The Ataman®* TCP Remote Logon Services User's Manual

This Manual is for the Free version of the ATRLS.

No support is available from Ataman Software, Inc. for this version.

IMPORTANT:
Read the *Security Considerations* section before you install these services.

See Ordering the ATRLS for ordering information.

Table of Contents

---

* Ataman is a registered trademark of Ataman Software, Inc.
All other trademarks herein are trademarks of their respective holders.

1. Overview

The Ataman TCP Remote Logon Services (ATRLS) contains server implementations of the Internet TCP telnet, rlogin, rsh, rcp and rexec protocols. These services are distributed as shareware. The ATRLS are not free! However, you may try the ATRLS before paying. If you are not familiar with the shareware concept, see the Shareware section below.

The telnetd service provides an implementation of the telnet protocol specified in the Internet document, RFC 854. The rlogind service provides an implementation of the rlogin protocol defined in the Internet document, RFC 1282. Both services allow users with an appropriate client program to remotely logon to Windowsf systems. Users log on using the same password as used in normal Windows user authentication. During a remote logon session, user processes run in the user's security context.

Additionally, if a user's telnet/rlogin client program provides support for ANSI terminal escape sequences, users may run full-screen console apps, such as text editors. Telnet and rlogin clients that use ANSI escape sequences include, but are not limited to, those that emulate the following terminal types: VT100, VT102, VT220, VT320, VT420, xterms, xterm. Color support is provided for telnet/rlogin clients that use ANSI-BBS-style escape sequences.

The ATRLS also contain rexecd and rshd services. These services implement the rexec, rcp and rsh protocols found on many Unix systems. (As in Unix, the rcp functionality is built on top of the rsh protocol.) Also provided is the C programming language source code of an rexec client program suitable for porting to any Unix system that does not have an rexec client. Due to the many possible places and conditions this rexec client program might be used, it is provided **without** technical support. If you have experience in porting code between Unix systems, you may find the provided source code a convenience.

2. Security Considerations

This manual assumes that you are familiar with the risks and benefits of the rshd, rexecd, rlogind and telnetd services found on Unix and other operating systems. The secure operation of these services is a complicated matter, and this manual provides only the details specific to this implementation. If you are unfamiliar with the security aspects of these services, Ataman advises you to consult a TCP/IP networking protocols tutorial.

The Ataman TCP Remote Logon Services (ATRLS) allow users to remotely logon within their own security context. However, several security issues remain because Microsoft Windows is not implemented with full support for remotely logged in users.

2.1 Potential Interaction Problems

2.1.1 Random sounding of the system bell

Remote users may run programs that cause an action to request the system bell to ring. The system bell associated with the main monitor will sound because Windows does not redirect this function. Locally logged-on users, unaware of remote users running programs, may think they have made a mistake since the bell seems to ring at random.

2.2 No Clean Process Termination

Microsoft provides no **clean** method of killing a process in Windows. There is a kill provided, but that kill does not notify DLLs that are attached to the killed process of the exit. This potentially leaves dead data inside those DLLs. We do use the kill provided to cleanup logon sessions that are unexpectedly terminated.

If your program uses DLLs and your connection is unexpectedly broken, you may encounter this problem. Unfortunately, Microsoft does not document the conditions where dead data is left inside DLLs. However, in practice, the vast majority of programs do not seem to be affected. Please understand, this problem is a shortcoming of Windows and **not** a flaw in Ataman’s services. Hopefully, Microsoft will become embarrassed about this obvious and serious shortcoming in Windows and provide a proper mechanism to cleanly terminate processes.

2.3 Other issues

The issues described above are the only security problems identified thus far. However, since Windows does not fully support remotely logged-on users, it is likely that new security holes will be discovered. In short, remotely logged-on users using the Ataman telnetd, rlogind or rexecd services should be limited. The security levels of these users should reflect their potential to gain privileged access to the system. The section Using the Ataman TCP Remote Logon Services section below covers the mechanism used to restrict the users allowed to logon.

3. Requirements

- Windows XP Home/XP Professional/2003 Server/2008 Server/
  Vista/ 7 /2008 Server R2
- Microsoft Winsock Version 2 installed and configured.

4. Installation

Your user account must have Administrators or Domain Admin privilege levels to install the ATRLS.

On the system that you wish to install the Ataman TCP Remote Logon Services, create a directory that is **local** to that system. For example:

```
mkdir c:\atrls
```

The directory you create must have its permissions set such that the executable (*.exe) files can be read and executed by the SYSTEM account and all user accounts that will be allowed to remotely logon. All directories in the path to the executables must be searchable by those accounts.

Change your working directory to this new directory. Unzip the archive into this directory.

To install the ATRLS type:

```
atrls install start
```

You should now proceed to the Using the Ataman TCP Remote Logon Services section in order to authorize users to logon.

5. Removal

Do NOT use the procedures in this section if you are upgrading or moving the software to a different location on the same machine. Instead, follow the information in the Upgrading or Reinstallation sections.

Ataman Software is committed to making the use of its software as easy as possible for the end user. Most users prefer software that removes as easily as it installed, thus we provide a procedure to uninstall the software. The uninstall procedure removes the services and all associated registry entries. It does **not** remove the disk files as you may simply be moving the software to a different machine.

If you need to remove the ATRLS from your system, type:

```
atrls stop remove
```

6. Reinstallation

If you want to move the ATRLS to a new location on your machine, stop the ATRLS service:

```
atrls stop
```

Move the files to the new location, then:

```
atrls reinstall start
```

Using this method will preserve all your old configuration settings.

7. Upgrading

You might want to first create backups of your current settings. You can do this using the “dump” option to the `auseradm.exe` and `aconfig.exe` commands. See Configuring the ATRLS from the Command Line for details.

To upgrade to a new version of the ATRLS, in the directory of the old version type:

```
atrls stop
```

Unzip the new archive into the directory of your choice, then in the new directory type:

```
atrls upgrade start
```

Using this method will preserve all your old configuration settings.

This upgrade method will work for the version 2.X -> 3.X -> 4.X → 5.X upgrade as well. However, if you were using the version 2.X environment file option, make sure you save your environment file and read the Version 3.X and newer doesn't have the Environment File option section.

If you are upgrading to a new major version, you need to install your new registration code. See the Registration section for details.

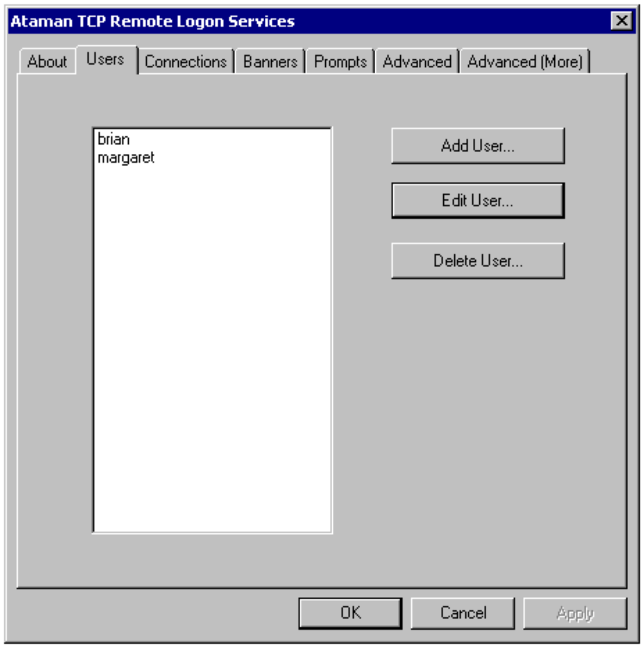
8. Registration

Registration is not needed for the Free version of the ATRLS.

9. Using the ATRLS

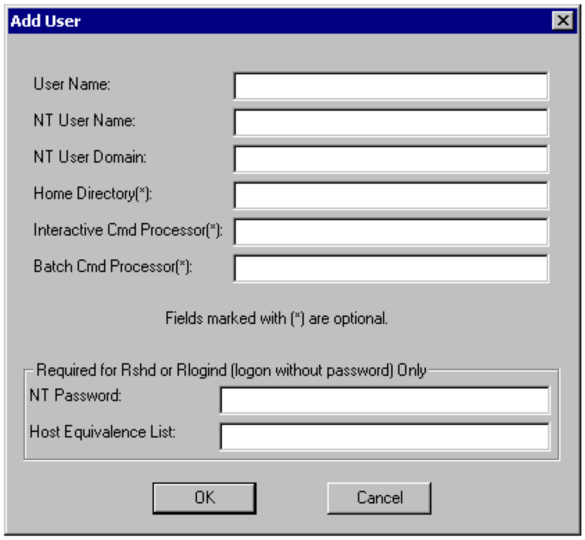
9.1 Configuring the ATRLS

To configure the ATRLS, invoke the "Ataman TCP R. L. Services" control panel, through the Windows Start button from the Control Panel item in the Settings menu. The various user and program settings are described below.



9.1.1 Users Page

To add a user, select the “Add User…” button. To edit a user, select the desired user account name, and then press the “Edit User…” button. (Alternatively, you can double-click the user account name you want to edit.) To delete a user, select the user account you want deleted, then press the “Delete User…” button. You will be asked to confirm the operation.



Fields marked with (*) are optional.

9.1.1.1 User Name

This is the user account name the remote client programs will use to allow remote users to logon. This name is the name used by the outside world. The ATRLS treats user account names that differ only in the case of letters as the same account.

9.1.1.2 NT User Name

This is the Windows user account that will execute for this remote user's logon. This can be an account either on the local machine, or in a Windows domain. This account name is the same name as the "Username" listed in the User Manager program.

9.1.1.3 NT User Domain

This is the domain name associated with the NT User Name above. If the NT User Name is not a domain account, but is local to the system you are configuring; you should place a single period or dot "**.**" in this field.

9.1.1.4 Home Directory

This is the initial working directory given to the commands initiated by this user account. This directory should be local to the system you are configuring. This directory will be automatically set into the environment variable HOME when the user logs on.

9.1.1.5 Interactive Command Processor and Batch Command Processor

By default, the ATRLS pass the command from the rsh client to the `cmd.exe` command processor that comes with Windows. If you have purchased an alternate command processor that you would rather use, enter the command processor's path and any switches in this field. Because the command processor often needs different arguments when used interactively, rather than in "batch" mode, we provide two forms of command processor string. The Interactive Command Processor is invoked by telnetd and rlogind. The Batch Command Processor is invoked by rshd and rexecd. When the Batch Command Processor value is used, the environment variable COMMAND is set to the command passed in from the remote rsh or rexec client program. In both instances, the value of the Command Processor field is environment expanded.

For example, you could have rshd invoke `cmd.exe` by entering:

```
%COMSPEC% /C %COMMAND%
```

When a remote user uses an rsh client program to issue the `dir` command, on most systems the resulting command will be:

```
C:\WINNT\SYSTEM32\CMD.EXE /C DIR
```

In the version 1 of the ATRLS, a script called REMOTE.CMD was automatically invoked when a user logged-on. You can achieve the same functionality in version 2 by setting these fields as follows:

Interactive Command Processor:

```
%COMSPEC% /K CALL %HOME%\REMOTE.CMD
```

Batch Command Processor:

```
%COMSPEC% /C CALL %HOME%\REMOTE.CMD&%COMMAND%
```

9.1.1.6 NT Password

This value is used only with rshd or rlogind if the user is allowed to logon without a password. If you do not want the rshd or rlogind (without password) functionality, **do not set this field**.

This is the password associated with the Windows user account specified above. This field is necessary because Windows has no facility that allows a security context change without the presence of a password. (In other words Windows does not have a capability similar to the Unix `setuid()` system call.) This password must be changed any time the Windows user account's password is changed.

9.1.1.7 Host Equivalence List

This value is used only with rshd, or rlogind if the user is allowed to logon without a password. If you do not want the rshd or rlogind (without password) functionality, **do not set this field**.

This is a comma-separated list of host names or TCP/IP addresses from which this user account name is allowed to execute commands. Limited wildcards are supported.

Specifically, names of the form "`*.univ.edu`" will allow all host names ending in "`.univ.edu`" execute programs under this user account name. Likewise, an address of the form: `205.238.107.*` allows execution from that TCP/IP subnet. The "`*`" must occur only at the beginning of host names and at the end of TCP/IP addresses. It must be immediately followed/proceeded by a period "`.`".

Example:

```
205.238.107.*,m1.some.com,*.xyz.gov,128.110.163.210
```

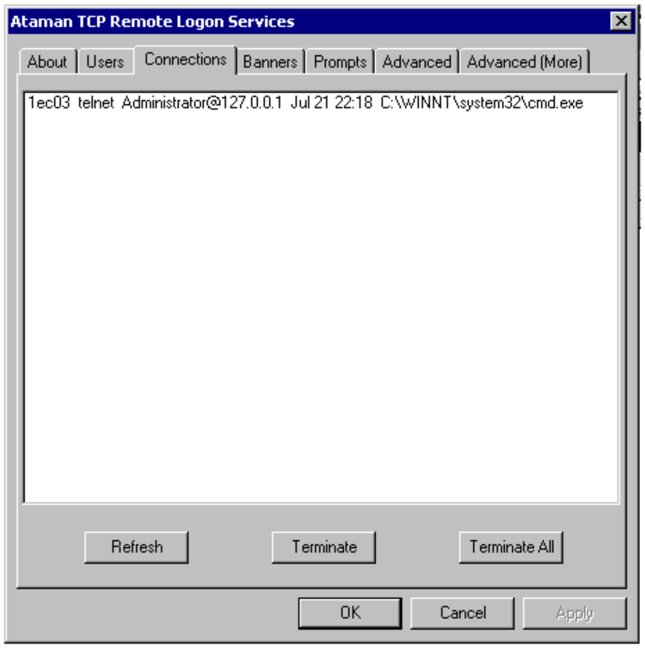
For best security, use only fully specified TCP/IP addresses. (I.e. don't use host names and don't use wildcards.)

Additionally, you can allow a remote account with a different account name access as this user. To do this precede the host entry with "Username@". If you want all accounts to access as this user from a given host, substitute a "*" for "Username".

Example:

```
*@205.238.107.*,brian@ataman.com
```

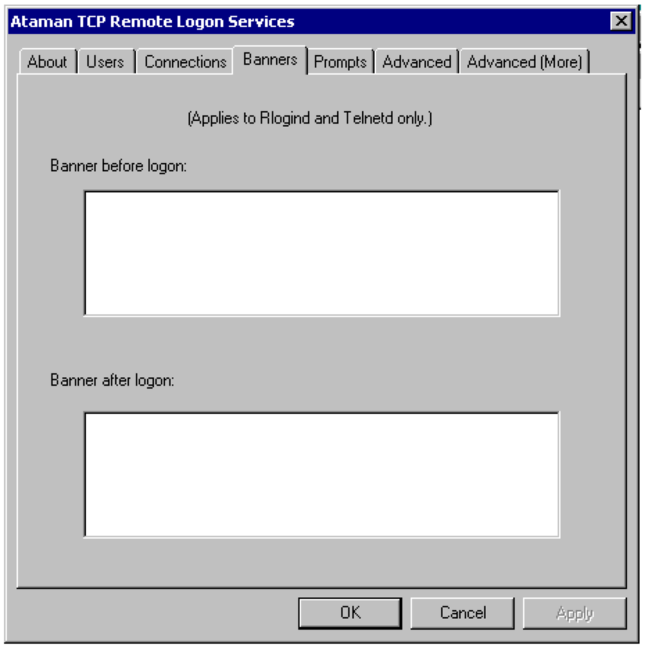
Again, for best security, don't use wildcards.



```
1ec03  telnet  Administrator@127.0.0.1  Jul 21 22:18  C:\WINNT\system32\cmd.exe
```

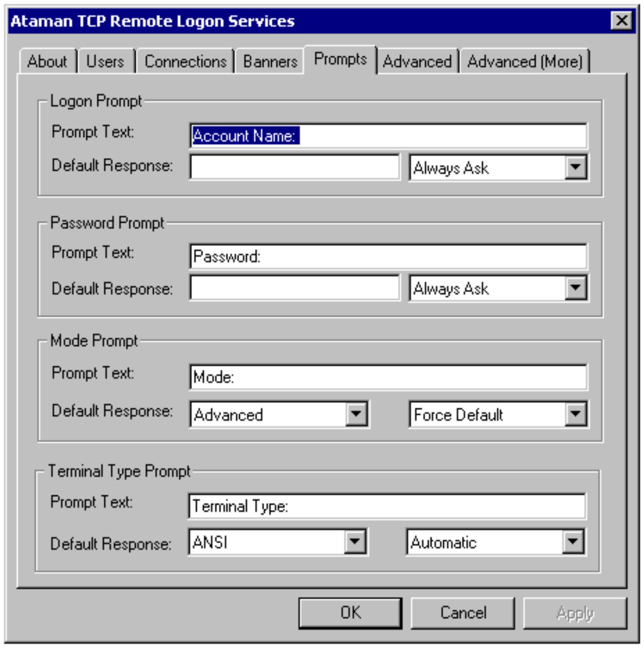
9.1.2 Connections Page

This page shows a list of the commands that are actively executing through the ATRLS. The "Refresh" button is used to update the list of active connections. The "Terminate" button causes the programs associated with a logon session to terminate. The "Terminate All" button causes all current logon sessions to terminate.



9.1.3 Banners Page

This page allows you to enter text that will be displayed to telnetd/rlogind users before and after logon.



Logon Prompt
Prompt Text: Account Name:
Default Response: Always Ask

Password Prompt
Prompt Text: Password:
Default Response: Always Ask

Mode Prompt
Prompt Text: Mode:
Default Response: Advanced Force Default

Terminal Type Prompt
Prompt Text: Terminal Type:
Default Response: ANSI Automatic

9.1.4 Prompts Page

This page controls the presentation and text of telnetd/rlogind prompts to users.

9.1.4.1 Logon Prompt

When the Default Handling is set to “Always Ask”, the user is prompted for a logon account name with the Prompt Text.

When the Default Handling is set to “Force Default”, the user is not prompted. The user name listed as the Default Response will be used.

9.1.4.2 Password Prompt

When the Default Handling is set to “Always Ask”, the user is prompted for a password with the Prompt Text.

When the Default Handling is set to “Force Default”, the user is not prompted. The password listed as the Default Response will be used.

It is very rare that you would want to set the Default Response field for the Password Prompt – in general, this would create a huge security hole. This feature was provided at the request of customers who need the ability to automatically logon all of their users to a fixed application when connecting to the telnet port.

9.1.4.3 Mode Prompt

For an explanation of modes, see the Using the Rlogind and Telnetd Services section below.

When the Default Handling is set to "Always Ask", the user is prompted for the mode with the Prompt Text. If the user responds by just hitting return, the mode listed as the Default Response will be used. The user should choose either Simple or Advanced and hit the return key. (Only a "s" or "a" followed by return has to be typed. Either case can be used.)

When the Default Handling is set to "Force Default", the user is not prompted. The mode listed as the Default Response will be used.

9.1.4.4 Terminal Type Prompt

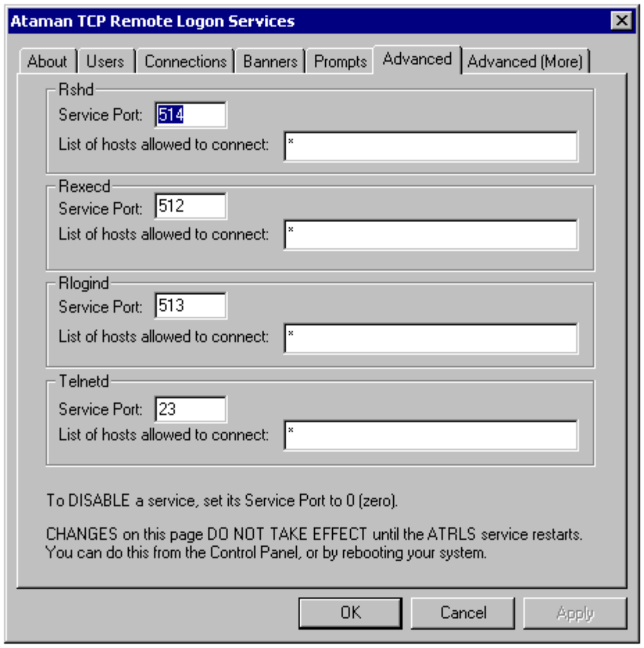
The Terminal Type Prompt processing occurs only if a connection uses Advanced Mode.

The ANSI terminal type should be used if your telnet/rlogin client supports the ANSI-BBS set of terminal escape sequences. (ANSI-BBS is the same set of escape sequences used by ANSI.SYS under DOS and also in OS/2. It uses the OEM character set and has limited color support.) Otherwise, the VTXXX terminal type should be used. This terminal type works with most VT100, VT102, VT220, VT320 and VT420 emulations and with the xterm program found on most Unix systems.

When the Default Handling is set to "Always Ask", the user is prompted for a terminal type. If the user responds by just hitting return, the mode listed as the Default Response will be used. The user should choose either VTXXX or ANSI and hit the return key. (Only a "v" or "a" followed by return has to be typed. Either case can be used.)

When the Default Handling is set to "Automatic", the terminal type sent by the client program is examined. If the terminal type begins with "ansi", then the ANSI terminal type is chosen. If the terminal type begins with "vt", "dec-vt", or "xterm" then the VTXXX type is chosen. Terminal type comparisons are all case-insensitive. Otherwise, the user will be prompted as in the case of "Always Ask" above. The "Automatic" style of terminal type handling cannot be used if all other prompts have been set to "Force Default".

When the Default Handling is set to "Force Default", the user is not prompted. The terminal type listed as the Default Response will be used.



Rshd
Service Port: 514
List of hosts allowed to connect: *

Rexecd
Service Port: 512
List of hosts allowed to connect: *

Rlogind
Service Port: 513
List of hosts allowed to connect: *

Telnetd
Service Port: 23
List of hosts allowed to connect: *

To DISABLE a service, set its Service Port to 0 (zero).

CHANGES on this page DO NOT TAKE EFFECT until the ATRLS service restarts. You can do this from the Control Panel, or by rebooting your system.

9.1.5 Advanced Page

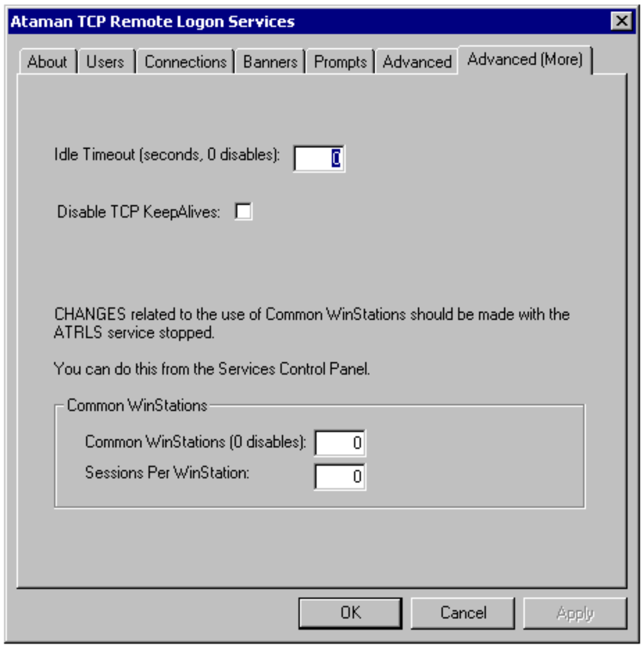
Most users do not need to use the settings on this page.

For each service in the ATRLS:

- “Service Port” is the TCP/IP port number the service will listen on. To disable a service, set this field to 0 (zero).
- “List of hosts allowed to connect” is a comma-separated list of TCP/IP addresses of the hosts allowed to access the service. The default value is “*” which allows all hosts to access the service. Limited wildcards are supported. Specifically, an address of the form: `205.238.107.*` allows execution from that TCP/IP subnet. The “`*`” must occur only at the end of TCP/IP addresses. It must be immediately followed/proceeded by a “`.`”.

  Example:

  ```
  205.238.107.*,128.110.163.210
  ```



Idle Timeout (seconds, 0 disables): 0

Disable TCP KeepAlives: ☐

CHANGES related to the use of Common WinStations should be made with the ATRLS service stopped.

You can do this from the Services Control Panel.

Common WinStations

Common WinStations (0 disables): 0

Sessions Per WinStation: 0

9.1.6 Advanced (More) Page

Most users do not need to use the settings on this page.

For each service in the ATRLS:

- “Idle Timeout” is the number of seconds a session is allowed to be idle before being automatically disconnected. To disable the timeout function, set this field to 0 (zero).
- “Disabling TCP KeepAlives” By default, Telnet and other services normally enable a TCP option that polls the remote host occasionally, to make sure that the connection is still available. Normally, this makes possible the cleanup of broken sessions that otherwise would not be detected. For most users this is very desirable. However, in VERY, VERY busy networks, sometimes this can lead to otherwise good connections being prematurely terminated (normally will happen only in very long running sessions). If you do have such a situation, use this option to disable TCP KeepAlives for all sessions.
- The “Common Winstations” group allows you to increase the number of possible simultaneous connections.

  In Windows security for “GUI objects” (those in User32.DLL and GDI32.DLL) is encapsulated in a securable object called a Window Station. By default, the ATRLS causes the creation of a new Window Station for every connection, giving each logon session its own set of securable GUI objects. While secure, there is a drawback. The number of available Window Stations is limited. Unfortunately, the actual number of Window Stations available varies from release to release of the operating system (including service packs), so we cannot tell you how many connections you can securely have. Generally speaking, if you need less than 10 simultaneous connections you do not want to ever use any of the options in this group.

If you need more simultaneous connections, you can use the settings below, but there is a security caveat. If your telnet users run programs that access the GUI subsystem [for example the clipboard or any window other than a command prompt (console) window], then these programs will open on a Window Station/Desktop combination that is completely open to manipulation by ALL users. Microsoft uses this approach in their Unix Services for Windows product, so presumably it is not extremely unsafe.

To enable the use of common Window Stations set non-zero values in the “Common WinStations” and “Sessions Per WinStation” fields. Barring Windows-imposed resource limits, the total number of possible simultaneous logon sessions will be: the two numbers multiplied together (i.e. “Common WinStations” * “Sessions Per WinStation”).

Using values of “CommonWinStations”:2, “Sessions Per WinStation”:50, we have been able to get 100 simultaneous connections under Windows. Please note that other installed software may also use the underlying resources in Windows, so your results may vary.

If you need even more simultaneous connections, you need to carefully consider one additional step as Windows tends to run out of other resources. A detailed explanation is beyond the scope of this manual. For more information, please see the following Microsoft Technical Notes: `Q142676, Q126962, Q169321, Q184802.` For example, by changing `SharedSection=1024,3072,512 to SharedSection=1024,3072,1024` (“Common WinStations”:4, “Sessions Per WinStation”:50) we were able to get 200 simultaneous connections.

To disable the use of common Window Stations, set “Common WinStations” to 0 (zero).

9.2 Configuring the ATRLS from the Command Line

As an complement to configuration via a Control Panel, the ATRLS come with two programs that allow you to configure the service from the command line: AUSERADM.EXE and ACONFIG.EXE. The AUSERADM.EXE program allows the addition, deletion, editing, and listing of user accounts for the ATRLS. The ACONFIG.EXE program configures the remaining parameters that can also be configured from the ATRLS Control Panel.

Both programs have a “dump” command that you can use to automatically generate “.CMD” that will allow you to easily duplicate configurations on multiple machines. This feature can be a big time saver. Example:

```
auseradm dump >uconfig.cmd
```

Creates a batch command file called uconfig.cmd, containing a series of AUSERADM.EXE command that when executed will duplicate the existing set of users.

To get usage information for either program, execute it with the "help" command (or /?) as in:

```
auseradm help
aconfig /?
```

9.3 Advanced Configuration

9.3.1 Automatic Logon.

It is possible to cause the ATRLS to use one of two forms of automatic logon configuration. We do not recommend the use of these features for most users because:

1. A detailed knowledge of Windows security is necessary to use the feature securely. In particular, if you do not understand well the relationship between Windows Local Groups, Global Groups and User Accounts, you should NOT use this feature. If you do not understand well the difference between a user name on the local machine and a user name in the domain, you should NOT use this feature.
2. Users allowed to logon using this feature are not able to use the Rshd service. Nor are they allowed to Rlogin without a password.
3. It is not possible to support user accounts with the same name, even if they are in different domains. The incoming protocols only support short names… it is not possible to properly support fully qualified Windows user names. If you add users with the same name (directly or indirectly) to the same local group only one will be able to logon. (The choice of which user will appear random.)

Due to the complexity of these features, only limited technical support is available. In particular, we will not help users that do not have the needed understanding of Windows security, gain this understanding. (It is a complex subject, far too complex to explain during the course of a technical support e-mail exchange.)

The configuration parameters for these features CANNOT be set from the ATRLS Control Panel.

If the issues above do not present a problem for you, chose ONE of the following:

9.3.1.1 Allowing any user account to logon without further configuration.

To enable this feature, issue the following command:

```
aconfig AllowAnyAccountToLogon 1
```

You should also set a home directory for these users as follows:

```
aconfig AutomaticLogonHomeDirectory "PathOfHomeDirectory"
```

Where *PathOfHomeDirectory* is the pathname of the home directory that will be given to all users when they logon via the ATRLS.

9.3.1.2 Allowing direct members of a group to logon without further configuration.

To enable this feature, issue the following command:

```
aconfig AutomaticLogonGroup “LocalGroupName”
```

Where *LocalGroupName* is the name of the Windows Local Group whose members are allowed to logon without an individual user configuration. Members of the group must be explicitly listed… indirect lookup of members of groups listed in the group is not done.

You should also set a home directory for these users as follows:

```
aconfig AutomaticLogonHomeDirectory “PathOfHomeDirectory”
```

Where *PathOfHomeDirectory* is the pathname of the home directory that will be given to all members of this group when they logon via the ATRLS.

9.3.2 8-bit characters need to be sent by faulty telnet client.

According to the RFC that governs telnet, by default, the telnet data stream is only 7-bits wide. In version 2.0 we corrected our earlier design to strip characters to 7 bits unless binary mode had been negotiated. Unfortunately, there are telnet clients in the world don’t support 8-bit mode, nor do they refuse to display 8-bit characters showing up in the data stream… so for some customers it may make sense to put are telnet service into a non-standard mode. (For example, the telnet client that comes with Windows versions prior to Windows 2000 is one such faulty telnet client.) To enable this non-standard mode, use the command:

```
aconfig SuppressCharacterStripping 1
```

9.3.3 Suppress cleanup of child processes.

For most users of the ATRLS, it is desirable to have processes executing at the time of an unexpected failure be killed automatically. However, some users want to use the ATRLS to start “background” tasks. We know of no good way to allow this, but we can suppress the normal process cleanup that occurs at the time of session close. If you enable this features, “background” processes will continue to run, but likewise will any processes running at an unexpected process close. In particular, it is important that you always exit or telnet sessions by issuing the “exit” command to CMD.EXE or the CMD.EXE will continue to execute in the “background”. To enable this feature:

```
aconfig DisableChildCleanup 1
```

9.3.4 Programs that use Job Objects may not work under the ATRLS.

Microsoft limits each process to belonging to only one Job Object. Due to problems reading Windows Performance Counters (to determine what processes to terminate) on some systems, we switched to using Job Objects to control process termination on connections that unexpectedly hang up. Few programs use Job Objects, so this generally should not be a problem. Past experience says that someone, somewhere will have a problem with this, so we left the old method of cleaning up stray processes available. To enable this feature:

```
aconfig UseOldStyleCleanup 1
```

9.3.5 Prompt after the banner after logon to give users a chance to read banners.

Several users have requested the ability to move the registration message until after a successful logon. To enable this option:

```
aconfig PromptAfterBannerAfterLogon 1
```

9.3.6 Move print of "registration" banner until after logon.

Several users have requested the ability to move the registration message until after a successful logon. To enable this option:

```
aconfig RegistrationBannerAfterLogon 1
```

9.3.7 Enable Tracing

To control debug information from the various services. Set the option TraceLevel to one of the following:

1. Minimal state information.
2. Some protocol details.
3. Most protocol details.
4. Multiple messages per character transmitted. (You probably don't want to use this one.)

To disable the tracing set TraceLevel to 0. The debug information is logged to the file: `tracelog.txt` in the directory where you installed the ATRLS.

Example:

```
aconfig TraceLevel 3
```

9.3.8 Sending Bells in Advanced Mode

You can reserve a character in Advanced Mode that will be translated to a bell character. The argument to BellCharacter is a decimal number representing the character that will cause a bell character to be sent to the telnet/rlogind client. On the resulting client screen, the BellCharacter (in addition to sounding the bell) will appears as a space character. See the Programming Considerations section for the details of how to use this feature.

In many situations a good character to use is 255 (this is the non-breakable space character used by old word processors… not usually used anymore).

To disable this option, set BellCharacter to zero.

Example:

```
aconfig BellCharacter 255
```

9.3.9 Suppress Successful Logon Messages.

Some users do not want a record of successful logons or command executions in the event log.

To enable this option:

```
aconfig SuppressSuccessfulLogonMessages 1
```

9.3.10 Suppress Event Log Messages about Telnet Probes.

Some management packages detect the presence of a telnet server on your system by opening, then immediately closing a connection to the telnet service. This same "probe" is also used by people attempting to break into your system. By default we flag this as an error. Some users want to allow the probes with no generated messages.

To enable this option:

```
aconfig SuppressProbeMessages 1
```

9.4 Using the Rlogind and Telnetd Services

See the "Prompts Page" subsection of the Configuring the ATRLS section for an explanation of the various user prompts and associated options.

9.4.1 Simple vs. Advanced

The telnetd and rlogind of the ATRLS work in two modes: simple and advanced.

Advanced Mode is the most powerful, but incurs more overhead for each logon session. Simple Mode is very low overhead and works well with custom software.

9.4.1.1 Advanced Mode

This mode allows you to run full-screen console programs such as text editors. In order to use this feature your client program must support ANSI terminal escape sequences. Most terminal emulation programs use ANSI escape sequences. You may use the Advanced Mode if you run a program emulating a VTXXX terminal or when using the "`rlogin`" or "`telnet`" programs from inside the "`xterm`" program found on most Unix systems.

The full range of DOSKEY-style command line editing is available in Advanced Mode. See the Sending Special Keys section below for information about how to send keys such as “Home”.

Due to the way Advanced Mode works and since the Win32 API does not provide good facilities for remote logon, the ^S and Pause keys do not suspend output as they do in a local command prompt window. If you are issuing a command that will have more than one screen of output, piping its output to the Windows “`more`” command is advisable. Example:

```
type longfile.txt | more
```

Advanced Mode works by “polling” the contents of a console window screen buffer and sending across any changes since the last “poll”. For this reason, Advanced Mode is somewhat CPU intensive. Also, this mode is rather data intensive because it frequently redraws your screen. The data intensity is normally only a problem for remote users with a slow modem link.

9.4.1.2 Simple Mode

Simple Mode has very low overhead. It allows you to use most console-mode programs that read from standard input and write to standard output.

Limited command line editing is available in Simple Mode:

<ESC>, ^U	Erase the current line.
^H, ^?	Erase the last character typed.
^C	Interrupt Process (as in CMD.EXE).
^S	Suspend Output (as in CMD.EXE).
^Z	Send End Of File (as in CMD.EXE).

Simple Mode is well suited for use with custom software. See the Programming Considerations section for details.

9.4.2 User Environment

1. When users logon, they will receive their normal user environment and other profile settings

Additionally, the Ataman TCP Remote Logon Services automatically set the following environment variables:

HOME	The path name of the ATRLS-defined home directory of the user. If the user’s home directory is listed as a local path name, the environment variables HOMEDRIVE and HOMEPATH will also be set as in a normal Windows logon.
REMOTEADDRESSS	The IP address of the remote host that made this connection.

TERM	When in Simple Mode, the value passed by your client program is put into the **TERM** environment variable. Advanced Mode works best with programs that use the native Win32 Console API instead of terminal escape sequences. For this reason the **TERM** variable is not set in Advanced Mode.
EMULATION	Set in Advanced Mode to the terminal type selected at logon.
BELL_CHARACTER	Set in Advanced Mode, but only if you have a bell character defined. See Sending Bells in Advanced Mode for details.

9.4.3 Sending Special Keys – (Advanced Mode Only).

The rlogin and telnet protocols are defined using only the ASCII character set. However, many DOS, OS/2 and Windows applications expect the availability of keys defined outside the ASCII set. Unfortunately, there is no ANSI specification for special keys. In place of such a standard, the following sequences were adopted. Ataman hopes they are reasonably easy to generate manually and to remember.

Check the documentation that came with your client rlogin or telnet program. Many such programs contain the ability to create keymaps.

Character Sequence Typed	**Special Character Generated**
`^A^A`	^A
`^Aa`	The next character sent will be sent as an “Alt” character. Example: to send Alt-F1, type: `^A^a^A1`.[*] This sequence may be combined with the Ctrl and Shift sequences. If you need to simulate the press and immediate release of the Alt key, see the sequence `^Az` below.
`^Az`	Simulate the pressing and immediate release of the Alt key. This sequence is needed in many CUA-compliant programs to activate the program's menu bar.
`^Ac`	The next character sent will be sent as a control character. Example: to send Ctrl-F1, type: `^A^c^A1`.[*] This sequence may be combined with the Alt and Shift sequences.
`^As`	The next character sent will be sent shifted. Example: to send Shift-F1, type: `^A^s^A1`.[*] This sequence may be combined with the Alt and Ctrl sequences.

---

[*] Due to the many different ways DOS programs handle special keys, Windows is not always able to send the Alt, Ctrl, and Shift modifiers in the manner the program expects. Experimentation with an application on files containing non-critical data is the only way we have found to know whether or not these keys can be reliably sent.

`^A^R`	Causes the screen to be redrawn. For applications that work in line input mode (for example the Command Prompt itself) ^R alone works too.
`^Au`	Up Arrow
`^Ad`	Down Arrow
`^Al`	Left Arrow
`^Ar`	Right Arrow
`^Ai`	Insert
`^Ax`	Delete
`^Ah`	Home
`^Ae`	End
`^Ap`	PageUp (Previous)
`^An`	PageDown (Next)
`^A1`	F1
`^A2`	F2
`^A3`	F3
`^A4`	F4
`^A5`	F5
`^A6`	F6
`^A7`	F7
`^A8`	F8
`^A9`	F9
`^A0`	F10
`^A-`	F11
`^A=`	F12

For the convenience of users that have VT100/VT102/VT220/VT320/VT420 emulators, we also support the limited subset of keys that the emulators provide. In the table below `<CSI>` can be `<ESC>[` or `\x9a` and `<SS3>` can be `<ESC>O` or `\x8f`.

VT Key	**PC Key**	**ANSI Escape Sequence Expected**
<UP ARROW>	<UP ARROW>	<CSI>A
<DOWN ARROW>	<DOWN ARROW>	<CSI>B

<RIGHT ARROW>	<RIGHT ARROW>	<CSI>C
<LEFT ARROW>	<LEFT ARROW>	<CSI>D
<FIND>	<HOME>	<CSI>1~
<INSERT HERE>	<INSERT>	<CSI>2~
<REMOVE>	<DELETE>	<CSI>3~
<SELECT>	<END>	<CSI>4~
<PREVIOUS SCREEN>	<PAGE UP>	<CSI>5~
<NEXT SCREEN>	<PAGE DOWN>	<CSI>6~
<PF1>	<F1>	<SS3>P
<PF2>	<F2>	<SS3>Q
<PF3>	<F3>	<SS3>R
<PF4>	<F4>	<SS3>S
<F5>	<F5>	Not Implemented. On real VTXXX terminals F5 cannot send a sequence, so there is no standard for a corresponding sequence. If you need to send F5, try using ^A5 as in the table above.
<F6>	<F6>	<CSI>17~
<F7>	<F7>	<CSI>18~
<F8>	<F8>	<CSI>19~
<F9>	<F9>	<CSI>20~
<F10>	<F10>	<CSI>21~
<F11>	<F11>	<CSI>23~
<F12>	<F12>	<CSI>24~

Note: Terminal emulators often differ on the definition of VTXXX keys. Therefore, your client program may not match our set. The arrow keys work with nearly all emulators.

9.4.4 Screen buffer size limits.

Due to both the method used to copy the console screen buffer to the remote screen and an internal limit in a Win32 API call used to perform the copy, screen buffer sizes are limited to a maximum of 60 lines and 128 columns. If the Ataman Telnetd Service or the

Ataman Rlogind Service is used via slower links (for example over a 14.4K modem), performance will be best when using smaller screen buffer sizes.

9.4.5 Manual resize necessary when screen buffer size is changed on the server end.

The rlogin and telnet protocols do no allow a server to communicate a window size change to a client. Thus, if you run a DOS or OS/2 character mode application (which can automatically cause screen buffer size changes) or use an application that explicitly changes the size of the screen buffer (i.e. the "mode con" command), you must manually resize the client to match the change in the screen buffer size.

If your rlogin or telnet client program implements resizing the client window, it will be passed to the rlogind/telnetd service. The remote screen buffer will be resized to match the client. To check the size of your remote screen buffer use the "mode con" command with no additional arguments.

9.5 Using the Rshd and Rexecd Services

The Rshd and Rexecd services are used by invoking an rsh or rexec client program of your choosing. The discussion in the User Environment section for the rlogind and telnetd services above applies to the rshd and rexecd services too.

9.6 Using the Command Line-Based Tools

The ATRLS provide 4 command-line-based tools: AWHO.EXE, AKILL.EXE and REXEC.EXE. You will need to cause your PATH variable to point to the directory where these tools are stored, or else switch to that directory when you run them.

The AWHO.EXE program lists the users currently logged-on to the ATRLS. The "-a" flag provides more detail.

The AKILL.EXE program terminates a logon session. It takes a session id argument. The session id arguments are listed by the AWHO.EXE program. Alternatively, the argument `"all"` causes all current logon sessions to be terminated.

The REXEC.EXE program is a client program that works with the rexecd service. The client program is similar to the one provide with Windows except that it has a `-p` flag which allows you to specify the password from the command-line. WARNING: Users with debug privileges will be able to see this password information in running programs. Also be careful with the security settings of any files which store this password information. Execute REXEC.EXE with the `-?` flag for a usage message.

10. Programming Considerations

A substantial number of our clients use the Ataman TCP Remote Logon Services (ATRLS) in conjunction with hand-held radio frequency terminals. These users typically write their own software and need assistance with making their software work with the ATRLS. This section answers most commonly asked questions.

- If you generate your own escape sequences, you want to use Simple Mode. Advanced Mode generates its own escape sequences recreating the contents of the console screen buffer on a remote host. These escape sequences will interfere with escape sequences that you generate.
- It is now possible to send bell characters in Advanced Mode. See Sending Bells in Advanced Mode for information on how to enable this feature.
  The discussion below assumes you have chosen 255 as your bell character.
  Because of the way that Advanced Mode works (screen scraping), you must generate the bell character in a special manner:
  1. The bell character is interpreted only on the top line of the console screen. Thus to use it, you must first position to the top line. This is to prevent repeated beeps caused by line scrolling.
  2. You should avoid character scrolling within the top line of the console screen. If you cause the screen locations containing bells to move, the bell characters may be resent.
  3. Because Advanced Mode works by screen scraping, it is important that you change a screen location each time you want to sound a bell. The following pseudo-code demonstrates a good way of doing this:

     ```
     static int flip = 0;
     MoveCursor(0, 0);
     if (flip == 0)  then
        print “ \xff”; # \xff in hex is 255 decimal
        flip = 1;
     else
        print “\xff “;
        flip = 0;
     endif
     ```

     The idea is to use two screen locations. Each time a bell is generated, alternating locations are used to print the new bell and to erase the previous bell by overprinting it with a space character.
- Simple Mode works by setting the standard handles to pipes. Some programming environments handle pipes differently from consoles. The most common difference is the tendency to buffer pipes. Please consult the programming environment documentation to discover how to override this buffering, since it will interfere with user interaction. In ANSI C, this is done with the setvbuf call.

  ```
  setvbuf(fp, NULL, _IONBF, 0);
  ```

- Simple Mode is configured from the Prompts Page of the ATRLS Control Panel.

- In Simple Mode, if your program needs to do character-at-a-time input (as opposed to line-at-a-time input), you will need to input using Win32 API calls. **Do not try to use the getch() or kbhit() routines supplied by your compiler vendor. Those routines are usually not designed to work with pipes.**

  The file SAMPLE.CPP contains sample source code of a working program that shows escape sequence output to stdout and character-at-a-time I/O from standard input. This file contains sample _kbhit() and _getch() subroutines using the appropriate Win32 API calls.

11. Troubleshooting / Technical Support

11.1 Where to Begin

11.1.1 Event Log

The ATRLS report error messages to the Application Event Log. This log can be viewed using the Event Viewer application, which is usually launched by double-clicking its icon. The icon is found in the Program Manager group under Administrative Tools. The Application Event Log must have a check mark beside the Log menu entry to be selected.

All ATRLS entries begin with the tag "Ataman". Most of the error messages are self-explanatory. If an error message requires explanation, please do not hesitate to send electronic mail to technical support (`support@ataman.com`).

On rare occasion, you may have a service failure. These are logged by the Service Control Manager in the System Log.

If you are having compatibility problems with a client program, it may help to enable tracing. See the Enable Tracing section for details.

11.2 Known Problems and FAQ

11.2.1 Summary of permissions requirements for the ATRLS.

- CMD.EXE (or other if you have overridden default command processor):
    - Readable by System account.
    - Readable by user accounts used for telnet.
- Home directories of incoming users:
    - Local to machine.
    - Readable by System account.
    - Readable by user.
- Files of the ATRLS:
    - Full Control by System account.

  - Readable by user accounts.
- All paths above must be searchable by telneting user accounts and the System account.
- The ATRLS service must run under the System account.

11.2.2 Remote users use CPU time even when they are idle.

This message occurs only when using rlogind or telnetd in Advanced Mode and is not a problem, but an artifact of providing full-screen support. Windows does not provide inherent remote full-screen support but does provide the ability to read a screen's current contents. Advanced Mode examines the screen periodically to monitor any changes. If the CPU usage by remote users produces a heavy load than consider switching some remote users to Simple Mode.

11.2.3 User does not have permission to logon.

By default, the Windows Server does not allow normal users to logon. See if this is your problem by using the failing user account to logon to the main console (i.e. logon normally). You can override Microsoft's default by using the User Manager for Domains to give the user account the right "Log on locally" as follows:

- Run the User Manager for Domains program located in the Administrative Tools program group.
- If the server, for which you are setting the rights, is a Primary Domain Controller (PDC) or a Backup Domain Controller (BDC), skip to the next step. Otherwise, it is necessary to tell "User Manager for Domains" that you want to set rights for the server machine, **not** the domain. To do this, select the Select the Domain item in the User menu. In the dialog box where it says "Domain:", type "\\MachineName" where MachineName is the host name of the system, which you wish to edit privileges. When you push the OK button, you can edit the privileges for that system
- Select the User Rights item in the Policies menu.
- Scroll the "Right:" drop-down list until you get to "Log on locally" entry.
- Add those users or groups you wish to allow to logon.
- If your server is not a BDC, you are finished.
- If your server is a BDC, run the Server Manager for Domains program located in the Administrative Tools program group.
- Select the Synchronize with Primary Domain Controller item in the Computer menu.

11.2.4 Running a text editor or program clears the screen and appears to hang.

Probably the program in question is using an Alternate Screen Buffer (a Win32 Console API feature). The ATRLS is unable to read the alternate screen buffer. Common editors with this problem: GNU emacs and some ports of "vi" clones. In both instances, you can find free (but unsupported) ports of similar editors on our web site that do work with our services. (Source code as well as Intel binaries are provided.)

11.2.5 Use of client-side screen “scroll-back” does not work.

This is a side effect of the Advanced Mode. Essentially Advanced Mode takes periodic snapshots of a console screen buffer and sends ANSI escape sequence across the network to duplicate the screen remotely. This is necessary since Windows only provides this method to intercept console I/O. Unfortunately using this method means that scroll-back buffers in the client programs are not used, since scrolling never occurs.

11.2.6 Process terminated immediately, probable CreateProcessAsUser failure.

We cannot give you an exact diagnosis of the problem because Windows does not return an error from the CreateProcessAsUser() API call when the problem occurs. Here are the possibilities that we know of:

1. You are out of some system resources, most commonly, window stations (under Windows Server you may only have about 15). If you suspect this is the problem, see: Increase number of simultaneous connections.
2. The Windows system file: `User32.DLL` had an initialization failure, see Microsoft Knowledge Base Article **`Q142676`** for fix you can make in the Windows registry.

11.2.7 With rshd/rexecd program exits immediately with no output.

Probably it is the same problem as Process terminated immediately, probable CreateProcessAsUser failure.. Because the CreateProcessAsUser() API call does not return an error indication, there is no way for us to warn you that this has occurred.

11.2.8 Rcp of extremely large files fails.

Our rcp server actually does support 63-bits of file size, unfortunately most rcp clients do not support more than 31 bits of file size (2 gigabytes).

There is no real standard for rcp. Should you find a client rcp that claims to support extremely large files but does not work with our rcp server, please let us know!

11.2.9 Version 3.X and newer doesn’t have the Environment File option

Since we now automatically load the user’s registry and environment, the functionality is not useful to most users. However if you do find you really do need that functionality, you can achieve it by:

1. Change the environment file to a series of SET commands.
2. Save it as ENVVARS.CMD in the user’s home directory (other directories can be used if desired).
3. Configure the user’s command processor options as follows:

Interactive Command Processor:

```
%COMSPEC% /K CALL %HOME%\ENVVARS.CMD
```

Batch Command Processor:

```
%COMSPEC% /C CALL HOME%\ENVVARS.CMD&%COMMAND%
```

11.2.10 rshd: (user1/user2@host)

Access denied.User2 needs user1@host added to its Host Equivalent List. See Host Equivalence List for details.

11.2.11 Cannot find control panel icon on an x86-64 machine (XP/2003 64-bit).

The ATRLS are still 32-bit applications. You can find the icon by looking in the Control Panel folder named “View x86 Control Panel Icons”.

11.3 The free version of the ATRLS does not have support.

Please do not contact Ataman Software about support for this version.