Albatros Console

User manual · Manuel d'utilisation · Manuale d'uso · Manual de utilizare · ユーザー マニュアル

Language · Langue · Lingua · Limba · 言語.....**Page · Page · Pagina · Pagina · ページ**

EN English.....2

FR Français.....17

IT Italiano.....32

RO Română.....46

JA 日本語.....61

DevService.net — Raffaele Romano

Albatros Console

SSH and SFTP terminal for Windows

User Manual

Version with credentials screen, host key verification and reconnection

October 2026

Contents

1. Introduction....4
1.1 What's new in this version....4
2. Requirements and building....5
2.1 Requirements....5
2.2 Opening and building the project....5
3. The credentials screen....6
3.1 The fields....6
3.2 The preset connection....6
3.3 Storing the passphrase or password....7
3.4 Checks before connecting....7
4. Host key verification....8
4.1 First connection....8
4.2 Changed key....8
5. The terminal window....9
5.1 Connessione menu....9
5.2 File menu....9
5.3 Status bar....10
5.4 Keyboard and mouse....10
6. Reboot monitor and disconnections....11
6.1 How the monitor works....11
6.2 When the connection is interrupted....11
7. Configuration files....13
8. Security....14
8.1 Default passphrase in the code....14
8.2 What DPAPI protects....14
8.3 Host keys....14
9. Troubleshooting....15
10. Notes for the developer....16
10.1 Structure....16
10.2 Connection generations....16
10.3 Editable parameters....16
10.4 Tests performed....16

1. Introduction

Albatros Console is an SSH terminal for Windows written in VB.NET (.NET Framework 4.8, SSH.NET library). In addition to the interactive xterm-style shell, it offers remote file browsing and SFTP transfer, and a monitor that detects when the server is rebooted.

This manual describes the version with the **initial credentials screen**. Compared with the previous version, in which server, user, key and passphrase were written in the code and the connection started by itself, a window now appears at startup that proposes the preset connection and lets you change it.

1.1 What's new in this version

- **Credentials screen** at startup, with the values of the preset connection already filled in and the **Predefiniti** (Defaults) button to restore them.
- Authentication with a **private key** (PuTTY PPK or OpenSSH format, with or without passphrase) or with a **password**.
- **Optional storage** of the passphrase or password, encrypted with Windows DPAPI for the current user.
- **Host key verification** of the server, as the ssh command does: on the first connection you are asked for confirmation, then a different key is reported.
- Background connection: the window does not freeze while the server responds, and **errors are explained** in Italian instead of closing the program.
- **Connessione** (Connection) menu with Nuova connessione, Riconnetti and Disconnetti (New connection, Reconnect, Disconnect); after an interruption the program offers to **reconnect** instead of closing.
- **Status bar** with the connected server and the state of the reboot monitor.
- More robust reboot monitor: a single network error is no longer enough to declare the connection lost.
- The private key is **decrypted only once** and reused by the terminal, monitor, file browser and transfers (previously it was read again at every SFTP operation and every check, which was slow with PPK v3 keys).
- Terminal: function keys F1–F12, PgUp/PgDn, Ins, Shift+Tab, application cursor mode (vim, less, htop), **Shift+PgUp/PgDn** to scroll the history; fixed the first line that remained hidden under the menu.

2. Requirements and building

2.1 Requirements

- Windows 10/11 or Windows Server 2016 and later, with .NET Framework 4.8.
- To build: Visual Studio 2019 or 2022 with the .NET desktop development workload.
- Remote server with SSH enabled. The reboot monitor requires a Linux server (it reads `/proc/sys/kernel/random/boot_id`).

2.2 Opening and building the project

1. Extract the zip file and open `AlbatrosConsole.sln` in Visual Studio.
2. The `packages` folder already contains the NuGet packages (SSH.NET 2026.0.0 and dependencies). If Visual Studio reports them as missing, use Restore NuGet Packages.
3. Build with Build → Rebuild Solution and start with F5.

Compared with the previous version, the project contains four new files and one additional reference; they are already included in the `AlbatrosConsole.vbproj` file:

Item	Content
`FormLogin.vb`	Credentials screen.
`SshConnectionProfile.vb`	Connection parameters, decrypted key cache, creation of SSH/SFTP clients.
`ConnectionStore.vb`	Values of the preset connection (`DefaultConnection` module) and saving of the last connection.
`HostKeyStore.vb`	Store of trusted host keys (equivalent of known_hosts).
Reference `System.Security`	Required for DPAPI encryption (`ProtectedData`).

3. The credentials screen

At startup the program opens the **Albatros Console – Connessione** window. The terminal waits behind the screen until the connection succeeds.

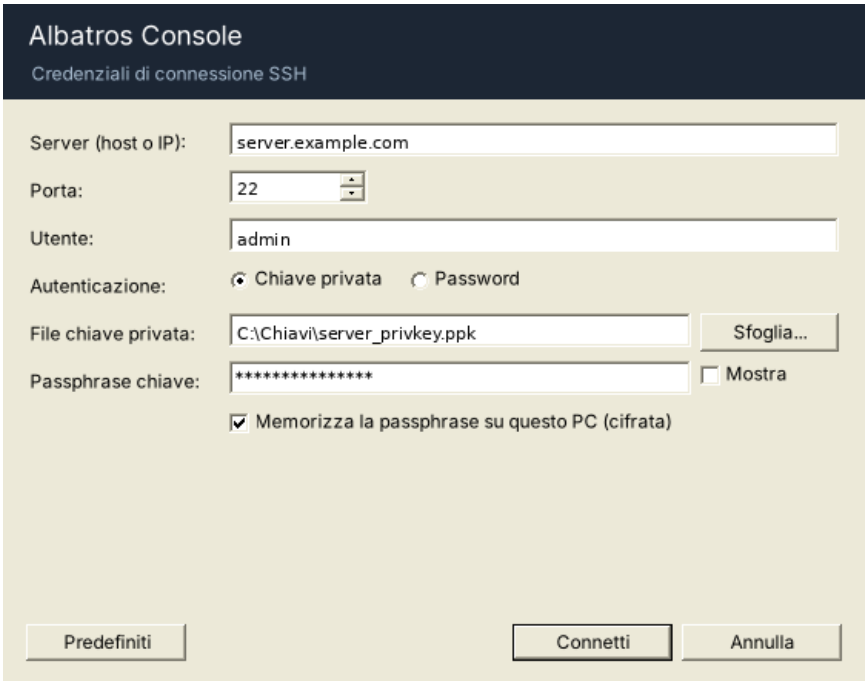

Figure 1 – Credentials screen with the preset connection (indicative appearance)

3.1 The fields

Field	Meaning
Server (host o IP)	DNS name or IP address of the SSH server.
Porta	TCP port of the SSH service (1–65535). The standard is 22.
Utente	Name of the remote user.
Autenticazione	**Chiave privata** (Private key) or **Password**. In Password mode the key file field is disabled.
File chiave privata	Path of the key file: PuTTY PPK (v2 or v3) or OpenSSH key. The **Sfoglia...** (Browse...) button opens the file selection.
Passphrase chiave / Password	Passphrase protecting the key, or the user's password. Leave empty if the key is not protected. **Mostra** (Show) makes the typed text visible.
Memorizza ... su questo PC (cifrata)	If selected, the passphrase or password is saved encrypted for the current Windows user (see 3.3).

Buttons: **Connetti** (Connect; also with Enter) starts the connection; **Annulla** (Cancel; also with Esc) at startup closes the program, while from the menu it closes only the screen, leaving the current session unchanged; **Predefiniti** (Defaults) restores the preset connection.

3.2 The preset connection

At first startup, and whenever no last connection has been saved, the screen proposes the connection preset at build time: server, port, user, private key file and any passphrase already filled in. In practice you just press Connetti.

After each successful connection, the values used are saved and proposed again at the next startup. The **Predefiniti** button returns to the preset values at any time, including the passphrase.

The preset values are in the `DefaultConnection` module at the beginning of `ConnectionStore.vb`: to change them, just modify the constants and rebuild.

3.3 Storing the passphrase or password

With the **Memorizza** (Store) check box selected, the passphrase or password is written to the `connessione.xml` file, encrypted with Windows DPAPI (current user scope). Only the same Windows user on the same PC can decrypt it: if the file is copied to another computer, or read with another account, the field is empty and must be entered again.

If the check box is cleared, the value is not saved: at the next startup the field will be empty. During the same session the program still remembers it for reconnections.

3.4 Checks before connecting

When you press **Connetti**, the screen checks the required fields and, for key authentication, immediately tries to read and decrypt the key. In this way a missing file or a wrong passphrase is reported in red below the fields without waiting for the server; the cursor is placed on the field to correct.

If the connection to the server fails (server off, wrong port, authentication rejected, host key not accepted), the screen is reopened with the reason for the error and with the data already entered, ready for a new attempt.

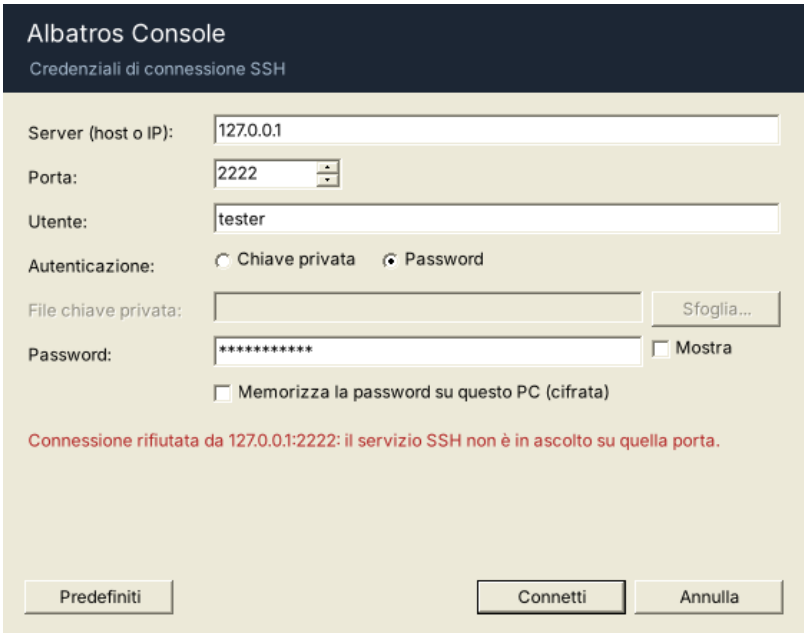

Figure 2 – Screen shown again after a connection error

4. Host key verification

Every SSH server has its own host key. Checking it guarantees that you are really talking to your own server and not to an intermediary (man-in-the-middle attack). The previous version accepted any key; now Albatros Console behaves like the ssh command.

4.1 First connection

The first time you connect to a server (or to a new port of the same server), a confirmation request appears with the key type and its SHA256 fingerprint.

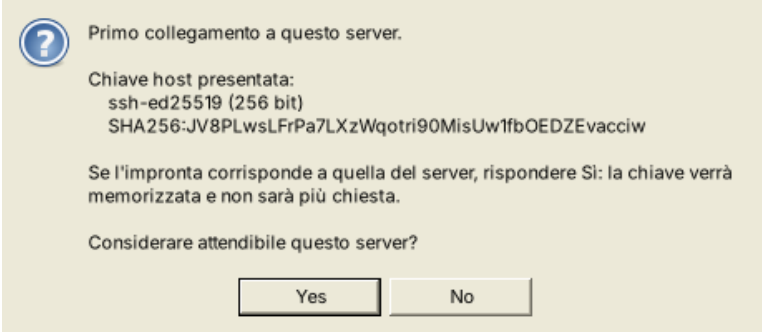

Figure 3 – Host key confirmation on the first connection (indicative appearance)

To verify the fingerprint, on the server, from an already trusted session (server console or an already configured PuTTY):

```
for f in /etc/ssh/ssh_host_*_key.pub; do ssh-keygen -lf "$f"; done
```

One of the lines must show the same `SHA256:...` fingerprint shown by the program. If you answer **Yes**, the key is stored and will not be asked for again; if you answer **No**, the connection is canceled.

> **Note.** After upgrading to this version, confirmation is requested once also for the preset server, because the list of trusted keys starts empty.

4.2 Changed key

If the server presents a key different from the stored one, a warning with the new fingerprint appears; the default button is **No**. The change is normal after reinstalling the operating system or regenerating the SSH keys; in any other case it is better not to continue and first find out the reason. If you answer **Yes**, the new key replaces the stored one.

The file browser, the SFTP transfer and the reboot monitor accept only keys that are already stored and ask no questions: confirmation always takes place in the main connection.

> **Caution.** To forget a server's key, just delete the corresponding line in the `known_hosts.txt` file (see chapter 7).

5. The terminal window

When the connection succeeds, the window title shows `utente@host:porta` and the terminal receives the server prompt.



Figure 4 – Connected terminal, with status bar (indicative appearance)

5.1 Connessione menu

Item	Function
Nuova connessione...	Reopens the credentials screen. If you confirm, the current session is closed and the new one opens; if you cancel, the current session stays as it is.
Riconnetti	Reconnects to the last server with the same credentials, without reopening the screen. If it fails, it opens the screen with the reason.
Disconnetti	Closes the session. The window stays open with the terminal content; you can reconnect from the menu.
Esci	Closes the program.

5.2 File menu

Item	Function
Trasferimento file SFTP...	Download, upload and server → server copy, with multiple transfers in parallel.
Esplora file remoto...	Browsing remote folders, creating and deleting files and folders, Unix permissions, opening files on the fly with Notepad++ or Notepad, download.

Both windows use the credentials of the current connection and are available only while connected. Their operation is unchanged from the previous version.

5.3 Status bar

On the left it shows the connection state (Connessione in corso, Connesso a ..., Disconnesso da ..., Non connesso: Connecting, Connected to ..., Disconnected from ..., Not connected); on the right the state of the reboot monitor (active, not available) or the reason for the last disconnection.

5.4 Keyboard and mouse

Key or action	Effect
Ctrl+C	Copies the selection; if there is no selection, sends the interrupt (SIGINT) to the remote program.
Ctrl+V, right-click without selection	Pastes the text from the clipboard.
Right-click with selection	Copies the selection.
Dragging with the left button	Selects text, also in the history.
Mouse wheel	Scrolls the history (5000 lines). A yellow bar on the right indicates that you are not at the bottom.
Shift+PgUp / Shift+PgDn	Scrolls the history by one page (new).
Arrows, Home, End	Sent to the server; in full-screen programs (vim, less, htop) they automatically use application cursor mode.
PgUp, PgDn, Ins, Del	Sent to the server (new for PgUp, PgDn, Ins).
F1–F12	Sent to the server as in xterm (new). F10 no longer opens the window menu: it is used, for example, to exit mc and htop.
Shift+Tab	Sent to the server (back tab).
Ctrl+D	End of input; in the shell it is equivalent to exit.
Any other key	Brings the view back to the bottom, at the prompt.

6. Reboot monitor and disconnections

6.1 How the monitor works

As soon as it is connected, the program reads the server's `boot_id`, an identifier that Linux changes at every boot. From then on, every 3 seconds, it opens a short, separate check connection and reads it again:

- if the `boot_id` has changed, the server has been rebooted and the disconnection is reported immediately;
- if the check fails because of a network problem, the disconnection is reported only after 2 consecutive errors (previously one was enough);
- if the server is not Linux and the `boot_id` cannot be read, the monitor stays disabled and the status bar shows «Monitor riavvio non disponibile» (Reboot monitor not available).

The check connection is separate from the main session because, after a quick reboot, SSH.NET may still consider a session that is in fact dead to be «connected».

6.2 When the connection is interrupted

A yellow line with the reason appears in the terminal and a choice window opens:

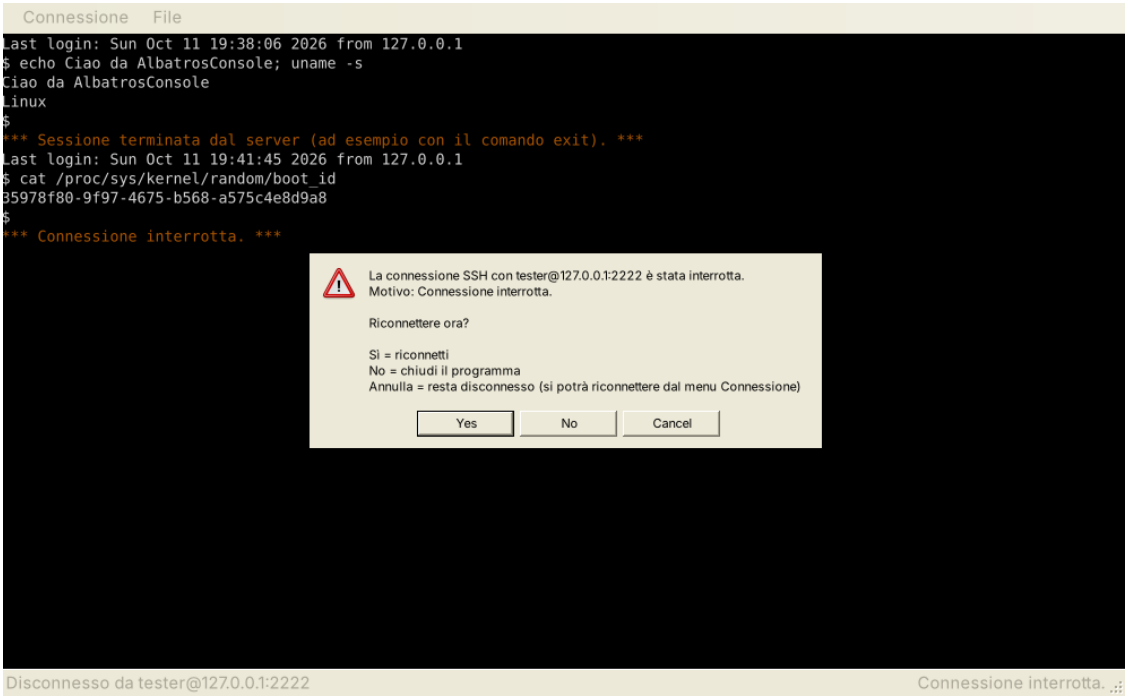

Figure 5 – Connection interrupted: choice between reconnecting, exiting or staying disconnected (indicative appearance)

Answer	Effect
Yes	Immediate reconnection with the same credentials. If the server is still not responding (for example, it is rebooting), the screen opens with the reason: wait a few seconds and press Connetti.
No	Closes the program (behavior of the previous version).
Cancel	Stays disconnected with the terminal content visible; reconnect from the Connessione menu.

The possible reasons are: server rebooted; session terminated by the server (for example with the `exit` command); connection interrupted (network, server off, SSH service stopped).

On Italian-language Windows the message box buttons appear as Sì, No and Annulla; the figures in this manual were captured in a test environment and show the English captions (Yes, No, Cancel).

7. Configuration files

The program saves its data in the Windows user's `%APPDATA%\AlbatrosConsole` folder (normally `C:\Users\<utente>\AppData\Roaming\AlbatrosConsole`).

File	Content	If it is deleted
`connessione.xml`	Last successful connection: server, port, user, authentication type, key file, storage choice and, if requested, passphrase or password encrypted with DPAPI.	At the next startup the preset values are proposed.
`known_hosts.txt`	One line per server: `[host]:porta tipo-chiave SHA256:impronta`.	At the next connection the host key is asked for again.

Example of a `known_hosts.txt` line:

```
[vps.example.org]:22 ssh-ed25519 SHA256:JV8PLwsLFrPa7LXzWqotri90MisUw1fbOEDZEvacciw
```

8. Security

8.1 Default passphrase in the code

To meet the requirement of having the preset connection ready to use, the passphrase of the default key remains written in the source code (`KeyPassphrase` constant in `ConnectionStore.vb`), exactly as in the previous version it was written in `Form1.vb`. As a consequence it can also be read in the compiled executable, with any .NET decompiler.

If the executable is to leave your own PC, it is recommended to:

1. empty the `KeyPassphrase` constant (leaving `""`) and rebuild;
2. connect once, entering the passphrase by hand with the **Memorizza** check box selected: from then on it stays saved, encrypted with DPAPI, and no longer appears in the program.

The private key file, on the other hand, remains on disk: anyone who has a copy of it and knows the passphrase can access the server. Keep it in a folder with permissions restricted to the user.

8.2 What DPAPI protects

DPAPI ties the data to the Windows account: another user of the same PC or another computer cannot decrypt it. It does not, however, protect against a program run with the same account as the user, which could decrypt it just as Albatros Console does.

8.3 Host keys

Host key verification protects the connection only if the fingerprint is actually compared with the server's on the first connection (section 4.1). A changed-key warning must never be accepted without knowing the reason.

9. Troubleshooting

Message	Probable cause	What to do
Il file della chiave privata non esiste	(The private key file does not exist.) Wrong path or file moved.	Correct the path or use Sfoglia...
La chiave privata è protetta: inserire la passphrase	(The private key is protected: enter the passphrase.) Encrypted key and empty passphrase field.	Enter the passphrase.
Impossibile leggere la chiave privata: passphrase errata oppure file non valido	(Unable to read the private key: wrong passphrase or invalid file.) Wrong passphrase or a file that is not a private key.	Retype the passphrase (use Mostra to check it) or choose the correct file.
Server non trovato	(Server not found.) Wrong host name or DNS not reachable.	Check the name or use the IP address.
Connessione rifiutata ... il servizio SSH non è in ascolto	(Connection refused ... the SSH service is not listening.) Wrong port or sshd stopped.	Check the port; on the server: `sudo systemctl status ssh`.
Il server ... non è raggiungibile / Tempo scaduto	(The server ... is not reachable / Timed out.) Server off or rebooting, firewall, no network.	Wait and try again; check the firewall and the connection.
Autenticazione rifiutata dal server	(Authentication rejected by the server.) Public key not authorized, wrong user or password.	Check `~/.ssh/authorized_keys` on the server, the user and the password.
Connessione annullata: la chiave host ... non è stata accettata	(Connection canceled: the host key ... was not accepted.) You answered No to the host key verification.	Verify the fingerprint (4.1) and try again.
«Host key could not be verified» in the file browser or in the transfer	The host key changed after the session was opened.	Reconnect from the Connessione menu and confirm the new key.
Monitor riavvio non disponibile	(Reboot monitor not available.) The server does not expose `/proc/sys/kernel/random/boot_id` (it is not Linux).	No action: the terminal works normally, without reboot detection.

10. Notes for the developer

10.1 Structure

Class / module	Role
`Form1`	Main window: menu, status bar, asynchronous connection, shell reading, monitor, disconnection handling.
`FormLogin`	Credentials screen; returns an already verified `SshConnectionProfile` and the `ConnectionSettings` to be saved.
`SshConnectionProfile`	Immutable connection parameters. `CreateSshClient` and `CreateSftpClient` return clients that are not yet connected, with the host key check already attached. The decrypted key is cached per thread (`ThreadLocal`), because SSH.NET key instances are not guaranteed to be thread-safe and parallel transfers run on different threads.
`ConnectionSettings`, `ConnectionStore`, `DefaultConnection`	Screen values, saving to `connessione.xml` with DPAPI, preset values.
`HostKeyStore`	Reading and writing of `known_hosts.txt`; `Attach` attaches the verification to the `HostKeyReceived` event.
`FormTransfer`, `RemoteBrowserDialog`	Logic unchanged; the constructor now receives an `SshConnectionProfile` instead of the five parameters host, port, user, key and passphrase.
`TerminalControl`, `TerminalBuffer`	Added function keys, PgUp/PgDn, DECCKM mode, scrolling with Shift+PgUp/PgDn, cached bold font and lighter background drawing.

10.2 Connection generations

Each connection receives a «generation» number. The shell reading thread and the monitor timer remember the generation they started with; when the user disconnects or opens a new connection, the generation changes and any late notifications from them are ignored. In this way, closing a session voluntarily does not cause false disconnection warnings.

10.3 Editable parameters

Constant (in Form1.vb)	Value	Meaning
`ConnectTimeoutSeconds`	15	Maximum time for the main connection.
`MonitorIntervalSeconds`	3	Interval between two monitor checks.
`MonitorTimeoutSeconds`	4	Maximum time for each check.
`MonitorMaxFailures`	2	Consecutive network errors before declaring the connection lost.

10.4 Tests performed

Build with no errors or warnings against .NET Framework 4.8. On a test OpenSSH server the following were verified: key with correct, wrong and missing passphrase; access with key and with password; new, stored and changed host key; SFTP also with six parallel connections; encrypted credentials; the cycle screen → host key → terminal → disconnection → reconnection; detection of the stopped server by the monitor.

Albatros Console

Terminal SSH et SFTP pour Windows

Manuel d'utilisation

Version avec fenêtre d'identifiants, vérification de la clé d'hôte et reconnexion

Octobre 2026

Sommaire

1. Introduction....19
1.1 Nouveautés de cette version....19
2. Configuration requise et compilation....20
2.1 Configuration requise....20
2.2 Ouvrir et compiler le projet....20
3. La fenêtre d'identifiants....21
3.1 Les champs....21
3.2 La connexion prédéfinie....21
3.3 Mémorisation de la phrase secrète ou du mot de passe....22
3.4 Contrôles avant la connexion....22
4. Vérification de la clé d'hôte....23
4.1 Première connexion....23
4.2 Clé modifiée....23
5. La fenêtre du terminal....24
5.1 Menu Connessione....24
5.2 Menu File....24
5.3 Barre d'état....25
5.4 Clavier et souris....25
6. Moniteur de redémarrage et déconnexions....26
6.1 Fonctionnement du moniteur....26
6.2 Lorsque la connexion s'interrompt....26
7. Fichiers de configuration....28
8. Sécurité....29
8.1 Phrase secrète prédéfinie dans le code....29
8.2 Ce que protège DPAPI....29
8.3 Clés d'hôte....29
9. Dépannage....30
10. Notes pour le développeur....31
10.1 Structure....31
10.2 Générations de connexion....31
10.3 Paramètres modifiables....31
10.4 Vérifications effectuées....31

1. Introduction

Albatros Console est un terminal SSH pour Windows écrit en VB.NET (.NET Framework 4.8, bibliothèque SSH.NET). Outre le shell interactif de style xterm, il offre l'exploration des fichiers distants et le transfert SFTP, ainsi qu'un moniteur qui détecte le redémarrage du serveur.

Ce manuel décrit la version avec **fenêtre initiale des identifiants**. Par rapport à la version précédente, dans laquelle le serveur, l'utilisateur, la clé et la phrase secrète étaient écrits dans le code et la connexion démarrait d'elle-même, une fenêtre apparaît désormais au démarrage ; elle propose la connexion prédéfinie et permet de la modifier.

1.1 Nouveautés de cette version

- **Fenêtre d'identifiants** au démarrage, avec les valeurs de la connexion prédéfinie déjà renseignées et le bouton **Predefiniti** pour les rétablir.
- Authentification par **clé privée** (PPK de PuTTY ou format OpenSSH, avec ou sans phrase secrète) ou par **mot de passe**.
- **Mémorisation facultative** de la phrase secrète ou du mot de passe, chiffrés avec DPAPI de Windows pour l'utilisateur actuel.
- **Vérification de la clé d'hôte** du serveur, comme le fait la commande ssh : à la première connexion, une confirmation est demandée, puis une clé différente est signalée.
- Connexion en arrière-plan : la fenêtre ne se bloque pas pendant que le serveur répond et les **erreurs sont expliquées** en italien au lieu de fermer le programme.
- Menu **Connessione** avec Nuova connessione, Riconnetti et Disconnetti ; après une interruption, le programme propose de **se reconnecter** au lieu de se fermer.
- **Barre d'état** avec le serveur connecté et l'état du moniteur de redémarrage.
- Moniteur de redémarrage plus robuste : une seule erreur réseau ne suffit plus à déclarer la connexion perdue.
- La clé privée est **déchiffrée une seule fois** et réutilisée par le terminal, le moniteur, l'explorateur et les transferts (auparavant, elle était relue à chaque opération SFTP et à chaque contrôle, ce qui était lent avec les clés PPK v3).
- Terminal : touches de fonction F1–F12, Pg préc./Pg suiv., Inser, Maj+Tab, mode curseur application (vim, less, htop), **Maj+Pg préc./Pg suiv.** pour faire défiler l'historique ; correction de la première ligne qui restait cachée sous le menu.

2. Configuration requise et compilation

2.1 Configuration requise

- Windows 10/11 ou Windows Server 2016 et versions ultérieures, avec .NET Framework 4.8.
- Pour compiler : Visual Studio 2019 ou 2022 avec la charge de travail Développement .NET Desktop.
- Serveur distant avec SSH actif. Le moniteur de redémarrage nécessite un serveur Linux (il lit `/proc/sys/kernel/random/boot_id`).

2.2 Ouvrir et compiler le projet

1. Extrayez le zip et ouvrez `AlbatrosConsole.sln` dans Visual Studio.
2. Le dossier `packages` contient déjà les packages NuGet (SSH.NET 2026.0.0 et dépendances). Si Visual Studio les signale comme manquants, utilisez Restaurer les packages NuGet.
3. Compilez avec Générer → Régénérer la solution et lancez avec F5.

Par rapport à la version précédente, le projet contient quatre nouveaux fichiers et une référence supplémentaire ; ils sont déjà inclus dans le fichier `AlbatrosConsole.vbproj` :

Élément	Contenu
`FormLogin.vb`	Fenêtre d'identifiants.
`SshConnectionProfile.vb`	Paramètres de connexion, cache de la clé déchiffrée, création des clients SSH/SFTP.
`ConnectionStore.vb`	Valeurs de la connexion prédéfinie (module `DefaultConnection`) et enregistrement de la dernière connexion.
`HostKeyStore.vb`	Archive des clés d'hôte de confiance (équivalent de known_hosts).
Référence `System.Security`	Nécessaire pour le chiffrement DPAPI (`ProtectedData`).

3. La fenêtre d'identifiants

Au démarrage, le programme ouvre la fenêtre **Albatros Console – Connessione**. Le terminal reste en attente derrière la fenêtre jusqu'à ce que la connexion réussisse.

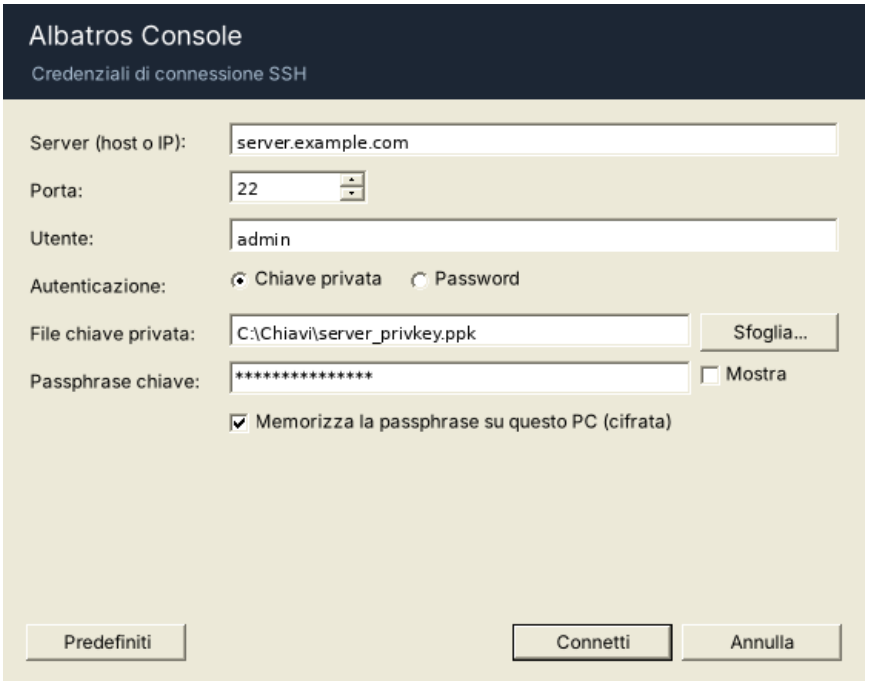

Figure 1 – Fenêtre d'identifiants avec la connexion prédéfinie (aspect indicatif)

3.1 Les champs

Champ	Signification
Server (host o IP)	Nom DNS ou adresse IP du serveur SSH.
Porta	Port TCP du service SSH (1–65535). La valeur standard est 22.
Utente	Nom de l'utilisateur distant.
Autenticazione	**Chiave privata** ou **Password**. En mode Password, le champ du fichier de clé est désactivé.
File chiave privata	Chemin du fichier de la clé : PPK de PuTTY (v2 ou v3) ou clé OpenSSH. Le bouton **Sfoglia...** ouvre la sélection du fichier.
Passphrase chiave / Password	Phrase secrète qui protège la clé, ou bien le mot de passe de l'utilisateur. Laissez vide si la clé n'est pas protégée. **Mostra** rend visible le texte saisi.
Memorizza ... su questo PC (cifrata)	Si elle est cochée, la phrase secrète ou le mot de passe sont enregistrés chiffrés pour l'utilisateur Windows actuel (voir 3.3).

Boutons : **Connetti** (également avec Entrée) lance la connexion ; **Annulla** (également avec Échap) ferme le programme au démarrage, tandis que depuis le menu il ferme seulement la fenêtre en laissant la session en cours inchangée ; **Predefiniti** rétablit la connexion prédéfinie.

3.2 La connexion prédéfinie

Au premier démarrage, et chaque fois qu'il n'existe pas de dernière connexion enregistrée, la fenêtre propose la connexion prédéfinie lors de la compilation : serveur, port, utilisateur, fichier de la clé privée et éventuelle phrase secrète déjà renseignés. En pratique, il suffit d'appuyer sur Connetti.

Après chaque connexion réussie, les valeurs utilisées sont enregistrées et reproposées au démarrage suivant. Le bouton **Predefiniti** ramène à tout moment aux valeurs prédéfinies, y compris la phrase secrète.

Les valeurs prédéfinies se trouvent dans le module `DefaultConnection` au début de `ConnectionStore.vb` : pour les changer, il suffit de modifier les constantes et de recompiler.

3.3 Mémorisation de la phrase secrète ou du mot de passe

Lorsque la case **Memorizza** est cochée, la phrase secrète ou le mot de passe sont écrits dans le fichier `connessione.xml`, chiffrés avec DPAPI de Windows (portée utilisateur actuel). Seul le même utilisateur Windows sur le même PC peut les déchiffrer : si vous copiez le fichier sur un autre ordinateur, ou le lisez avec un autre compte, le champ apparaît vide et doit être ressaisi.

Si vous décochez la case, la valeur n'est pas enregistrée : au prochain démarrage, le champ sera vide. Pendant la même session, le programme la retient tout de même pour les reconnexions.

3.4 Contrôles avant la connexion

Lorsque vous appuyez sur **Connetti**, la fenêtre contrôle les champs obligatoires et, pour l'authentification par clé, essaie immédiatement de lire et de déchiffrer la clé. Ainsi, un fichier inexistant ou une phrase secrète erronée sont signalés en rouge sous les champs sans attendre le serveur ; le curseur se place sur le champ à corriger.

Si la connexion au serveur échoue (serveur éteint, port erroné, authentification refusée, clé d'hôte non acceptée), la fenêtre est rouverte avec le motif de l'erreur et les données déjà saisies, prête pour une nouvelle tentative.



Figure 2 – Fenêtre reproposée après une erreur de connexion

4. Vérification de la clé d'hôte

Chaque serveur SSH possède sa propre clé d'hôte. La contrôler garantit que vous communiquez vraiment avec votre serveur et non avec un intermédiaire (attaque de l'homme du milieu). La version précédente acceptait n'importe quelle clé ; désormais, Albatros Console se comporte comme la commande ssh.

4.1 Première connexion

La première fois que vous vous connectez à un serveur (ou à un nouveau port du même serveur), une demande de confirmation apparaît avec le type de clé et son empreinte SHA256.

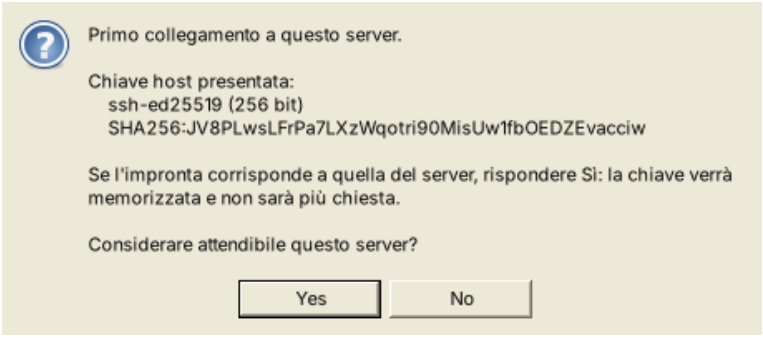

Figure 3 – Confirmation de la clé d'hôte à la première connexion (aspect indicatif)

Pour vérifier l'empreinte, sur le serveur, depuis une session déjà sûre (console du serveur ou PuTTY déjà configuré) :

```
for f in /etc/ssh/ssh_host_*_key.pub; do ssh-keygen -lf "$f"; done
```

L'une des lignes doit indiquer la même empreinte `SHA256:...` que celle affichée par le programme. Si vous répondez **Oui**, la clé est mémorisée et ne sera plus demandée ; si vous répondez **Non**, la connexion est annulée.

> **Remarque.** Après la mise à jour vers cette version, la confirmation est demandée une fois également pour le serveur prédéfini, car la liste des clés de confiance part vide.

4.2 Clé modifiée

Si le serveur présente une clé différente de celle qui est mémorisée, un avertissement apparaît avec la nouvelle empreinte ; le bouton proposé est **Non**. Le changement est normal après la réinstallation du système d'exploitation ou la régénération des clés SSH ; dans tous les autres cas, il vaut mieux ne pas continuer et comprendre d'abord la raison. Si vous répondez **Oui**, la nouvelle clé remplace celle qui est mémorisée.

L'explorateur de fichiers, le transfert SFTP et le moniteur de redémarrage n'acceptent que les clés déjà mémorisées et ne posent pas de questions : la confirmation a toujours lieu dans la connexion principale.

> **Attention.** Pour oublier la clé d'un serveur, il suffit de supprimer la ligne correspondante dans le fichier `known_hosts.txt` (voir le chapitre 7).

5. La fenêtre du terminal

Une fois la connexion réussie, le titre de la fenêtre affiche `utente@host:porta` et le terminal reçoit l'invite du serveur.



Figure 4 – Terminal connecté, avec barre d'état (aspect indicatif)

5.1 Menu Connessione

Commande	Fonction
Nuova connessione...	Rouvre la fenêtre d'identifiants. Si vous confirmez, la session actuelle est fermée et la nouvelle s'ouvre ; si vous annulez, la session actuelle reste telle quelle.
Riconnetti	Se reconnecte au dernier serveur avec les mêmes identifiants, sans rouvrir la fenêtre. En cas d'échec, ouvre la fenêtre avec le motif.
Disconnetti	Ferme la session. La fenêtre reste ouverte avec le contenu du terminal ; vous pouvez vous reconnecter depuis le menu.
Esci	Ferme le programme.

5.2 Menu File

Commande	Fonction
Trasferimento file SFTP...	Téléchargement, chargement et copie serveur → serveur, avec plusieurs transferts en parallèle.
Esplora file remoto...	Navigation dans les dossiers distants, création et suppression de fichiers et de dossiers, autorisations Unix, ouverture à la volée avec Notepad++ ou le Bloc-notes, téléchargement.

Les deux fenêtres utilisent les identifiants de la connexion en cours et ne sont disponibles qu'en étant connecté. Leur fonctionnement est inchangé par rapport à la version précédente.

5.3 Barre d'état

À gauche, elle affiche l'état de la connexion (Connessione in corso, Connesso a ..., Disconnesso da ..., Non connesso) ; à droite, l'état du moniteur de redémarrage (actif, non disponible) ou le motif de la dernière déconnexion.

5.4 Clavier et souris

Touche ou action	Effet
Ctrl+C	Copie la sélection ; s'il n'y a pas de sélection, envoie l'interruption (SIGINT) au programme distant.
Ctrl+V, clic droit sans sélection	Colle le texte du presse-papiers.
Clic droit avec sélection	Copie la sélection.
Glisser avec le bouton gauche	Sélectionne le texte, y compris dans l'historique.
Molette de la souris	Fait défiler l'historique (5000 lignes). Une barre jaune à droite indique que vous n'êtes pas en bas.
Maj+Pg préc. / Maj+Pg suiv.	Fait défiler l'historique d'une page (nouveauté).
Flèches, Origine, Fin	Envoyées au serveur ; dans les programmes en plein écran (vim, less, htop), elles utilisent automatiquement le mode curseur application.
Pg préc., Pg suiv., Inser, Suppr	Envoyées au serveur (nouveauté pour Pg préc., Pg suiv., Inser).
F1–F12	Envoyées au serveur comme dans xterm (nouveauté). F10 n'ouvre plus le menu de la fenêtre : elle sert par exemple à quitter mc et htop.
Maj+Tab	Envoyée au serveur (tabulation arrière).
Ctrl+D	Fin de saisie ; dans le shell, équivaut à exit.
Toute autre touche	Ramène la vue en bas, sur l'invite.

6. Moniteur de redémarrage et déconnexions

6.1 Fonctionnement du moniteur

Dès la connexion, le programme lit le `boot_id` du serveur, un identifiant que Linux change à chaque démarrage. À partir de ce moment, toutes les 3 secondes, il ouvre une brève connexion de contrôle séparée et le relit :

- si le `boot_id` a changé, le serveur a été redémarré et la déconnexion est signalée immédiatement ;
- si le contrôle échoue à cause d'un problème réseau, la déconnexion n'est signalée qu'après 2 erreurs consécutives (auparavant, une seule suffisait) ;
- si le serveur n'est pas sous Linux et que le `boot_id` n'est pas lisible, le moniteur reste désactivé et la barre d'état indique « Monitor riavvio non disponibile ».

La connexion de contrôle est séparée de la session principale car, après un redémarrage rapide, SSH.NET peut encore considérer comme « connectée » une session désormais morte.

6.2 Lorsque la connexion s'interrompt

Une ligne jaune avec le motif apparaît dans le terminal et une fenêtre de choix s'ouvre :

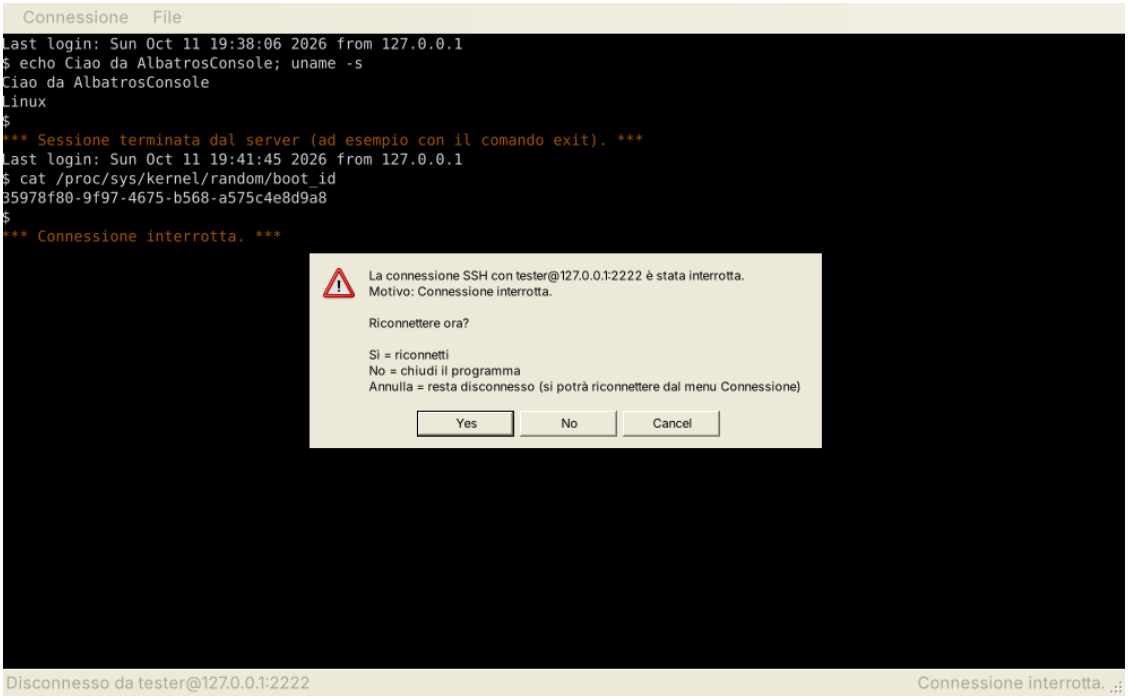

Figure 5 – Connexion interrompue : choix entre reconnexion, sortie ou rester déconnecté (aspect indicatif)

Réponse	Effet
Oui	Reconnexion immédiate avec les mêmes identifiants. Si le serveur ne répond pas encore (par exemple parce qu'il est en cours de redémarrage), la fenêtre s'ouvre avec le motif : attendez quelques secondes et appuyez sur Connetti.
Non	Ferme le programme (comportement de la version précédente).
Annuler	Reste déconnecté avec le contenu du terminal visible ; la reconnexion se fait depuis le menu Connessione.

Les motifs possibles sont : serveur redémarré ; session terminée par le serveur (par exemple avec la commande `exit`) ; connexion interrompue (réseau, serveur éteint, service SSH arrêté).

Sous Windows en français, les boutons des boîtes de message apparaissent comme Oui, Non et Annuler ; les figures de ce manuel ont été capturées dans un environnement de test et montrent les libellés anglais.

7. Fichiers de configuration

Le programme enregistre ses données dans le dossier `%APPDATA%\AlbatrosConsole` de l'utilisateur Windows (normalement `C:\Users\<utente>\AppData\Roaming\AlbatrosConsole`).

Fichier	Contenu	S'il est supprimé
`connessione.xml`	Dernière connexion réussie : serveur, port, utilisateur, type d'authentification, fichier de clé, choix de mémorisation et, si demandé, phrase secrète ou mot de passe chiffrés avec DPAPI.	Au prochain démarrage, les valeurs prédéfinies sont proposées.
`known_hosts.txt`	Une ligne par serveur : `[host]:porta tipo-chiave SHA256:impronta`.	À la prochaine connexion, la clé d'hôte est de nouveau demandée.

Exemple de ligne de `known_hosts.txt` :

```
[vps.example.org]:22 ssh-ed25519 SHA256:JV8PLwsLFrPa7LXzWqotri90MisUw1fbOEDZEvacciw
```

8. Sécurité

8.1 Phrase secrète prédéfinie dans le code

Pour satisfaire l'exigence d'une connexion prédéfinie prête à l'emploi, la phrase secrète de la clé prédéfinie reste écrite dans le code source (constante `KeyPassphrase` dans `ConnectionStore.vb`), exactement comme elle était écrite dans `Form1.vb` dans la version précédente. Par conséquent, elle est aussi lisible dans l'exécutable compilé, avec n'importe quel décompilateur .NET.

Si l'exécutable doit sortir de votre PC, il est conseillé de :

1. vider la constante `KeyPassphrase` (en laissant `""`) et recompiler ;
2. vous connecter une fois en saisissant la phrase secrète à la main avec la case **Memorizza** cochée : à partir de ce moment, elle reste enregistrée chiffrée avec DPAPI et n'apparaît plus dans le programme.

Le fichier de la clé privée, en revanche, reste sur le disque : quiconque en possède une copie et connaît la phrase secrète peut accéder au serveur. Conservez-le dans un dossier dont les autorisations sont limitées à l'utilisateur.

8.2 Ce que protège DPAPI

DPAPI lie les données au compte Windows : un autre utilisateur du même PC ou un autre ordinateur ne parviennent pas à les déchiffrer. En revanche, DPAPI ne protège pas contre un programme exécuté avec le même compte que l'utilisateur, qui pourrait les déchiffrer comme le fait Albatros Console.

8.3 Clés d'hôte

La vérification de la clé d'hôte ne protège la connexion que si l'empreinte est réellement comparée à celle du serveur lors de la première connexion (section 4.1). Un avertissement de clé modifiée ne doit jamais être accepté sans en connaître la raison.

9. Dépannage

Message	Cause probable	Que faire
Il file della chiave privata non esiste	Chemin erroné ou fichier déplacé.	Corrigez le chemin ou utilisez Sfoglia...
La chiave privata è protetta: inserire la passphrase	Clé chiffrée et champ de la phrase secrète vide.	Saisissez la phrase secrète.
Impossibile leggere la chiave privata: passphrase errata oppure file non valido	Phrase secrète erronée ou fichier qui n'est pas une clé privée.	Ressaisissez la phrase secrète (utilisez Mostra pour la contrôler) ou choisissez le bon fichier.
Server non trovato	Nom d'hôte erroné ou DNS inaccessible.	Vérifiez le nom ou utilisez l'adresse IP.
Connessione rifiutata ... il servizio SSH non è in ascolto	Port erroné ou sshd arrêté.	Vérifiez le port ; sur le serveur : `sudo systemctl status ssh`.
Il server ... non è raggiungibile / Tempo scaduto	Serveur éteint ou en cours de redémarrage, pare-feu, réseau absent.	Attendez et réessayez ; vérifiez le pare-feu et la connexion.
Autenticazione rifiutata dal server	Clé publique non autorisée, utilisateur ou mot de passe erronés.	Contrôlez `~/.ssh/authorized_keys` sur le serveur, l'utilisateur, le mot de passe.
Connessione annullata: la chiave host ... non è stata accettata	Vous avez répondu Non à la vérification de la clé d'hôte.	Vérifiez l'empreinte (4.1) et réessayez.
« Host key could not be verified » dans l'explorateur ou dans le transfert	La clé d'hôte a changé après l'ouverture de la session.	Reconnectez-vous depuis le menu Connessione et confirmez la nouvelle clé.
Monitor riavvio non disponibile	Le serveur n'expose pas `/proc/sys/kernel/random/boot_id` (il n'est pas sous Linux).	Aucune action : le terminal fonctionne normalement, sans détection du redémarrage.

10. Notes pour le développeur

10.1 Structure

Classe / module	Rôle
`Form1`	Fenêtre principale : menus, barre d'état, connexion asynchrone, lecture du shell, moniteur, gestion des déconnexions.
`FormLogin`	Fenêtre d'identifiants ; renvoie un `SshConnectionProfile` déjà vérifié et les `ConnectionSettings` à enregistrer.
`SshConnectionProfile`	Paramètres de connexion immuables. `CreateSshClient` et `CreateSftpClient` renvoient des clients non encore connectés avec le contrôle de la clé d'hôte déjà associé. La clé déchiffrée est mise en cache par thread (`ThreadLocal`), car les instances de clé de SSH.NET ne sont pas garanties thread-safe et les transferts parallèles s'exécutent sur des threads différents.
`ConnectionSettings`, `ConnectionStore`, `DefaultConnection`	Valeurs de la fenêtre, enregistrement dans `connessione.xml` avec DPAPI, valeurs prédéfinies.
`HostKeyStore`	Lecture et écriture de `known_hosts.txt` ; `Attach` associe la vérification à l'événement `HostKeyReceived`.
`FormTransfer`, `RemoteBrowserDialog`	Logique inchangée ; le constructeur reçoit désormais un `SshConnectionProfile` au lieu des cinq paramètres hôte, port, utilisateur, clé et phrase secrète.
`TerminalControl`, `TerminalBuffer`	Ajout des touches de fonction, Pg préc./Pg suiv., mode DECCKM, défilement avec Maj+Pg préc./Pg suiv., police grasse en cache et dessin plus léger de l'arrière-plan.

10.2 Générations de connexion

Chaque connexion reçoit un numéro de « génération ». Le thread de lecture du shell et le minuteur du moniteur retiennent la génération avec laquelle ils ont démarré ; lorsque l'utilisateur se déconnecte ou ouvre une nouvelle connexion, la génération change et leurs éventuels signalements tardifs sont ignorés. Ainsi, la fermeture volontaire d'une session ne provoque pas de faux avertissements de déconnexion.

10.3 Paramètres modifiables

Constante (dans Form1.vb)	Valeur	Signification
`ConnectTimeoutSeconds`	15	Durée maximale pour la connexion principale.
`MonitorIntervalSeconds`	3	Intervalle entre deux contrôles du moniteur.
`MonitorTimeoutSeconds`	4	Durée maximale de chaque contrôle.
`MonitorMaxFailures`	2	Erreurs réseau consécutives avant de déclarer la connexion perdue.

10.4 Vérifications effectuées

Compilation sans erreurs ni avertissements avec .NET Framework 4.8. Sur un serveur OpenSSH de test, les éléments suivants ont été vérifiés : clé avec phrase secrète correcte, erronée et manquante ; accès par clé et par mot de passe ; clé d'hôte nouvelle, mémorisée et modifiée ; SFTP y compris avec six connexions parallèles ; identifiants chiffrés ; le cycle fenêtre → clé d'hôte → terminal → déconnexion → reconnexion ; la détection du serveur arrêté par le moniteur.

Albatros Console

Terminale SSH e SFTP per Windows

Manuale d'uso

Versione con maschera delle credenziali, verifica della chiave host e riconnessione

Ottobre 2026

Indice

1. Introduzione....34
1.1 Novità di questa versione....34
2. Requisiti e compilazione....35
2.1 Requisiti....35
2.2 Aprire e compilare il progetto....35
3. La maschera delle credenziali....36
3.1 I campi....36
3.2 Il collegamento preimpostato....36
3.3 Memorizzazione della passphrase o della password....37
3.4 Controlli prima della connessione....37
4. Verifica della chiave host....38
4.1 Primo collegamento....38
4.2 Chiave cambiata....38
5. La finestra del terminale....39
5.1 Menu Connessione....39
5.2 Menu File....39
5.3 Barra di stato....40
5.4 Tastiera e mouse....40
6. Monitor del riavvio e disconnessioni....41
6.1 Come funziona il monitor....41
6.2 Quando la connessione si interrompe....41
7. File di configurazione....42
8. Sicurezza....43
8.1 Passphrase predefinita nel codice....43
8.2 Cosa protegge DPAPI....43
8.3 Chiavi host....43
9. Risoluzione dei problemi....44
10. Note per lo sviluppatore....45
10.1 Struttura....45
10.2 Generazioni di connessione....45
10.3 Parametri modificabili....45
10.4 Verifiche eseguite....45

1. Introduzione

Albatros Console è un terminale SSH per Windows scritto in VB.NET (.NET Framework 4.8, libreria SSH.NET). Oltre alla shell interattiva in stile xterm offre l'esplorazione dei file remoti e il trasferimento SFTP, e un monitor che si accorge quando il server viene riavviato.

Questo manuale descrive la versione con **maschera iniziale delle credenziali**. Rispetto alla versione precedente, in cui server, utente, chiave e passphrase erano scritti nel codice e la connessione partiva da sola, ora all'avvio compare una finestra che propone il collegamento preimpostato e permette di modificarlo.

1.1 Novità di questa versione

- **Maschera delle credenziali** all'avvio, con i valori del collegamento preimpostato già compilati e il pulsante **Predefiniti** per ripristinarli.
- Autenticazione con **chiave privata** (PPK di PuTTY o formato OpenSSH, con o senza passphrase) oppure con **password**.
- **Memorizzazione facoltativa** di passphrase o password, cifrata con DPAPI di Windows per l'utente corrente.
- **Verifica della chiave host** del server, come fa il comando ssh: al primo collegamento viene chiesta conferma, poi una chiave diversa viene segnalata.
- Connessione in background: la finestra non si blocca mentre il server risponde e gli **errori vengono spiegati** in italiano invece di chiudere il programma.
- Menu **Connessione** con Nuova connessione, Riconnetti e Disconnetti; dopo un'interruzione il programma propone di **riconnettersi** invece di chiudersi.
- **Barra di stato** con il server collegato e lo stato del monitor del riavvio.
- Monitor del riavvio più robusto: un singolo errore di rete non basta più a dichiarare la connessione persa.
- La chiave privata viene **decifrata una sola volta** e riusata da terminale, monitor, esploratore e trasferimenti (prima veniva riletta a ogni operazione SFTP e a ogni controllo, cosa lenta con le chiavi PPK v3).
- Terminale: tasti funzione F1–F12, PagSu/PagGiù, Ins, Maiusc+Tab, modalità cursore applicazione (vim, less, htop), **Maiusc+PagSu/PagGiù** per scorrere lo storico; corretta la prima riga che restava nascosta sotto il menu.

2. Requisiti e compilazione

2.1 Requisiti

- Windows 10/11 o Windows Server 2016 e successivi, con .NET Framework 4.8.
- Per compilare: Visual Studio 2019 o 2022 con il carico di lavoro per applicazioni desktop .NET.
- Server remoto con SSH attivo. Il monitor del riavvio richiede un server Linux (legge `/proc/sys/kernel/random/boot_id`).

2.2 Aprire e compilare il progetto

1. Estrarre lo zip e aprire `AlbatrosConsole.sln` in Visual Studio.
2. La cartella `packages` contiene già i pacchetti NuGet (SSH.NET 2026.0.0 e dipendenze). Se Visual Studio li segnala come mancanti, usare Ripristina pacchetti NuGet.
3. Compilare con Compila → Ricompila soluzione e avviare con F5.

Rispetto alla versione precedente il progetto contiene quattro file nuovi e un riferimento in più; sono già inseriti nel file `AlbatrosConsole.vbproj`:

Elemento	Contenuto
`FormLogin.vb`	Maschera delle credenziali.
`SshConnectionProfile.vb`	Parametri di connessione, cache della chiave decifrata, creazione dei client SSH/SFTP.
`ConnectionStore.vb`	Valori del collegamento preimpostato (modulo `DefaultConnection`) e salvataggio dell'ultima connessione.
`HostKeyStore.vb`	Archivio delle chiavi host attendibili (equivalente di known_hosts).
Riferimento `System.Security`	Necessario per la cifratura DPAPI (`ProtectedData`).

3. La maschera delle credenziali

All'avvio il programma apre la finestra **Albatros Console – Connessione**. Il terminale resta in attesa dietro la maschera finché la connessione non riesce.

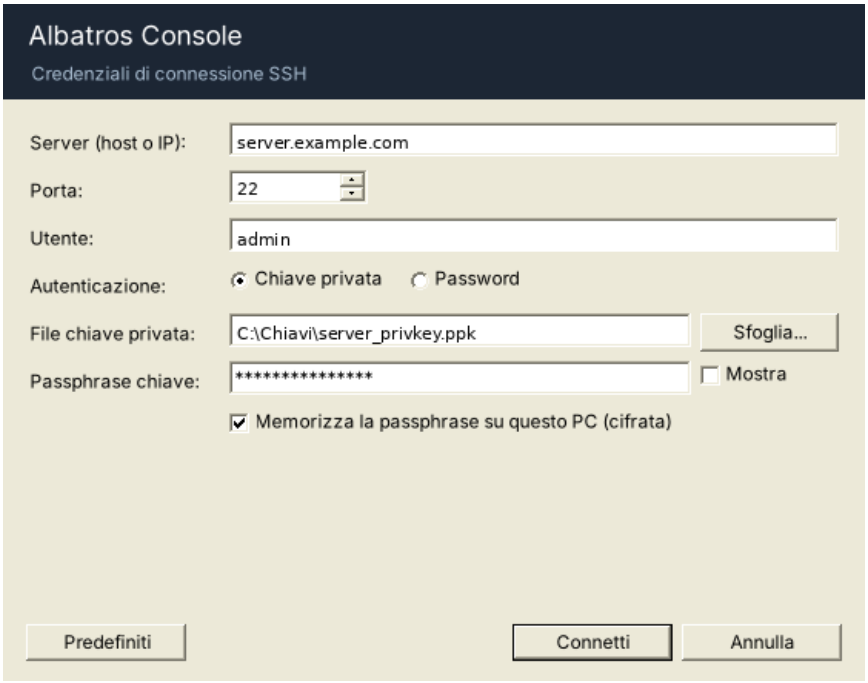

Figura 1 – Maschera delle credenziali con il collegamento preimpostato (aspetto indicativo)

3.1 I campi

Campo	Significato
Server (host o IP)	Nome DNS o indirizzo IP del server SSH.
Porta	Porta TCP del servizio SSH (1–65535). Lo standard è 22.
Utente	Nome dell'utente remoto.
Autenticazione	**Chiave privata** oppure **Password**. In modalità Password il campo del file chiave è disattivato.
File chiave privata	Percorso del file della chiave: PPK di PuTTY (v2 o v3) o chiave OpenSSH. Il pulsante **Sfoglia...** apre la scelta del file.
Passphrase chiave / Password	Passphrase che protegge la chiave, oppure la password dell'utente. Lasciare vuoto se la chiave non è protetta. **Mostra** rende visibile il testo digitato.
Memorizza ... su questo PC (cifrata)	Se spuntata, passphrase o password vengono salvate cifrate per l'utente Windows corrente (vedere 3.3).

Pulsanti: **Connetti** (anche con Invio) avvia il collegamento; **Annulla** (anche con Esc) all'avvio chiude il programma, mentre da menu chiude solo la maschera lasciando invariata la sessione in corso; **Predefiniti** ripristina il collegamento preimpostato.

3.2 Il collegamento preimpostato

Al primo avvio, e ogni volta che non esiste un'ultima connessione salvata, la maschera propone il collegamento preimpostato in fase di compilazione: server, porta, utente, file della chiave privata ed eventuale passphrase già compilati. In pratica basta premere Connetti.

Dopo ogni connessione riuscita i valori usati vengono salvati e riproposti all'avvio successivo. Il pulsante **Predefiniti** riporta in qualsiasi momento ai valori preimpostati, compresa la passphrase.

I valori preimpostati si trovano nel modulo `DefaultConnection` all'inizio di `ConnectionStore.vb`: per cambiarli basta modificare le costanti e ricompilare.

3.3 Memorizzazione della passphrase o della password

Con la casella **Memorizza** spuntata, passphrase o password vengono scritte nel file `connessione.xml` cifrate con DPAPI di Windows (ambito utente corrente). Solo lo stesso utente Windows sullo stesso PC può decifrarle: copiando il file su un altro computer, o leggendolo con un altro account, il campo risulta vuoto e va reinserito.

Togliendo la spunta il valore non viene salvato: al prossimo avvio il campo sarà vuoto. Durante la stessa sessione il programma lo ricorda comunque per le riconnessioni.

3.4 Controlli prima della connessione

Premendo **Connetti** la maschera controlla i campi obbligatori e, per l'autenticazione a chiave, prova subito a leggere e decifrare la chiave. Così un file inesistente o una passphrase errata vengono segnalati in rosso sotto i campi senza attendere il server; il cursore si posiziona sul campo da correggere.

Se la connessione al server non riesce (server spento, porta errata, autenticazione rifiutata, chiave host non accettata) la maschera viene riaperta con il motivo dell'errore e con i dati già inseriti, pronta per un nuovo tentativo.

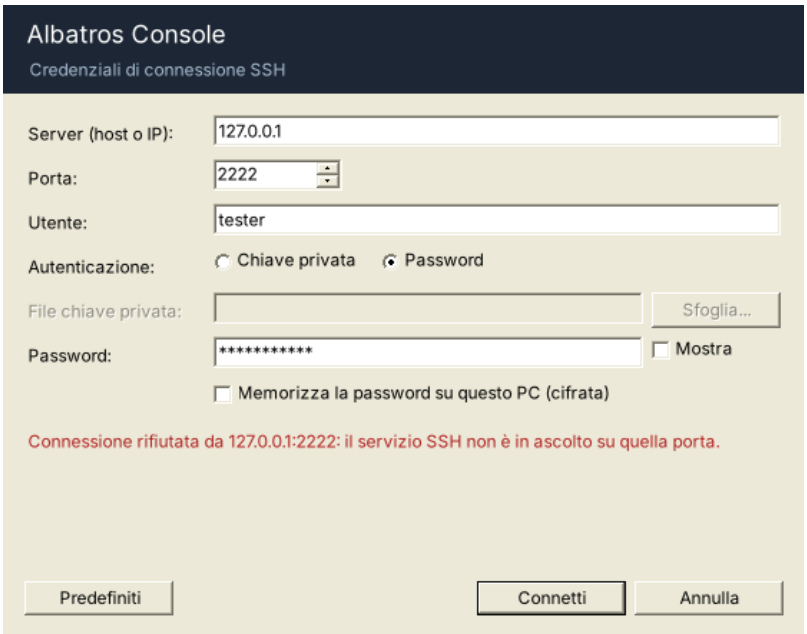

Figura 2 – Maschera riproposta dopo un errore di connessione

4. Verifica della chiave host

Ogni server SSH ha una propria chiave host. Controllarla garantisce di parlare davvero con il proprio server e non con un intermediario (attacco man-in-the-middle). La versione precedente accettava qualunque chiave; ora Albatros Console si comporta come il comando ssh.

4.1 Primo collegamento

La prima volta che ci si collega a un server (o a una nuova porta dello stesso server) compare la richiesta di conferma con il tipo di chiave e la sua impronta SHA256.

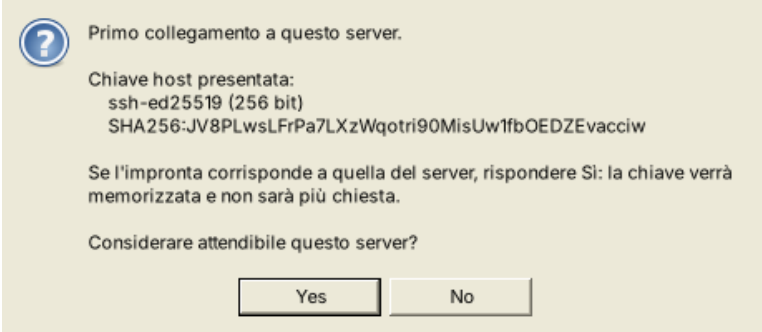

Figura 3 – Conferma della chiave host al primo collegamento (aspetto indicativo)

Per verificare l'impronta, sul server, da una sessione già fidata (console del server o PuTTY già configurato):

```
for f in /etc/ssh/ssh_host_*_key.pub; do ssh-keygen -lf "$f"; done
```

Una delle righe deve riportare la stessa impronta `SHA256:...` mostrata dal programma. Rispondendo **Sì** la chiave viene memorizzata e non sarà più chiesta; rispondendo **No** la connessione viene annullata.

> **Nota.** Dopo l'aggiornamento a questa versione la conferma viene chiesta una volta anche per il server preimpostato, perché l'elenco delle chiavi attendibili parte vuoto.

4.2 Chiave cambiata

Se il server presenta una chiave diversa da quella memorizzata compare un avviso con l'impronta nuova; il pulsante proposto è **No**. Il cambiamento è normale dopo la reinstallazione del sistema operativo o la rigenerazione delle chiavi SSH; in ogni altro caso conviene non proseguire e capire prima il motivo. Rispondendo **Sì** la nuova chiave sostituisce quella memorizzata.

L'esploratore file, il trasferimento SFTP e il monitor del riavvio accettano solo chiavi già memorizzate e non fanno domande: la conferma avviene sempre nella connessione principale.

> **Attenzione.** Per dimenticare la chiave di un server basta cancellare la riga corrispondente nel file `known_hosts.txt` (vedere il capitolo 7).

5. La finestra del terminale

A connessione riuscita il titolo della finestra mostra `utente@host:porta` e il terminale riceve il prompt del server.



Figura 4 – Terminale collegato, con barra di stato (aspetto indicativo)

5.1 Menu Connessione

Voce	Funzione
Nuova connessione...	Riapre la maschera delle credenziali. Se si conferma, la sessione attuale viene chiusa e si apre la nuova; se si annulla, la sessione attuale resta com'è.
Riconnetti	Si ricollega all'ultimo server con le stesse credenziali, senza riaprire la maschera. Se non riesce, apre la maschera con il motivo.
Disconnetti	Chiude la sessione. La finestra resta aperta con il contenuto del terminale; ci si può ricollegare dal menu.
Esci	Chiude il programma.

5.2 Menu File

Voce	Funzione
Trasferimento file SFTP...	Download, upload e copia server → server, con più trasferimenti in parallelo.
Esplora file remoto...	Navigazione tra le cartelle remote, creazione ed eliminazione di file e cartelle, permessi Unix, apertura al volo con Notepad++ o Notepad, download.

Entrambe le finestre usano le credenziali della connessione in corso e sono disponibili solo da connessi. Il loro funzionamento è invariato rispetto alla versione precedente.

5.3 Barra di stato

A sinistra mostra lo stato della connessione (Connessione in corso, Connesso a ..., Disconnesso da ..., Non connesso); a destra lo stato del monitor del riavvio (attivo, non disponibile) oppure il motivo dell'ultima disconnessione.

5.4 Tastiera e mouse

Tasto o azione	Effetto
Ctrl+C	Copia la selezione; se non c'è selezione invia l'interruzione (SIGINT) al programma remoto.
Ctrl+V, clic destro senza selezione	Incolla il testo degli appunti.
Clic destro con selezione	Copia la selezione.
Trascinamento con il tasto sinistro	Seleziona il testo, anche nello storico.
Rotellina del mouse	Scorre lo storico (5000 righe). Una barra gialla a destra indica che non si è in fondo.
Maiusc+PagSu / Maiusc+PagGiù	Scorre lo storico di una pagina (novità).
Frecce, Home, Fine	Inviati al server; nei programmi a schermo intero (vim, less, htop) usano automaticamente la modalità cursore applicazione.
PagSu, PagGiù, Ins, Canc	Inviati al server (novità per PagSu, PagGiù, Ins).
F1–F12	Inviati al server come in xterm (novità). F10 non apre più il menu della finestra: serve per esempio a uscire da mc e htop.
Maiusc+Tab	Inviato al server (tab all'indietro).
Ctrl+D	Fine input; sulla shell equivale a exit.
Qualsiasi altro tasto	Riporta la vista in fondo, sul prompt.

6. Monitor del riavvio e disconnessioni

6.1 Come funziona il monitor

Appena collegato, il programma legge il `boot_id` del server, un identificativo che Linux cambia a ogni avvio. Da quel momento, ogni 3 secondi, apre una breve connessione di controllo separata e lo rilegge:

- se il `boot_id` è cambiato, il server è stato riavviato e la disconnessione viene segnalata subito;
- se il controllo fallisce per un problema di rete, la disconnessione viene segnalata solo dopo 2 errori consecutivi (prima ne bastava uno);
- se il server non è Linux e il `boot_id` non è leggibile, il monitor resta disattivato e la barra di stato riporta «Monitor riavvio non disponibile».

La connessione di controllo è separata dalla sessione principale perché, dopo un riavvio rapido, SSH.NET può considerare ancora «connessa» una sessione ormai morta.

6.2 Quando la connessione si interrompe

Nel terminale compare una riga gialla con il motivo e si apre una finestra di scelta:

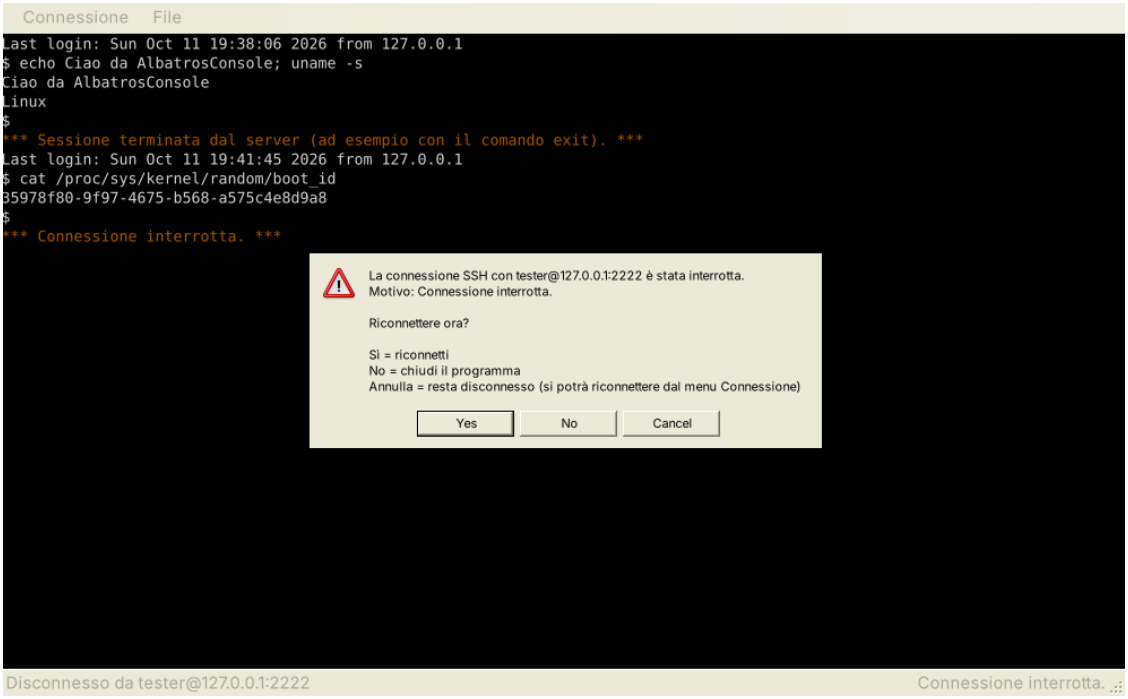

Figura 5 – Connessione interrotta: scelta tra riconnessione, uscita o restare disconnessi (aspetto indicativo)

Risposta	Effetto
Sì	Riconnessione immediata con le stesse credenziali. Se il server non risponde ancora (per esempio è in fase di riavvio) si apre la maschera con il motivo: attendere qualche secondo e premere Connetti.
No	Chiude il programma (comportamento della versione precedente).
Annulla	Resta disconnessi con il contenuto del terminale visibile; ci si ricollega dal menu Connessione.

I motivi possibili sono: server riavviato; sessione terminata dal server (per esempio con il comando `exit`); connessione interrotta (rete, server spento, servizio SSH fermato).

Su Windows in italiano i pulsanti delle finestre di messaggio appaiono come Sì, No e Annulla; le figure di questo manuale sono state riprese in un ambiente di prova e mostrano le diciture inglesi.

7. File di configurazione

Il programma salva i propri dati nella cartella `%APPDATA%\AlbatrosConsole` dell'utente Windows (di norma `C:\Users\<utente>\AppData\Roaming\AlbatrosConsole`).

File	Contenuto	Se viene cancellato
`connessione.xml`	Ultima connessione riuscita: server, porta, utente, tipo di autenticazione, file chiave, scelta di memorizzazione e, se richiesto, passphrase o password cifrata con DPAPI.	Al prossimo avvio vengono proposti i valori preimpostati.
`known_hosts.txt`	Una riga per server: `[host]:porta tipo-chiave SHA256:impronta`.	Al prossimo collegamento la chiave host viene chiesta di nuovo.

Esempio di riga di `known_hosts.txt`:

```
[vps.example.org]:22 ssh-ed25519 SHA256:JV8PLwsLFrPa7LXzWqotri90MisUw1fbOEDZEvacciw
```

8. Sicurezza

8.1 Passphrase predefinita nel codice

Per rispettare la richiesta di avere il collegamento preimpostato già pronto, la passphrase della chiave predefinita resta scritta nel sorgente (costante `KeyPassphrase` in `ConnectionStore.vb`), esattamente come nella versione precedente era scritta in `Form1.vb`. Di conseguenza è leggibile anche nell'eseguibile compilato, con un qualunque decompilatore .NET.

Se l'eseguibile deve uscire dal proprio PC, si consiglia di:

1. svuotare la costante `KeyPassphrase` (lasciando `""`) e ricompilare;
2. collegarsi una volta inserendo la passphrase a mano con la casella **Memorizza** spuntata: da quel momento resta salvata cifrata con DPAPI e non compare più nel programma.

Il file della chiave privata, invece, resta sul disco: chi ne possiede una copia e conosce la passphrase può accedere al server. Conservarlo in una cartella con permessi limitati all'utente.

8.2 Cosa protegge DPAPI

DPAPI lega i dati all'account Windows: un altro utente dello stesso PC o un altro computer non riescono a decifrarli. Non protegge invece da un programma eseguito con lo stesso account dell'utente, che potrebbe decifrarli come fa Albatros Console.

8.3 Chiavi host

La verifica della chiave host protegge la connessione solo se l'impronta viene davvero confrontata con quella del server al primo collegamento (paragrafo 4.1). Un avviso di chiave cambiata non va mai accettato senza conoscerne il motivo.

9. Risoluzione dei problemi

Messaggio	Causa probabile	Cosa fare
Il file della chiave privata non esiste	Percorso errato o file spostato.	Correggere il percorso o usare Sfoglia...
La chiave privata è protetta: inserire la passphrase	Chiave cifrata e campo passphrase vuoto.	Inserire la passphrase.
Impossibile leggere la chiave privata: passphrase errata oppure file non valido	Passphrase sbagliata o file che non è una chiave privata.	Ridigitare la passphrase (usare Mostra per controllarla) o scegliere il file corretto.
Server non trovato	Nome host errato o DNS non raggiungibile.	Verificare il nome o usare l'indirizzo IP.
Connessione rifiutata ... il servizio SSH non è in ascolto	Porta errata o sshd fermo.	Verificare la porta; sul server: `sudo systemctl status ssh`.
Il server ... non è raggiungibile / Tempo scaduto	Server spento o in riavvio, firewall, rete assente.	Attendere e riprovare; verificare firewall e connessione.
Autenticazione rifiutata dal server	Chiave pubblica non autorizzata, utente o password errati.	Controllare `~/.ssh/authorized_keys` sul server, l'utente, la password.
Connessione annullata: la chiave host ... non è stata accettata	Si è risposto No alla verifica della chiave host.	Verificare l'impronta (4.1) e riprovare.
«Host key could not be verified» nell'esploratore o nel trasferimento	La chiave host è cambiata dopo l'apertura della sessione.	Riconnettersi dal menu Connessione e confermare la nuova chiave.
Monitor riavvio non disponibile	Il server non espone `/proc/sys/kernel/random/boot_id` (non è Linux).	Nessuna azione: il terminale funziona normalmente, senza rilevazione del riavvio.

10. Note per lo sviluppatore

10.1 Struttura

Classe / modulo	Ruolo
`Form1`	Finestra principale: menu, barra di stato, connessione asincrona, lettura della shell, monitor, gestione delle disconnessioni.
`FormLogin`	Maschera delle credenziali; restituisce un `SshConnectionProfile` già verificato e i `ConnectionSettings` da salvare.
`SshConnectionProfile`	Parametri immutabili di connessione. `CreateSshClient` e `CreateSftpClient` restituiscono client non ancora connessi con il controllo della chiave host già collegato. La chiave decifrata è tenuta in cache per thread (`ThreadLocal`), perché le istanze di chiave di SSH.NET non sono garantite thread-safe e i trasferimenti paralleli girano su thread diversi.
`ConnectionSettings`, `ConnectionStore`, `DefaultConnection`	Valori della maschera, salvataggio in `connessione.xml` con DPAPI, valori preimpostati.
`HostKeyStore`	Lettura e scrittura di `known_hosts.txt`; `Attach` collega la verifica all'evento `HostKeyReceived`.
`FormTransfer`, `RemoteBrowserDialog`	Invariati nella logica; il costruttore riceve ora un `SshConnectionProfile` al posto dei cinque parametri host, porta, utente, chiave e passphrase.
`TerminalControl`, `TerminalBuffer`	Aggiunti tasti funzione, PagSu/PagGiù, modalità DECCKM, scorrimento con Maiusc+PagSu/PagGiù, font grassetto in cache e disegno più leggero dello sfondo.

10.2 Generazioni di connessione

Ogni connessione riceve un numero di «generazione». Il thread di lettura della shell e il timer del monitor ricordano la generazione con cui sono partiti; quando l'utente si disconnette o apre una nuova connessione la generazione cambia e le loro eventuali segnalazioni tardive vengono ignorate. In questo modo la chiusura volontaria di una sessione non provoca falsi avvisi di disconnessione.

10.3 Parametri modificabili

Costante (in Form1.vb)	Valore	Significato
`ConnectTimeoutSeconds`	15	Tempo massimo per la connessione principale.
`MonitorIntervalSeconds`	3	Intervallo tra due controlli del monitor.
`MonitorTimeoutSeconds`	4	Tempo massimo di ciascun controllo.
`MonitorMaxFailures`	2	Errori di rete consecutivi prima di dichiarare la connessione persa.

10.4 Verifiche eseguite

Compilazione senza errori né avvisi contro .NET Framework 4.8. Su un server OpenSSH di prova sono stati verificati: chiave con passphrase corretta, errata e mancante; accesso con chiave e con password; chiave host nuova, memorizzata e cambiata; SFTP anche con sei connessioni parallele; credenziali cifrate; il ciclo maschera → chiave host → terminale → disconnessione → riconnessione; la rilevazione del server fermo da parte del monitor.

Albatros Console

Terminal SSH și SFTP pentru Windows

Manual de utilizare

Versiunea cu fereastra de acreditări, verificarea cheii gazdei și reconectare

Octombrie 2026

Cuprins

1. Introducere....48
1.1 Noutățile acestei versiuni....48
2. Cerințe și compilare....49
2.1 Cerințe....49
2.2 Deschiderea și compilarea proiectului....49
3. Fereastra de acreditări....50
3.1 Câmpurile....50
3.2 Conexiunea presetată....50
3.3 Memorarea frazei de acces sau a parolei....51
3.4 Verificări înainte de conectare....51
4. Verificarea cheii gazdei....52
4.1 Prima conectare....52
4.2 Cheie schimbată....52
5. Fereastra terminalului....53
5.1 Meniul Connessione....53
5.2 Meniul File....53
5.3 Bara de stare....54
5.4 Tastatură și mouse....54
6. Monitorul de repornire și deconectări....55
6.1 Cum funcționează monitorul....55
6.2 Când conexiunea se întrerupe....55
7. Fișiere de configurare....57
8. Securitate....58
8.1 Fraza de acces implicită din cod....58
8.2 Ce protejează DPAPI....58
8.3 Cheile gazdelor....58
9. Depanare....59
10. Note pentru dezvoltator....60
10.1 Structură....60
10.2 Generații de conexiune....60
10.3 Parametri modificabili....60
10.4 Verificări efectuate....60

1. Introducere

Albatros Console este un terminal SSH pentru Windows scris în VB.NET (.NET Framework 4.8, biblioteca SSH.NET). Pe lângă shell-ul interactiv în stil xterm, oferă explorarea fișierelor la distanță și transferul SFTP, precum și un monitor care detectează când serverul este repornit.

Acest manual descrie versiunea cu **fereastră inițială de acreditări**. Față de versiunea anterioară, în care serverul, utilizatorul, cheia și fraza de acces erau scrise în cod și conexiunea pornea singură, acum la pornire apare o fereastră care propune conexiunea presetată și permite modificarea ei.

1.1 Noutățile acestei versiuni

- **Fereastră de acreditări** la pornire, cu valorile conexiunii presetate deja completate și butonul **Predefiniti** (Implicite) pentru a le restabili.
- Autentificare cu **cheie privată** (PPK din PuTTY sau format OpenSSH, cu sau fără frază de acces) sau cu **parolă**.
- **Memorare opțională** a frazei de acces sau a parolei, criptată cu DPAPI din Windows pentru utilizatorul curent.
- **Verificarea cheii gazdei** serverului, ca la comanda ssh: la prima conectare se cere confirmare, apoi o cheie diferită este semnalată.
- Conectare în fundal: fereastra nu se blochează cât timp serverul răspunde, iar **erorile sunt explicate** în italiană în loc ca programul să se închidă.
- Meniul **Connessione** cu Nuova connessione, Riconnetti și Disconnetti; după o întrerupere, programul propune **reconectarea** în loc să se închidă.
- **Bară de stare** cu serverul conectat și starea monitorului de repornire.
- Monitor de repornire mai robust: o singură eroare de rețea nu mai este suficientă pentru a declara conexiunea pierdută.
- Cheia privată este **decriptată o singură dată** și reutilizată de terminal, monitor, explorator și transferuri (anterior era recitită la fiecare operațiune SFTP și la fiecare verificare, lucru lent cu cheile PPK v3).
- Terminal: tastele funcționale F1–F12, PgUp/PgDn, Ins, Shift+Tab, modul cursor aplicație (vim, less, htop), **Shift+PgUp/PgDn** pentru derularea istoricului; corectat primul rând care rămânea ascuns sub meniu.

2. Cerințe și compilare

2.1 Cerințe

- Windows 10/11 sau Windows Server 2016 și versiuni ulterioare, cu .NET Framework 4.8.
- Pentru compilare: Visual Studio 2019 sau 2022 cu sarcina de lucru pentru aplicații desktop .NET.
- Server la distanță cu SSH activ. Monitorul de repornire necesită un server Linux (citește `/proc/sys/kernel/random/boot_id`).

2.2 Deschiderea și compilarea proiectului

1. Extrageți arhiva zip și deschideți `AlbatrosConsole.sln` în Visual Studio.
2. Folderul `packages` conține deja pachetele NuGet (SSH.NET 2026.0.0 și dependențele). Dacă Visual Studio le semnalează ca lipsă, folosiți Restore NuGet Packages.
3. Compilați cu Build → Rebuild Solution și porniți cu F5.

Față de versiunea anterioară, proiectul conține patru fișiere noi și o referință în plus; acestea sunt deja incluse în fișierul `AlbatrosConsole.vbproj`:

Element	Conținut
`FormLogin.vb`	Fereastra de acreditări.
`SshConnectionProfile.vb`	Parametri de conectare, cache-ul cheii decriptate, crearea clienților SSH/SFTP.
`ConnectionStore.vb`	Valorile conexiunii presetate (modulul `DefaultConnection`) și salvarea ultimei conexiuni.
`HostKeyStore.vb`	Arhiva cheilor de gazdă de încredere (echivalentul known_hosts).
Referința `System.Security`	Necesară pentru criptarea DPAPI (`ProtectedData`).

3. Fereastra de acreditări

La pornire, programul deschide fereastra **Albatros Console – Connessione**. Terminalul rămâne în așteptare în spatele ferestrei până când conectarea reușește.

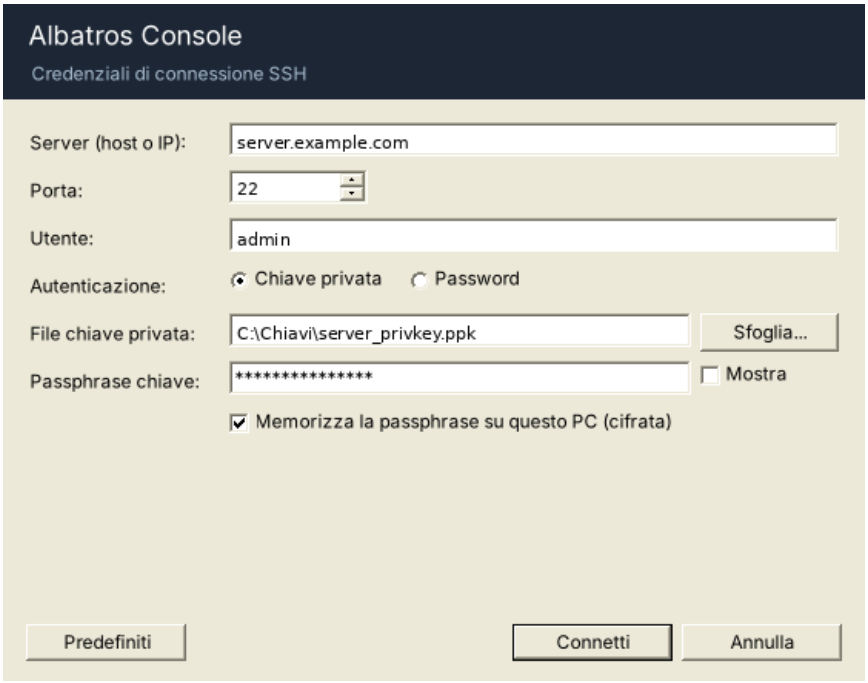

Figura 1 – Fereastra de acreditări cu conexiunea presetată (aspect orientativ)

3.1 Câmpurile

Câmp	Semnificație
Server (host o IP)	Numele DNS sau adresa IP a serverului SSH.
Porta	Portul TCP al serviciului SSH (1–65535). Standardul este 22.
Utente	Numele utilizatorului la distanță.
Autenticazione	**Chiave privata** sau **Password**. În modul Password, câmpul fișierului cheii este dezactivat.
File chiave privata	Calea fișierului cheii: PPK din PuTTY (v2 sau v3) sau cheie OpenSSH. Butonul **Sfoglia...** (Răsfoire...) deschide selecția fișierului.
Passphrase chiave / Password	Fraza de acces care protejează cheia sau parola utilizatorului. Lăsați gol dacă cheia nu este protejată. **Mostra** (Afișare) face vizibil textul tastat.
Memorizza ... su questo PC (cifrata)	Dacă este bifată, fraza de acces sau parola sunt salvate criptate pentru utilizatorul Windows curent (vezi 3.3).

Butoane: **Connetti** (Conectare; și cu Enter) pornește conectarea; **Annulla** (Anulare; și cu Esc) la pornire închide programul, în timp ce din meniu închide doar fereastra, lăsând neschimbată sesiunea curentă; **Predefiniti** restabilește conexiunea presetată.

3.2 Conexiunea presetată

La prima pornire, și de fiecare dată când nu există o ultimă conexiune salvată, fereastra propune conexiunea presetată la compilare: serverul, portul, utilizatorul, fișierul cheii private și eventuala frază de acces sunt deja completate. Practic, este suficient să apăsați Connetti.

După fiecare conectare reușită, valorile utilizate sunt salvate și propuse din nou la pornirea următoare. Butonul **Predefiniti** readuce oricând valorile presetate, inclusiv fraza de acces.

Valorile presetate se află în modulul `DefaultConnection` de la începutul fișierului `ConnectionStore.vb`: pentru a le schimba, este suficient să modificați constantele și să recompilați.

3.3 Memorarea frazei de acces sau a parolei

Cu caseta **Memorizza** (Memorare) bifată, fraza de acces sau parola sunt scrise în fișierul `connessione.xml` criptate cu DPAPI din Windows (domeniul utilizatorului curent). Numai același utilizator Windows pe același PC le poate decripta: dacă fișierul este copiat pe alt computer sau citit cu alt cont, câmpul apare gol și trebuie reintrodus.

Dacă debifați caseta, valoarea nu este salvată: la următoarea pornire câmpul va fi gol. În cadrul aceleiași sesiuni, programul o reține totuși pentru reconectări.

3.4 Verificări înainte de conectare

La apăsarea butonului **Connetti**, fereastra verifică câmpurile obligatorii și, pentru autentificarea cu cheie, încearcă imediat să citească și să decripteze cheia. Astfel, un fișier inexistent sau o frază de acces greșită sunt semnalate cu roșu sub câmpuri fără a aștepta serverul; cursorul se poziționează pe câmpul de corectat.

Dacă conectarea la server nu reușește (server oprit, port greșit, autentificare refuzată, cheia gazdei neacceptată), fereastra este redeschisă cu motivul erorii și cu datele deja introduse, pregătită pentru o nouă încercare.



Figura 2 – Fereastra afișată din nou după o eroare de conectare

4. Verificarea cheii gazdei

Fiecare server SSH are propria cheie de gazdă. Verificarea ei garantează că comunicați într-adevăr cu propriul server și nu cu un intermediar (atac man-in-the-middle). Versiunea anterioară accepta orice cheie; acum Albatros Console se comportă ca la comanda ssh.

4.1 Prima conectare

Prima dată când vă conectați la un server (sau la un port nou al aceluiași server), apare solicitarea de confirmare cu tipul cheii și amprenta sa SHA256.

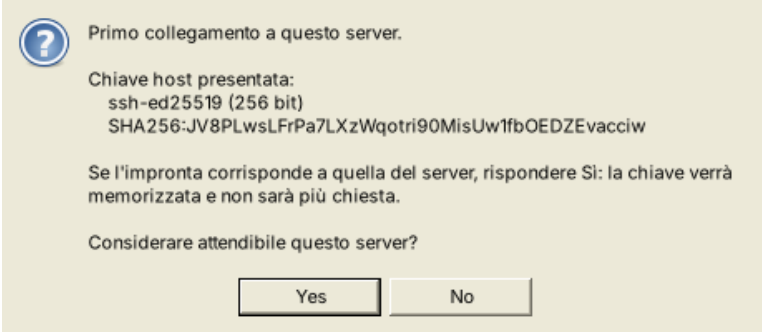

Figura 3 – Confirmarea cheii gazdei la prima conectare (aspect orientativ)

Pentru a verifica amprenta, pe server, dintr-o sesiune deja de încredere (consola serverului sau PuTTY deja configurat):

```
for f in /etc/ssh/ssh_host_*_key.pub; do ssh-keygen -lf "$f"; done
```

Unul dintre rânduri trebuie să conțină aceeași amprentă `SHA256:...` afișată de program. Răspunzând **Da**, cheia este memorată și nu va mai fi solicitată; răspunzând **Nu**, conectarea este anulată.

> **Notă.** După actualizarea la această versiune, confirmarea este solicitată o dată și pentru serverul presetat, deoarece lista cheilor de încredere pornește goală.

4.2 Cheie schimbată

Dacă serverul prezintă o cheie diferită de cea memorată, apare un avertisment cu noua amprentă; butonul propus este **Nu**. Schimbarea este normală după reinstalarea sistemului de operare sau regenerarea cheilor SSH; în orice alt caz este bine să nu continuați și să aflați mai întâi motivul. Răspunzând **Da**, noua cheie o înlocuiește pe cea memorată.

Exploratorul de fișiere, transferul SFTP și monitorul de repornire acceptă numai chei deja memorate și nu pun întrebări: confirmarea are loc întotdeauna în conexiunea principală.

> **Atenție.** Pentru a uita cheia unui server, este suficient să ștergeți rândul corespunzător din fișierul `known_hosts.txt` (vezi capitolul 7).

5. Fereastra terminalului

După conectarea reușită, titlul ferestrei afișează `utente@host:porta`, iar terminalul primește promptul serverului.



Figura 4 – Terminal conectat, cu bara de stare (aspect orientativ)

5.1 Meniul Connessione

Element	Funcție
Nuova connessione... (Conexiune nouă...)	Redeschide fereastra de acreditări. Dacă confirmați, sesiunea actuală este închisă și se deschide cea nouă; dacă anulați, sesiunea actuală rămâne neschimbată.
Riconnetti (Reconectare)	Se reconectează la ultimul server cu aceleași acreditări, fără a redeschide fereastra. Dacă nu reușește, deschide fereastra cu motivul.
Disconnetti (Deconectare)	Închide sesiunea. Fereastra rămâne deschisă cu conținutul terminalului; vă puteți reconecta din meniu.
Esci (Ieșire)	Închide programul.

5.2 Meniul File

Element	Funcție
Trasferimento file SFTP... (Transfer fișiere SFTP...)	Descărcare, încărcare și copiere server → server, cu mai multe transferuri în paralel.
Esplora file remoto... (Explorare fișiere la distanță...)	Navigare prin folderele la distanță, crearea și ștergerea fișierelor și folderelor, permisiuni Unix, deschidere rapidă cu Notepad++ sau Notepad, descărcare.

Ambele ferestre folosesc acreditările conexiunii curente și sunt disponibile numai când sunteți conectat. Funcționarea lor este neschimbată față de versiunea anterioară.

5.3 Bara de stare

În stânga afișează starea conexiunii (Connessione in corso, Connesso a ..., Disconnesso da ..., Non connesso); în dreapta, starea monitorului de repornire (activ, indisponibil) sau motivul ultimei deconectări.

5.4 Tastatură și mouse

Tastă sau acțiune	Efect
Ctrl+C	Copiază selecția; dacă nu există selecție, trimite întreruperea (SIGINT) programului de la distanță.
Ctrl+V, clic dreapta fără selecție	Lipește textul din clipboard.
Clic dreapta cu selecție	Copiază selecția.
Glisare cu butonul stâng	Selectează textul, inclusiv în istoric.
Rotița mouse-ului	Derulează istoricul (5000 de rânduri). O bară galbenă în dreapta indică faptul că nu sunteți la final.
Shift+PgUp / Shift+PgDn	Derulează istoricul cu o pagină (noutate).
Săgeți, Home, End	Trimise serverului; în programele pe ecran complet (vim, less, htop) folosesc automat modul cursor aplicație.
PgUp, PgDn, Ins, Del	Trimise serverului (noutate pentru PgUp, PgDn, Ins).
F1–F12	Trimise serverului ca în xterm (noutate). F10 nu mai deschide meniul ferestrei: servește, de exemplu, la ieșirea din mc și htop.
Shift+Tab	Trimis serverului (tab înapoi).
Ctrl+D	Sfârșit de intrare; în shell echivalează cu exit.
Orice altă tastă	Readuce vizualizarea la final, pe prompt.

6. Monitorul de repornire și deconectări

6.1 Cum funcționează monitorul

Imediat după conectare, programul citește `boot_id` al serverului, un identificator pe care Linux îl schimbă la fiecare pornire. Din acel moment, la fiecare 3 secunde, deschide o scurtă conexiune de control separată și îl recitește:

- dacă `boot_id` s-a schimbat, serverul a fost repornit și deconectarea este semnalată imediat;
- dacă verificarea eșuează din cauza unei probleme de rețea, deconectarea este semnalată numai după 2 erori consecutive (anterior era suficientă una);
- dacă serverul nu este Linux și `boot_id` nu poate fi citit, monitorul rămâne dezactivat, iar bara de stare afișează „Monitor riavvio non disponibile”.

Conexiunea de control este separată de sesiunea principală deoarece, după o repornire rapidă, SSH.NET poate considera încă „conectată” o sesiune deja moartă.

6.2 Când conexiunea se întrerupe

În terminal apare un rând galben cu motivul și se deschide o fereastră de alegere:

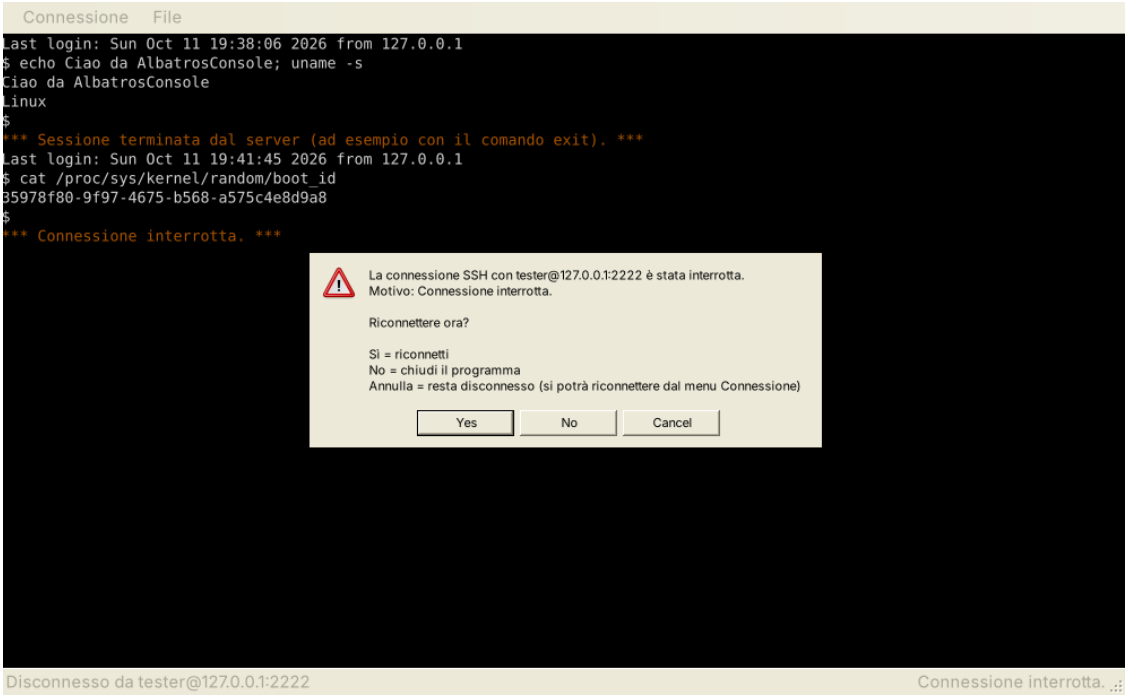

Figura 5 – Conexiune întreruptă: alegere între reconectare, ieșire sau rămânerea deconectat (aspect orientativ)

Răspuns	Efect
Da	Reconectare imediată cu aceleași acreditări. Dacă serverul încă nu răspunde (de exemplu, este în curs de repornire), se deschide fereastra cu motivul: așteptați câteva secunde și apăsați Connetti.
Nu	Închide programul (comportamentul versiunii anterioare).
Anulare	Rămâneți deconectat, cu conținutul terminalului vizibil; vă reconectați din meniul Connessione.

Motivele posibile sunt: server repornit; sesiune încheiată de server (de exemplu, cu comanda `exit`); conexiune întreruptă (rețea, server oprit, serviciu SSH oprit).

Butoanele casetelor de mesaj apar în limba Windows (pe un Windows în italiană: Sì, No și Annulla; în română: Da, Nu și Anulare); figurile din acest manual au fost realizate într-un mediu de test și afișează denumirile în engleză.

7. Fișiere de configurare

Programul își salvează datele în folderul `%APPDATA%\AlbatrosConsole` al utilizatorului Windows (de obicei `C:\Users\<utente>\AppData\Roaming\AlbatrosConsole`).

Fișier	Conținut	Dacă este șters
`connessione.xml`	Ultima conexiune reușită: serverul, portul, utilizatorul, tipul de autentificare, fișierul cheii, opțiunea de memorare și, dacă s-a solicitat, fraza de acces sau parola criptată cu DPAPI.	La următoarea pornire sunt propuse valorile presetate.
`known_hosts.txt`	Un rând pentru fiecare server: `[host]:porta tipo-chiave SHA256:impronta`.	La următoarea conectare, cheia gazdei este solicitată din nou.

Exemplu de rând din `known_hosts.txt`:

```
[vps.example.org]:22 ssh-ed25519 SHA256:JV8PLwsLFrPa7LXzWqotri90MisUw1fbOEDZEvacciw
```

8. Securitate

8.1 Fraza de acces implicită din cod

Pentru a respecta cerința de a avea conexiunea presetată deja pregătită, fraza de acces a cheii implicite rămâne scrisă în codul sursă (constanta `KeyPassphrase` din `ConnectionStore.vb`), exact cum în versiunea anterioară era scrisă în `Form1.vb`. În consecință, poate fi citită și în executabilul compilat, cu orice decompilator .NET.

Dacă executabilul trebuie să părăsească propriul PC, se recomandă:

1. să goliți constanta `KeyPassphrase` (lăsând `""`) și să recompilați;
2. să vă conectați o dată introducând manual fraza de acces, cu caseta **Memorizza** bifată: din acel moment rămâne salvată criptată cu DPAPI și nu mai apare în program.

Fișierul cheii private, în schimb, rămâne pe disc: cine deține o copie a lui și cunoaște fraza de acces poate accesa serverul. Păstrați-l într-un folder cu permisiuni limitate la utilizator.

8.2 Ce protejează DPAPI

DPAPI leagă datele de contul Windows: un alt utilizator al aceluiași PC sau un alt computer nu le pot decripta. În schimb, nu protejează împotriva unui program executat cu același cont al utilizatorului, care le-ar putea decripta așa cum face Albatros Console.

8.3 Cheile gazdelor

Verificarea cheii gazdei protejează conexiunea numai dacă amprenta este într-adevăr comparată cu cea a serverului la prima conectare (secțiunea 4.1). Un avertisment de cheie schimbată nu trebuie acceptat niciodată fără a-i cunoaște motivul.

9. Depanare

Mesaj	Cauză probabilă	Ce trebuie făcut
Il file della chiave privata non esiste (Fișierul cheii private nu există)	Cale greșită sau fișier mutat.	Corectați calea sau folosiți Sfoglia...
La chiave privata è protetta: inserire la passphrase (Cheia privată este protejată: introduceți fraza de acces)	Cheie criptată și câmpul frazei de acces gol.	Introduceți fraza de acces.
Impossibile leggere la chiave privata: passphrase errata oppure file non valido (Cheia privată nu poate fi citită: frază de acces greșită sau fișier nevalid)	Frază de acces greșită sau fișier care nu este o cheie privată.	Tastați din nou fraza de acces (folosiți Mostra pentru a o verifica) sau alegeți fișierul corect.
Server non trovato (Server negăsit)	Nume de gazdă greșit sau DNS inaccesibil.	Verificați numele sau folosiți adresa IP.
Connessione rifiutata ... il servizio SSH non è in ascolto (Conexiune refuzată ... serviciul SSH nu ascultă)	Port greșit sau sshd oprit.	Verificați portul; pe server: `sudo systemctl status ssh`.
Il server ... non è raggiungibile / Tempo scaduto (Serverul ... nu poate fi accesat / Timp expirat)	Server oprit sau în curs de repornire, firewall, rețea absentă.	Așteptați și reîncercați; verificați firewallul și conexiunea.
Autenticazione rifiutata dal server (Autentificare refuzată de server)	Cheie publică neautorizată, utilizator sau parolă greșite.	Verificați `~/.ssh/authorized_keys` pe server, utilizatorul, parola.
Connessione annullata: la chiave host ... non è stata accettata (Conexiune anulată: cheia gazdei ... nu a fost acceptată)	S-a răspuns Nu la verificarea cheii gazdei.	Verificați amprenta (4.1) și reîncercați.
„Host key could not be verified” în explorator sau în transfer	Cheia gazdei s-a schimbat după deschiderea sesiunii.	Reconectați-vă din meniul Connessione și confirmați noua cheie.
Monitor riavvio non disponibile (Monitor de repornire indisponibil)	Serverul nu expune `/proc/sys/kernel/random/boot_id` (nu este Linux).	Nicio acțiune: terminalul funcționează normal, fără detectarea repornirii.

10. Note pentru dezvoltator

10.1 Structură

Clasă / modul	Rol
`Form1`	Fereastra principală: meniu, bară de stare, conectare asincronă, citirea shell-ului, monitor, gestionarea deconectărilor.
`FormLogin`	Fereastra de acreditări; returnează un `SshConnectionProfile` deja verificat și `ConnectionSettings` de salvat.
`SshConnectionProfile`	Parametri de conectare imuabili. `CreateSshClient` și `CreateSftpClient` returnează clienți încă neconectați, cu verificarea cheii gazdei deja atașată. Cheia decriptată este păstrată în cache pentru fiecare fir de execuție (`ThreadLocal`), deoarece instanțele de cheie SSH.NET nu sunt garantat thread-safe, iar transferurile paralele rulează pe fire diferite.
`ConnectionSettings`, `ConnectionStore`, `DefaultConnection`	Valorile ferestrei, salvarea în `connessione.xml` cu DPAPI, valorile presetate.
`HostKeyStore`	Citirea și scrierea fișierului `known_hosts.txt`; `Attach` leagă verificarea de evenimentul `HostKeyReceived`.
`FormTransfer`, `RemoteBrowserDialog`	Logică neschimbată; constructorul primește acum un `SshConnectionProfile` în locul celor cinci parametri host, port, utilizator, cheie și frază de acces.
`TerminalControl`, `TerminalBuffer`	Adăugate tastele funcționale, PgUp/PgDn, modul DECCKM, derularea cu Shift+PgUp/PgDn, fontul aldin în cache și desenarea mai ușoară a fundalului.

10.2 Generații de conexiune

Fiecare conexiune primește un număr de „generație”. Firul de citire a shell-ului și temporizatorul monitorului reține generația cu care au pornit; când utilizatorul se deconectează sau deschide o nouă conexiune, generația se schimbă, iar eventualele lor semnalări întârziate sunt ignorate. Astfel, închiderea voluntară a unei sesiuni nu provoacă avertismente false de deconectare.

10.3 Parametri modificabili

Constantă (în Form1.vb)	Valoare	Semnificație
`ConnectTimeoutSeconds`	15	Timpul maxim pentru conexiunea principală.
`MonitorIntervalSeconds`	3	Intervalul dintre două verificări ale monitorului.
`MonitorTimeoutSeconds`	4	Timpul maxim al fiecărei verificări.
`MonitorMaxFailures`	2	Erori de rețea consecutive înainte de a declara conexiunea pierdută.

10.4 Verificări efectuate

Compilare fără erori sau avertismente pentru .NET Framework 4.8. Pe un server OpenSSH de test au fost verificate: cheie cu frază de acces corectă, greșită și lipsă; acces cu cheie și cu parolă; cheie de gazdă nouă, memorată și schimbată; SFTP inclusiv cu șase conexiuni paralele; acreditări criptate; ciclul fereastră → cheia gazdei → terminal → deconectare → reconectare; detectarea serverului oprit de către monitor.

Albatros Console

Windows 用 SSH および SFTP ターミナル

ユーザー マニュアル

資格情報画面、ホスト キーの検証、再接続に対応したバージョン

2026 年 10 月

目次

1. はじめに 63

1.1 このバージョンの新機能 63

2. 要件とビルド 64

2.1 要件 64

2.2 プロジェクトを開いてビルドする 64

3. 資格情報画面 65

3.1 フィールド 65

3.2 既定の接続 66

3.3 パスフレーズまたはパスワードの保存 66

3.4 接続前のチェック 66

4. ホスト キーの検証 67

4.1 初回接続 67

4.2 キーが変更された場合 67

5. ターミナル ウィンドウ 68

5.1 Connessione メニュー 68

5.2 File メニュー 68

5.3 ステータス バー 69

5.4 キーボードとマウス 69

6. 再起動モニターと切断 70

6.1 モニターのしくみ 70

6.2 接続が中断したとき 70

7. 構成ファイル 72

8. セキュリティ 73

8.1 コード内の既定のパスフレーズ 73

8.2 DPAPI が保護するもの 73

8.3 ホスト キー 73

9. トラブルシューティング 74

10. 開発者向けの注意事項 75

10.1 構造 75

10.2 接続の世代 75

10.3 変更可能なパラメーター 75

10.4 実施した検証 76

1. はじめに

Albatros Console は、VB.NET (.NET Framework 4.8、SSH.NET ライブラリ) で作成された Windows 用の SSH ターミナルです。xterm スタイルの対話型シェルに加えて、リモート ファイルの参照と SFTP 転送、およびサーバーの再起動を検知するモニターを備えています。

このマニュアルでは、**起動時の資格情報画面**を備えたバージョンについて説明します。サーバー、ユーザー、キー、パスフレーズがコードに記述され、接続が自動的に開始されていた以前のバージョンとは異なり、起動時に既定の接続を提示し、それを変更できるウィンドウが表示されるようになりました。

1.1 このバージョンの新機能

- **起動時の資格情報画面**: 既定の接続の値があらかじめ入力されており、**Predefiniti** ボタンで既定値に戻すことができます。
- **秘密キー** (PuTTY の PPK または OpenSSH 形式、パスフレーズの有無を問わない) または**パスワード**による認証。
- **パスフレーズまたはパスワードの保存 (省略可能)**: 現在のユーザーに対して Windows の DPAPI で暗号化されます。
- **サーバーのホスト キーの検証**: ssh コマンドと同様に、初回接続時に確認を求め、以降はキーが異なる場合に警告します。
- バックグラウンドでの接続: サーバーの応答待ちの間もウィンドウが固まらず、プログラムを終了する代わりに**エラーの説明**をイタリア語で表示します。
- **Connessione** メニュー (Nuova connessione、Riconnetti、Disconnetti)。中断後、プログラムは終了する代わりに**再接続**を提案します。
- **ステータス バー**: 接続中のサーバーと再起動モニターの状態を表示します。
- より堅牢な再起動モニター: 1 回のネットワーク エラーだけでは接続が失われたと判断しなくなりました。

- 秘密キーは**一度だけ復号**され、ターミナル、モニター、エクスプローラー、転送で再利用されます (以前は SFTP 操作やチェックのたびに読み直しており、PPK v3 キーでは時間がかかっていました)。
- ターミナル: ファンクション キー F1 〜 F12、PageUp/PageDown、Insert、Shift+Tab、アプリケーション カーソル モード (vim、less、htop)、履歴をスクロールする **Shift+PageUp/PageDown**。メニューの下に隠れていた最初の行を修正しました。

2. 要件とビルド

2.1 要件

- .NET Framework 4.8 がインストールされた Windows 10/11 または Windows Server 2016 以降。
- ビルドするには: .NET デスクトップ開発ワークロードを含む Visual Studio 2019 または 2022。
- SSH が有効なリモート サーバー。再起動モニターには Linux サーバーが必要です (`/proc/sys/kernel/random/boot_id` を読み取ります)。

2.2 プロジェクトを開いてビルドする

1. zip を展開し、Visual Studio で `AlbatrosConsole.sln` を開きます。
2. `packages` フォルダーには NuGet パッケージ (SSH.NET 2026.0.0 と依存関係) が既に含まれています。Visual Studio がパッケージの不足を報告する場合は、[NuGet パッケージの復元] を使用します。
3. [ビルド] → [ソリューションのリビルド] でビルドし、F5 キーで起動します。

以前のバージョンと比べて、プロジェクトには 4 つの新しいファイルと 1 つの追加参照が含まれています。これらは既に `AlbatrosConsole.vbproj` ファイルに追加されています:

要素	内容
`FormLogin.vb`	資格情報画面。
`SshConnectionProfile.vb`	接続パラメーター、復号済みキーのキャッシュ、SSH/SFTP クライアントの作成。
`ConnectionStore.vb`	既定の接続の値 (`DefaultConnection` モジュール) と最後の接続の保存。
`HostKeyStore.vb`	信頼済みホスト キーのストア (known_hosts に相当)。
参照 `System.Security`	DPAPI による暗号化 (`ProtectedData`) に必要です。

3. 資格情報画面

起動時に、プログラムは **Albatros Console** – **Connessione** ウィンドウを開きます。接続が成功するまで、ターミナルは画面の背後で待機します。

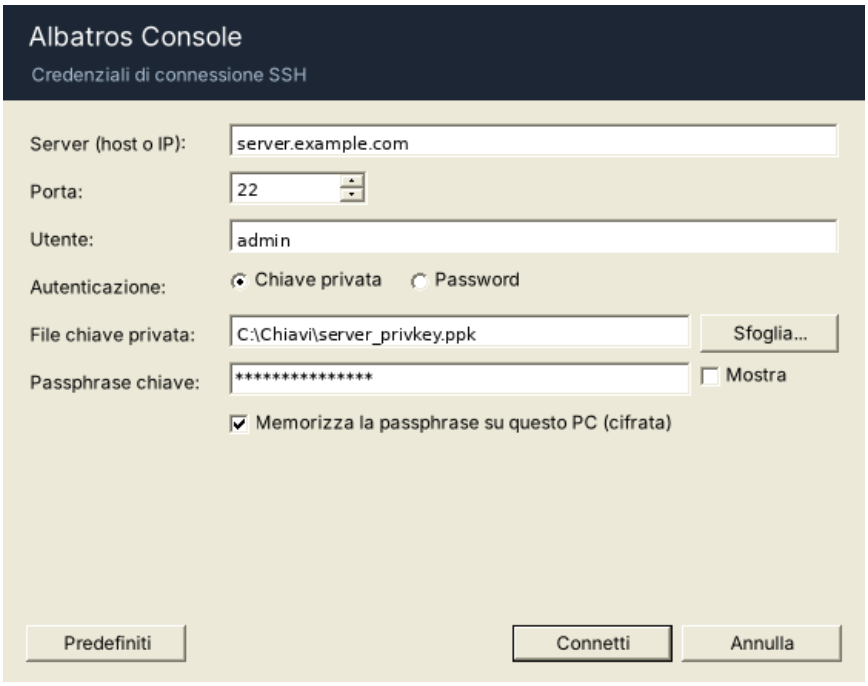

図 1 – 既定の接続が表示された資格情報画面 (外観はイメージです)

3.1 フィールド

フィールド	意味
Server (host o IP)	SSH サーバーの DNS 名または IP アドレス。
Porta	SSH サービスの TCP ポート (1 ～ 65535)。標準は 22 です。
Utente	リモート ユーザーの名前。
Autenticazione	**Chiave privata** または **Password**。Password モードでは、キー ファイルのフィールドは無効になります。
File chiave privata	キー ファイルのパス: PuTTY の PPK (v2 または v3) または OpenSSH キー。**Sfoglia...** ボタンでファイルの選択を開きます。
Passphrase chiave / Password	キーを保護するパスフレーズ、またはユーザーのパスワード。キーが保護されていない場合は空のままにします。**Mostra** で入力したテキストを表示します。

フィールド	意味
Memorizza ... su questo PC (cifrata)	オンにすると、パスフレーズまたはパスワードが現在の Windows ユーザー用に暗号化されて保存されます (3.3 を参照)。

ボタン: **Connetti** (Enter キーでも可) で接続を開始します。**Annulla** (Esc キーでも可) は、起動時にはプログラムを終了し、メニューから開いた場合は画面だけを閉じて現在のセッションはそのままにします。**Predefiniti** は既定の接続に戻します。

3.2 既定の接続

初回起動時、および保存された最後の接続がない場合は常に、ビルド時に設定された既定の接続が画面に提示されます。サーバー、ポート、ユーザー、秘密キー ファイル、パスフレーズ (ある場合) が入力済みなので、実際には Connetti をクリックするだけです。

接続が成功するたびに使用した値が保存され、次回の起動時に提示されます。**Predefiniti** ボタンを使うと、パスフレーズを含め、いつでも既定値に戻すことができます。

既定値は `ConnectionStore.vb` の先頭にある `DefaultConnection` モジュールにあります。変更するには、定数を編集して再コンパイルするだけです。

3.3 パスフレーズまたはパスワードの保存

Memorizza チェック ボックスをオンにすると、パスフレーズまたはパスワードは Windows の DPAPI (現在のユーザーのスコープ) で暗号化されて `connessione.xml` ファイルに書き込まれます。復号できるのは、同じ PC 上の同じ Windows ユーザーだけです。ファイルを別のコンピューターにコピーしたり、別のアカウントで読み取ったりすると、フィールドは空になり、再入力が必要になります。

チェックを外すと値は保存されず、次回の起動時にはフィールドが空になります。ただし、同じセッション中は再接続のためにプログラムが値を記憶しています。

3.4 接続前のチェック

Connetti をクリックすると、画面は必須フィールドを確認し、キー認証の場合はすぐにキーの読み取りと復号を試みます。これにより、存在しないファイルや間違ったパスフレーズが、サーバーを待たずにフィールドの下に赤で表示されます。カーソルは修正が必要なフィールドに移動します。

サーバーへの接続に失敗した場合 (サーバーの電源がオフ、ポートの誤り、認証の拒否、ホスト キーが受け入れられない)、エラーの理由と入力済みのデータとともに画面が再び開き、再試行できる状態になります。



図 2 – 接続エラー後に再表示された画面

4. ホスト キーの検証

各 SSH サーバーには固有のホスト キーがあります。これを確認することで、仲介者 (中間者攻撃) ではなく、本当に自分のサーバーと通信していることが保証されます。以前のバージョンはどのキーでも受け入れていましたが、現在の Albatros Console は ssh コマンドと同じように動作します。

4.1 初回接続

サーバー (または同じサーバーの新しいポート) に初めて接続するときは、キーの種類とその SHA256 フィンガープリントが表示された確認の要求が表示されます。

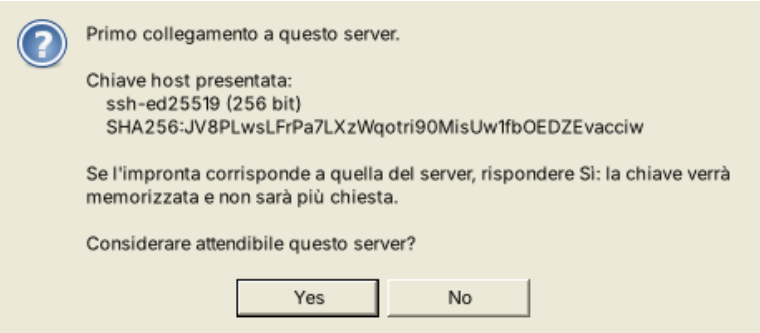

図 3 – 初回接続時のホスト キーの確認 (外観はイメージです)

フィンガープリントを確認するには、サーバー上で、既に信頼できるセッション (サーバーのコンソール、または構成済みの PuTTY) から次を実行します:

```
for f in /etc/ssh/ssh_host_*_key.pub; do ssh-keygen -lf "$f"; done
```

いずれかの行に、プログラムが表示したものと同じ `SHA256:...` フィンガープリントが含まれている必要があります。**はい** と答えるとキーが保存され、以降は確認されません。**いいえ** と答えると接続は取り消されます。

> **注:** このバージョンに更新した後は、信頼済みキーの一覧が空の状態から始まるため、既定のサーバーについても一度だけ確認が求められます。

4.2 キーが変更された場合

サーバーが保存済みのキーと異なるキーを提示した場合、新しいフィンガープリントとともに警告が表示されます。既定のボタンは **いいえ** です。オペレーティング システムの再インストールや SSH キーの再生成の後であれば、変更は正常です。それ以外の場合は、続行せずにまず原因を確認することをお勧めします。**はい** と答えると、新しいキーが保存済みのキーを置き換えます。

ファイル エクスプローラー、SFTP 転送、再起動モニターは、既に保存されているキーだけを受け入れ、確認は行いません。確認は常にメインの接続で行われます。

> **注意:** サーバーのキーを削除するには、`known_hosts.txt` ファイル内の該当する行を削除するだけです (第 7 章を参照)。

5. ターミナル ウィンドウ

接続が成功すると、ウィンドウのタイトルに `utente@host:porta` が表示され、ターミナルにサーバーのプロンプトが表示されます。



図 4 – ステータス バー付きの接続済みターミナル (外観はイメージです)

5.1 Connessione メニュー

項目	機能
Nuova connessione...	資格情報画面を再び開きます。確定すると現在のセッションが閉じられて新しいセッションが開き、キャンセルすると現在のセッションはそのまま残ります。
Riconnetti	画面を開かずに、同じ資格情報で最後のサーバーに再接続します。失敗した場合は、理由とともに画面を開きます。
Disconnetti	セッションを閉じます。ウィンドウはターミナルの内容を表示したまま開いており、メニューから再接続できます。
Esci	プログラムを終了します。

5.2 File メニュー

項目	機能
Trasferimento file SFTP...	ダウンロード、アップロード、サーバー間のコピー。複数の転送を並行して実行できます。
Esplora file remoto...	リモート フォルダーの参照、ファイルとフォルダーの作成と削除、Unix のアクセス許可、Notepad++ またはメモ帳での直接オープン、ダウンロード。

どちらのウィンドウも現在の接続の資格情報を使用し、接続中にのみ使用できます。動作は以前のバージョンから変わっていません。

5.3 ステータス バー

左側には接続の状態 (Connessione in corso、Connesso a ...、Disconnesso da ...、Non connesso) が、右側には再起動モニターの状態 (有効、使用不可) または最後の切断の理由が表示されます。

5.4 キーボードとマウス

キーまたは操作	効果
Ctrl+C	選択範囲をコピーします。選択範囲がない場合は、リモート プログラムに割り込み (SIGINT) を送信します。
Ctrl+V、選択範囲がないときの右クリック	クリップボードのテキストを貼り付けます。
選択範囲があるときの右クリック	選択範囲をコピーします。
左ボタンでドラッグ	テキストを選択します (履歴内でも可能)。
マウス ホイール	履歴 (5000 行) をスクロールします。右側の黄色のバーは、末尾にいないことを示します。
Shift+PageUp / Shift+PageDown	履歴を 1 ページ分スクロールします (新機能)。

キーまたは操作	効果
方向キー、Home、End	サーバーに送信されます。全画面のプログラム (vim、less、htop) では、自動的にアプリケーション カーソル モードが使用されます。
PageUp、PageDown、Insert、Delete	サーバーに送信されます (PageUp、PageDown、Insert は新機能)。
F1 ～ F12	xterm と同様にサーバーに送信されます (新機能)。F10 でウィンドウのメニューが開かなくなりました。たとえば mc や htop を終了するのに使用します。
Shift+Tab	サーバーに送信されます (逆タブ)。
Ctrl+D	入力の終了。シェルでは exit と同じです。
その他のキー	表示を末尾のプロンプトに戻します。

6. 再起動モニターと切断

6.1 モニターのしくみ

接続するとすぐに、プログラムはサーバーの `boot_id` を読み取ります。これは Linux が起動のたびに変更する識別子です。それ以降、3 秒ごとに別個の短いチェック用接続を開いて、これを読み直します:

- `boot_id` が変わっていれば、サーバーが再起動されたので、ただちに切断が通知されます。
- ネットワークの問題でチェックが失敗した場合、切断が通知されるのは 2 回連続でエラーになった後だけです (以前は 1 回で十分でした)。
- サーバーが Linux ではなく `boot_id` を読み取れない場合、モニターは無効のままとなり、ステータス バーに「Monitor riavvio non disponibile」(再起動モニターは使用できません) と表示されます。

チェック用の接続をメイン セッションと分けているのは、短時間で再起動した後、SSH.NET が既に切れたセッションをまだ「接続済み」と見なすことがあるためです。

6.2 接続が中断したとき

ターミナルに理由を示す黄色の行が表示され、選択用のウィンドウが開きます:

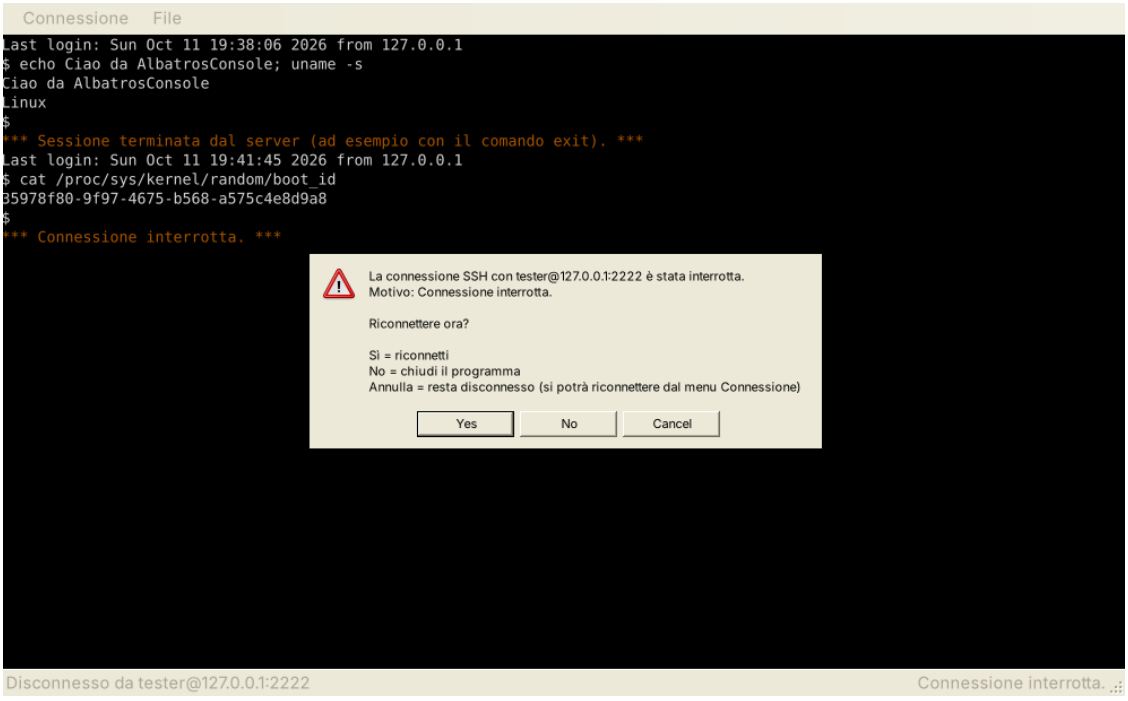

図 5 – 接続の中断: 再接続、終了、切断状態のままにするかの選択 (外観はイメージです)

応答	効果
はい	同じ資格情報ですぐに再接続します。サーバーがまだ応答しない場合 (たとえば再起動中)、理由とともに画面が開きます。数秒待ってから Connetti をクリックしてください。
いいえ	プログラムを終了します (以前のバージョンの動作)。
キャンセル	ターミナルの内容を表示したまま切断状態にとどまります。Connessione メニューから再接続できます。

考えられる理由: サーバーの再起動、サーバーによるセッションの終了 (たとえば `exit` コマンド)、接続の中断 (ネットワーク、サーバーの電源オフ、SSH サービスの停止)。

日本語版の Windows では、メッセージ ボックスのボタンは「はい」「いいえ」「キャンセル」と表示されます。このマニュアルの図はテスト環境で取得したもので、英語の表記になっています。

7. 構成ファイル

プログラムは、Windows ユーザーの `%APPDATA%\AlbatrosConsole` フォルダー (通常は `C:\Users\<utente>\AppData\Roaming\AlbatrosConsole`) にデータを保存します。

ファイル	内容	削除した場合
`connessione.xml`	最後に成功した接続: サーバー、ポート、ユーザー、認証の種類、キー ファイル、保存の選択、および要求された場合は DPAPI で暗号化されたパスフレーズまたはパスワード。	次回の起動時に既定値が提示されます。
`known_hosts.txt`	サーバーごとに 1 行: `[host]:porta tipo-chiave SHA256:impronta`。	次回の接続時に、ホストキーの確認が再び求められます。

`known_hosts.txt` の行の例:

```
[vps.example.org]:22 ssh-ed25519 SHA256:JV8PLwsLFrPa7LXzWqotri90MisUw1fbOEDZEvacciw
```

8. セキュリティ

8.1 コード内の既定のパスフレーズ

既定の接続をすぐに使える状態にするという要件を満たすため、既定のキーのパスフレーズはソースに記述されたままです (`ConnectionStore.vb` 内の定数 `KeyPassphrase`)。以前のバージョンで `Form1.vb` に記述されていたのとまったく同じです。そのため、コンパイル済みの実行可能ファイルからも、任意の .NET 逆コンパイラーで読み取ることができます。

実行可能ファイルを自分の PC の外に出す場合は、次のことをお勧めします:

1. 定数 `KeyPassphrase` を空にして (`""` のままにして) 再コンパイルする。
2. **Memorizza** チェック ボックスをオンにして、パスフレーズを手動で入力して一度接続する。それ以降、パスフレーズは DPAPI で暗号化されて保存され、プログラム内に現れることはなくなります。

一方、秘密キー ファイルはディスク上に残ります。そのコピーを持ち、パスフレーズを知っている人はサーバーにアクセスできます。アクセス許可をユーザーに限定したフォルダーに保管してください。

8.2 DPAPI が保護するもの

DPAPI はデータを Windows アカウントに結び付けます。同じ PC の別のユーザーや別のコンピューターでは復号できません。ただし、ユーザーと同じアカウントで実行されるプログラムからは保護されず、そのようなプログラムは Albatros Console と同様に復号できてしまいます。

8.3 ホスト キー

ホスト キーの検証が接続を保護するのは、初回接続時にフィンガープリントを実際にサーバーのものと照合した場合だけです (4.1 節)。キー変更の警告は、理由がわからないまま受け入れてはいけません。

9. トラブルシューティング

メッセージ	考えられる原因	対処方法
Il file della chiave privata non esiste	パスが間違っているか、ファイルが移動されています。	パスを修正するか、Sfoglia... を使用します。
La chiave privata è protetta: inserire la passphrase	キーが暗号化されていて、パスフレーズのフィールドが空です。	パスフレーズを入力します。
Impossibile leggere la chiave privata: passphrase errata oppure file non valido	パスフレーズが間違っているか、ファイルが秘密キーではありません。	パスフレーズを入力し直す (Mostra で確認します) か、正しいファイルを選択します。
Server non trovato	ホスト名が間違っているか、DNS に到達できません。	名前を確認するか、IP アドレスを使用します。
Connessione rifiutata ... il servizio SSH non è in ascolto	ポートが間違っているか、sshd が停止しています。	ポートを確認します。サーバー上で: `sudo systemctl status ssh`。
Il server ... non è raggiungibile / Tempo scaduto	サーバーの電源がオフか再起動中、ファイアウォール、ネットワークなし。	しばらく待ってから再試行します。ファイアウォールと接続を確認します。
Autenticazione rifiutata dal server	公開キーが承認されていないか、ユーザーまたはパスワードが間違っています。	サーバー上の `~/.ssh/authorized_keys`、ユーザー、パスワードを確認します。
Connessione annullata: la chiave host ... non è stata accettata	ホスト キーの検証で「いいえ」と答えました。	フィンガープリントを確認して (4.1)、再試行します。
エクスプローラーまたは転送での「Host key could not be verified」	セッションを開いた後にホストキーが変更されました。	Connessione メニューから再接続し、新しいキーを確認します。
Monitor riavvio non disponibile	サーバーが `/proc/sys/kernel/random/boot_id` を公開していません (Linux ではありません)。	対処は不要です。ターミナルは再起動の検出なしで通常どおり動作します。

10. 開発者向けの注意事項

10.1 構造

クラス / モジュール	役割
`Form1`	メイン ウィンドウ: メニュー、ステータス バー、非同期接続、シェルの読み取り、モニター、切断の処理。
`FormLogin`	資格情報画面。検証済みの `SshConnectionProfile` と、保存する `ConnectionSettings` を返します。
`SshConnectionProfile`	不変の接続パラメーター。`CreateSshClient` と `CreateSftpClient` は、ホスト キーのチェックを既に組み込んだ、未接続のクライアントを返します。SSH.NET のキー インスタンスはスレッド セーフであることが保証されておらず、並列転送は別々のスレッドで実行されるため、復号済みキーはスレッドごとにキャッシュされます (`ThreadLocal`)。
`ConnectionSettings`、`ConnectionStore`、`DefaultConnection`	画面の値、DPAPI を使用した `connessione.xml` への保存、既定値。
`HostKeyStore`	`known_hosts.txt` の読み書き。`Attach` は検証を `HostKeyReceived` イベントに接続します。
`FormTransfer`、`RemoteBrowserDialog`	ロジックは変更なし。コンストラクターは、ホスト、ポート、ユーザー、キー、パスフレーズの 5 つのパラメーターの代わりに `SshConnectionProfile` を受け取るようになりました。
`TerminalControl`、`TerminalBuffer`	ファンクション キー、PageUp/PageDown、DECCKM モード、Shift+PageUp/PageDown によるスクロール、太字フォントのキャッシュ、背景の軽量な描画を追加しました。

10.2 接続の世代

各接続には「世代」番号が付与されます。シェルの読み取りスレッドとモニターのタイマーは、開始時の世代を記憶しています。ユーザーが切断するか新しい接続を開くと世代が変わり、それらからの

遅れて届いた通知は無視されます。これにより、意図的にセッションを閉じたときに誤った切断警告が出ることはありません。

10.3 変更可能なパラメーター

定数 (Form1.vb 内)	値	意味
`ConnectTimeoutSeconds`	15	メイン接続の最大時間。
`MonitorIntervalSeconds`	3	モニターの 2 回のチェックの間隔。
`MonitorTimeoutSeconds`	4	各チェックの最大時間。
`MonitorMaxFailures`	2	接続が失われたと判断するまでの連続したネットワーク エラーの回数。

10.4 実施した検証

.NET Framework 4.8 を対象に、エラーも警告もなくビルドできることを確認しました。テスト用の OpenSSH サーバーで次の項目を検証しました: 正しいパスフレーズ、間違ったパスフレーズ、パスフレーズなしのキー、キーおよびパスワードによるアクセス、新規・保存済み・変更されたホスト キー、6 本の並列接続を含む SFTP、暗号化された資格情報、画面 → ホスト キー → ターミナル → 切断 → 再接続のサイクル、モニターによるサーバー停止の検出。