A). BASIC readers cache this list instead of probing all 27 possible metadata files. A legacy one-line file remains readable and falls back to probing all IDs. The native message ABI fixes this path at D1:>MAIL>; changing the file is not a supported relocation mechanism. Its packaged value is:

D1:>MAIL>

D1:>DATA>SCHED.CFG stores generic scheduled jobs. Message jobs use registered application keys, not a message-specific hard-coded table. The packaged defaults declare MSGVER daily, MSGNET daily, and MSGCOMP weekly. WFC dispatches one eligible job per RTC minute while idle with no carrier or pending ring.

D1:>DATA>SYSOP.CFG controls remote sysop chat availability and snoop default. Current format:

SYSC,2
available
snoop_default

0 means remote callers who use Y see SYSOP NOT AVAILABLE. and CORE stays loaded. 1 allows CORE to dispatch the SYSCHAT pager. The local console rings three times, five seconds apart; the SysOp presses the Atari START console key to answer. The caller can cancel the page with Ctrl-X. If no answer arrives during the three-ring window, the caller is told the SysOp is not available but has been notified, then returns to CORE. Legacy one-number files remain readable as availability-only files.

D1:>DATA>ACL.CFG is comma-separated:

level,session_minutes,daily_minutes

Level 0 is banned. Level 9 is sysop. LOGIN applies session and daily limits from this file during login/registration.

Compact Permission Authorities

Mutineer v1 uses compact minimum-level fields rather than resident per-level permission matrices:

- APPREG.DAT stores the minimum level for each app.
- DOORCAT.DAT stores the minimum level for each door.
- MSG?.MET stores read and post levels plus moderation mode.
- FILE?.MET stores view, download, and upload levels plus protocol policy.
- USERS.DAT stores caller access level; LOGIN enforces profile/user-edit rules.

Authors can delete their own messages; level 8+ can moderate/delete other users' messages; level 9 owns sysop-only user edits including handle changes. Full per-level matrices, roles, groups, and deny-overrides are not part of the current runtime format.

Users

Files:

- D1:>DATA>USERS.DAT: fixed-width user records.
- D1:>DATA>USERS.IDX: fixed-width handle index records.
- D1:>DATA>NEXTUSER.DAT: next user number as text.

USERS.DAT records are 128 bytes:

- status byte;
- user number high/low;
- handle, 16 bytes;
- full name, 24 bytes;
- email, 40 bytes;
- password, 12 bytes;
- access level;
- flags;
- call count high/low;
- daily date, 8 bytes;
- daily used minutes high/low;
- reserved tail bytes.

USERS.IDX records contain normalized handle text, user number high/low, status, and a reserved byte. Registration creates the first user as access level 9; later self-registered users are level 1. Fresh ATRs preallocate 128 zero-filled record slots in both files. LOGIN opens them in update mode and positions to the selected fixed slot before writing; a zero first byte marks an unused index slot. Ordinary callers cannot change handles after creation. Level 9 sysops can change a handle through LOGIN profile edit; the save rewrites the matching USERS.DAT record and its fixed USERS.IDX slot together. NEXTUSER.DAT is rejected once all 128 slots are allocated.

Caller History

D1:>DATA>CALLERS.DAT is a text file. The packaged file starts with *. WFC recalibrates the append pointer on startup and appends caller identity, date, and time. CORE reads it for the caller-list command.

Feedback

D1:>DATA>FEEDBACK.DAT is an append-only text file created the first time a caller saves feedback. CORE writes one record per feedback entry:

```
+MM/DD/YY HH:MM:SS
user_number
handle
email_or_secondary_identity
feedback text
```

The E feedback command and confirmed G goodbye path both use this file.

Application Registry

Packed runtime: D1:>DATA>APPREG.DAT.

APPREG.DAT is a binary packed image. APPMGR bulk-loads it once, validates its length and checksum, and binary-searches its key-sorted records while all live communications continue through REMMS.

Logical rows:

handle, key, return, type, flags, access, route, term, name, path

Validation rules:

- up to 32 records;
- handle range 1..254, unique;
- stable key unique, hexadecimal in the source file;
- return key must be 0000 or match another app key;
- access must be 0..9;
- path must fit 24 bytes and match supported SDX path syntax;
- name must fit 8 bytes.

Packed format:

- 12-byte header: MAPR, version 1, record size 48, record count, generation, little-endian 16-bit additive body checksum, and two zero bytes;
- key-sorted 48-byte records: handle/type/flags/access, little-endian key and return key, path/name lengths, route/term, 24-byte path, 8-byte name, and four reserved zero bytes.

D1:>DATA>FILEBASE.CFG is headed by FCFG,1; each following ATASCII record is one unique uppercase file-area ID in display order. FILEBASE uses this list for its resident area selector instead of probing all 26 possible metadata files, and FILEMGR updates it atomically after a successful area creation.

Transfer Protocol Registry

Packed runtime: D1:>DATA>PROTOREG.DAT.

PROTOREG.DAT begins with MPRT,1; each remaining ATASCII record contains:

id, name, send_key, recv_key, features, min_level, max_block, description

The packaged registry exposes XMODEM checksum, XMODEM CRC, XMODEM-1K, and YMODEM batch. Native binaries are physically packaged under D1:>PROTO> as XMSEND.COM, XMRECV.COM, YMSEND.COM, and YMRECV.COM. ZMODEM remains reserved and is not user-selectable.

Message Bases

Default public/private base files:

- D1:>MAIL>MESGA.MET
- D1:>MAIL>MESGA.DAT
- D1:>MAIL>MESGA.IDX
- D1:>MAIL>MESG0.MET
- D1:>MAIL>MESG0.DAT
- D1:>MAIL>MESG0.IDX

Format 1.1 MESG?.MET stores two 17-field MMET metadata blocks. Each block contains:

- signature MMET;
- format version;
- base ID;

- generation;
- base name and short name;
- read and post levels;
- moderation mode;
- next message number;
- active, deleted, and held counts;
- DAT end sector/byte;
- IDX record count;
- last compaction marker.

Format 1.1 MESG?.DAT records start with MM11, followed by record state, message number, author and recipient user numbers, parent and root message numbers, display names, subject, date, time, line count, body lines, and END.

Runtime states are A (active), H (held), and D (deleted). Moderation and deletion update both DAT state and index flags so recovery preserves the decision.

MESG?.IDX records are 16 bytes: message number, DAT pointer, status flags, line count, parent/root, author, recipient, and one reserved zero byte.

Auxiliary message files:

- D1:>MAIL>DRAFTS.DAT/.IDX: disk-backed message drafts.
- D1:>MAIL>MSGREAD.DAT: append-updated per-user/base high-water table. Each 8-byte record contains version, base ID, big-endian user number, big-endian high-water message number, a reserved zero byte, and an additive checksum. Readers use the last valid matching record and ignore corruption.
- D1:>MAIL>MSGREAD.XCP: sparse read exceptions.
- D1:>MAIL>MAILSTAT.DAT: physical 8.3 name for private-mail state records.
- D1:>MAIL>MAILTO.IDX and D1:>MAIL>MAILFROM.IDX: rebuildable mail accelerators.
- D1:>MAIL>MSGAUDIT.DAT: moderation/private-access audit trail.
- D1:>NET>NETIN.Q and NETOUT.Q: filesystem transport queues.
- D1:>NET>NETSEEN.IDX, NETSENT.IDX, and NETMAP.DAT: 16-byte network identity and suppression records.
- D1:>NET>NETNODE.CFG and NETROUTE.CFG: local node and area,base,direction,peer routes.
- D1:>NET>IN, OUT, BAD, and DONE: packet lifecycle directories.

MAILSTAT.DAT uses 16-byte version-1 records containing message number, sender, recipient, sender-deleted, recipient-deleted, recipient-read, and delivered state. Sender and recipient deletion are independent. Only when both delete a message is the authoritative MESG0 record marked deleted.

Interactive message modules:

- MESGAREA: list, scan, read, and dispatch.
- MSGEDIT: compose using disk-backed drafts and commit MM11 records. / opens the save/edit/display/abort/continue menu and // stores a literal slash. Public addressees may name an active user without making the post private. ANSI and ATASCII cursor, insert, delete, and backspace input is interpreted by the editor and is never stored in message text. Unsafe modem bursts discard and restart the whole line; no received prefix is committed.
- MSGMAIL: private mail compose, inbox, sent listing, and readback for base 0.
- MSGMOD: level 8/9 held queue, approve, soft-delete, restore, thread lock/unlock, and audit logging.

Utility/job modules live under D1:>TOOLS>:

- MSGVER: read-only structural message-base verification.
- MSGRECOV: thin manual UI for MSGRCV.COM. The native worker scans valid active, held, and deleted MM11 records, rebuilds the full index and mail accelerators, retains backups, and installs through an MSWP manifest.
- MSGCOMP: thin manual/scheduled UI for MSGCMP.COM. The native worker

rewrites retained DAT records and every index pointer, drops fully deleted private mail, retains .DBK/.IBK/.MBK, and installs through MSWP.

- MSGNET: validates and exchanges MNET v1 ASCII7 packets through filesystem queues. Modem transport remains external.

MSGSTORE.COM is the only runtime owner of live message metadata, data, indexes, locks, mail state, moderation audit, and network maps.

Interactive BASIC modules write only request and staging files. Its transient MSG1 control block occupies session offsets 340..351.

Routine message operations use a short-lived .LCK, not a transaction journal. DAT is append-only, and MSGRCV.COM rebuilds index and metadata from valid MM11 records after an interrupted operation. A failed operation keeps the lock visible until manual recovery. Legacy MTXN files remain recognized so bases created by earlier builds can still be recovered.

Base creation uses a versioned MBASE request in D1:>MAIL>MSGREQ.DAT. INIT supplies names and access levels; MSGSTORE.COM validates the request and constructs both 17-field MMET blocks. BASIC never writes an MMET image. Normal commits append a validated 16-byte index record and install a verified dual-block metadata replacement. Recovery and compaction rebuild from authoritative MM11 records and verify staged index cardinality and DAT pointers before installation.

SCHED.CFG is executed by the SCHED overlay while WFC has no carrier or ring. DATA>JOBCTX.DAT uses MJOB,2 followed by job ID, module key, argument, attempt count, retry limit, state, result, and detail. It records one running/retried job. Verification, networking, and conditional compaction run without prompts; recovery is manual-only.

MSGMGR is fenced as a format 1.1 maintenance launcher. Its old create, post, delete, undelete, and rebuild writers are disabled because they emitted the pre-1.1 metadata, body, and index layouts.

Caller commands in MESGAREA:

- Q: quick scan.
- S: summary.
- R: read.
- E: enter.
- M: private mail.
- O: moderation for level 8+.
- K: kill.
- A or L: select/list bases.
- X: return to main menu.

File Areas

Default files:

- D1:>DATA>FILEA.MET
- D1:>DATA>FILEA.DAT
- D1:>DATA>FILEA.IDX

FILE?.MET version 2 stores the original eleven fields:

- version;
- area ID;
- area name;
- view level;
- download level;

- upload level;
- data file path;
- index file path;
- upload path;
- next file number;
- active file count.

It then stores generation, protocol mask and defaults, batch/partial policy, upload/free-space/batch limits, and the area quarantine path. The packaged area uses D1:>DATA>QUAR>.

File records include status, file number, filename, description, type, location, uploader, upload date, file size, and download count. Index records are 16 bytes and use states 0=unused, 1=active, 2=held, and 4=deleted. Caller uploads are always installed in the configured quarantine path as held records. Held records are excluded from browse, search, marks, and downloads. FILEMGR creates and verifies areas, approves held uploads by moving them to the public upload path, rejects files, restores deleted records, and invokes native recovery/compaction. Permanent catalog and transaction writes belong only to the native file workers.

XFER.TXN is an append-only FTXN,2 journal. Its fixed context retains the area and operation, file and actor numbers, four transaction-ID bytes, original DAT/index/generation positions, committed batch count, and source/temp/final paths. Durable state lines follow END: TRANSFER_OK, FILE_OK, DAT_OK, IDX_OK, MET_OK, LOG_OK, and COMMITTED. FILERCV.COM validates the journal paths, accepts an already-installed payload, rebuilds the authoritative index before appending a missing DAT record, avoids duplicate audit entries, and removes the lock/journal only after convergence. Rejected held uploads have their quarantine payload removed before the deleted catalog record is committed.

Caller commands in FILEBASE:

- B, F, A, N: browse/list configured areas and files.
- D: download by number or name.
- U: upload into an area.
- I, H, ?: file-area information.
- L: lists marked files.
- C: clears marked files.
- M: marks a file.
- S: searches files.
- P: selects the transfer protocol.

Runtime modules print short status lines before slow disk/database operations, such as loading menus, scanning message/file indexes, looking up users, and saving records. These messages are caller-visible and are expected output.

Door Catalog

Source file: D1:>DATA>DOORS.CFG.

Runtime file: D1:>DATA>DOORCAT.DAT.

Rows are comma-separated:

```
id, door name, filename, path, access level
```

The path must end in >. Offline/package tooling validates DOORS.CFG, generates DOORCAT.DAT, and generates the visible DOORMNU menu files. Callers use DOORCAT.DAT; they do not pay the cost of parsing CSV rows.

DOORCAT.DAT starts with DORC, version byte 1, a two-digit ASCII count, and a space. Each record is 50 bytes: selector, required access level, and a 48-byte space-padded target path. Missing door programs return DOOR UNAVAILABLE. and leave the caller in the door menu.

Text Assets

Default packaged directories, configurable through PATHREG.DAT:

- D1:>ASCII>
- D1:>ATASCII>

Menus/static files include MAINMNU, FILESMNU, MESGMNU, MAILMNU, MODMNU, DOORMNU, INITMNU, FILEMGR, MSGMGR, BULLMNU, BULL1, BULL2, BULL4, INFO, OBBS, WELC, and PRELOGIN. PRELOGIN is authored as the paired source files prelogin.asc and prelogin.ata and is displayed once before the login/new-user/goodbye prompt. The packager also writes generated branded text files HELP, NEWU, and FUNC, and stages TEXTF article files.

Menus and static files are reread through REMMS RTEXT whenever their module menu loop returns. BASIC does not keep menu text resident or open these assets directly while the modem owns R:.

TROUBLESHOOTING

docs/mutineer/TROUBLESHOOTING.md

Troubleshooting

Troubleshoot the active REMMS path. Do not switch a live board back to an old MMS communication layer to work around a failure.

Altirra Does Not Boot The Board

- Confirm the Mutineer ATR is mounted as D1:.
- Confirm SpartaDOS X is mounted as the cartridge.
- Confirm BASIC is enabled.
- Confirm fast boot is enabled.
- If you only want to inspect the ATR, boot a utility disk as D1: and mount Mutineer as a secondary disk so it does not auto-start.

Expected startup:

```
AUTOEXEC.BAT -> RHND850.COM -> REMMS.COM -> BASIC -> WFC
```

No Listener Or No Connection

- Confirm the Atari 850 device is enabled.
- Confirm 850 emulation is Full.
- Confirm the modem is a child device of the 850.
- Confirm the modem is inbound/listen on 127.0.0.1:6507.
- Confirm no other Altirra instance owns port 6507.
- Keep Telnet LF translation off.
- Leave CIO R: patch off.

The caller should see:

```
CONNECTED PRESS [RETURN]
```

If the caller sees WFC dashboard text before that prompt, the BBS is leaking local dashboard output to the modem and should not be used as a release image.

Return Does Not Work

Use a normal CR or CR/LF Return from the terminal client. In PuTTY, use Telnet, set local echo to Auto or Force off, and leave local line editing off.

If the A command says HIT and then ignores Return, the runtime is using an older CORE/REMMS combination. Use a current release ATR.

First User Is Not Sysop

The first completed registration becomes user number 1 and access level 9. If a test user was created first, restore your clean ATR backup or use offline maintenance to repair the user/access files.

Files involved:

- D1:>DATA>USERS.DAT
- D1:>DATA>USERS.IDX
- D1:>DATA>NEXTUSER.DAT
- D1:>DATA>ACL.CFG

Cannot Reach A Module

- Confirm the main menu prompt is visible.
- Use ? to redraw the main menu.
- Use M for message bases, F for file areas, and D for doors.
- Use X inside message, file, and door menus to return to CORE.
- If the board disconnects, reconnect and confirm WFC answers normally.

Online modules must preserve REMMS session state. If a custom module loses the caller, review its session restore, ROPEN,0, route, and return path.

Door Is Missing

Missing doors print:

DOOR UNAVAILABLE.

Check:

- the caller's access level meets the door catalog level;
- the door path in the offline DOORS.CFG source ends in >;
- D1:>DATA>DOORCAT.DAT was rebuilt after catalog edits;
- the 16MB Mutineer ATR contains the expected D1:>DOORS> directory;
- the runtime catalog target agrees with the bundled D1: path.

Message Or File Area Looks Wrong

- Confirm message bases use MSG?.MET, MSG?.DAT, and MSG?.IDX.
- Confirm file areas use FILE?.MET, FILE?.DAT, and FILE?.IDX.
- Confirm the metadata paths point to existing files.
- Use offline maintenance tools for rebuild/compact work.
- Keep a backup before changing live data files.

Dropped Calls

REMMS should detect carrier loss and return to WFC. If a custom module hangs after a caller disconnects, it is probably waiting for input without polling REMMS status.

At input boundaries, caller-facing modules should call RSTAT, RTGET, or REVENT, handle structured results at or above 256, and return fatal or carrier-loss states to WFC.

Backups

Keep a clean release ATR and a separate live ATR. Back up the live ATR before:

- complete initialization;
- editing user files;
- editing message or file metadata;
- adding door catalog rows;
- replacing text assets;
- changing bundled door programs or data.