SEATTLELAB
COMMUNICATION & SECURITY SOLUTIONS

SLNet

Installation, Configuration, Administration and User Guide

Version 6.7

High Performance Telnet Server for Windows

Published by:

SeatlleLab

www.SeattleLab.com

© SeattleLab Limited 2023. All Rights Reserved.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of SeattleLab.

SLNet®, SLNetRF®, and SeattleLab® are trademarks of SeattleLab. Other brands or products are the trademarks or registered trademarks of their respective holders and should be treated as such.

Any errors, corrections, suggestions, or omissions should be sent to support@seattlelab.co.uk

This page intentionally left blank.

Table of Contents

Introduction **1**
Full-Featured Telnet Service 1
Telnet Security 1
Hand Held RF Device Support 1
SAP Console compatibility 1

Architecture **2**

SLNet Requirements **2**
PDF Reader 2

Upgrading from a Previous Release **3**

SLNet Features **4**
High Performance 4
Access to Windows's Common Command Shell 4
Direct Access to Custom Applications 4
Security 4
Secure Logon 4
Terminal Emulation 4
Logging 4
Remote Administration 4
Disconnects 4

What's New in the SLNet 6.7 Release **5**
Windows 2012, 2016, 2019 and 2022 support 5
64 bit application support 5
Improved performance and reliability 5
Improved support for double byte character set 5
Improved Child Process Monitoring 5
Improved compatibility with applications that read VT escape codes 5

SSH Security **6**

Installation of SLNet 6.7 ..... **7**

Configuration **8**
Service tab 9
Logging: 10
Logon Banner: 11
Status: 11
Stop, Start or Pause SLNet: 11
Alternative Ways to Start and Stop SLNet 11
Console Tab 12
Auto Reconnect: 13
User Prompts: 14
Default User: 14
Auto Logon IPs 15

Enable Auto Logon 15
Add a single IP address and User ID. 16
Add IP addresses using a mask. 16
Add a range of IP addresses and User ID. 17
Edit an auto logon entry. 17
Delete an auto logon entry. 17
Terminal Types Tab 18
Override Service and Console settings 19
User Redirection Characters: 20
User Profiles Tab 22
Override settings 23
Default Shell: 24
Default Start Application: 24
Use Redirection Characters: 24
Include / Exclude Tab 27
Connected Users Tab 28
Send a message 28
License Tab 29
Get a License Key 29
Load a License Key 31
About Tab 32

Logging on to SLNet 33
Logon Welcome Message 33
User Prompts 33
Manual Logon 34
Logon with Telnet Client 34
Logon with PuTTy Client 35
Automatic Logon 36
Automatic Logon by IP address 36
Automatic Reconnect 36
Exiting an SLNet session 36

SLNet Terminal Emulation 37

SLNet Entries in Windows Application Log 38
Logging Message examples 38

SLNet Error Codes 39

Appendix A. Troubleshooting 40
Before You Contact SLNet Technical Support 40
SeattleLab Professional Services 40
Contacting SLNet Specialists 40

Appendix B. Customization 41
Named Pipes 41
Raw Keyboard Mode 44
SLNet and Windows Security 45
Authentication types 45

This page intentionally left blank.

Introduction

SLNet is the most efficient, most flexible and best value telnet server solution available. The architecture, performance and features are robust enough for large enterprises, yet affordable and flexible enough for any small business. .SLNet enables an unlimited number of users to work invisibly and simultaneously on any Windows server without interfering with one another or the server console user. Additionally, SLNet optimizes the telnet protocol between Radio Frequency (RF) devices and Microsoft Windows servers.

SLNet v6.7 has been optimized for x64 windows server operating systems - Windows 2012, 2016, 2019 and 2022.

Full-Featured Telnet Service

- Allows **unlimited concurrent connections** for any device running terminal emulation, including RF devices.
- Users can be placed **directly** into custom applications or access the Common Command Shell.
- Use **existing** character-based and DOS applications to run multi-users without modification.
- Written in open Internet standards to work with any telnet client. **No proprietary** client software is required.
- Remote users can work invisibly and simultaneously on the same Windows 2012, 2016, 2019 or 2022 server without interfering with the server console user.
- Features two levels of 'time-out' including: the **clean-up** of processes when a connection is dropped and **inactivity time-outs** to log clients off after a specified period of no activity.
- System administrators can include more information such as security warnings for logon banners through the multi-line banner support.
- Gives administrators the ability to **customize the logon and password prompts**.
- Allows administrators to **send messages** to logged on users without interrupting their work flow.

Telnet Security

- Completely **preserves** Windows security protocols.
- Requires user authentication by the **Windows security** authority.
- Gives administrators the ability to configure SLNet **permissions** on a user-by-user basis.
- **Restricts** logon rights to users with "log-on locally" permissions.
- Gives users the same permissions as if they were working at the console.
- Allow only **specified** users to log on via Telnet.
- **Restricts** logon rights to specific IP addresses or address ranges.
- Allows the logon sequence to be **encrypted**

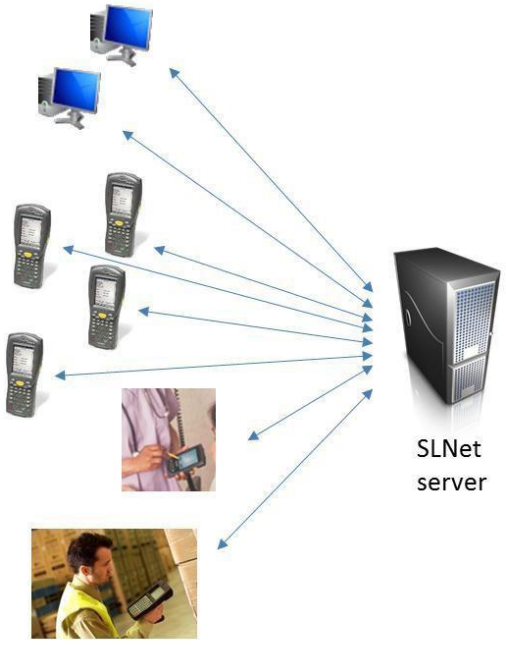
Hand Held RF Device Support

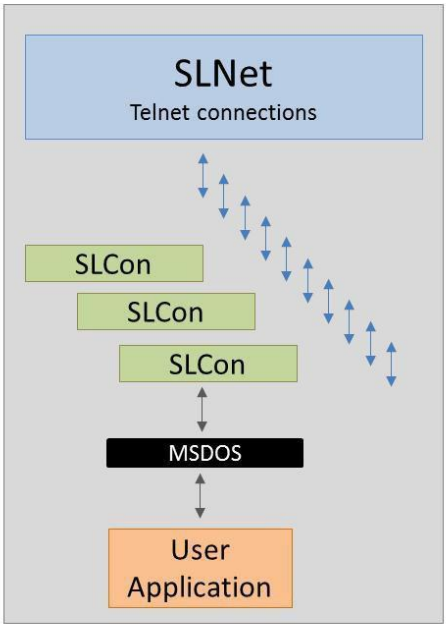
- SLNet allows **unlimited, concurrent RF devices** connections.
- Ready for real-time use with any wireless radio frequency (RF) device that supports the telnet protocol.
- Users can access Windows 2012, 2016, 2019 and 2022 hosts from a device.
- Support small screen sizes.
- **No extensive application rewrites** or the investment of learning new and cumbersome wireless APIs.

SAP Console compatibility

SLNet is the standard for interfacing to the SAP Console from RF devices. It is in use at major organizations worldwide. They chose SLNet because of its outstanding performance and reliability. It installs in a few minutes and delivers a powerful solution for organizations from a few people to thousands.

Architecture





The SLNet server handles a large number of connections (thousands) by handling all connections, terminal emulation and character translation through an SLCon.exe for each user that connects up to the user's application through an MS-DOS prompt.

SLNet Requirements

SLNet 6.7 supports Windows Server 2012, 2016, 2019 and 2022.

The SLNet service itself uses only a minimal amount of system resources, requiring four megabytes of disk space in your root system directory (the directory where Windows is installed on your system) and 2 megabytes of system RAM for the initial connection.

PDF Reader

You will need to install software such as Adobe Reader or Foxit Reader to view the help documentation (PDF format).

Upgrading from a Previous Release

If you are a previous SLNet user of 2.6, 4.1, 5.0, 5.3, 6.0, 6.2 or 6.3 versions, you should upgrade to the 6.7 release for improved operation and security, additional platform support, and new features.

If upgrading from SLNet versions up to and including 6.0, and the installation runs on Windows Server 2008 or earlier Windows Server versions, you must install SLNet 6.7 onto a new server which is either Windows Server 2012, 2016, 2019 or 2022**.** To transfer your settings from the previous version, export the following Windows registry key:

Previous versions - 32 bit server registry location: *HKLM\Software\Seattlelab\SLNet*
Previous versions - 64 bit server registry location: *HKLM\Software\Wow6432Node\Seattle Lab\SLNet*

SLNet 6.7 registry is located at *HKLM\Software\Seattlelab\SLNet* , so if exporting from a 64 bit server ensure you edit the registry file with the correct path before importing.

You can upgrade on the same server if the earlier SLNet version is running on Windows Server 2012, 2016, 2019 or 2022. To upgrade on the same server, stop the Seattle Lab Telnet Service and then install 6.7 over the current installation.

A new installation of SLNet 6.7 will install a temporary license key which gives full functionality for 14 days. To get a permanent license key contact sales@seattlelab.com. Instructions for requesting a license can be found on page 29.

SLNet Features

High Performance
SLNet communicates via Winsock Sockets to the terminals and via Pipes to SLCon and the MSDos Command Prompts. To minimize impact on the CPU resources, the SLNet programs are heavily multi-threaded and asynchronous.

Access to Windows's Common Command Shell
Users who Telnet into SLNet can have full access to Windows' Common Command Shell. SLNet is completely configurable by the administrator, who can choose to disable access to the Command Shell on a user-by-user basis.

Direct Access to Custom Applications
SLNet offers you the power to define default custom applications for users as soon as they log in.

Security
Users run in their own appropriate security context, with all their own permissions in place, just as if they had logged in from the main console. This completely preserves all Windows's C2 security features.

Secure Logon
SLNet encrypts user IDs and passwords for secure logon over the Internet. With SLNet's AuthTypes registry value you can specify the level of encryption (and the encryption protocol) for the entire system, or on a user-by-user basis.

Terminal Emulation
SLNet provides ANSI terminal sequences (compatible with the VT-100 and VT-220) for cursor positioning and formatted screen handling. It supports colors, UTF-8 characters, Unicode characters and custom key sequences.

Logging
SLNet can insert a record in the Windows event log every time a user connects or disconnects along with a record of what process the user executed and the resulting data and screens from that process.

Remote Administration
SLNet can be administered locally or remotely.

Disconnects
In a ***controlled*** disconnect, a telnet packet from the terminal lets SLNet know the terminal is about to disconnect. This allows SLNet to tell the various threads that the events they are waiting for will not be occurring.

In an ***uncontrolled*** disconnect, the threads will either go into a temporary wait mode or exit cleanly and close the sockets, pipes, and MSDOS Command prompt and applications as needed.

A terminal user or device can reconnect securely to a temporarily disconnected application.

What's New in the SLNet 6.7 Release

Windows 2012, 2016, 2019 and 2022 support
SLNet 6.7 has been optimized for x64 windows operating systems and fully supports Windows 2012, 2016, 2019 and 2022 servers

64 bit application support
SLNet 6.7 fully supports 64 bit applications such as SAP Console 7.3 whilst still supporting legacy 32bit applications.

Improved performance and reliability
SLNet 6.7 has improved handling of the pass-through pipe. It also has improved handling of the opening and closing of sessions and orphaned sessions

Improved support for double byte character set
SLNet 6.7 has improved support for International languages that use the double byte character set

Improved Child Process Monitoring
SLNet 6.7 has improved monitoring of any child processes that are spawned by the user's primary application if Child Process Monitoring is enabled. This ensures that all user processes are properly stopped and cleaned up before exiting

Improved compatibility with applications that read VT escape codes
By default, key press output in SLNet 6.7 is translated from Ascii Video Terminal Escape Codes (VT codes) to Windows scan codes. SLNet 6.7 provides functionality to change the key press output to VT code.

SSH Security

The wireless data transmitted between the RF Guns and the Base Station is usually encrypted at the network level using W EP, W PA, W PA-PSK, W PA-PSK2, etc.

In the past this was only way to secure Telnet traffic passing through the network. If you wish to have a secure encrypted connection between the RF device and the SLNet Server we recommend that you install our SSH product - SLNet Secure. With SLNet Secure the connection is protected using the SSH protocol. SSH provides strong encryption and integrity. For further security you can authenticate using SSH key pairs.

SLNet Secure can be installed on the same server as SLNet Server 6.7 and both versions can run in parallel. This means you can migrate RF terminals to the SSH protocol, whilst leaving other devices running as they did previously on Telnet. SLNet Secure requires a separate license key.



Installation of SLNet 6.7

Before installing SLNet 6.7, you must first log on as a member of the Administrator's user group.

Download the SLNet 6.7 installation file from the SeattleLab web site or obtain it from a reseller.

Put it on the local drive on the Windows server.

Right click on the .msi file and select "Install" to start the installation process.

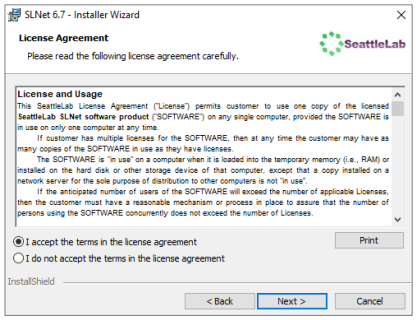


Click ***Next*** to continue.



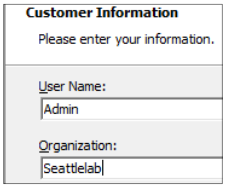
Read the License Agreement.

Check "I accept the terms in the license agreement."



Click ***Next*** to continue.

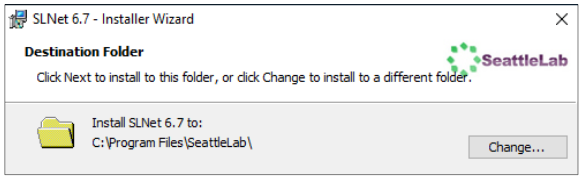
Enter a User Name and Organization.



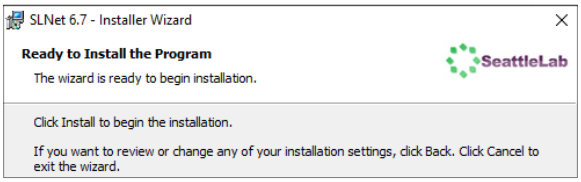
Click ***Next*** to continue.

Change the destination folder for the SLNet software if needed.

Click ***Next*** to continue.



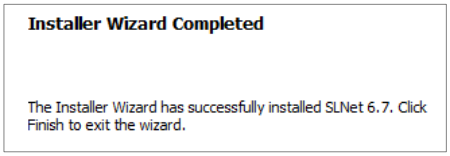
Click ***Install*** to continue.





Installation of SLNet is complete.

Click ***Finish***.

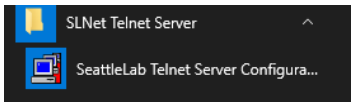




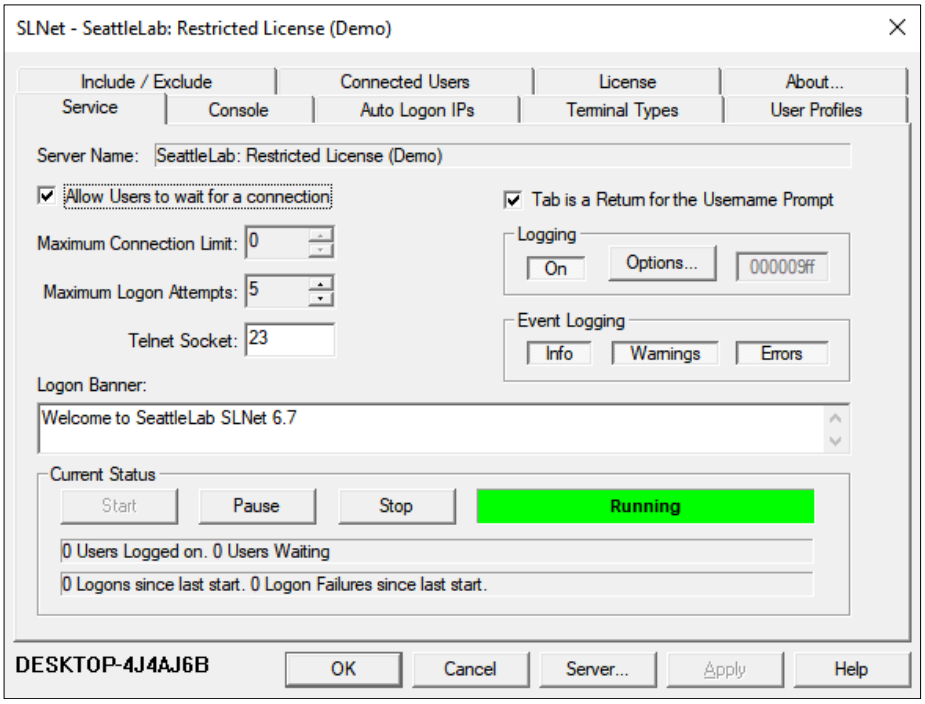
Configuration

SLNet offers a powerful selection of options for customization of your Telnet needs.

Go to Start, Programs, SLNet Telnet Server group. Then select SeattleLab Telnet Server Configuration



The SLNet configuration panel is displayed.

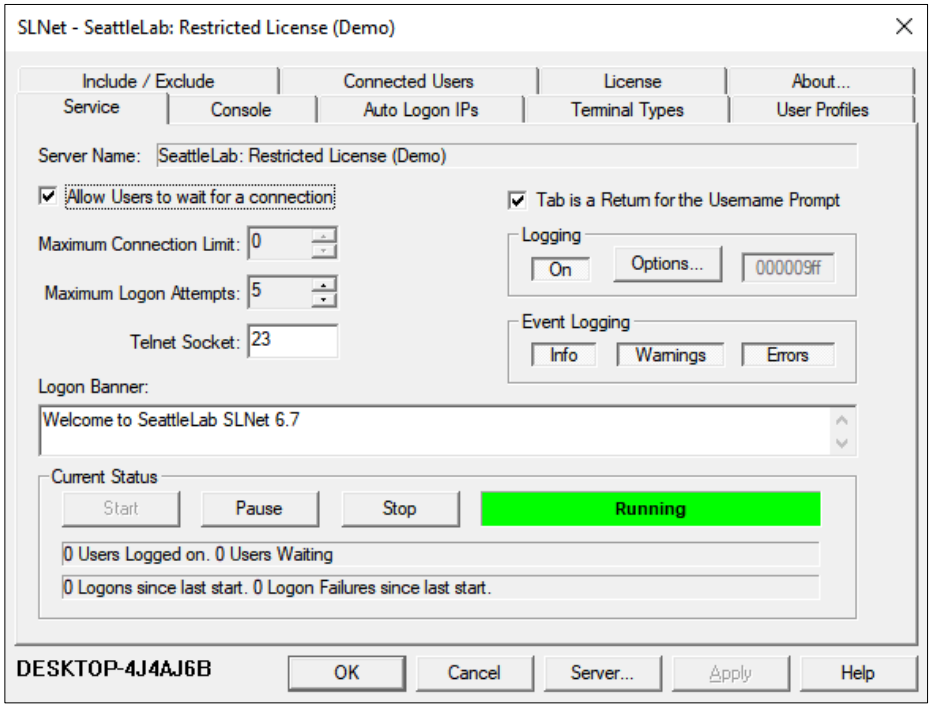


Until you enter a valid license key, SLNet is in a demonstration mode with a temporary license key that expires.

If you see the License Key expires warning prompt, click ***OK*** to continue.

If you have a License Key, click on the License tab to load your valid SLNet License Key file.

Service tab



Server Name:

The Server Name is a description of the server that SLNet is running on. It is a combination of your organization name and the registered server / location name that you supplied to generate a license key.



This is the default Server Name for an unlicensed server. SLNet is installed automatically as a Demonstration license with a limited number of users and a time expiration.

You can change to a different server by clicking on **Server…** button and entering a different server name. The server must be connected to your network. The server's computer name appears on the left bottom of the panel.



User Connections:

You can allow users to wait for a connection if all licenses are in use.



If this box is **Unchecked**, the users will see a message,
`There are currently too many Telnet sessions. Please try again later.`
They will then be returned to the Telnet prompt.

Your license will specify the maximum number of user connections allowed.

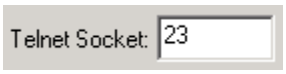


You can change it to a lower number if needed.

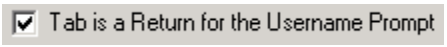
You can set the security limit for logon attempts.



Specify the Telnet socket (port) that SLNet will monitor for connections. *Changes you make to the Telnet Socket in the Service Options do not take effect until you restart the SLNet service.*

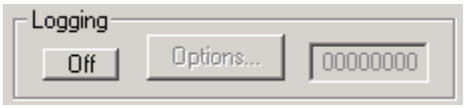


Miscellaneous: The tab key can be set as a return for the user name prompt.



Logging:

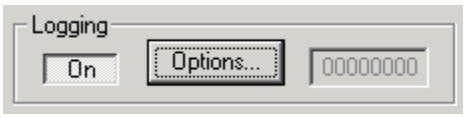
Logging is **Off** by default. *Logging should be left **Off** on production systems unless directed by a SeattleLab technician.*



Click the Logging button to turn **On** logging.

Select **Options** for logging parameters.

The total number of log entries is displayed.

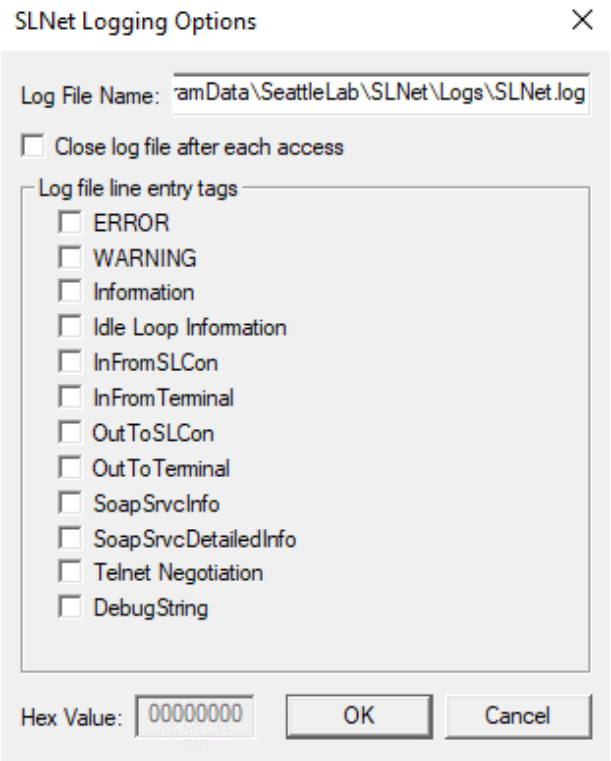


SLNet uses the Windows Application Log in the Event Viewer to keep a record of everything it does by entering messages as events occur.

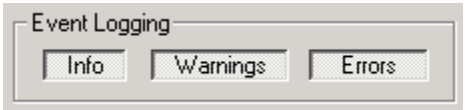
If you set Logging On, you can set which events you want logged.

Check the box next to an item to log its information.

Logs are written to C:\ProgramData\SeattleLab\SLNet\Logs These settings are usually used by SLNet technical support staff to assist you in resolving issues.



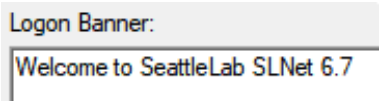
By default, SLNet logs three types of events: Info, Warnings and Errors.
Info messages (such as which user logged on) are the least serious and Errors (such as your demo copy of SLNet has expired) are the most serious.



You can turn off the recording of Events. Toggle each button to turn off its Event recording.

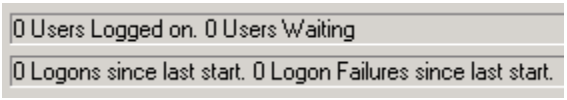
Logon Banner:

You can change the welcome message sent to each device when it connects to SLNet - *maximum number of characters is 2048*.



Status:

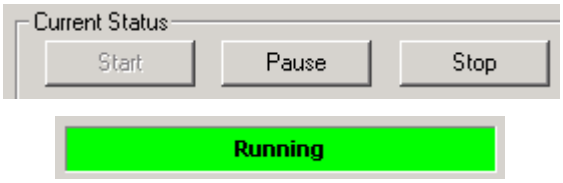
SLNet tracks current active users and any users waiting for connections (may need to increase your licensed amount of users).



Stop, Start or Pause SLNet:

You can Pause, Stop or Start SLNet.

If you stop the SLNet service, the service is completely inactivated. It is removed from memory and is no longer listening on the Telnet socket. Telnet users who attempt to connect to your Windows system will receive a connection refusal message.



Alternative Ways to Start and Stop SLNet

The SLNet service can also be controlled from the Services applet in the system control panel. In the Services scroll box, choose SeattleLab Telnet Server. You may then start, stop or pause the service.

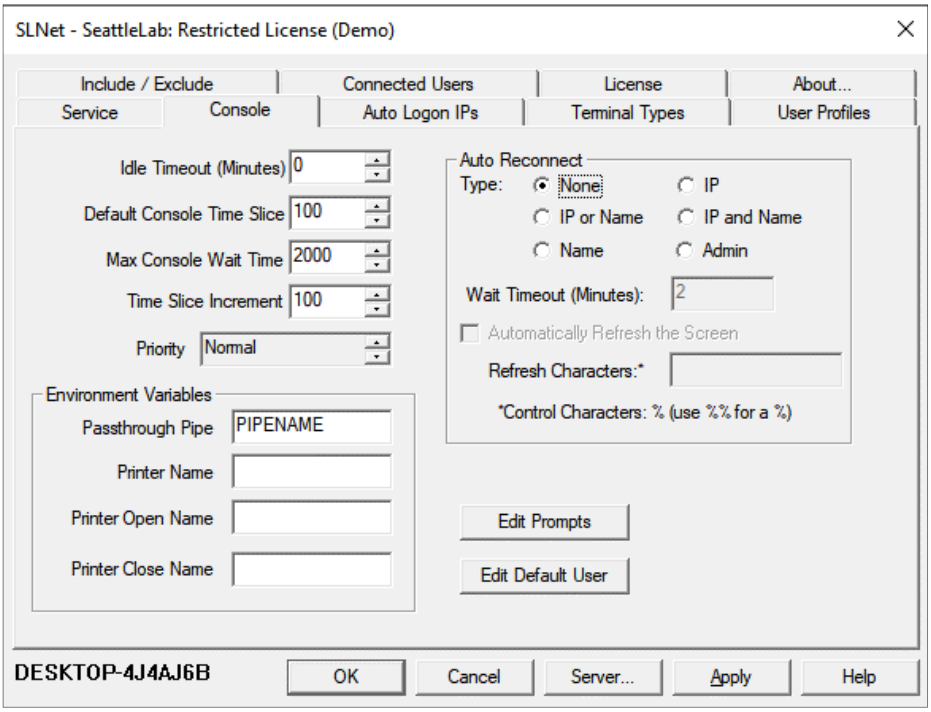


SLNet can also be started or stopped directly from the Windows Common Command Shell.

```
net start slnet
net stop slnet
```

Console Tab

The Console tab of the SLNet Configuration Control Panel allows you to set internal parameters used in the virtual consoles created by the SLNet service.



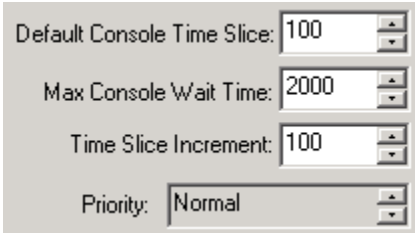
Do not adjust any of these parameters without advice from a trained SeattleLab technician.
Doing so will change performance, latency, and throughput of SLNet and your Windows system as a whole.

You may set Idle Timeout for as many minutes as you choose (0 value means **do not** timeout). If there is no activity from a user in the amount of time entered, SLNet will close the process.

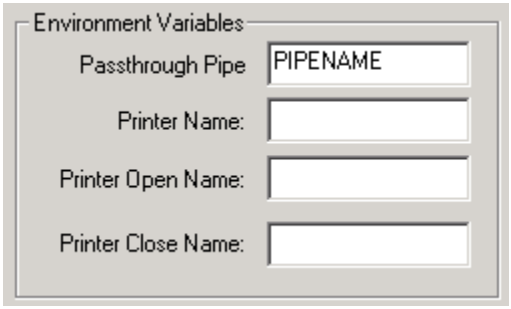


Suppose you cannot back up your system if there are processes running. If your users have a habit of not logging off at the end of the day, you can set this option in order to end the processes so that you can back up your system.

Only adjust these settings with the help of a trained SeattleLab technician.



The SLNet service monitors the screen buffer of a user's console for changes, so that those changes may be sent over Telnet to update the screen of the client. If SLNet monitors these changes too infrequently, the user will experience sluggish and choppy performance. If SLNet monitors those changes too frequently, it will consume too much of the system's resources and reduce the performance of the SLNet user's task and all other tasks running on the system. The algorithm used by SLNet to attempt to optimize these conflicting constraints is to begin polling relatively frequently, and to poll progressively less frequently when no screen activity is detected



Only adjust these settings with the help of a trained SeattleLab technician.

The **Passthrough Pipe** allows you to customize the environment variable name of the main pipe used by SLNet. The default name for this field is PIPENAME.

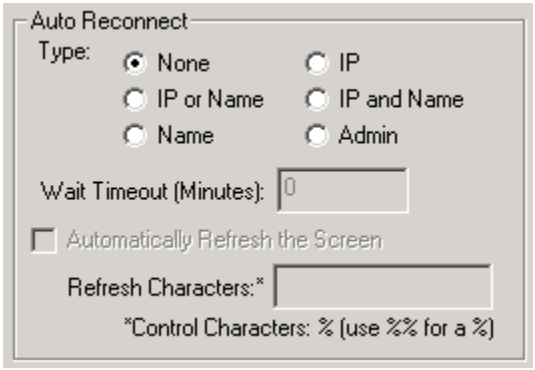
The **Printer Name** is the environment variable to contain the printer name associated with the user or terminal.

The **Printer Open Name** is the environment variable to contain the Escape sequence that enables the user printer.

The **Printer Close Name** is the environment variable to contain the Escape sequence that disables the user printer.

See Appendix B. for customization options.

Auto Reconnect:



SLNet can be configured to automatically reconnect to devices that get disconnected. Set the Auto Reconnect values:

Security - Reconnect only if match on

- None – *no auto reconnect*
- IP address or User Name
- User Name
- IP address
- IP address and User Name
- Admin user

Wait Timeout – keep connection available – then exit and clean up.

To refresh the screen upon a reconnect enter the character string to send to the application.

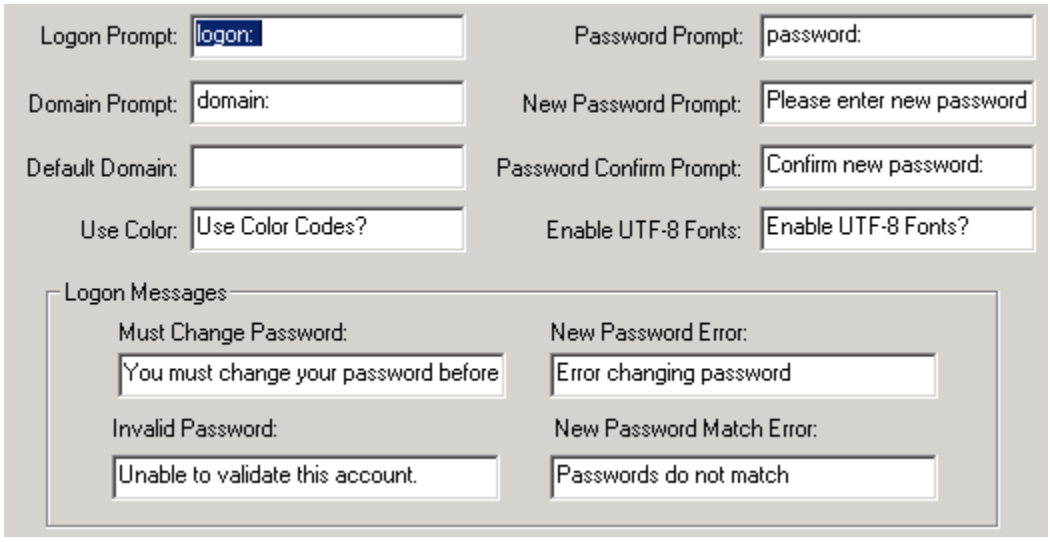
- %R will cause SLNet to redraw the screen from its last saved screen buffer in SLCon.
- You can enter any command that will cause the user's application to refresh the screen.

User Prompts:

You can change any of the SLNet prompts.

If you enter a Windows domain name in the Default Domain box, users will not see a domain prompt and SLNet will access the default domain automatically.





These prompts can be changed to any text (language independent).

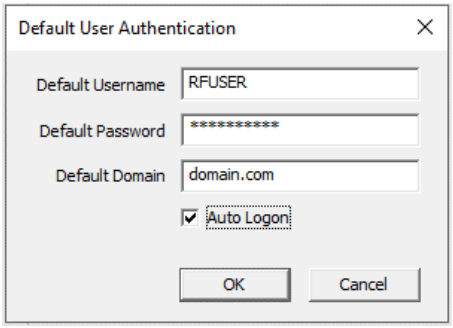
Default User:

If you want all telnet connections to log in with the same Windows user account:

Enable Auto Logon and set the Default Username, Default Password, and Default Domain. All terminals will be logged on as this Windows user account.

You can also set only the Default Domain (leaving Auto Logon disabled) so that your users will not be prompted for the domain during manual logons.

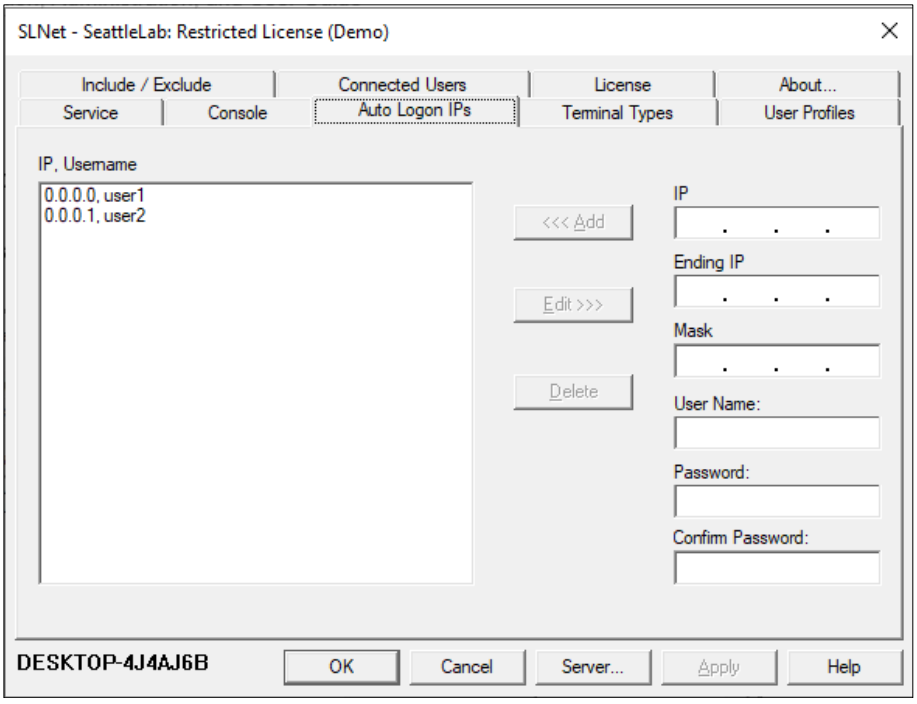




Auto Logon IPs

SLNet can automatically logon a user from a device that connects from a specific IP address or range of addresses.

This enables you to have a shared user ID be used from a single IP address or a secured range of IP addresses.



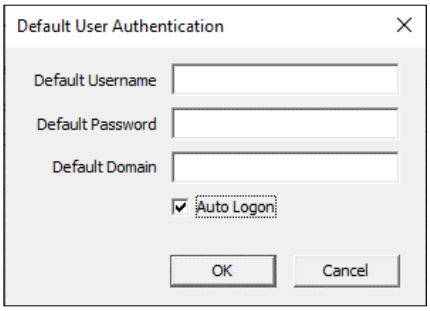
Enable Auto Logon



On the Console tab, click on the ***Default User…*** button.

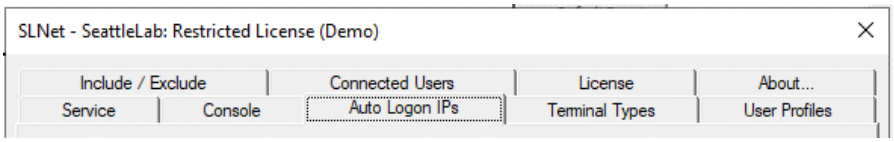


Do not enter any information for the Default Username, Password or Domain.



Check the **Auto Logon** box, click ***OK***.

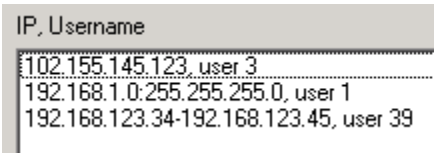
Select the Auto Logon IPs tab.



Use a **single IP address** to log on with a specific user ID.

The format is **`IPaddress,userID`**.

If you need a IP Mask, the format will be `IPaddress:Mask,userID.`



Use a **range of IP addresses** to log on with a specific user ID.

The format is **`IPaddress-IPaddress,userID`**.

Add a single IP address and User ID.

Enter the **IP address**.

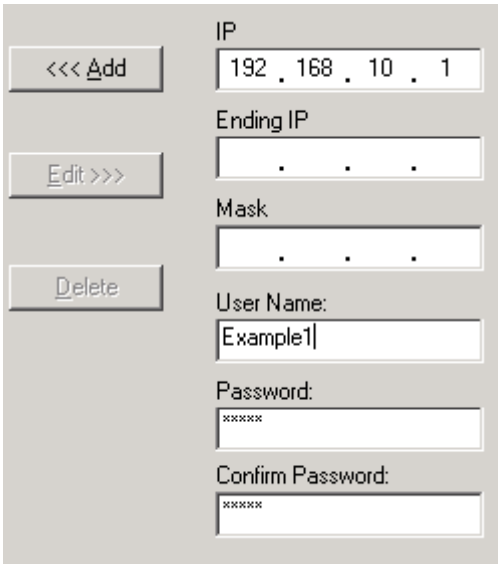
Enter a **User Name**.

Enter a **Password** (not required).

Confirm the Password.

Click the ***<<<Add*** button.





Add IP addresses using a mask.

Enter the starting **IP address**.

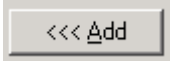
Enter the **mask** to populate the IP address range.

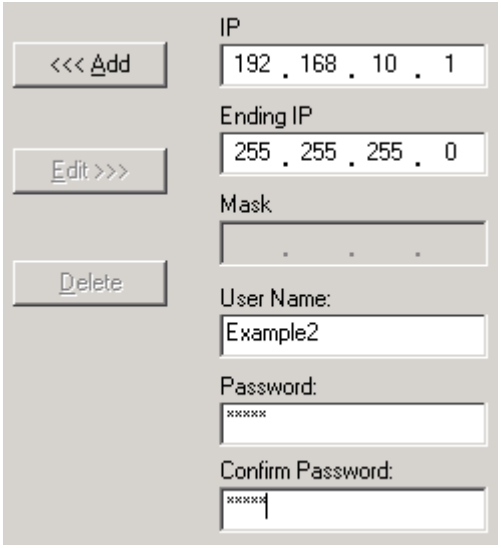
Enter a **User Name**.

Enter a **Password** (not required).

Confirm the Password.

Click the ***<<<Add*** button.





Add a range of IP addresses and User ID.

Enter the beginning **IP address**.

Enter the ending **IP address**.

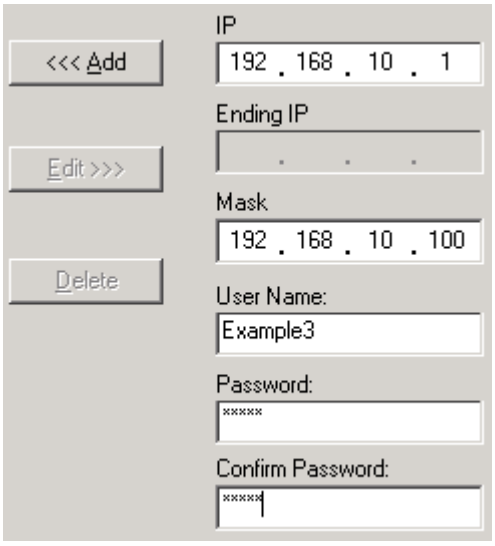
Enter a **User Name**.

Enter a **Password** (not required).

Confirm the Password.

Click the ***<<<Add*** button.





Edit an auto logon entry.

Highlight **IP address**.

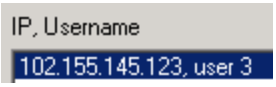
Click the ***<<<Edit*** button.

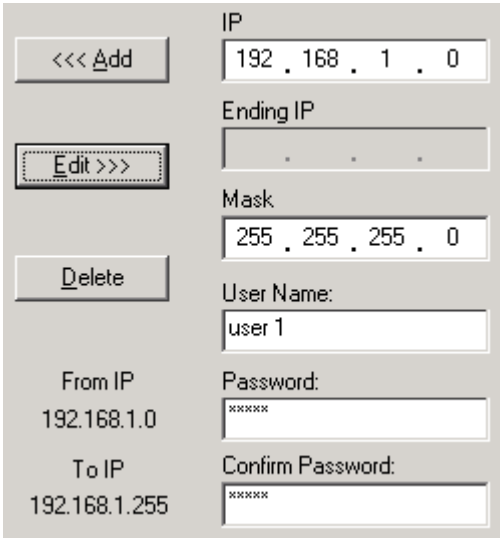


Make any changes.

Click the ***<<<Add*** button.







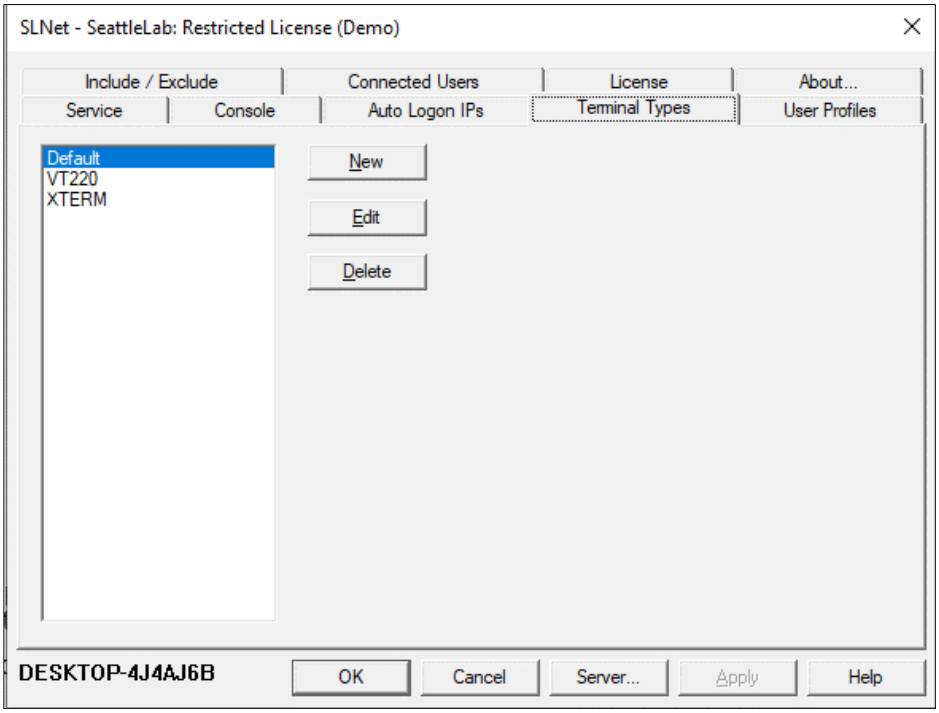
Delete an auto logon entry.

Highlight **IP address**.

Click the ***Delete*** button.



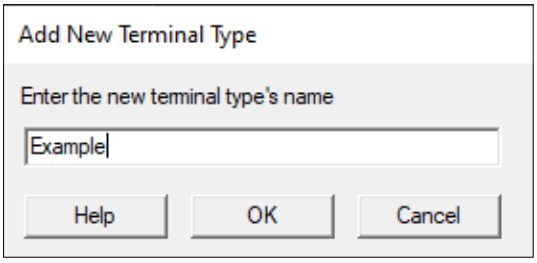
Terminal Types Tab

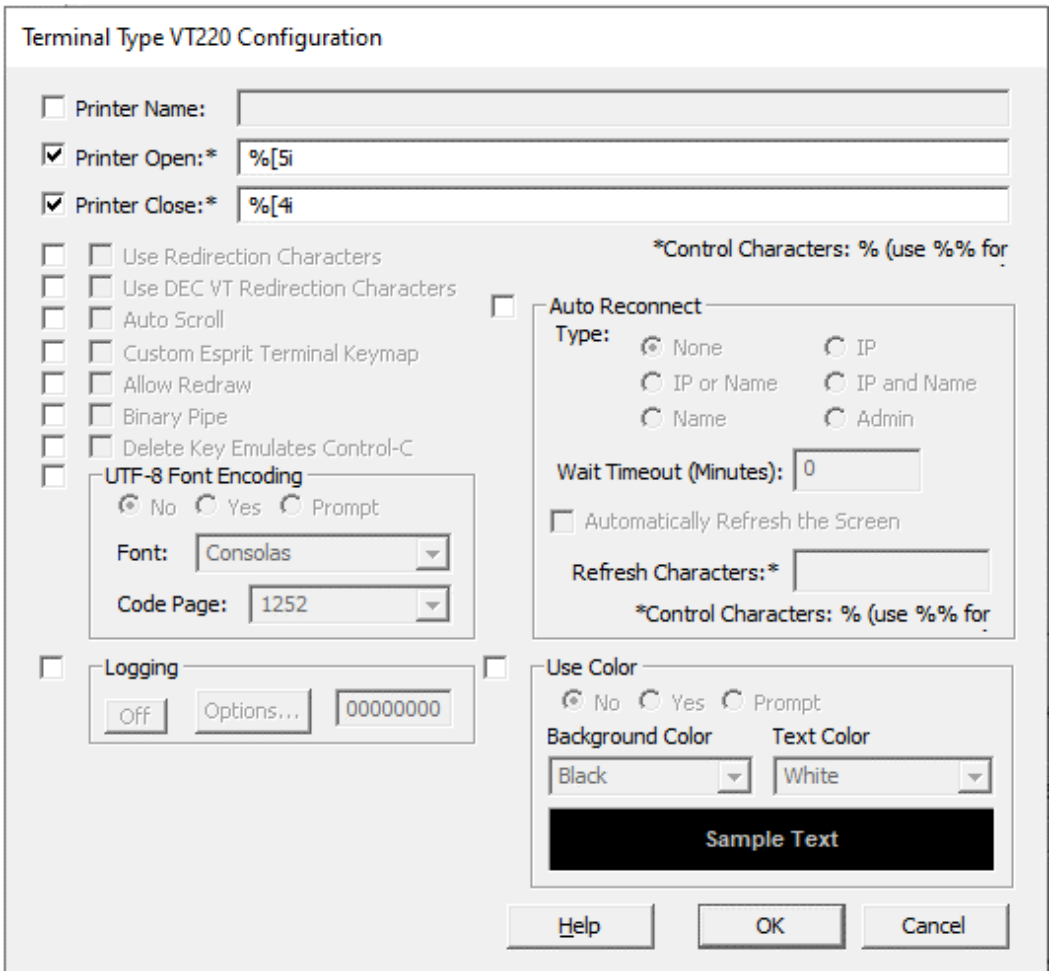


SLNet can support an unlimited number of terminal types. You can set a default terminal type or enter any new terminal types that you need. ***It must match the ID String from the terminal or Port Server.***

The information entered in the terminal settings is passed to the user.

Select a terminal type and click **Edit** to make changes or **Delete** to remove a terminal type from the list. Click **New** to add a new terminal type.





Override Service and Console settings

The flexibility of SLNet allows you to override the default Service and Console settings. The two columns of check boxes allow you to override a setting's value and then be able to activate it.



Not checked and grayed out – use Service or Console setting.



Checked and not grayed out – setting can be modified but is **disabled**.

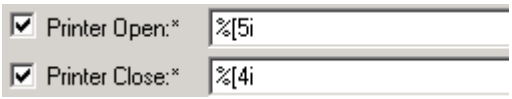


Checked and Checked – setting can be modified and setting is **enabled**.

Set the default printer for this terminal type. Check the box to enter the UNC or IP address



Enter the command string to open the printer or close the printer.



User Redirection Characters:



If this option is enabled, non-printable characters (characters less than 32 decimal) are mapped to printable characters. The 32 non-printable characters will be converted to printable characters based on the following table. When a non- printable character is encountered, the printable equivalent will be substituted in its place. If this option is disabled, all non-printable characters will be represented with spaces.

Decimal	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Non-Printable Character	☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
RNT Equivalent	@	*	n	*	*	$	*	o	*	o	o	o	!	|	*

Decimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Non-Printable Character	►	◄	↕	‼	¶	§	■	↨	↑	↓	→	←	∟	↔	▲	▼
RNT Equivalent	>	<		#	#	$	-	|	^	v	>	<	-	%	^	v

If this is enabled, the screen is cleared if you switch from screen scraping mode to using a custom name pipe.



If this is enabled, all translation of characters in the input stream is disabled. This would be appropriate, for example, for users writing custom communications software such as a file transfer protocol.

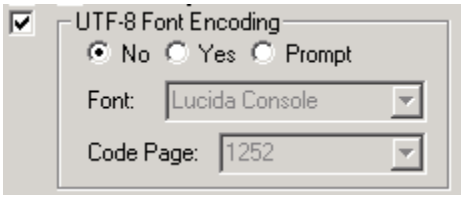


If this is enabled, the Delete key sends a CTRL-C to the application.



Enable UTF-8 fonts. Select a font and a Windows code page to use.

This allows Unicode characters to display on UTF-8 compatible terminals.



Set the Auto Reconnect values:

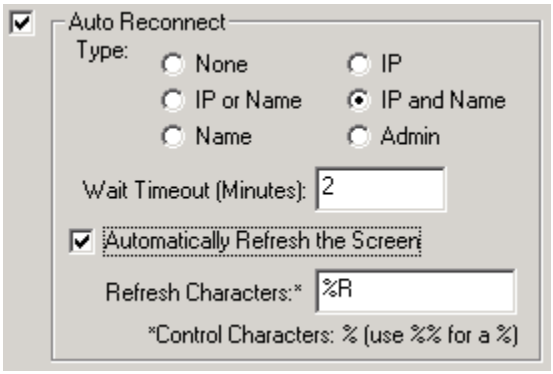
Security - Reconnect only if match on

- None = *No Auto Reconnect*
- IP address or User Name
- User Name
- IP address
- IP address and User Name
- Admin user

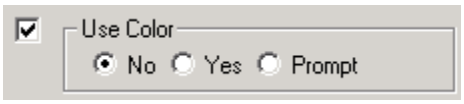
Wait Timeout – keep connection available – then exit and clean up.

To refresh the screen upon a reconnect enter the character string to send to the application.

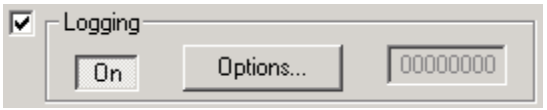
- %R will cause SLNet to redraw the screen from its last saved screen buffer in SLCon.
- You can enter any command that will cause the user's application to refresh the screen.



Activate color.



Set logging on – for all sessions using this terminal type.

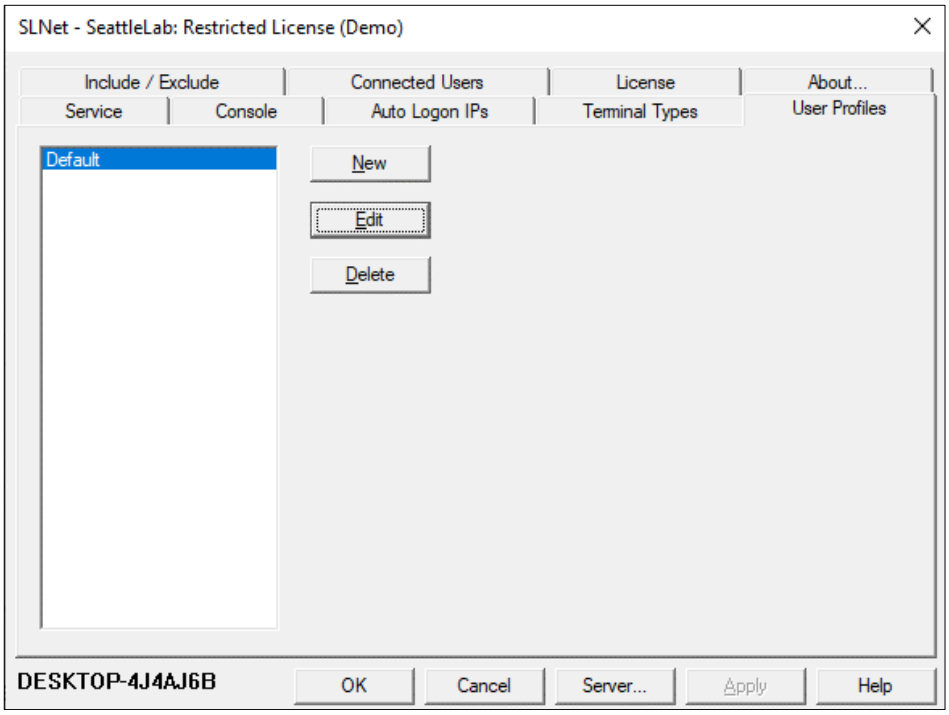


User Profiles Tab

You can add as many custom user profiles to SLNet as needed.

All SLNet users must have an existing Windows user account.

SLNet allows you to define system access on a user-by-user basis. It also allows you to automatically customize the start for each user - for example, all users are presented with the Windows Common Command Shell or SAP Console is started.

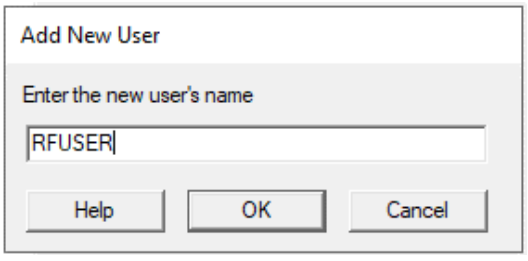


The **Default** user configuration is used for any user who logs onto SLNet but does not have a custom profile. Most organizations use the **Default** user.

Warning! If the Default configuration is ever removed, only individual users who have been specifically added to the SLNet user list will be allowed to log on via SLNet.

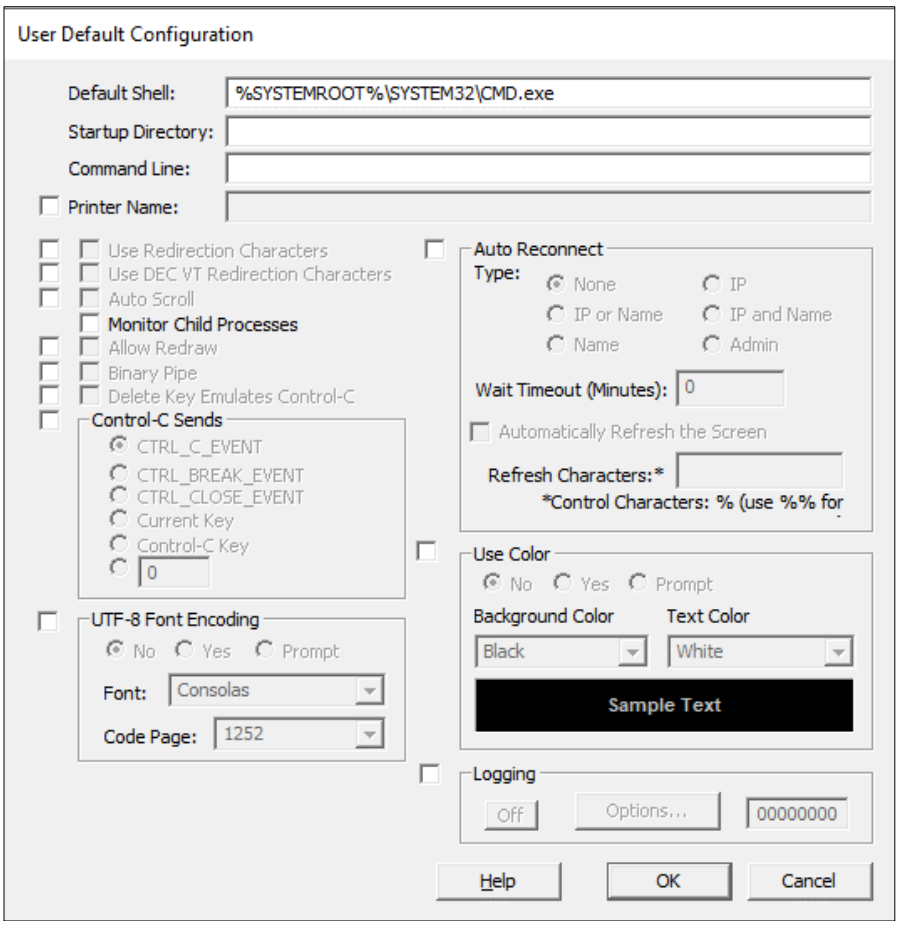
Select a user type and click **Edit** to make changes or **Delete** to remove a user from the list.

Click **New** to add a new user. It must be a valid Windows user account.



If Auto Logon is enabled, then all users will be logged in as the user configured in the Default User Authentication box

If Auto Logon is not selected, then users will be prompted for their Windows / Active Directory User ID and Password.



Override settings

The flexibility of SLNet allows you to override the default Service, Console and Terminal settings. The two columns of check boxes allow you to override a setting's value and then be able to activate it.



Not checked and greyed out – use Service or Console setting.

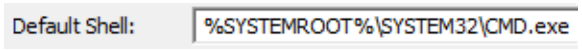


Checked and not greyed out – setting can be modified but is **disabled**.



Checked and Checked – setting can be modified and setting is **enabled**.

Default Shell:



For use with most applications the *Default Shell* field should be left as configured upon installation of SLNet. This default path allows access to the system command shell. Opening the command shell allows console applications to run properly as well as giving access to the prompt in the command shell if desired.

When used with SAPConsole 7.10 and above the *Default Shell* field should be changed to point to sapcnsl.exe.

Default Start Application:



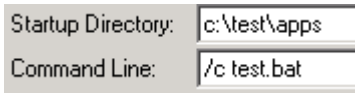
To put users directly into your own custom application, specify the directory and application startup command.

Enter the filename in the text box preceded by a switch command (`/c` or `/k`) followed by any required parameters for the application.

The complete paths *Startup Directory* fields must be entered in the format of either the Universal Naming Convention (UNC) or drive letters.

Paths entered must be LOCAL.

Example:



This is the default printer to use for this user. It will be assigned to the printer name environment variable defined on the Console tab.



Use Redirection Characters:

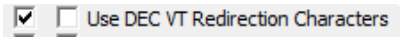


If this option is enabled, non-printable characters (characters less than 32 decimal) are mapped to printable characters. The 32 non-printable characters will be converted to printable characters based on the following table. When a non-printable character is encountered, the printable equivalent will be substituted in its place. If this option is disabled, all non-printable characters will be represented with spaces.

Decimal	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Non-Printable Character	☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
RNT Equivalent	@	*	n	*	*	$	*	o	*	o	o	o	!	|	*

Decimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Non-Printable Character	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
RNT Equivalent	>	<		#	#	$	-	|	^	v	>	<	-	%	^	v

Enable this option to provide DEC VT emulation.



If Auto Scroll option is enabled, when a character is output to the last position on the screen (the 80th column of the 25th row), the screen scrolls up.

Most Telnet clients support auto scrolling. If the Telnet client is set to perform scrolling, the Auto Scroll option should be disabled in SLNet to prevent incorrect screen formatting.



When this option is enabled any child processes that are spawned by the user's primary application will also be monitored by SLNet. With this option, SLNet can ensure that all user processes are properly stopped and cleaned up before exiting.

Some applications will not run if this option is enabled. If an application functions directly from the Windows system console but will not function with SLNet, disable this option.



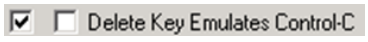
If this is enabled, the screen is cleared if you switch from screen scraping mode to using a custom name pipe.



If this option is enabled, all translation of characters in the input stream is disabled. This would be appropriate, for example, for users writing custom communications software such as a file transfer protocol.

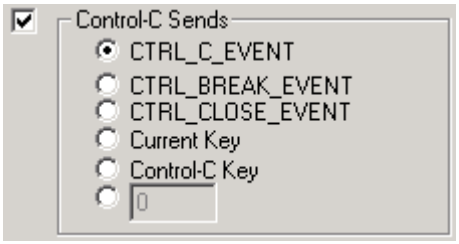


If this is enabled the Delete key to send a CTRL-C to the application.



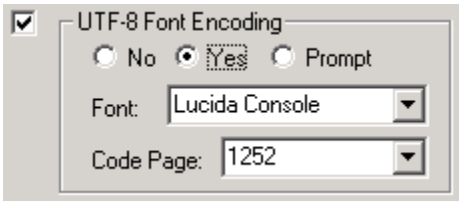
You can customize the Control –C character string passed to the application.

Make a selection or enter an ASCII value.

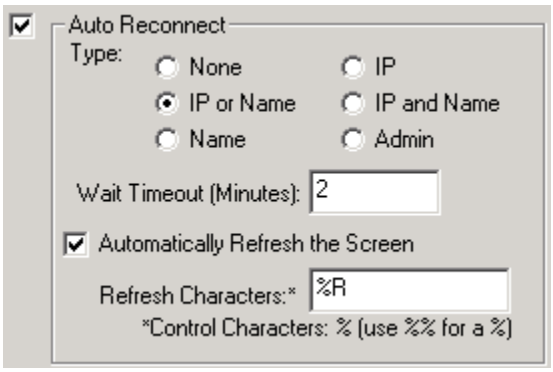


Enable UTF-8 fonts. Select a font and a Windows code page to use.

This allows Unicode characters to display on UTF-8 compatible terminals.



Set the Auto Reconnect values:



Security - Reconnect only if match on

- None – *no auto reconnect*
- IP address or User Name
- User Name
- IP address
- IP address and User Name
- Admin user

Wait Timeout – keep connection available – then exit and clean up.

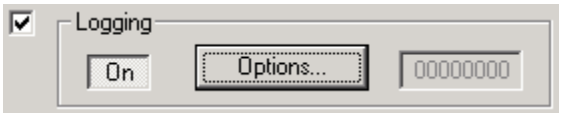
To refresh the screen upon a reconnect enter the character string to send to the application.

- %R will cause SLNet to redraw the screen from its last saved screen buffer in SLCon.
- You can enter any command that will cause the user's application to refresh the screen.

Activate color.

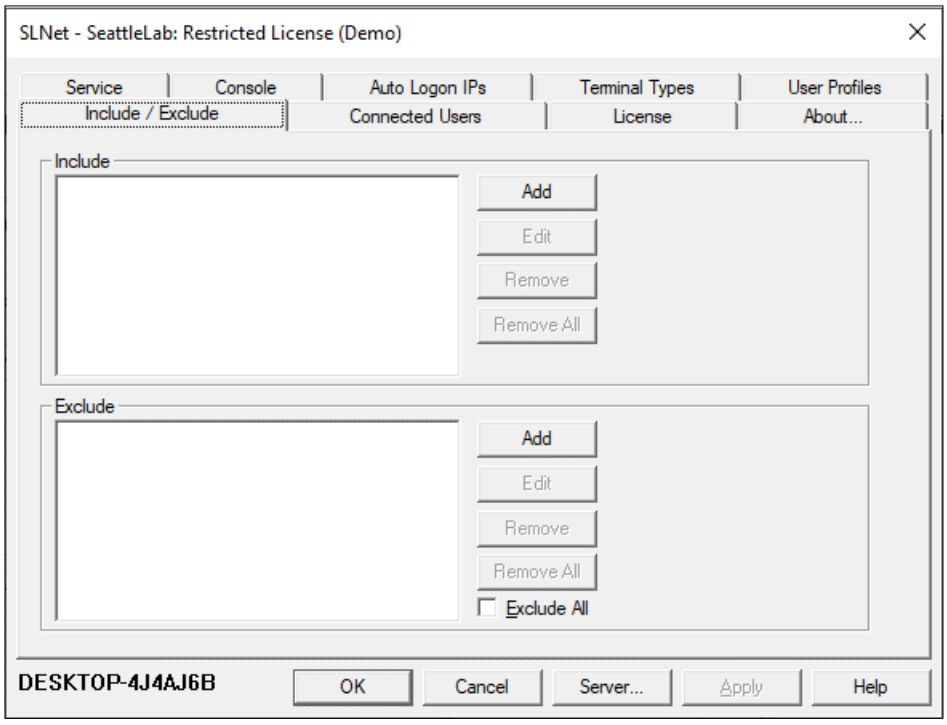


Set logging on for this user.



Include / Exclude Tab

Using the Include / Exclude tab you can control the locations that can access your server – from a single device to a range of addresses.



Click **Add** to add a new device or address range.



Choose **Include** to allow access.

Enter an IP address or range of addresses.

Choose **Exclude** to deny access.

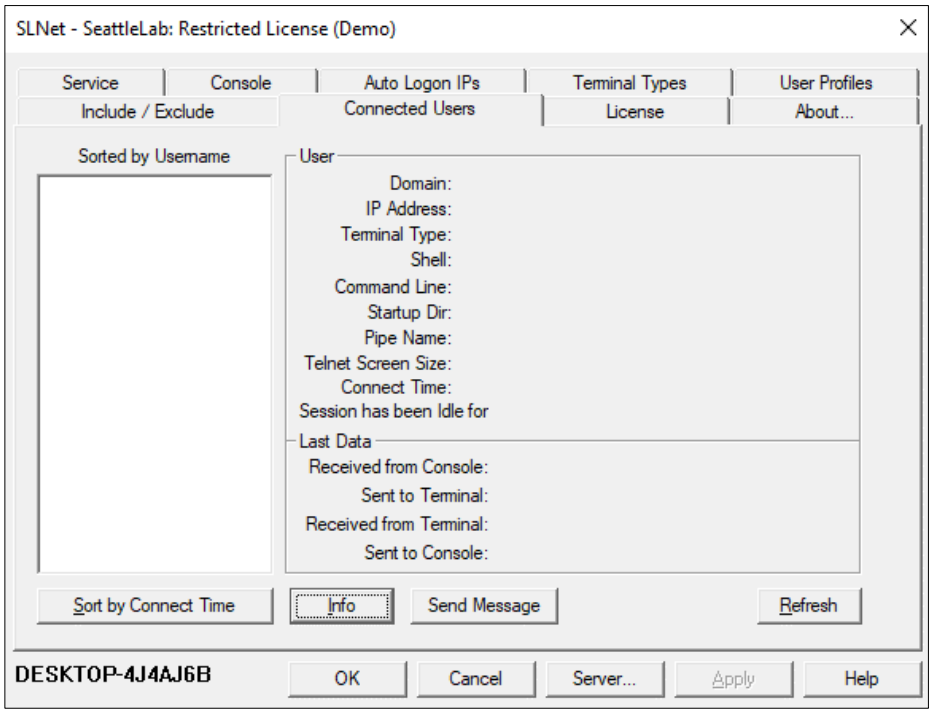
Enter an IP address or range of addresses.



Select a location and click **Edit** to make changes or **Remove** to remove a device or range of addresses from the list.

Connected Users Tab

Using the Connected Users tab you can monitor which users are connected to your server.



Select a user to see the details on that user. Click the ***Info*** button.



By clicking on the Connect Time field
you can switch the displayed time between Local and UTC.

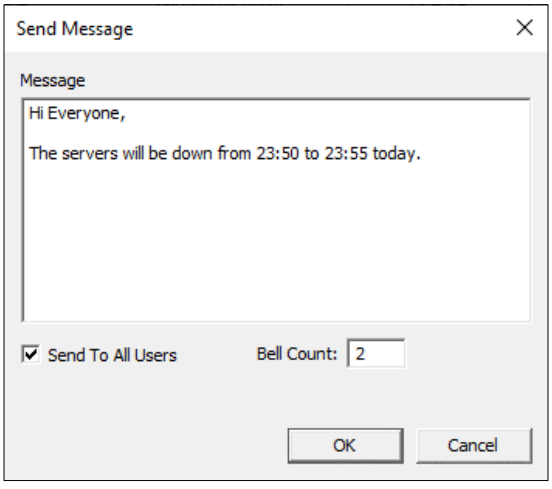
Send a message

You can send a message to any user. Select the user and click the **Send Message** button



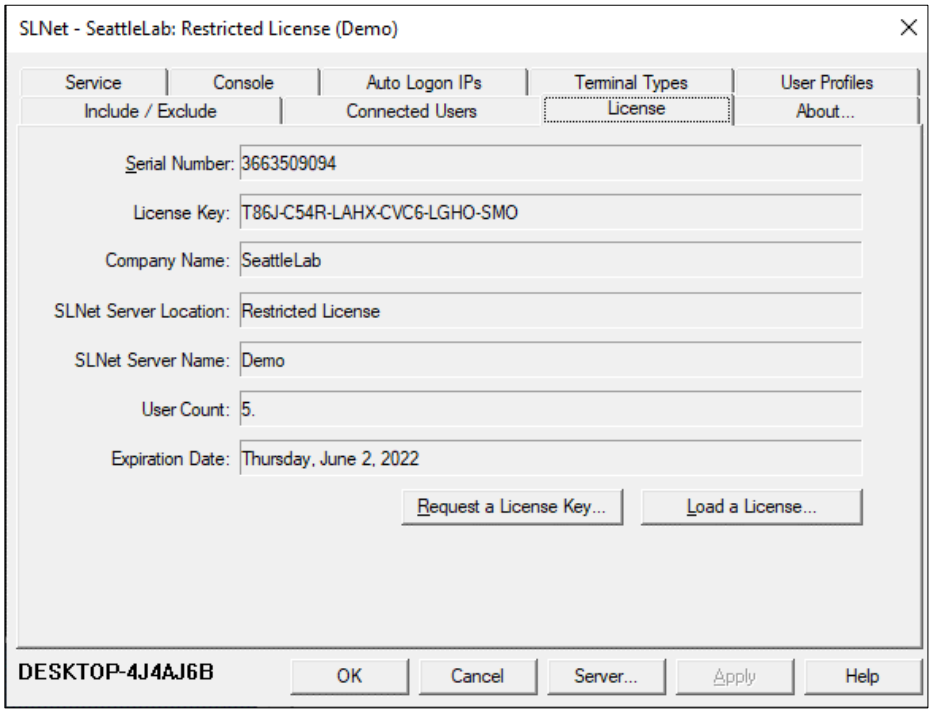
Check the box **Send to All Users** to do a broadcast to all connected users.

Broadcast messages do not interrupt the terminals input flow. The application will continue to execute. Characters sent from the terminal will go to the application even while the broadcast message is displayed.



License Tab

SLNet is licensed by server and the number of concurrent users.



Get a License Key

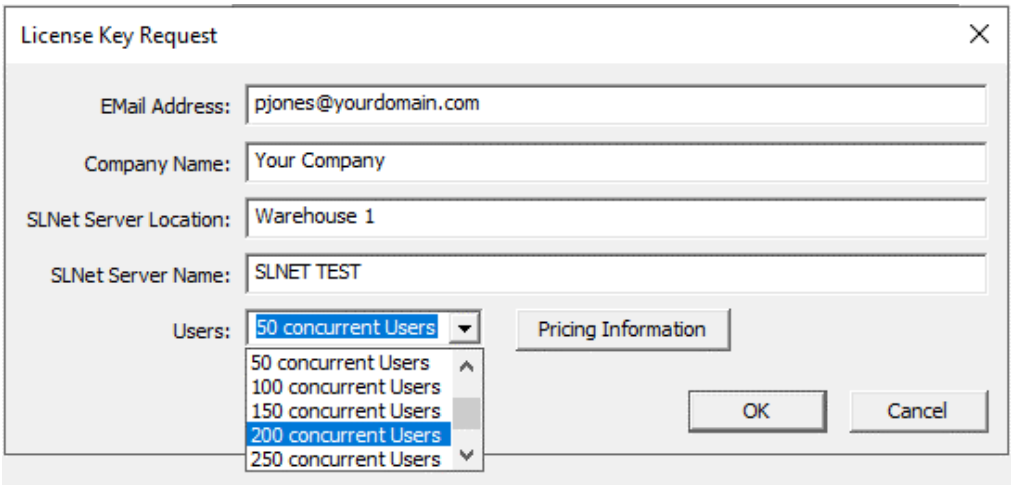
After you have installed SLNet on a server, click the ***Request a License Key*** button 

to generate a file to email to the SeattleLab license department. An email will be returned to you with the license key.

You must pay for your SLNet license first before a license key will be sent to you.

Enter the Email Address that the license key should be sent to.

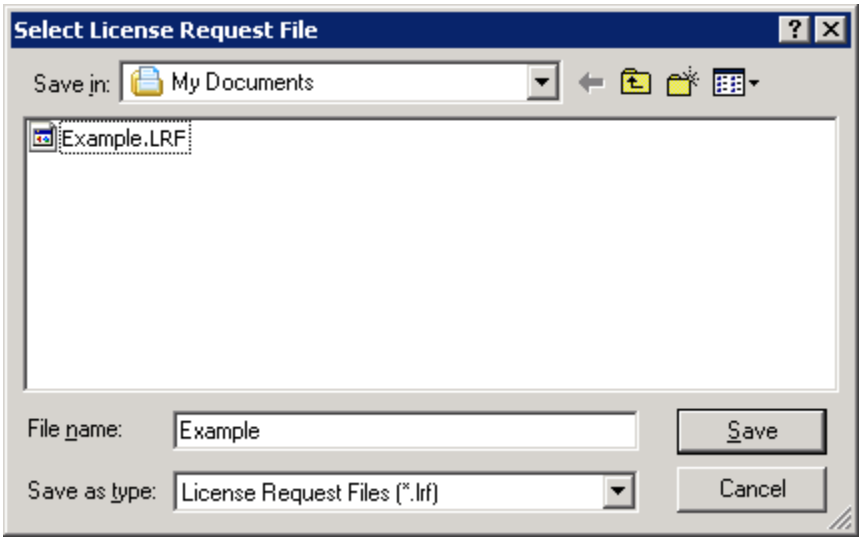
The License Key is based on the **Company Name, Server Location, Server name,** and **User Count**.



Click ***OK*** to save a license request file on this server.

You must pay for your SLNet license first before a license key will be sent to you.

Enter a name for the file, ***Save*** it to this server, and then email that **`.LRF`** file to licensing@seattlelab.com.



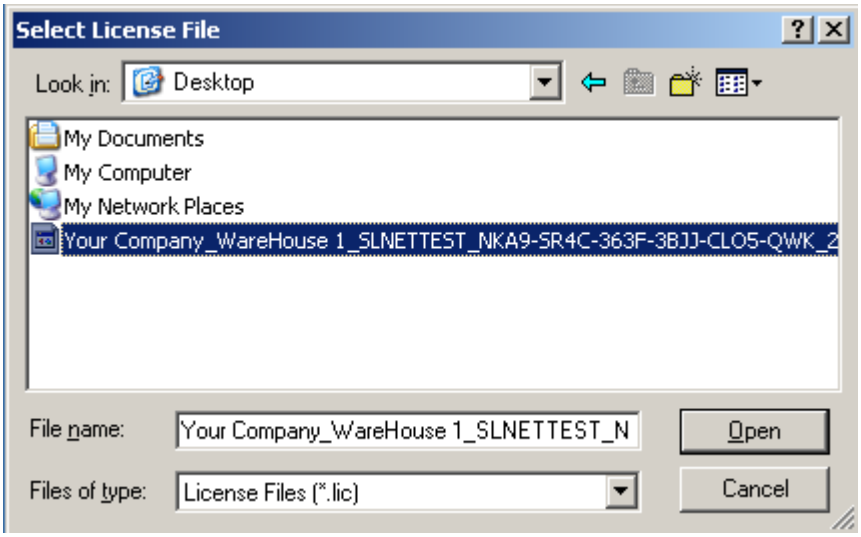
You will receive a **`.LIC`** license file in your email within two business days.

You must pay for your SLNet license first before a license key will be sent to you.

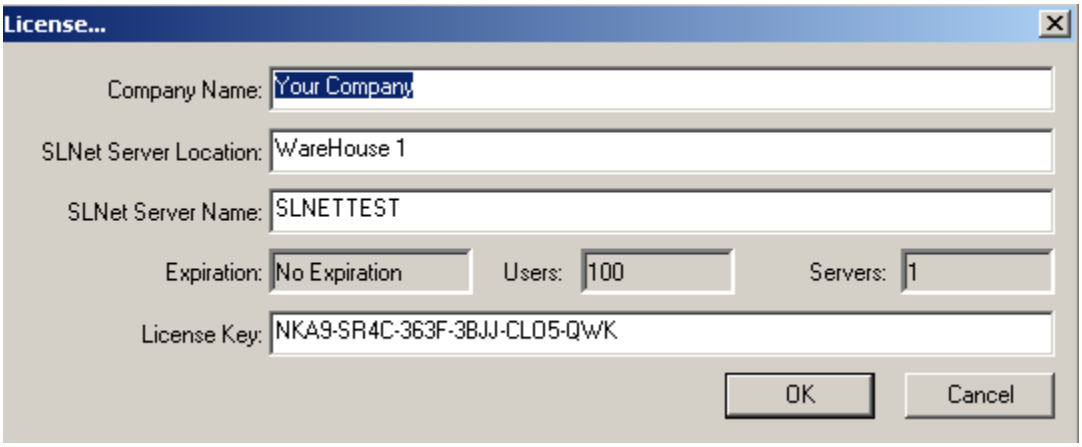
Load a License Key

To load your license file into SLNet, click on the ***Load a License*** button. 

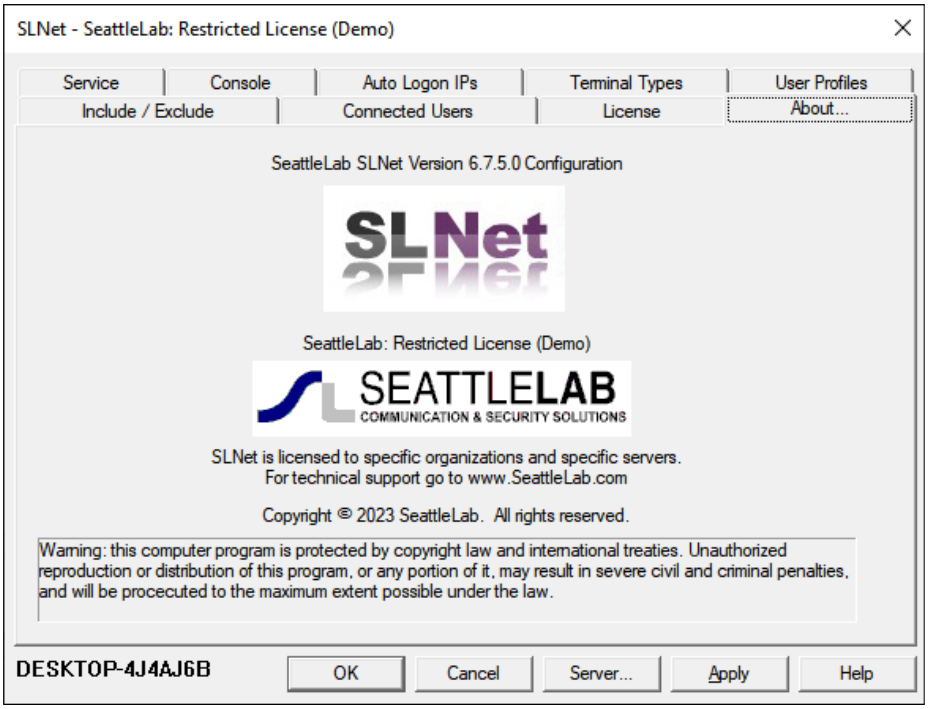
Open the license file (saved from the email you received).



Your license authorization is displayed. Click ***OK***.



About Tab



The About tab displays the SLNet version information. If you have any questions about SLNet then first take a note of your version from this tab.

Logging on to SLNet

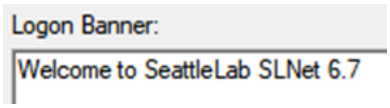
A Telnet client can be any of a variety of devices: a process on a UNIX machine, a Macintosh running MacTCP, a network terminal, a dumb terminal attached to Terminal Server, a Windows terminal emulator, a PC running PuTTY etc.

Many SLNet users connect through handheld terminals to their applications such as warehouse inventory systems, SAP (SLNet usage is recommended in SAP's documentation), etc. These RF devices can emulate a variety of terminal types.

When you initially launch a Telnet client, it is not connected to anything. You need to initiate a Telnet connection. From some Telnet clients, you do so by typing commands at a Telnet command prompt. For example, `connect hostname` or `open hostname` (where hostname is the server name you are connecting to).

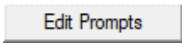
Logon Welcome Message

SLNet will display a logon welcome message to the user. This message can be easily customized on the SLNet Service tab.

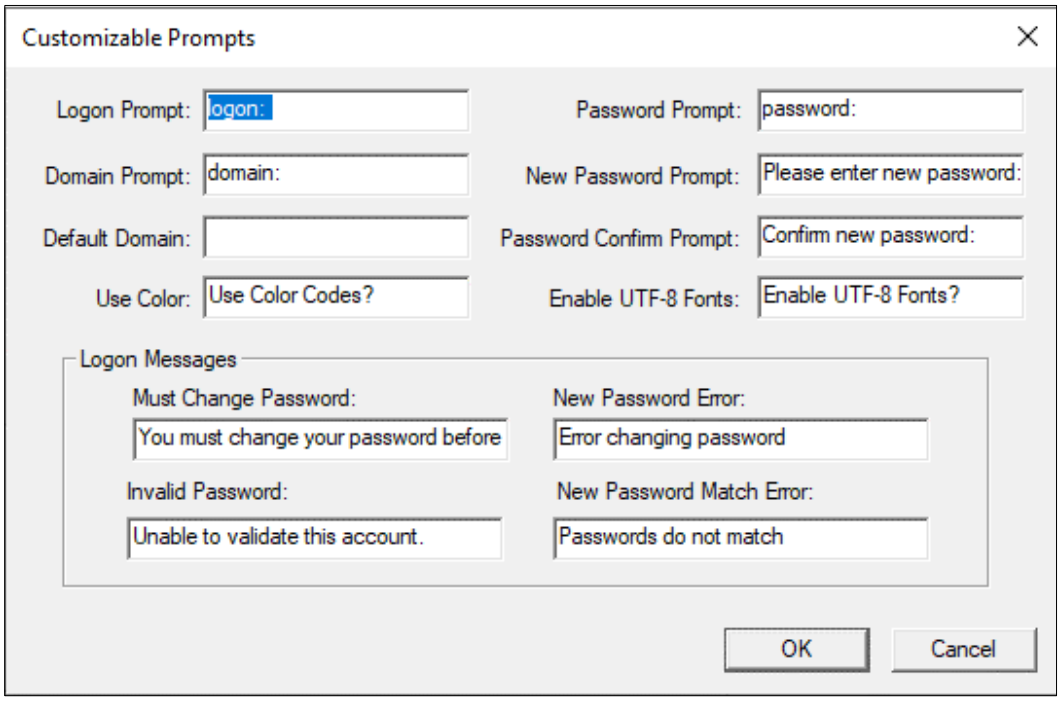


The user will be prompted for their user name and password.

User Prompts

You can change any of the SLNet user prompts to meet your needs. Click on Edit Prompts 

on the Console tab.



SLNet defaults to the domain name on the Customizable Prompts. If the Default is blank, then the user is prompted for a domain name.

A user can log onto another domain by entering the domain name before the user name at the logon prompt, separated by a back slash (\). For example: `PRODUCTION\MYUSERID`.

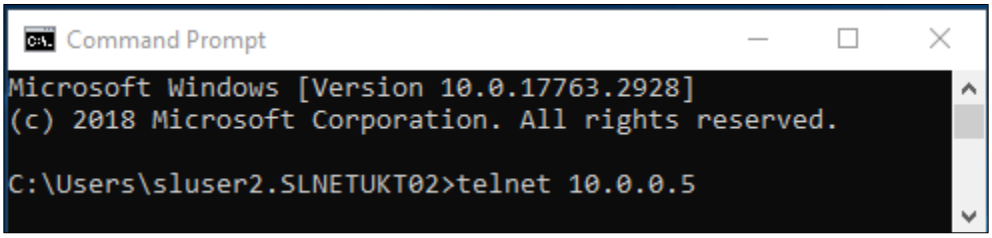
When a user has successfully logged on, they will be taken to the shell specified in the Configuration Screen for your user. (The default is the Windows Common Command Shell.)

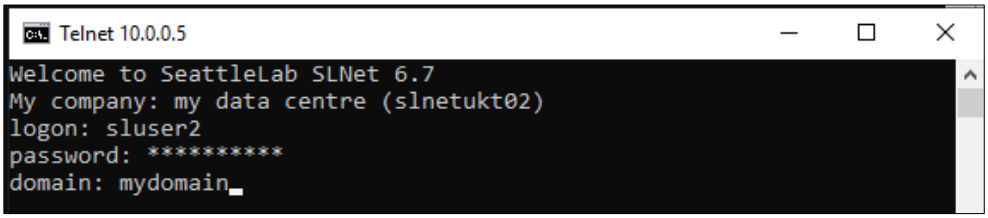
Manual Logon

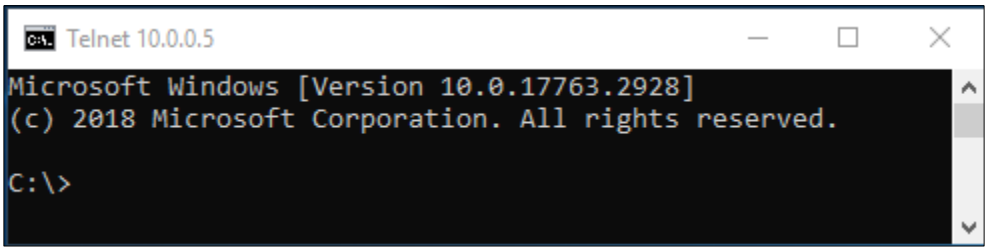
When a user connects to the SLNet server via a telnet session, they will be prompted for:
Logon: (username)
Password: (password)
Domain: (domain or local machine name if not is a domain) (unless a Default Domain has been set)
They might also be prompted for Use Color Codes? or Enable UTF-8 Fonts? if these options are set to “Prompt” in their SLNet profile.

Logon with Telnet Client

At a command prompt type : **`telnet`** ***`IP_of_SLnetServer`***



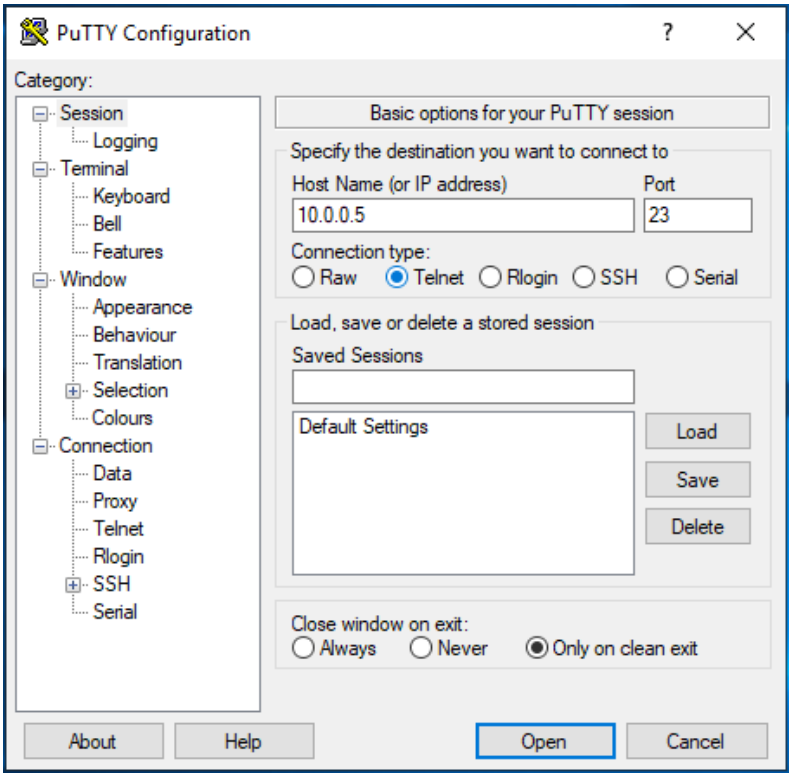




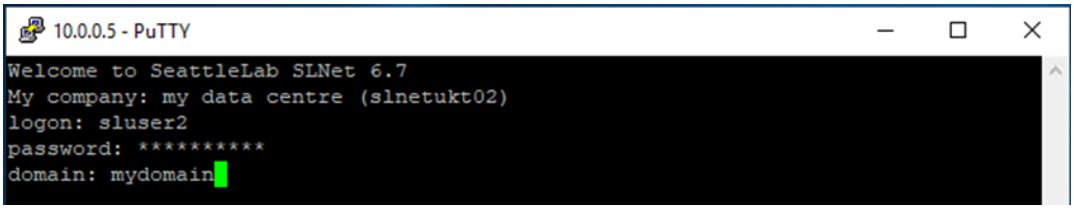
Logon with PuTTy Client

The PuTTY client supports UTF-8 encoding.

Install and launch the PuTTY client
The Session category should be loaded as default
Select Connection type ***Telnet*** - this will enter the default port number ***23***
In the **Host Name (or IP address)** box enter ***SLNetServer_ip_address***
Select **[Open]**



At the Enable UTF-8 Fonts? prompt enter Y or N





Automatic Logon

With SLNet you can set up an auto logon using a default user account for authentication or by IP address. With auto logon any device connecting to SLNet will automatically be logged on without having to enter a user name or password and will be placed in the shell specified in the User Profile.
Instructions for configuring a global Default User Logon can be found on page 14 of this manual.

Automatic Logon by IP address

SLNet can automatically logon a user from a device that connects from a specific IP address or a range of addresses*.* This enables you to have a shared user ID be used from a single IP address or a secured range of IP addresses.
Instructions for configuring Auto Logon by IP Address can be found on pages 15 – 17 of this manual.

Automatic Reconnect

SLNet can automatically connect a user back to their application if there is a temporary telnet disconnection (for example, an RF device in a warehouse has its signal interrupted) and the Windows IP stack makes SLNet aware of the interruption.

Instructions for configuring Automatic Reconnect can be found on page 13 of this manual.

Exiting an SLNet session

If logged into a backend application such as SAPConsole or Oracle, you should first properly exit the application. If at that time the SLNet session does not disconnect you can either turn of your RF terminal or type EXIT at the MSDOS prompt if connecting from a PC.

SLNet Terminal Emulation

SLNet can be customized to meet any terminal emulation / application character needs.

ANSI terminal emulators lack all the keys on a PC keyboard, yet software written to be run on PCs makes extensive use of these keys. To allow terminal emulator users to simulate the pressing of keys they do not have, SLNet provides mapping sequences. These sequences begin with `CTRL-A` (A for *Attention*). You first press `CTRL-A`, then another key, to simulate the pressing of a single PC keyboard key.

The keys that need to be simulated fall into three basic groups: Cursor keys (Home, Pg Up, etc.); ALT-Keys (keystrokes entered while the ALT key is held down); and PC function keys.

This chart summarizes the `CTRL-A` sequences used to simulate these keys. Where ellipses (...) are used, they indicate that the pattern set by the preceding simulation is followed. For example, to simulate Function Key 2, you would use the same sequence as for Function Key 1, only replacing the 1 with a 2. The sequence must be completed in less than one second, or no key will be simulated, and the sequence that you typed will be sent to the application verbatim.

The Special Keys allow you to control processes you have generated from SLNet. It is possible from SLNet to invoke a process of which you lose control. For example, suppose that from a terminal emulator via SLNet you type `Notepad` at the command line. SLNet will invoke the Notepad applet, which is a Windows program. From your terminal emulator, you have no way of getting input to a Windows program. The Special Keys allow you to terminate this process.

The special keys affect only processes you have created. You cannot use them to list or kill another user's processes

Sequence	Simulates
Cursor Keys	
`CTRL-A i`	Insert
`CTRL-A d`	Delete
`CTRL-A h`	Home
`CTRL-A e`	End
`CTRL-A u`	PgUp
`CTRL-A n`	PgDn
`CTRL-A t`	Tab
ALT Keys	
`CTRL-A A`	ALT-A
`...`	...
`CTRL-A Z`	ALT-Z
Function Keys	
`CTRL-A 1`	F1
`...`	...
`CTRL-A 9`	F9
`CTRL-A 0`	F10
`CTRL-A - (minus)`	F11
`CTRL-A = (equal)`	F12
Sticky Keys (Active until second key pressed)	
`CTRL-A a`	ALT
`CTRL-A c`	CTRL
`CTRL-A s`	SHIFT – Caps Lock On
Special Keys	
`CTRL-A p`	List Processes
`CTRL-A k`	Display processes
`CTRL-R`	Refresh Screen
`CTRL-A SPACE`	Refresh Screen
`CTRL-A l`	Display processes
`CTRL A CTRL K`	Keyboard analysis – key ASCII codes are displayed
`CTRL A CTRL R`	Raw keyboard input. Keys are not translated to Windows keys.

SLNet Entries in Windows Application Log

SLNet keeps a record of its activities by entering messages in the Windows Event Log as events occur.

To view event logs for SLNet, go to the Log menu on the tool bar and select *Application.*

You may select which types of events you wish recorded in the Event Log by going to the SLNet Configuration panel *Service* tab. Check Logging - set Logging On and adjust the Options.

Logging Message examples

The [%1] and [%2] tags in the following examples are values that are filled in by SLNet at the time the event is generated

- If the socket SLNet is trying to use is already in use by another program the user will get an Error message. The number *1000* will appear in the Event column and the Error message will be: “The [%1] service could not be started.”
- If a user who does not have permission to use SLNet attempts to log on, the number *2000* will appear in the Event column, and the Warning message will be: “User [%1] attempted to log on with no profile. There was no default.”
- When a user logs onto SLNet, the number *2002* will appear in the Event column and the Information message for that event is: “Accepting connection from IP: [%1]. Connection Number [%2].”
- When the Telnet service is successfully started, the number *3000* will appear in the Event column and the Success message for that event is: “[%1] has been successfully started.”

If the user is connecting via a proxy server, the IP address will always be the address of the proxy server rather than the client machine on the other side of the proxy server.

SLNet Error Codes

The [%1] and [%2] tags in the Error Message column are values that are filled in by SLNet at the time the event is generated. Generally speaking, [%1] tags represent applications and [%2] tags represent machine IP addresses.

Please contact our technical support team for assistance. Most error codes are from authentication security.

Event	Type	Error Message
1000	ERROR	The %1 service could not be started.
1001	ERROR	Unable to find socket library.
1002	ERROR	Unable to allocate memory for Socket Info Structure.
1003	ERROR	An error occurred while attempting to allocate memory.
1004	ERROR	%1 has not been installed correctly. Please re-install.
1005	ERROR	%1 %2 (generic)
1006	ERROR	%1
1007	ERROR	The user's process returned %1.
1008	ERROR	Assert in file %1.
1009	ERROR	Invalid registration data. The service is shutting down.
1010	ERROR	The demo version of %1 has expired.
10000	ERROR	The Socket error is undefined.
10004	ERROR	The Socket has been interrupted.
10024	ERROR	No more file descriptors are available.
10036	ERROR	A blocking Windows Sockets operation is in progress.
10041	ERROR	The specified protocol is the wrong type for this socket.
10043	ERROR	The specified protocol is not supported.
10044	ERROR	The specified socket type is not supported in this address family.
10047	ERROR	The specified address family is not supported.
10050	ERROR	The Windows Sockets implementation has detected that the network subsystem has failed.
10052	ERROR	The network has been reset.
10053	ERROR	The connection was aborted.
10054	ERROR	The connection was reset.
10055	ERROR	No buffer space is available. The socket cannot be created.
10060	ERROR	A timeout has occurred.
10093	ERROR	A successful WSAStartup must occur before using this API.
2000	WARNING	User %1 attempted to logon with no logon profile. There was no default.
2000	INFORMATION	The terminal type is: %1.
2001	ERROR	A connection from address %1 was rejected because it is not in the accepted address range.
2002	INFORMATION	Accepting connection from IP: %1. Connection Number %2.
2003	INFORMATION	IP %1 Disconnected.
2004	INFORMATION	The demonstration period will expire on %1.
2005	INFORMATION	%1 user version initialized.
3000	SUCCESS	%1 has been successfully started.
3001	SUCCESS	%1 has been successfully stopped.

Appendix A. Troubleshooting

SLNet is designed to be a powerful solution to Telnet needs.

Problems usually occur when you have made a change in your environment. This may be an operating service pack, a security change, new device, or changing users.

Before You Contact SLNet Technical Support

In order to do our best for you, we require accurate and timely information in the formats we request. Our records show that over 50% of all incidents reported are resolved as permission problems or not following the directions in the documentation.

The Windows Event log has many clues to resolving SLNet issues.

Here are steps to follow before you call our technical support team:

1. Review the Windows Event log. It is a good source for error messages on security and connectivity.
2. Validate that a user has a good connection to your network and a valid Windows account.
3. Verify that the user has permission to log on locally to the server that SLNet is installed on.
4. Turn on SLNet logging for a user and review the log details. Often you will find the user has entered invalid characters or data.

When you contact SLNet technical support, please let us know your Windows operating system release and hardware environment, SLNet version, and a complete description of your question or problem (let us know if you can recreate it), and have access to the SLNet server.

SeattleLab Professional Services

If you need special assistance with customization of SLNet, please contact our technical support group to arrange for billable professional services time.

Contacting SLNet Specialists

Customers with active maintenance support contracts have access to technical specialists.

Email Technical Support: slticket@SeattleLab.co.uk

Telephone numbers and hours of business for our offices are available on our web site –

www.seattlelab.com

We'll "do what it takes" to assist you with your questions or problems.

Appendix B. Customization

Named Pipes

SLNet works with unmodified off-the-shelf character applications by reading the console output buffer at periodic intervals, noting any changes made in the buffer, translating those changes into a character stream, and directing that character stream to the Telnet socket. The character stream contains ANSI escape sequences for cursor control to correctly format the screen. This design allows SLNet to support the widest possible range of third-party products, without requiring customers to make any modification to their software.

For customers who have the ability to modify their software, however, or who are writing new software, SLNet provides a more direct output connection. This connection takes the form of a Named Pipe. There are two advantages to using direct output to the named pipe:

1. It is more efficient, allowing given hardware to support more processes in addition to allowing for each process to complete more quickly.
2. It allows support for non-ANSI terminal emulations. The SLNet Named Pipe can be handled in the following way:

   When SLNet creates a process, it places the name of the pipe it creates for that process in a variable in the process's environment. The Pipe Name field defines the name of this environment variable. To read this environment variable from C code, a program could call the `getenv` function. The name received from the environment can then be used in a call to the W in32 API `CreateFile` (which is used to open files and other devices as well as to create them).

   If your application shells out to run other applications that do not know about the Named Pipe, you should close the pipe, and SLNet will return to screen-scraping mode. When your application regains control, it can then re-open the pipe. The Win32 API `CloseHandle` is used to close the pipe:

   Characters may be directed to the named pipe by calling the Win32 API function `WriteFile`.

The following C code sample demonstrates the basic steps for using named pipe to output all user interface information.

```
/*
This code is an example to show the basic steps for using named pipe
to output all user interface information.

printf() is only used for fatal errors, all other output is via pipe.

Note, during runtime the SLNET end of the pipe will be open/closed outside
of the control of this code.

The code shows the required method of checking for a pipe write and
reopening the pipe on failure. Failure is expected behaviour so note
expected mitigation.

THIS CODE IS NOT PRODUCTION STANDARD, HAS MINIMAL ERROR AND BUFFER
CHECKING. FOR GENERAL GUIIDANCE ONLY.
*/
#define _CRT_SECURE_NO_WARNINGS 1

#include <Windows.h>
#include <fcntl.h>
#include <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
```

```
void main(void)
{
      char PipeName[256] = { 0 };
      char* psEnvPipeName = { 0 };
      char TempIn[256] = { 0 };
      char TempOut[256] = { 0 };
      int PipeHandle = 0;
      int BytesRead = 0;
      const char* CLEAR_SCREEN = "\x1B[2J";

      /*Turn off console echo so that the only output is via the pipe.
      Only fatal erors user printf below*/
      HANDLE hStdin = GetStdHandle(STD_INPUT_HANDLE);
      DWORD mode = 0;
      GetConsoleMode(hStdin, &mode);
      SetConsoleMode(hStdin, mode & (~ENABLE_ECHO_INPUT));

      //Get pipe name from ENV Vars
      psEnvPipeName = getenv("PIPENAME");
      /* the pipename on Windows Server is \\.\pipe\<pipename> where
      <pipename> is the value stored **in the PIPENAME environment variable
set by SLNet */

      if (psEnvPipeName)
      {
            sprintf(PipeName, "\\\\.\\PIPE\\%s", psEnvPipeName);

            PipeHandle = _open(PipeName, _O_WRONLY);

            if (PipeHandle == -1)
            {
                  printf("Error: Unable to open the named pipe %s.\n",
PipeName); exit(-1);
            }

            sprintf(TempOut, "%sEcho Test.\nThis program now controls the
output to the terminal. Type some characters and press Return.\nType 'exit'
to end.\n", CLEAR_SCREEN);

            //Send this message to the terminal via pipe
            int bytesWritten = _write(PipeHandle, TempOut, strlen(TempOut));
            /*NOTE a critical point here.
            Having written to the pipe the pipe will now close.
            It may take a few tenths of a second but the
            pipe may be closed on next usage. See below*/

            //clear buffer
            memset(TempOut, 0, sizeof(TempOut));

            // read keyboard input
            while ((BytesRead = _read(0, TempIn, 256)))
            {/* File handle 0 is standard in or the keyboard */
```

```
                if (BytesRead == -1) /* an error occurred */
                     break;

                if (_strnicmp(TempIn, "exit", 4) == 0) /* The user chose to
exit */
                     break;

                TempIn[BytesRead] = '\0';//chop off extra new line (it has
two!)
                //format output
                sprintf(TempOut, "You typed: %s", TempIn);

                //attempt to do write to pipe. It may not succeed
                bytesWritten = _write(PipeHandle, TempOut,
strlen(TempOut));

                //did it write?
                if (bytesWritten == -1)
                {
                     //No. Reopen the pipe
                     PipeHandle = _open(PipeName, _O_WRONLY);
                     if (PipeHandle == -1)
                     {
                          printf("Reopen Pipe Failed %ld\n",
GetLastError());
                          break;
                     }
                     //now do the write again
                     bytesWritten = _write(PipeHandle, TempOut,
strlen(TempOut));
                     //should succeed but do your error checking here
                }

                //clear down buffers
                memset(TempOut, 0, sizeof(TempOut));
                memset(TempIn, 0, sizeof(TempIn));

           }

           _close(PipeHandle);

     }
     else
     {
          printf("SLNet is not running\n");
     }

     exit(0);

}
```

Raw Keyboard Mode

By default, key press output in SLNet 6.7 is translated from Ascii Video Terminal Escape Codes (VT codes) to Windows scan codes. SLNet 6.7 provides functionality to change the key press output to VT code.

To change the key press output to VT code you will need to create a registry entry for each SLNet user profile that requires this functionality. If all users use the Default SLNet profile create the key here:

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Seattle Lab\SLNet\Default

If you use specific named profiles e.g. if there is a user called RFUser, create the key here:

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Seattle Lab\SLNet\RFUser

Registry entry type = DWORD
DWORD name = RawKeyBoardMode
DWORD value = 1

SLNet and Windows Security

SLNet abides by all Windows security features. It does not alter nor change any security settings or privilege levels set by normal Windows mechanisms.

When a Telnet user logs into your system via SLNet, he or she must be a valid user that you or another administrator created using the Windows User Manager or Active Directory Manager. SLNet sets that user's process to run in the security context appropriate to that user. The user then has exactly the same privileges he or she would have on your machine if they sat down in front of it and logged in on the console by pressing the `CTRL ALT DELETE` keys.

Windows allows you to grant or withhold privileges on a user-by-user basis to every object on the system. Using Windows Explorer, for example, you may choose which individual users can have access to any given file you create. SLNet abides by these exact same access restrictions. It does not and cannot alter an y privileges set by Windows.

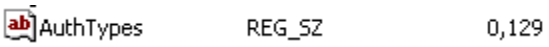
Authentication types

SLNet allows you to encrypt the user name and password in the data stream with the **AuthTypes** registry value.

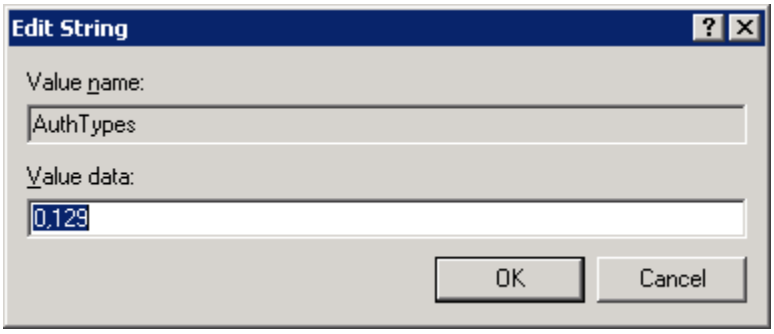
You will need to modify this registry value in the root of the SLNet registry (`HKEY_LOCAL_MACHINE > SOFTWARE > SeattleLab> SLNet`). Open the Registry Editor through the Start Menu (Start > Run and then entering `REGEDIT`).



In the right frame of the registry editor, right-click on the **AuthTypes** string value and select Modify from the popup menu.



By default the key has 0,129 (IP Address, decimal equivalent for the **AuthType** required), which allows all users to use the SeattleLab encryption or fall back to clear text.



Enter `<IP address, decimal equivalent for the authorization type; IP address,...>` and then click OK to update the string. The string pairs must be separated by a semi colon. For example: `0,129;127.0.0.1,128;129.168.0.1,65`

The authentication types are:

- Null (no encryption)—bit 1, decimal 1
- Microsoft NTLM—bit 7, decimal 64
- SeattleLab proprietary—bit 8, decimal 128

Note: Multiple encryption types can be specified by combining the bits. For example for Null (clear text) and SeattleLab encryption the decimal equivalent would be 129. All single encryption types and pairs are possible.

This page intentionally left blank.