Introduction

OrbitHub is a native desktop application for organizing connection profiles and launching SSH, RDP, and VNC sessions from one place, in a single tabbed window. It runs on Windows, Linux, and macOS.

This guide covers everyday use: creating and organizing profiles, connecting over SSH, RDP, and VNC, managing active sessions, and what to do when something goes wrong. It does not cover installation or building from source — see the project's `README.md` and `docs/BUILDING.md` for that.

VNC support covers standard VNC Authentication and no-auth servers (TigerVNC, x11vnc, TightVNC, and similar). macOS's built-in Screen Sharing server uses two undocumented, reverse-engineered authentication schemes that are implemented but not yet confirmed working end to end against a real macOS server — see Connecting via VNC.

Getting Started

When OrbitHub opens, you land on the **Profiles** tab — a searchable, sortable list of every connection you've saved. Two toolbar controls in the top-right change how that list is presented:

- **View**: `List` shows every profile in one flat table; `Folders` groups them into the folder tree you've organized them into.
- **Sort**: orders the list by Name, Protocol, or Host.

Along the top of the window, a **Search** box filters by name, host, folder path, or tag as you type, and **Protocol**/**Tag** dropdowns narrow the list further. These filters, plus your last view mode and sort order, are remembered the next time you open OrbitHub.

Along the bottom of the Profiles tab: **New**, **Edit**, and **Delete** buttons for managing the selected profile. The same actions are available by right-clicking a profile or folder, and from the **File** menu (`New Profile`, `New Folder`).

To connect, double-click a profile, or select it and press the connect control — this opens a new tab for that session and connects automatically.

Managing Profiles

A profile stores everything needed to reach one remote host. Open **New** (or **Edit** on an existing profile) to fill in:

- **Name** — a label for the profile; shown on its tab and in the list.
- **Host** — hostname or IP address.
- **Port** — defaults to 22 for SSH, 3389 for RDP, `5900` for VNC.
- **Username** — the account to log in as. Optional for every protocol — leave it blank and OrbitHub asks for one inline the first time you connect that profile, the same way it already asks for a password. (VNC servers mostly ignore username entirely; macOS Screen Sharing is the exception and requires one — see Connecting via VNC.)
- **Domain** — Windows domain for RDP logins (leave blank for local/ workgroup accounts, SSH, or VNC profiles).

- **Tags** — free-form, comma-separated labels for filtering and grouping (e.g. `prod, linux, db`).
- **Folder** — where the profile lives in the Folders view.
- **Protocol** — SSH, RDP, or VNC.

Fields below Protocol change based on what you pick:

SSH: **Auth Mode** (`Password` or `Private Key`), and if Private Key, a **Private Key** file path with a **Browse** button, plus **Known Hosts** policy — see Connecting via SSH for what each policy means.

RDP: **RDP Security** and **RDP Performance** — see Connecting via RDP.

VNC has no protocol-specific fields beyond Host/Port/Username above — see Connecting via VNC.

Passwords are never saved in the profile — OrbitHub asks for them each time you connect (unless you've set up private-key SSH auth, which needs no password to be entered per-connection if the key itself has none).

Organizing Profiles

As your profile list grows, two independent tools keep it manageable:

Folders. Switch the Profiles tab to `Folders` view to see profiles grouped into a tree. Create a folder from the **File** menu or by right-clicking in the tree (`New Folder`), and drag any profile onto a folder to move it there. Folders can nest inside other folders.

Tags. Tags are independent of folders — a profile can be in one folder but carry several tags (e.g. `prod`, `linux`, `db` all at once). Use the **Tag** filter dropdown in the toolbar to instantly narrow the list to everything sharing a tag, regardless of which folder it's filed under.

Combine both with the **Search** box (matches name, host, folder path, or tags) and the **Sort** control (Name / Protocol / Host) to find what you need quickly even with a large profile list.

Importing and Exporting Profiles

The **File** menu has three options for moving profiles in or out of OrbitHub:

- **Export Profiles...** — saves your entire profile list, including folder structure and tags, to a JSON file. Passwords are never included, since OrbitHub never stores them in the first place.
- **Import Profiles...** — reads that same JSON format back in, recreating folders and profiles. Existing profiles aren't touched; entries missing a name or host are skipped and reported in the summary.
- **Import from mRemoteNG...** — reads an mRemoteNG connections `.xml` file and creates equivalent OrbitHub profiles. Passwords are never imported — you'll be prompted the first time you connect each imported profile, same as a new one. Connections using a protocol OrbitHub doesn't support are skipped and listed in the import summary. This is a one-way conversion; there's no export back to mRemoteNG's format.

Connecting via SSH

Double-clicking an SSH profile opens a new tab with an embedded, fully interactive terminal — type directly into it as you would any terminal emulator. A **theme** selector lets you switch between `Dark`, `Light`, and `Solarized Dark`; your choice is remembered for future sessions.

Authentication. Set in the profile itself:

- **Username** — if left blank in the profile, OrbitHub asks for one inline the first time you connect, right alongside the password prompt.
- **Password** — OrbitHub prompts for a password each time you connect. It is never stored.
- **Private Key** — point at a key file (via the profile's Browse button); no password prompt unless the key itself is passphrase-protected.

Known Hosts policy. This controls how OrbitHub reacts to a server's SSH host key — the mechanism that protects against a different machine silently impersonating a host you've connected to before:

Policy	Behavior
`Ask`	Prompts you to confirm trust the first time a host is seen, and on any later change. Recommended default.
`Accept New`	Silently trusts a host the first time it's seen, but still stops and warns if a previously-trusted host's key later changes.
`Strict`	Never trusts an unknown host automatically — the connection fails until you've manually confirmed the host key some other way.
`Ignore`	Skips host-key checking entirely. Only use this for throwaway/test environments — it removes protection against on-path attacks.

Trusted host keys are recorded in your system's normal SSH `known_hosts` file (the same one the `ssh` command line tool uses), so trust decisions made through OrbitHub or a terminal `ssh` session carry over to each other.

Connecting via RDP

RDP sessions render in an embedded display surface inside the tab — no external RDP client window opens. Keyboard and mouse input go straight to the remote desktop while the tab has focus, and resizing the OrbitHub window renegotiates the remote resolution to match. Clipboard content syncs between your machine and the remote session automatically.

Authentication uses the profile's Username/Domain fields; OrbitHub prompts for the password at connect time, and for a username too if the profile's own Username field was left blank.

RDP Security controls which transport-security layer is used to negotiate the connection:

Mode	Behavior
`Negotiate`	Lets the client and server agree on the strongest mutually-supported option automatically. Recommended default.
`NLA`	Requires Network Level Authentication (credentials verified before a full session starts) — the modern standard for current Windows versions.
`TLS`	Requires TLS-only security, without NLA.

RDP	The legacy RDP-native security layer, for older servers that don't support TLS/NLA.

RDP Performance trades visual fidelity for responsiveness: `Balanced` (default), `Best Quality`, `Best Performance`, or `Auto Detect` (adapts based on the detected connection).

Server certificate verification. The first time you connect to an RDP host, OrbitHub trusts and remembers its TLS certificate — the same trust-on-first-use model SSH uses for host keys. If that certificate ever changes on a later connection, OrbitHub refuses the connection rather than connecting anyway, since a changed certificate can mean either a legitimate server certificate renewal or an active machine-in-the-middle presenting a different one. The event log (see below) shows the specific fingerprints involved. If the change is expected — you rotated the server's certificate yourself — reconnecting after clearing the old entry from FreeRDP's certificate store will trust the new one.

Connecting via VNC

VNC sessions render in an embedded display surface inside the tab, like RDP — no external VNC client window opens.

Authentication. VNC servers don't have a fixed username/password convention the way SSH and RDP do, so OrbitHub always prompts for a password at connect time (leave it blank if the server doesn't require one — no-auth servers exist and are supported). Most VNC servers ignore username entirely; the one common exception is **macOS's built-in Screen Sharing**, which does require one — see the Apple Screen Sharing note below if you're connecting to a Mac.

Display mode. Right-click a VNC session's tab to choose:

- **Scale to Fit** (default) — the whole remote screen is scaled to fit the tab, like RDP.
- **Actual Size (Scrollbars)** — renders at the remote's native pixel size, with scrollbars for panning. Useful when scaling would make small text illegible.

Your choice is remembered across sessions. Unlike RDP, VNC has no way to renegotiate the remote screen's resolution to match your window size — resizing the OrbitHub window changes how much of a scaled-down remote screen you can see, not the remote resolution itself.

Clipboard and cursor. Clipboard content syncs between your machine and the remote session in both directions, and the remote's cursor shape (not just position) is mirrored locally when the server supports it.

Encodings. OrbitHub negotiates whichever of Raw, CopyRect, Hextile, ZRLE, or Tight the server prefers — this is automatic and requires no configuration; more efficient encodings (Tight, ZRLE) simply mean better performance over slower links.

Apple Screen Sharing (macOS). macOS's built-in Screen Sharing server doesn't use a standard, documented VNC authentication method — it uses two undocumented schemes that OrbitHub implements based on reverse-engineering work, but neither is yet confirmed to authenticate successfully against a real macOS server. If you're connecting to a Mac's Screen Sharing and see an authentication failure, this is the likely cause rather than a wrong username or password; check docs/PROGRESS.md (Milestone 6) for the current status, or a

VNC server you install yourself on the Mac (e.g. TigerVNC) as a workaround in the meantime.

Managing Sessions

Every open connection lives in its own tab in the same window, alongside the Profiles tab. A tab's title and a colored state indicator show whether it's connecting, connected, disconnected, or failed. Closing a tab disconnects that session; opening a profile again starts a fresh one.

When OrbitHub needs a username or password from you, the tab shows a highlighted prompt bar with a Connect/Cancel pair — this happens for a blank-username SSH/RDP profile, any RDP or SSH password prompt, and any VNC connection (VNC always asks for a password, even if blank). If that prompt appears on a tab you're not currently viewing, the tab's title gets a `(Needs input)` suffix and its color changes so it's easy to spot among several open sessions.

Each session tab includes a collapsible **event log** beneath the connection surface — a timestamped record of connection state changes, warnings, and errors for that session. Controls above the log let you:

- **Show/Hide Events** — collapse the panel when you don't need it.
- **Filter** — a text box to search event text, and an `All` / `Warnings` / `Errors` severity dropdown to narrow what's shown.
- **Export Events** — save the current session's full event log to a file, useful when reporting a connection problem.
- **Clear Events** — empty the log for that tab.

Settings & Preferences

OrbitHub remembers your preferences across restarts without any separate settings screen — they're saved automatically as you use the app:

- Profile list: search text, view mode (List/Folders), Protocol/Tag filters, and sort order.
- Session tabs: terminal theme choice, whether the events panel is shown or hidden for new tabs, and VNC display mode (Scale to Fit / Actual Size).

Profile data itself (names, hosts, tags, folder structure, and so on) is stored in a local SQLite database — see the README for its exact path on your platform. Passwords are never part of that stored data.

Troubleshooting

"Host could not be resolved" — the hostname in the profile can't be looked up by DNS. Check for typos, or try the host's IP address directly to confirm whether it's a DNS problem or something else.

SSH connection fails immediately, no prompt — double-check the profile's Port (default 22) and that a firewall or network path isn't blocking that port from your machine.

RDP: "Authentication failed. Check username and password." — confirm Username and Domain are correct for the target server; some servers require the domain to be set explicitly even for local accounts.

RDP: "RDP security negotiation failed. Try a different RDP security mode." — the server

doesn't support the security mode selected in the profile. Try `Negotiate` first, or a more specific mode if you know what the server requires.

RDP: connection refused with a certificate-changed message — see Connecting via RDP above; this is expected, protective behavior, not a bug, whenever a previously-trusted server's certificate is replaced.

SSH: connection hangs at "Ask" waiting for host-key trust — check that policy's behavior under Connecting via SSH; switching to `Accept New` avoids the prompt for genuinely new hosts while still protecting against a later key change.

VNC: "Authentication or authorization failure" — for most VNC servers this means a wrong password. If you're connecting to **macOS Screen Sharing** specifically, this is the expected result right now — see the Apple Screen Sharing note under Connecting via VNC.

VNC: connection closes immediately with no prompt — some servers (including macOS Screen Sharing in some configurations) close the connection outright rather than negotiating; confirm the server is actually running and reachable on the profile's port (default `5900`), and that no allowlist on the server side is blocking your account or machine.

If none of this covers what you're seeing, a session's exported event log (see Managing Sessions) is the most useful thing to include when asking for help or filing an issue.

About & Support

OrbitHub is open source under the MIT license. Source code, issue tracking, and releases are hosted at git.darksingularity.org/DarkSingularity/orbithub.

For a list of bundled third-party libraries and their licenses, see **Help → About OrbitHub** inside the app.

To report a bug or request a feature, open an issue at git.darksingularity.org/DarkSingularity/orbithub/issues.