Terminal Services Manager

Administrator Guide

Table of Contents

Building your computer list **4**

Discovering computers in bulk 4

Generate names from a pattern 4

Discover from a source 5

Organizing and connecting 7

Group your servers 7

Connecting under the right account 7

Bulk operations 8

Importing and exporting 9

Export the list 9

Import a list 9

A simple rollout 10

Server administration tools **11**

Power and Remote Desktop 11

Connect over Remote Desktop 11

Restart or shut down 12

Enable or disable Remote Desktop 13

RDS configuration 13

Remote Desktop Services properties 13

Remote Desktop Services licensing 14

Auditing sessions 15

Session history 15

Failed logons 16

User activity 17

User profiles 18

Custom command-line tools 18

Add a tool 18

Read every property with entity details 20

Application log **21**

Reading the log 21

What the levels mean 22

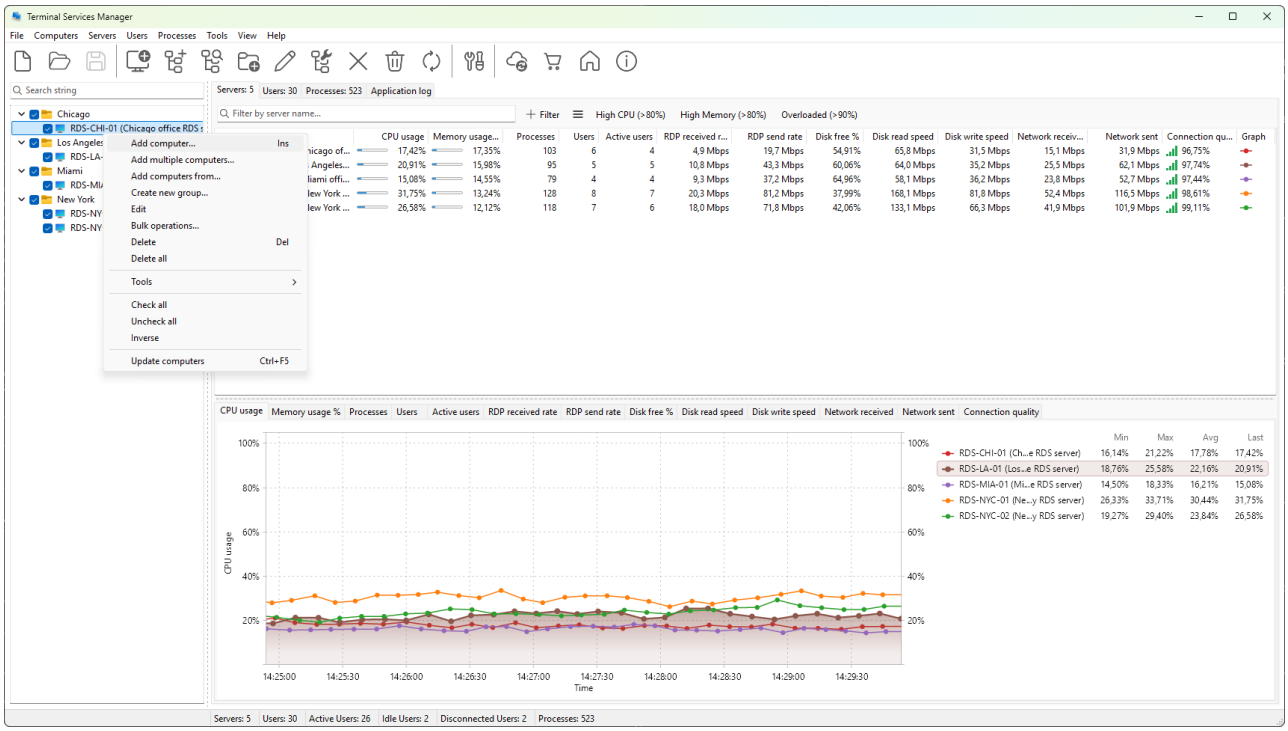
Filtering and exporting 22

Narrow the log 23

Copy and export 23
Updates and licensing **24**
Keeping up to date 24
Check on demand 24
Automatic checks 25
Rolling out to several machines 25
Registering your license 25
Enter your license key 25
If you do not have a key yet 26

Building your computer list

The computer list is the foundation of everything else in Terminal Services Manager. For a few servers you can add them one at a time, but for dozens or hundreds you will want to discover them in bulk, group them so the list stays readable, and reshape the whole list with a single operation when names or structure change.



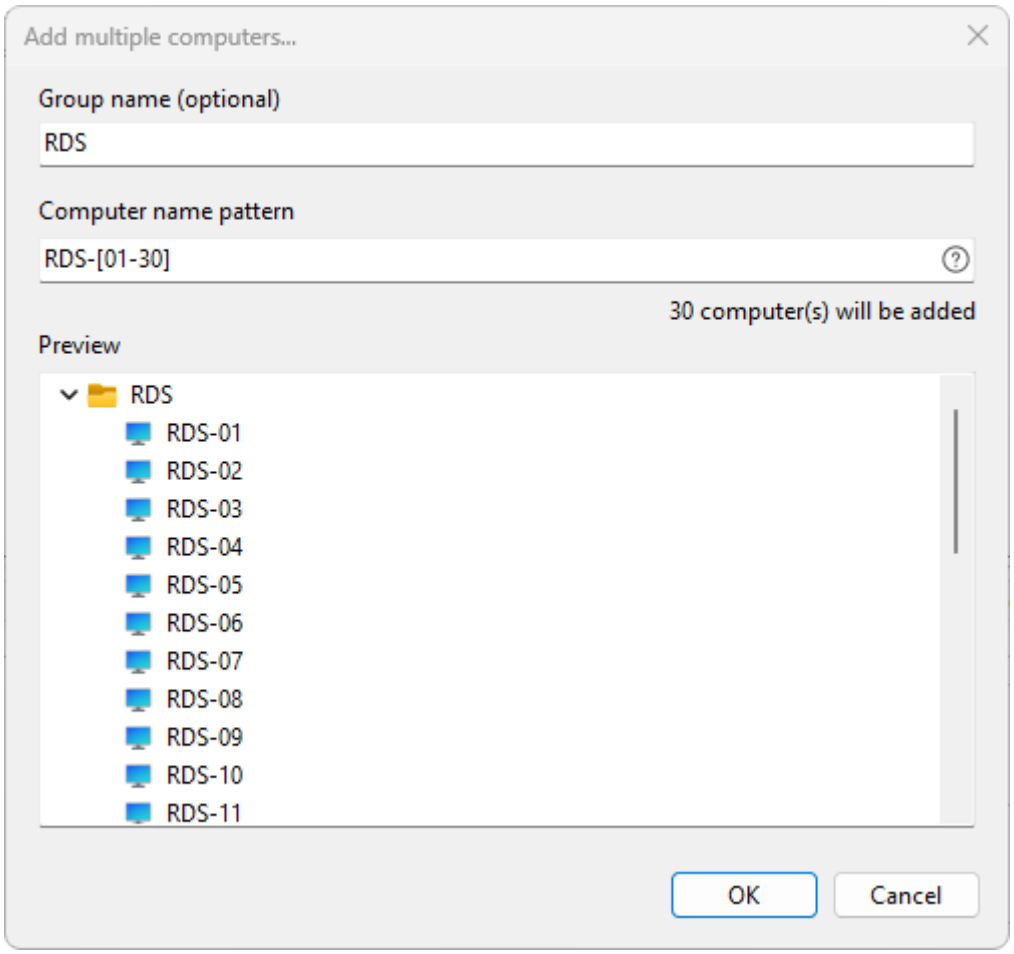
This section covers the tools that make a large list manageable. For the step-by-step basics of a single entry, see adding a single computer in the handbook.

Discovering computers in bulk

Typing servers in one by one does not scale. Terminal Services Manager gives you two faster paths: generate a batch of names from a pattern, or run the discovery wizard against a source that already knows your machines.

Generate names from a pattern

When your servers follow a naming convention, **Computers > Add multiple computers...** turns a pattern into a batch of entries. A live preview shows the names as you type.



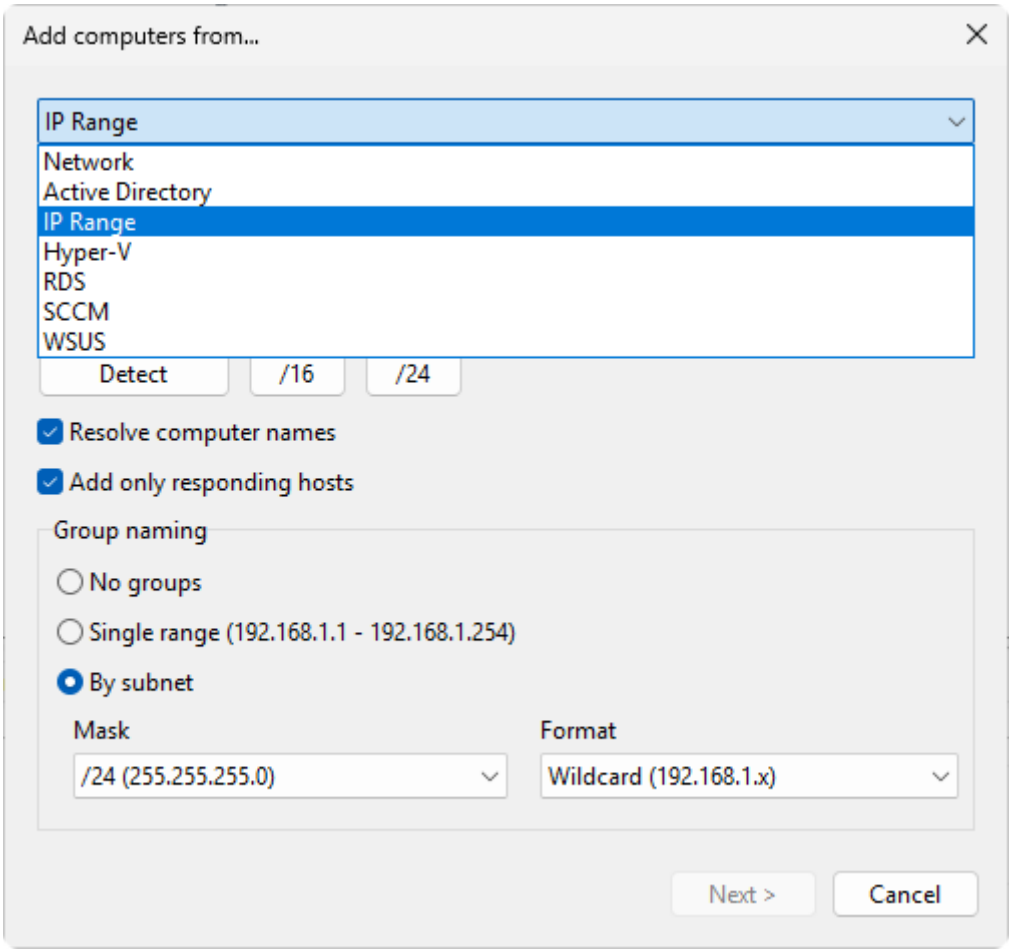
The pattern syntax expands ranges and lists:

Pattern	Expands to
`server[01-10]`	`server01` through `server10`
`rds{web,app,db}`	`rdsweb`, `rdsapp`, `rdsdb`
`rds[1-10:2]`	`rds1`, `rds3`, `rds5`, and so on, in steps of 2
`host[A-C][1-3]`	`hostA1`, `hostA2`, ... `hostC3`

Set an optional **Group name** to drop the whole batch into one group. The preview lists the first 1,000 names; the list accepts up to 10,000 entries in total.

Discover from a source

When you do not know every name, **Computers > Add computers from...** opens a three-step wizard that pulls servers from a source you already run.



On the first page, pick the source from **Import from**:

- **Network** - browse the Windows network for computers
- **Active Directory** - query a domain for computer accounts
- **IP Range** - scan a range of IP addresses
- **Hyper-V** - list virtual machines on a Hyper-V host
- **RDS** - pull session hosts from a Remote Desktop Services deployment
- **SCCM** - read devices from a Configuration Manager database
- **WSUS** - read computers from a WSUS server

Fill in the fields for that source and click **Next >**. The wizard searches and fills the **Found computers** list; you can click **Next >** before the scan finishes to stop accepting new results.

On the last page, choose where the results land. Under **Target group**, keep the source's own grouping, create a **New group**, or add them to an **Existing group**. Under **Duplicates**, choose **Skip duplicates** (the default), **Add duplicate**, or **Replace existing**, then click **Finish**.

Each source has its own page in the handbook, starting with adding computers using the wizard.

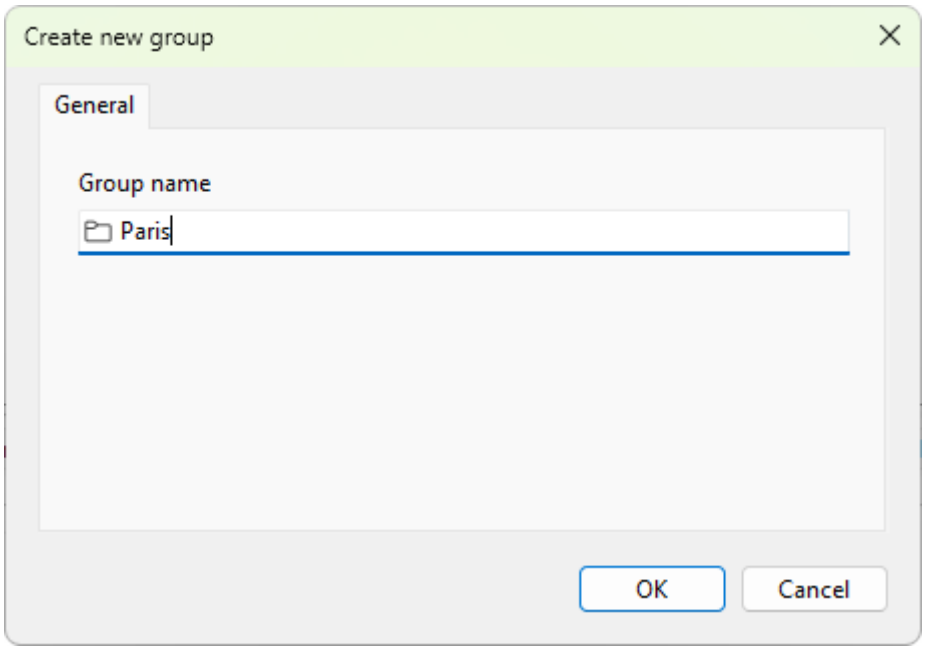
Once the servers are on the list, organize them into groups so it stays readable.

Organizing and connecting

A flat list of a hundred servers is hard to work with. Groups give the list structure, so it stays navigable as it grows. A separate question is which account the program connects with, and the answer is simpler than you might expect: one account for the whole list.

Group your servers

Create a group with **Computers > Create new group...**, or right-click in the list and choose **Create new group...**. Give it a name and click **OK**. Drag servers and other groups onto it to build the hierarchy you want; drag a node to the root to pull it back out.



Group by whatever matches how you work: by site, by domain, by role (session hosts, brokers, license servers), or by customer. Groups are organizational only and do not change what is monitored, so you can regroup freely. The expand and collapse state travels with the list when you export it.

To remove a group, right-click it and choose **Delete**. Deleting a group removes the servers inside it too, so move anything you want to keep first.

Connecting under the right account

Groups organize the list; they do not hold credentials. Terminal Services Manager connects to every server under the single Windows account it is running as, the same account for the whole list. There is no per-server or per-group user name and password to set.

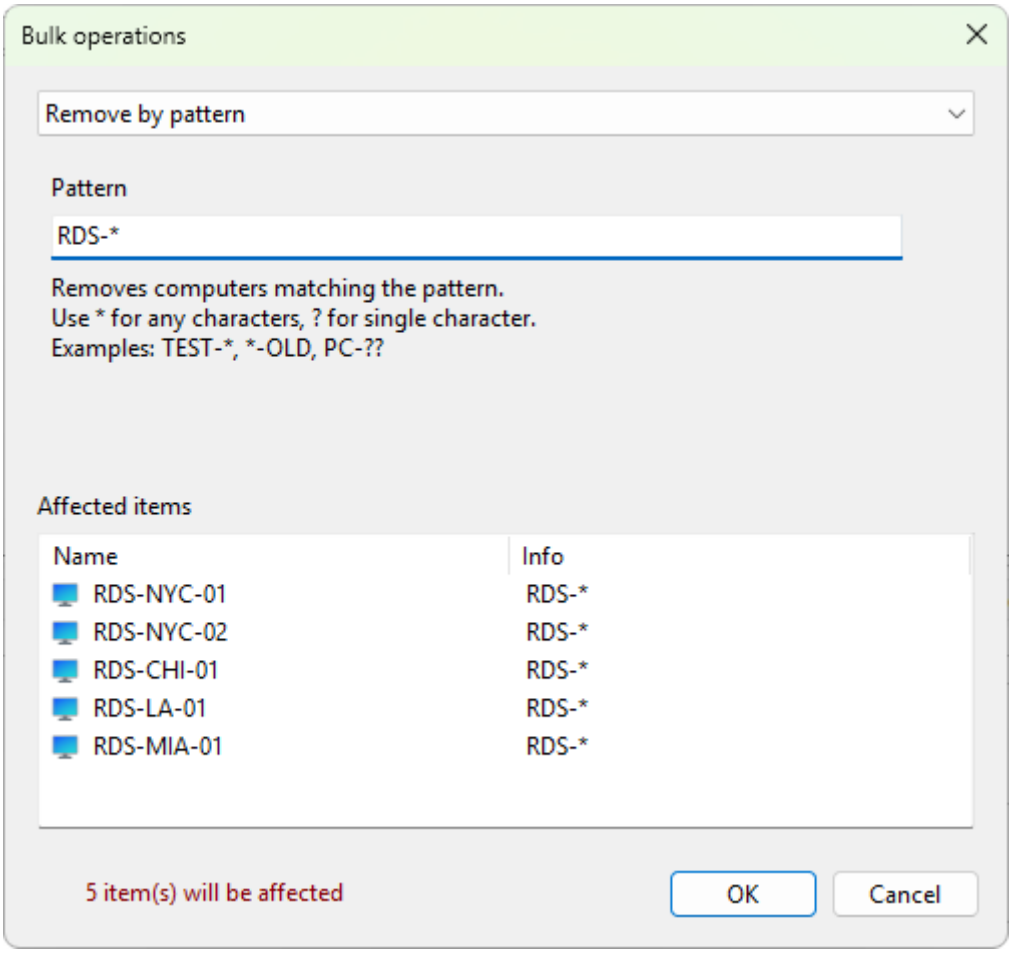
So the way to reach hosts on a different domain, or hosts your everyday account cannot administer, is to run the whole program under an account that can. Hold **Shift**, right-click the program shortcut, choose **Run as different user**, and sign in with an account that is a local administrator on those hosts. The handbook covers this in connection credentials.

At scale, the practical pattern is to keep one management workstation, or a session on a jump host, signed in under an account that administers your RDS fleet, and run Terminal Services Manager from there. An `Access denied` result or empty data from a server almost always means the account you are running as lacks rights on that host.

When the structure or names need a wider change, reach for bulk operations.

Bulk operations

When a list grows by import and discovery, it collects duplicates, stale entries, and inconsistent names. **Computers > Bulk operations...** reshapes the entire list in one pass, with a preview of exactly what will change before you commit.



Pick an operation from the dropdown and the **Affected items** preview updates to show the result. The available operations are:

Operation	What it does
Remove inactive computers	Drops servers that do not respond.
Remove duplicate computers	Keeps the first entry of each name and removes the rest (case-insensitive).
Remove empty groups	Deletes groups that hold no servers or subgroups.
Remove by pattern	Removes names matching a wildcard, such as `TEST-*` or `*-OLD`.

Flatten group structure	Moves every server to the root and removes all groups.
Normalize computer names	Converts names to one case style: UPPERCASE, lowercase, or Title Case.
Rename computers	Adds or removes a prefix or suffix, or replaces text across names.
Group computers	Builds groups automatically from a name prefix, an IP subnet, or a domain.
Format descriptions	Sets, clears, or rewrites the description field in bulk.

Group computers is the fast way to impose structure on a freshly imported flat list: group by domain to split a mixed list by site, or by name prefix to gather each server role together. **Rename computers** and **Normalize computer names** fix the naming drift that builds up when several people add servers over time.

Bulk operations apply immediately and cannot be undone from inside the dialog. If you are not certain, export the list first so you can restore it. The list is only written to disk when you exit the program normally, so closing without saving also rolls a change back.

The handbook documents each operation in detail under bulk operations.

Importing and exporting

A computer list you built once should not stay on one machine. Export it to roll the same list out to a team, move it to a new workstation, or keep it under version control so changes are tracked.

Export the list

Choose **Computers > Export computer list** and pick a format:

- **JSON** - the default. It diffs cleanly in version control and keeps the full structure: names, descriptions, notes, groups, and the expand and collapse state.
- **XML** - for interoperability with other tools.
- **Text** - one entry per line as `name|description`. The simplest format to hand-edit or generate from a script.
- **RHF** - Radmin Host File, kept for compatibility with older lists.

The export captures the whole tree, so a colleague who imports it sees the same groups you do.

Import a list

Choose **Computers > Import computer list** and select a file; the format is detected from the extension. You then choose how it merges with the current list:

- **Add new computers only** - the default and the safest. Your current list stays, and only names not already present are added.
- **Replace entire list** - clears the list and loads the file in its place.
- **Import all** - adds everything, including duplicates.

Import does not test connectivity or credentials; it loads the names as written, and the program connects to them afterward. As with bulk operations, **Replace entire list** takes effect on disk only when you exit normally, so you can back out by closing without saving.

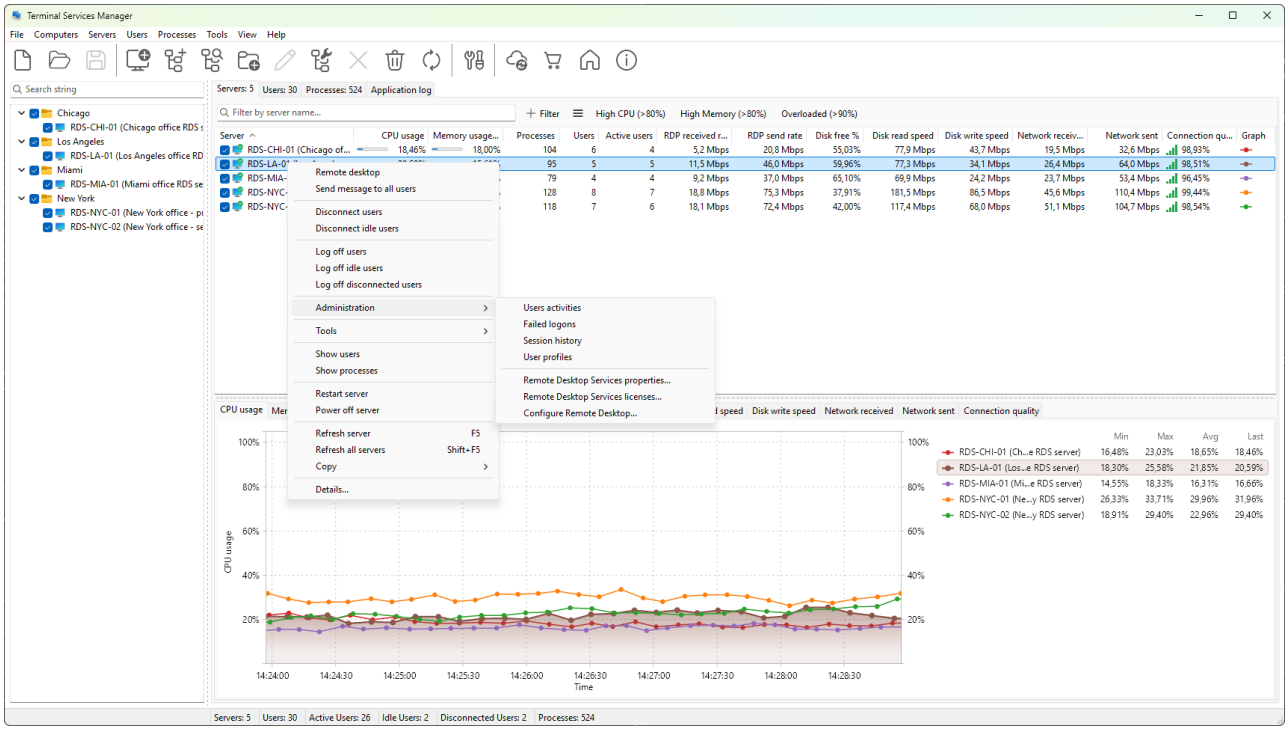
A simple rollout

To standardize a team on one list, build and groom it once, export to JSON, and share the file. Each person imports with **Replace entire list** to match it exactly, or with **Add new computers only** to merge it into a list they already keep. Storing that JSON in a repository gives you a history of every change.

For more detail, see importing and exporting the computer list in the handbook.

Server administration tools

Terminal Services Manager does more than watch your servers. From the **Servers** menu and the **Administration** submenu you can run the day-to-day RDS administration tasks that would otherwise mean opening a remote desktop or RSAT console on each host, one at a time.



This section groups those tools by the job they do.

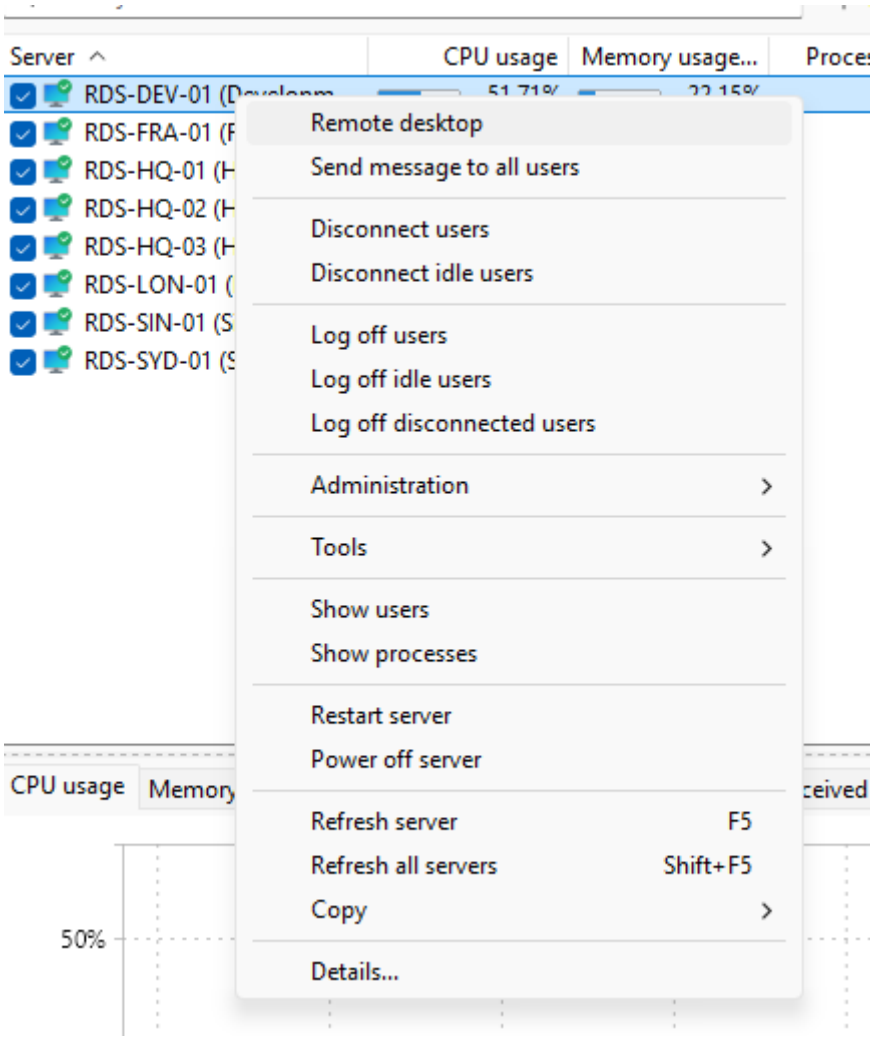
Most tools accept more than one server at a time. Select several hosts on the **Servers** tab and the action runs against all of them.

Power and Remote Desktop

These are the actions that get you onto a server or change its power and connectivity state. Each one is on the **Servers** menu or the server's right-click menu, and each accepts more than one server at a time.

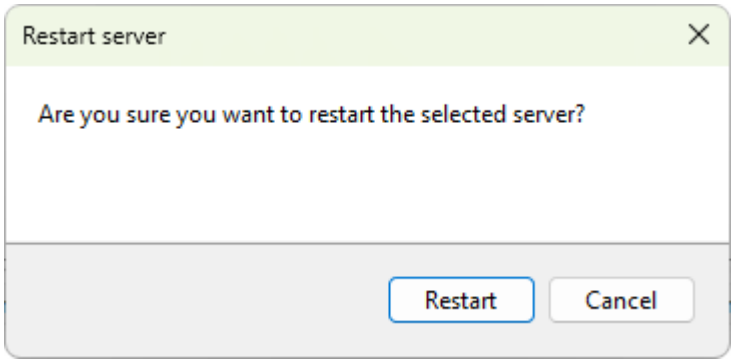
Connect over Remote Desktop

When you need a full desktop on a host, select it and choose **Servers > Remote desktop**. Terminal Services Manager opens a standard Remote Desktop connection to that server, so you do not have to keep separate shortcuts for each one.



Restart or shut down

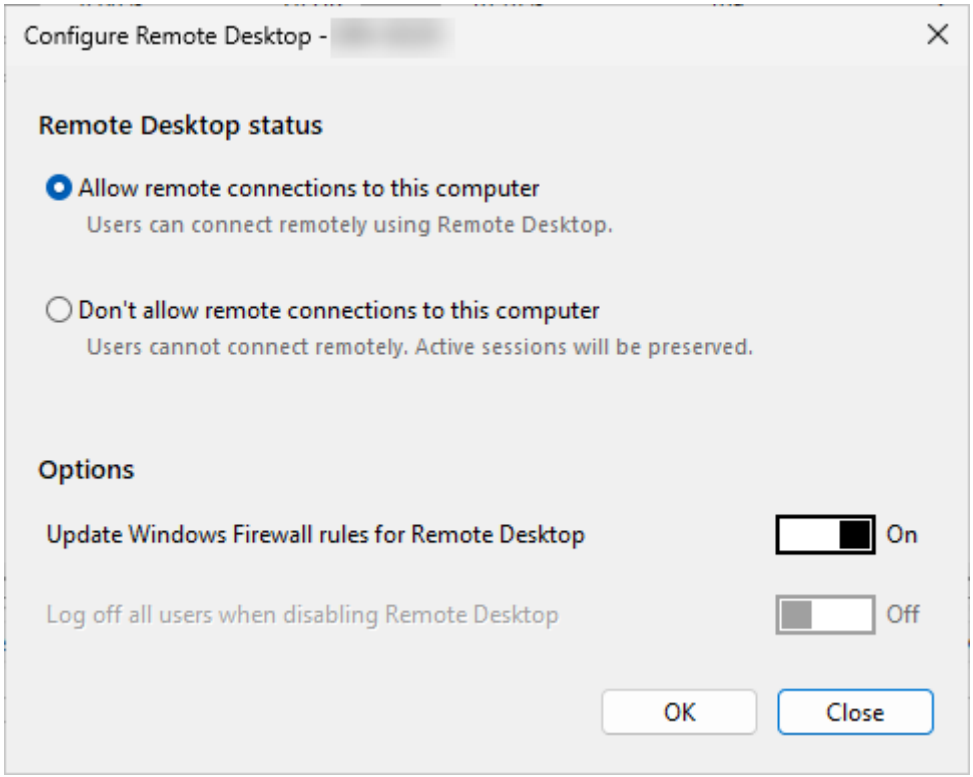
Select one or more servers and choose **Servers > Restart server** or **Servers > Power off server**. A confirmation prompt appears; click **Restart** or **Power off** to go ahead, or **Cancel** to stop. When you select more than one server, the prompt tells you how many.



Because you can select several servers first, a maintenance restart across a group is one action rather than one per host. There is no message field on the prompt, so if you need to warn users, send them a message before you restart.

Enable or disable Remote Desktop

To change whether a host accepts Remote Desktop connections at all, choose **Servers > Administration > Configure Remote Desktop...**. The **Configure Remote Desktop** dialog lets you **Allow remote connections to this computer** or **Don't allow remote connections to this computer**, update the Windows Firewall rules to match, and optionally log off all users when you turn Remote Desktop off.



The handbook covers these in connecting via Remote Desktop, rebooting and shutting down, and enabling or disabling Remote Desktop.

Next, review how each host is configured with the RDS configuration tools.

RDS configuration

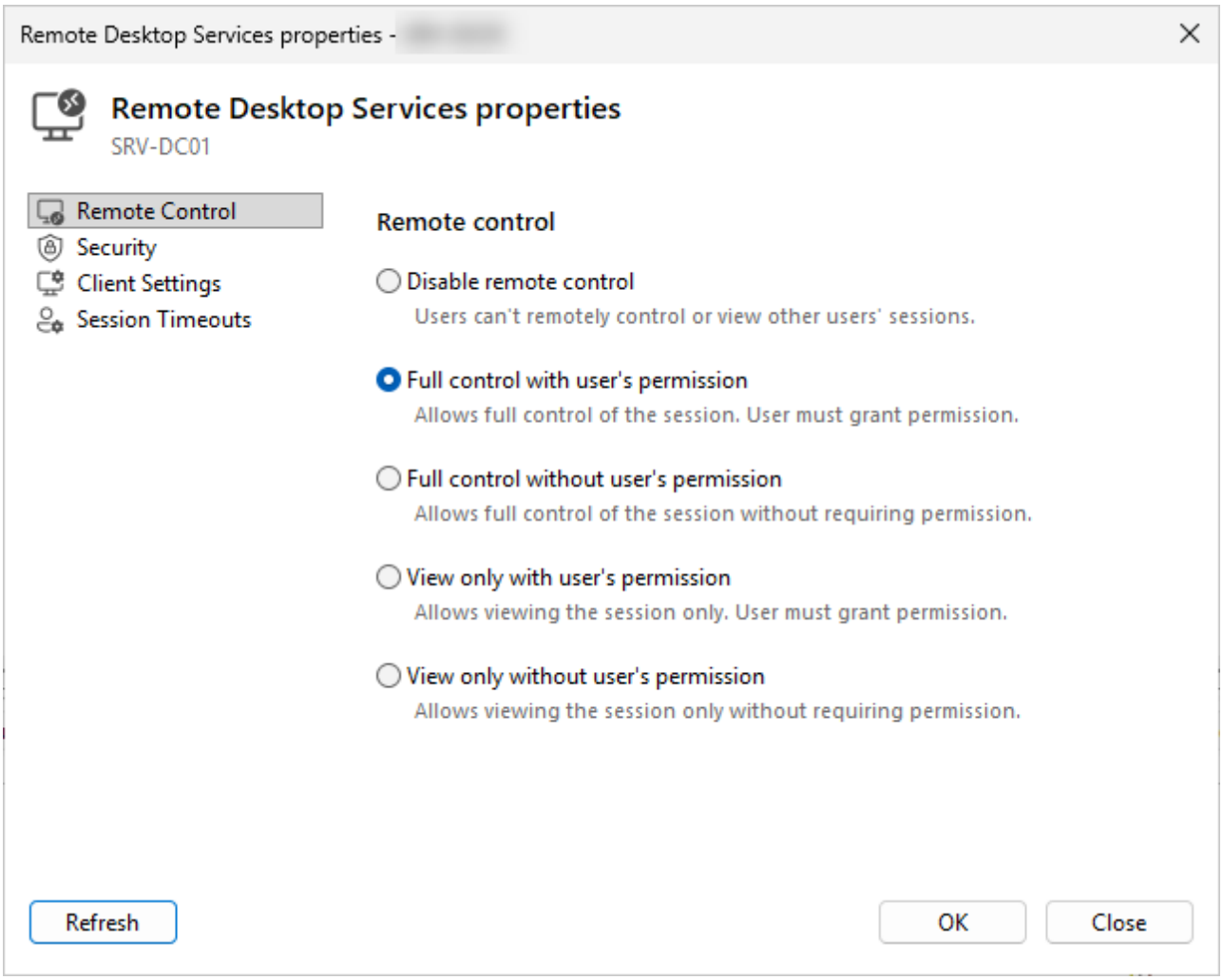
When you are responsible for many hosts, you need to see how each one is configured without logging into it. These two dialogs read that configuration straight from the server.

Remote Desktop Services properties

Select a server and choose **Servers > Administration > Remote Desktop Services properties...**. The dialog reads the host's RDS settings, grouped on four pages you pick from the list on the left:

- **Remote control** - how shadowing is allowed: disabled, full control, or view only, with or without the user's permission. This is the one page you can change here; the rest are read-only.
- **Security** - the security layer, encryption level, and certificate the host uses.

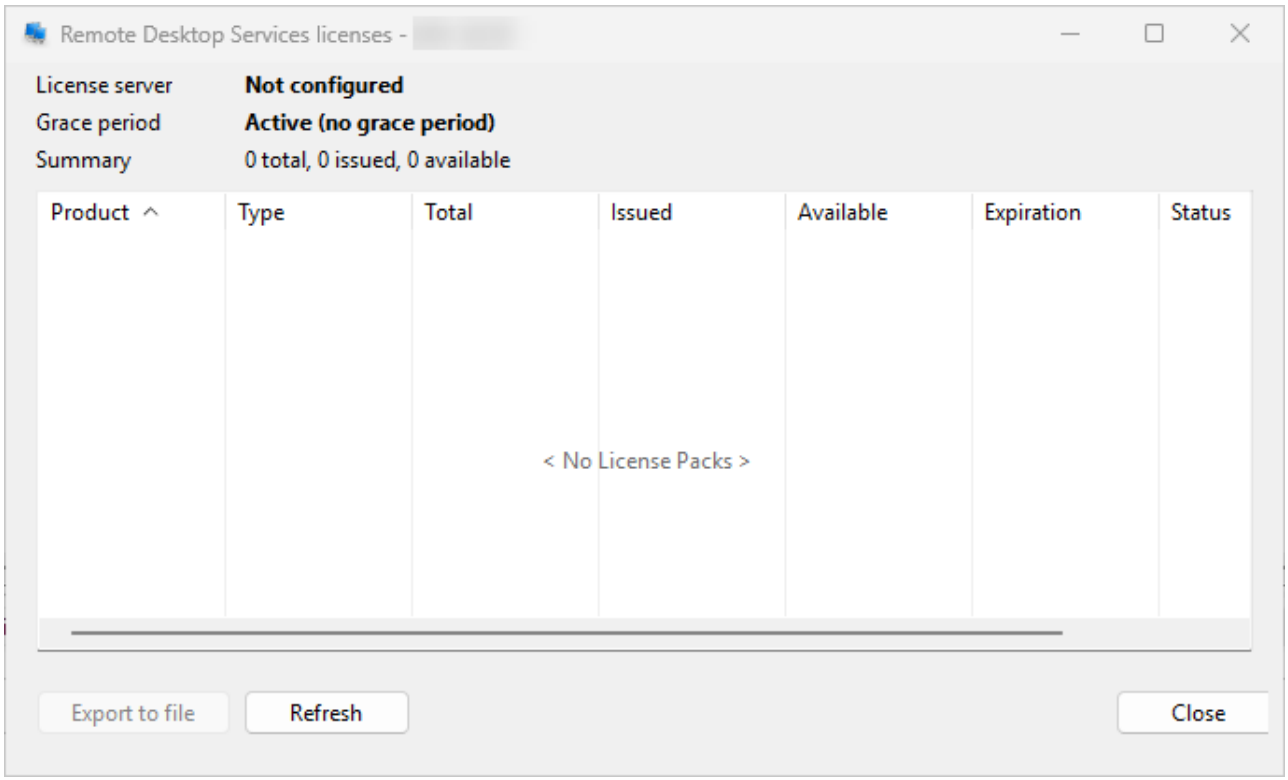
- **Client settings** - which redirections are allowed (drive, clipboard, printer, audio, COM port) and the color depth.
- **Session timeouts** - the active, disconnected, and idle session limits.



Reading these side by side across your hosts is the fastest way to spot one server whose timeouts or redirection settings drift from the rest. Click **Refresh** to re-read a host after a change.

Remote Desktop Services licensing

Choose **Servers > Administration > Remote Desktop Services licenses...** to see how a host's RDS Client Access Licenses are being used. The **RDS licenses** dialog shows the license server it points to, the grace period state, and a per-license breakdown of total, issued, and available counts with expiration dates.



This is where you catch a host still in its grace period, or a license pool running low before users start getting turned away. Use **Export to file** to keep a record or feed it into a report.

The handbook has the full reference in RDS properties and RDS licensing.

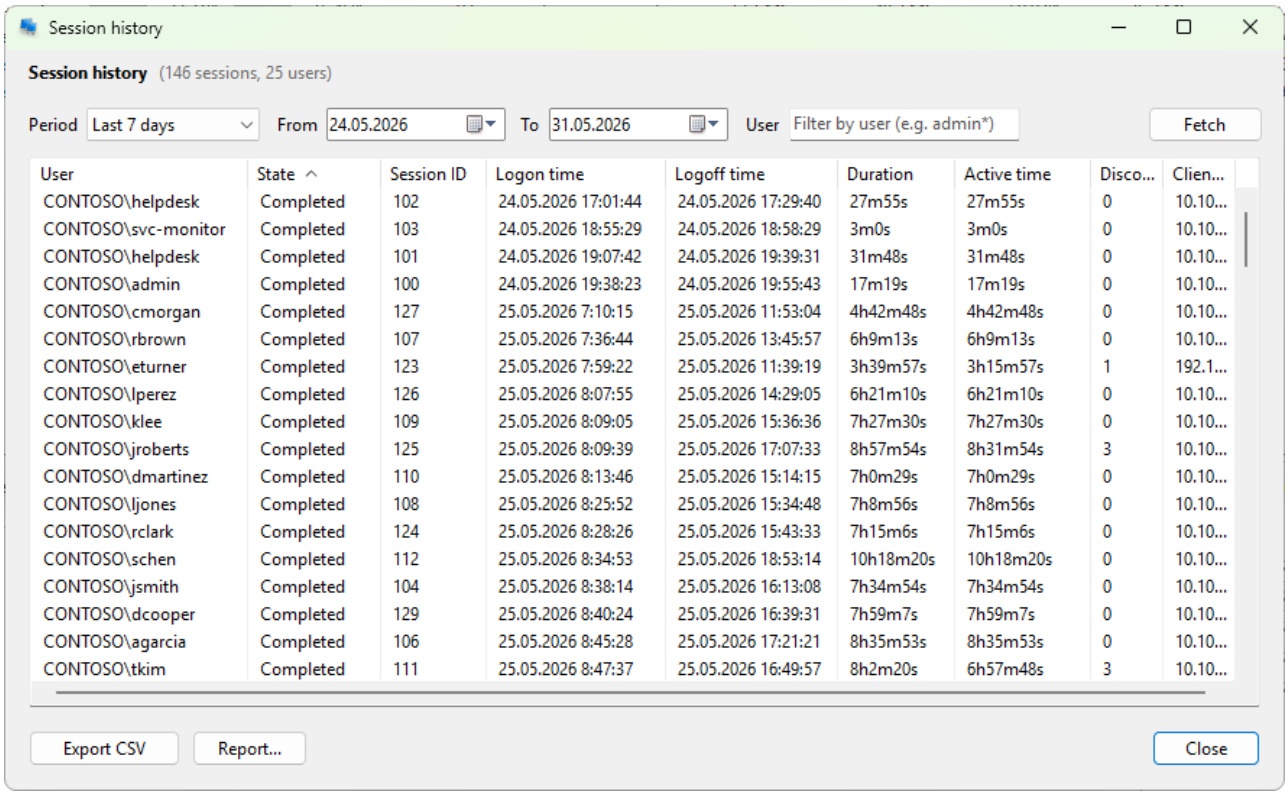
To investigate who connected and when, move on to auditing sessions.

Auditing sessions

Live tabs tell you what is happening now. These four tools answer what happened: who logged on, who failed to, what each session did, and what profiles are left behind. Each one can be run from a server on the **Servers** tab or scoped to a single user from the **User sessions** tab, and each exports to a file for reporting.

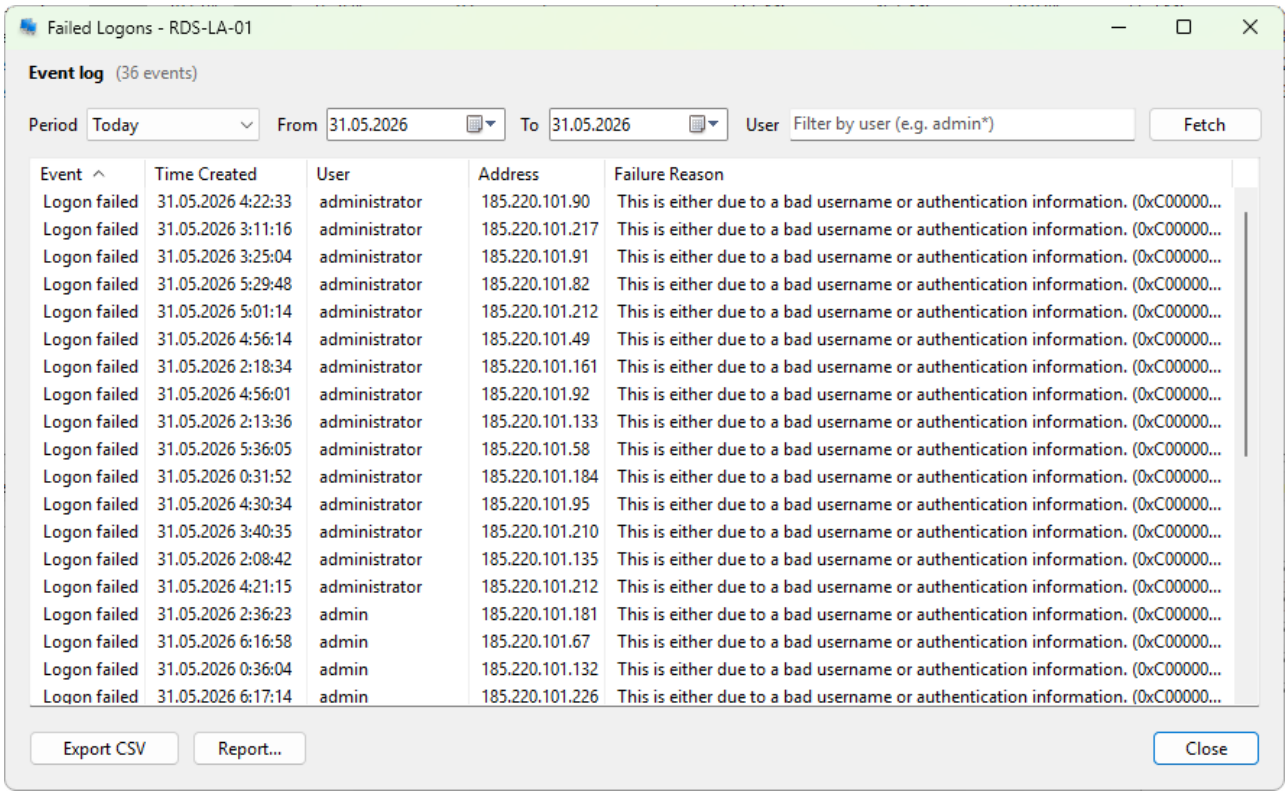
Session history

Choose **Servers > Administration > Session history** to see logon and logoff events over a period you set with the **From** and **To** dates. For each session you get the user, state, logon and logoff times, duration, active time, the number of disconnects, and the client address. Filter by user name with * and ? wildcards, then click **Export CSV** for a record.



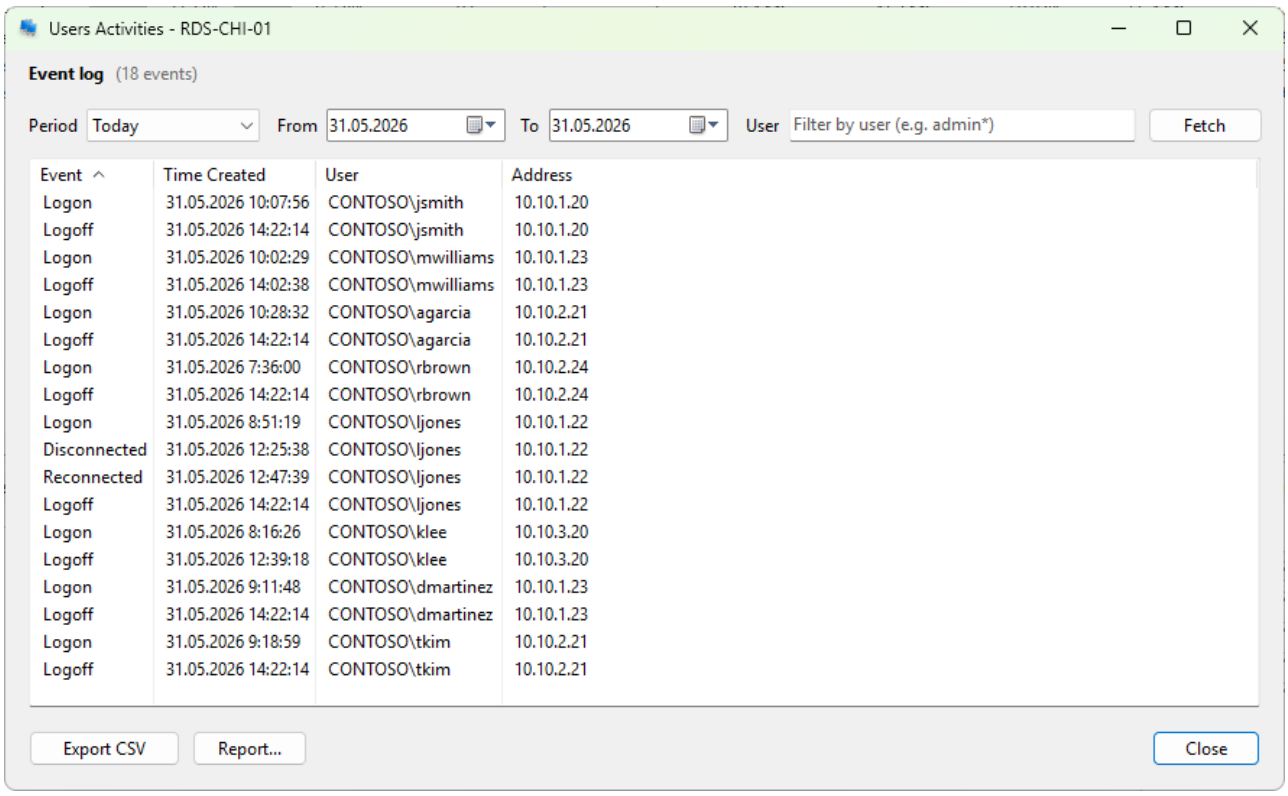
Failed logons

Choose **Servers > Administration > Failed logons** to review sign-in attempts that did not succeed. Each row gives the time, the user name tried, the source address, and a plain-language failure reason such as a bad password, a locked account, or an attempt outside permitted hours. This is the first place to look when you suspect a brute-force attempt or a misconfigured account.



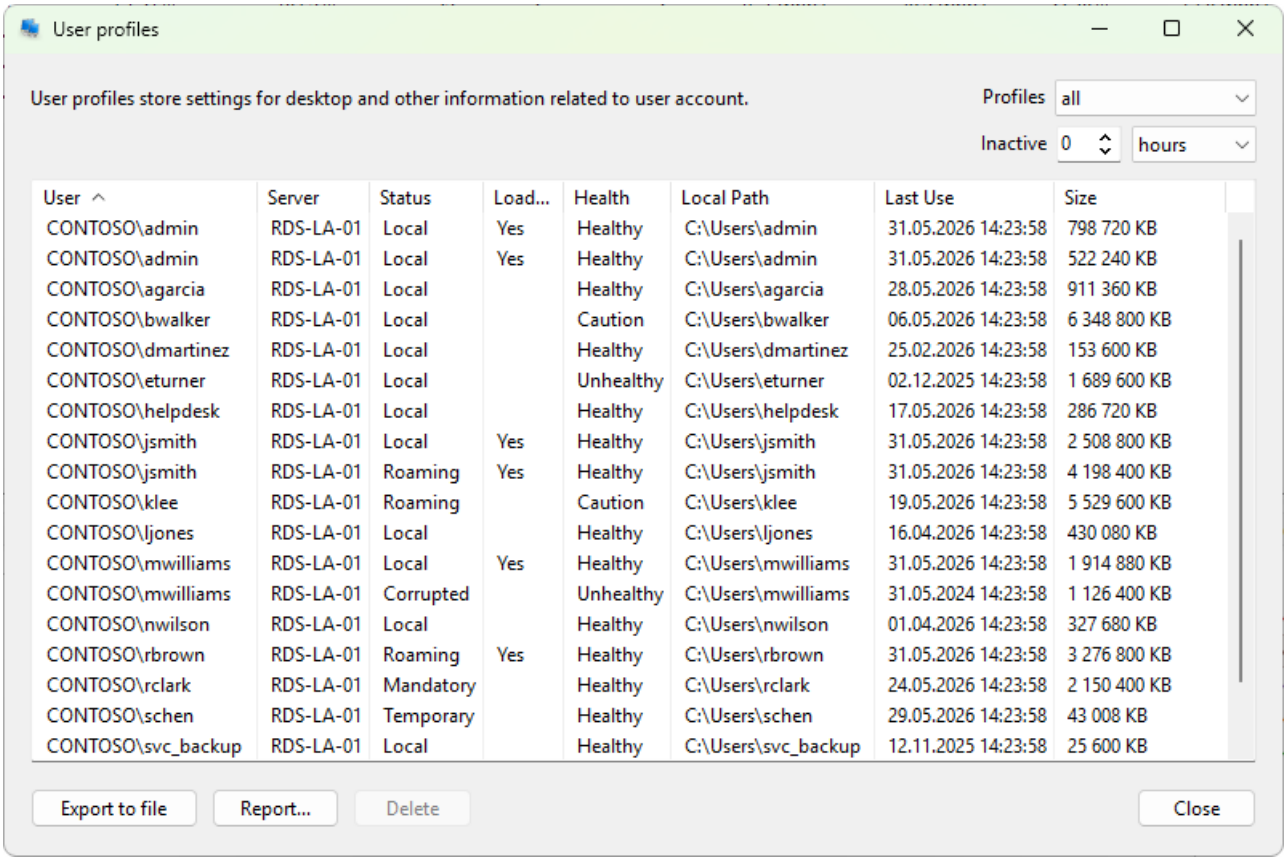
User activity

Choose **Servers > Administration > Users activities** to read the host's session log: logon, logoff, disconnect, and reconnect events with the time, user, and client address. It draws on the Windows Remote Desktop Services operational log, so it reflects what the server itself recorded.



User profiles

Choose **Servers > Administration > User profiles** to list the profiles stored on a host, with their type (local, roaming, mandatory, temporary, or corrupted), size, and last use. Filter to inactive profiles older than a threshold you set, then select and **Delete** the ones you no longer need to reclaim disk space. You can run this across several servers at once.



For the full reference, see session history, failed logons, users activities, and user profiles.

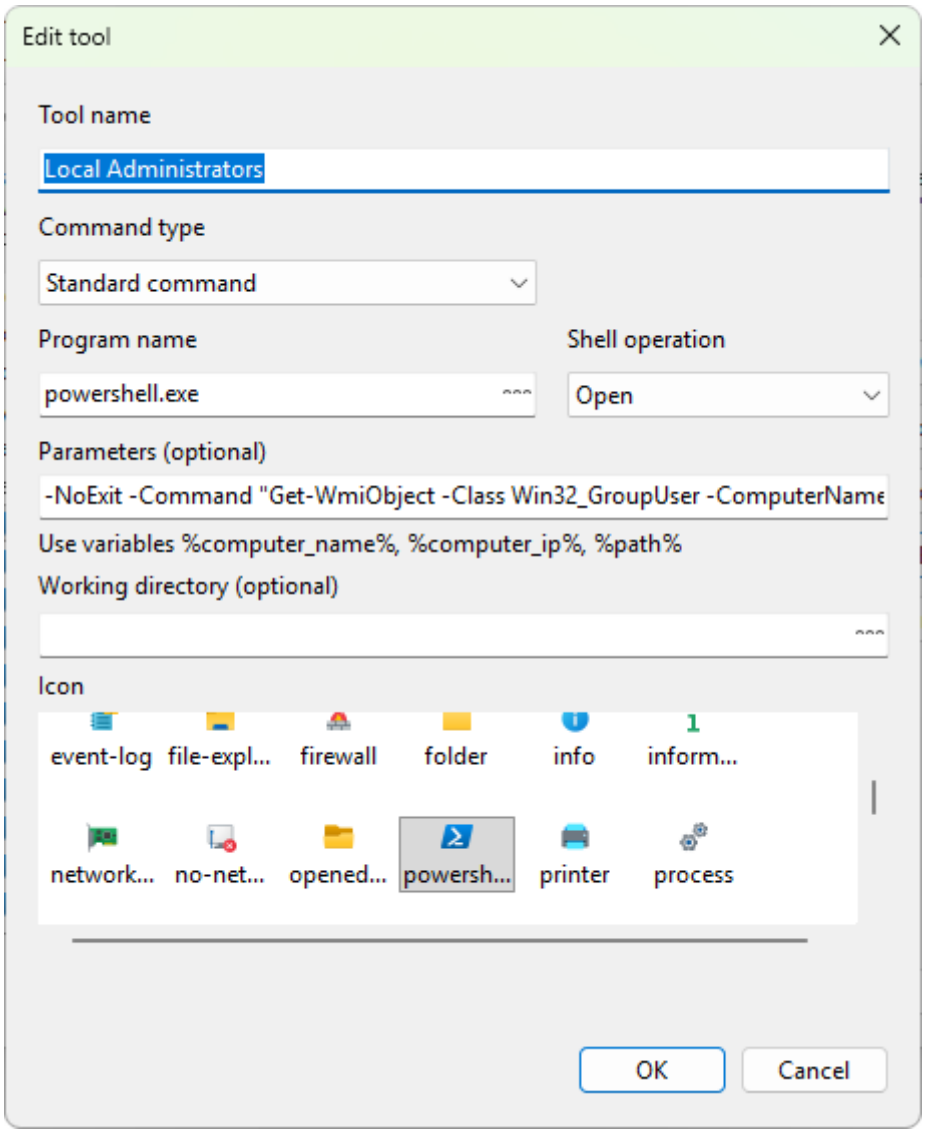
To extend the menu with your own utilities, see custom command-line tools.

Custom command-line tools

No tool ships with every utility an administrator uses. Terminal Services Manager lets you add your own to the **Tools** menu, so the scripts and executables you already rely on run against whatever server or session you have selected.

Add a tool

Choose **Tools > Manage tools** to open the tool manager, then **Add item** to create one. In the tool editor you give it a name, point **Program name** at the executable or script, and set the **Parameters** it runs with.



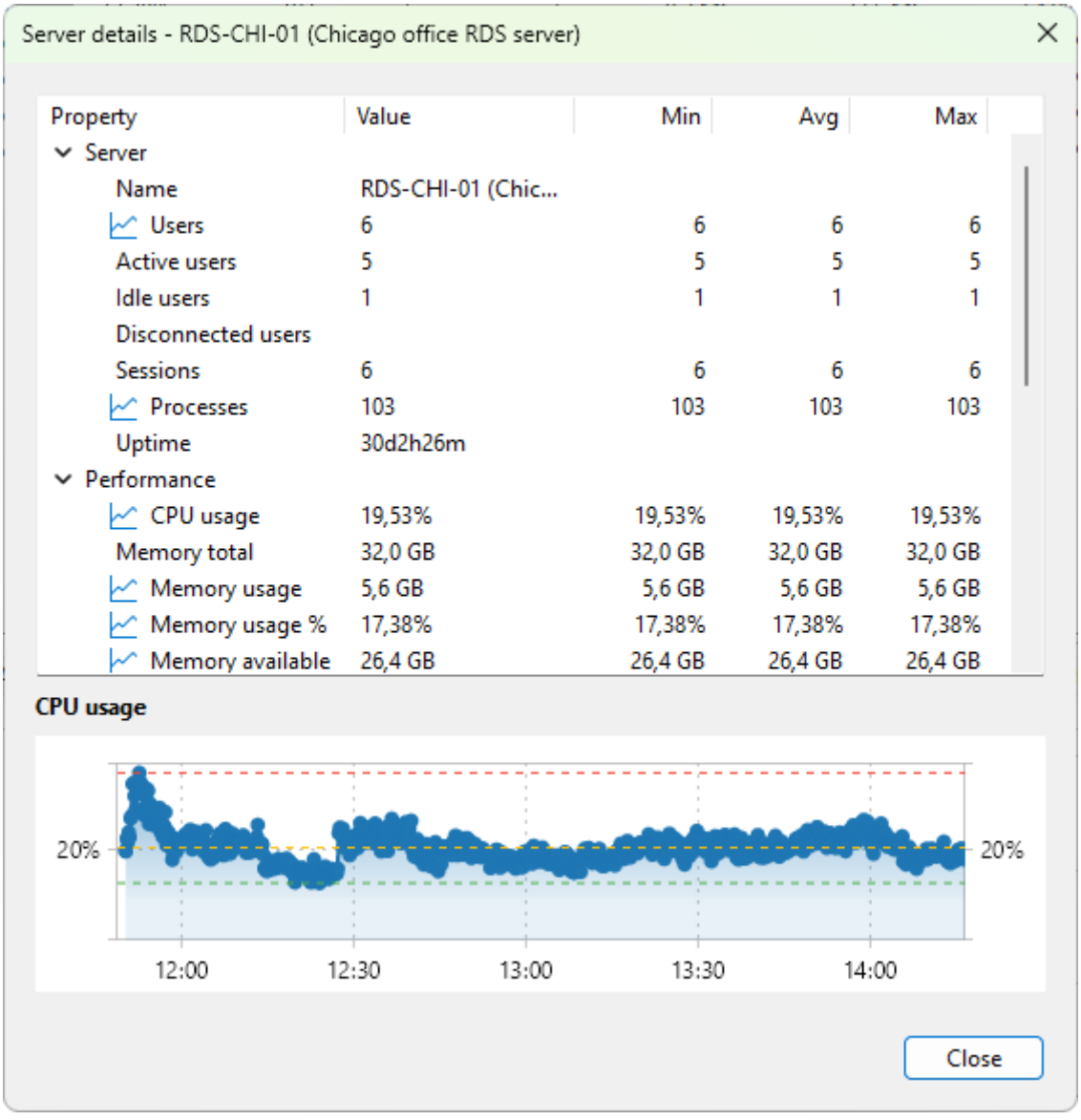
The power is in the parameters. Variables are filled in from the selected server or computer when the tool runs, so one entry works against whichever host you have selected. The editor's hint lists the variables you can use:

Variable	Replaced with
`%computer_name%`	the selected server or computer name
`%computer_ip%`	the selected computer's address
`%path%`	a UNC path to the computer, in the form `\\name`

For example, a tool that runs `mstsc.exe /v:%computer_name%` opens a remote desktop to whatever server is selected, and a PowerShell script that takes `%computer_name%` as an argument runs against the host you are looking at. Group related tools under categories with **Add category**, and reorder them with **Up** and **Down**. You can export your tools to share the set with colleagues.

Read every property with entity details

To see everything Terminal Services Manager knows about a server, session, or process, double-click the row, or right-click it and choose **Details...**. The entity details window lists every property grouped by category, with the current, minimum, average, and maximum value for each metric, and a chart of the one you select over time.

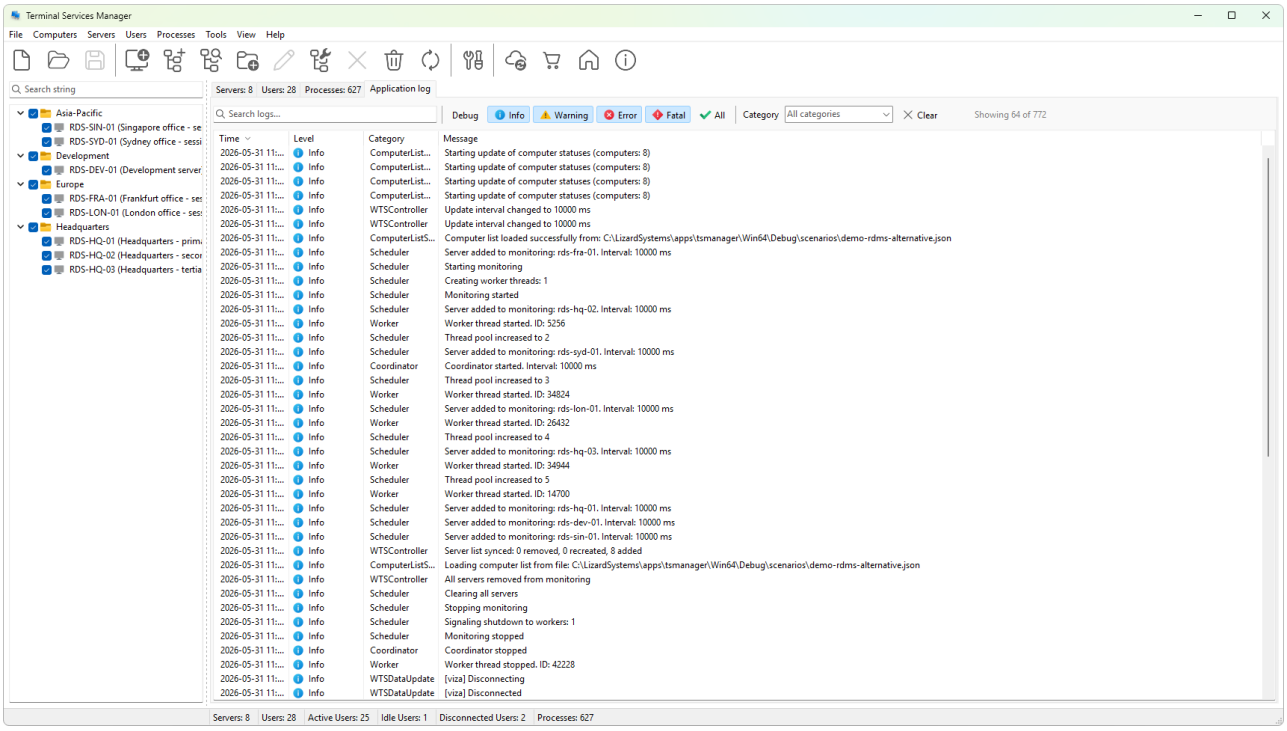


This is the tool for a deep look at a single item: the full performance picture for a server that is misbehaving, or every attribute of a process you are about to end.

See custom command-line tools and entity details in the handbook for the complete reference.

Application log

The **Application log** tab records what Terminal Services Manager itself is doing: the servers it connects to, the actions it runs, and any errors it hits along the way. When a connection fails or an action does not behave as expected, this is where you find out why.

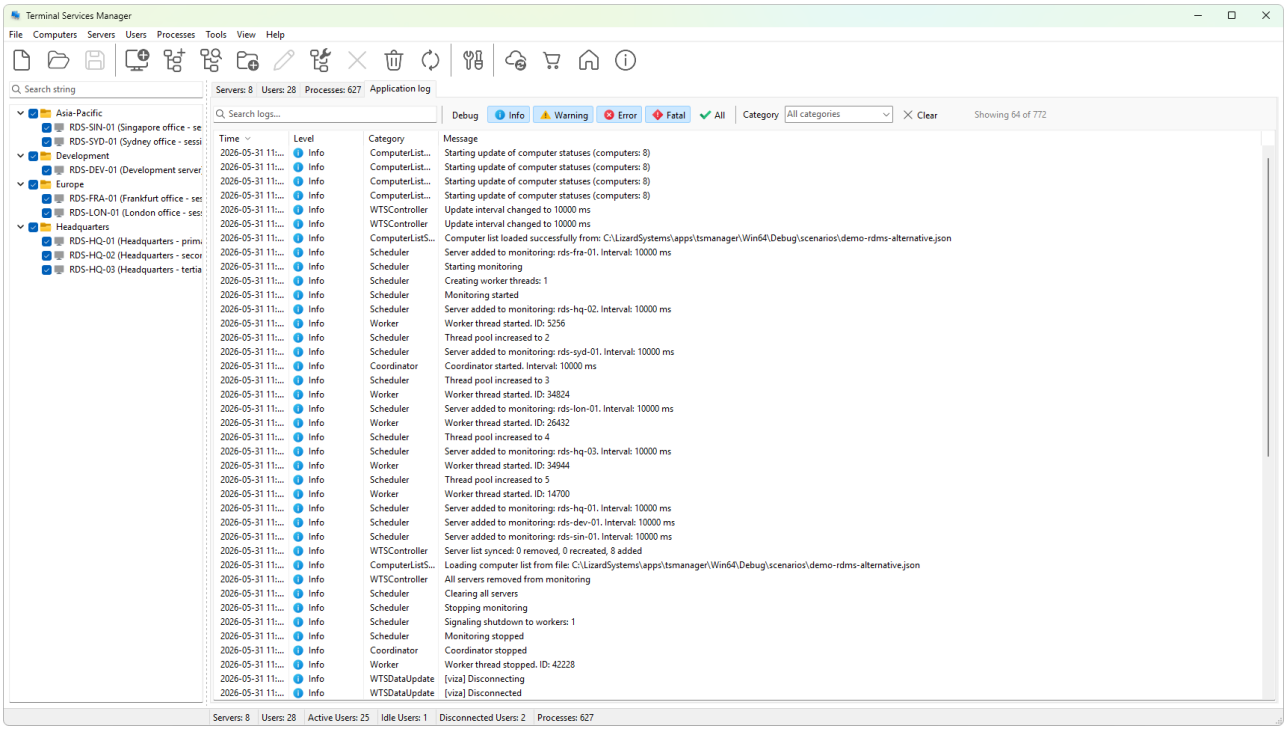


This section covers reading the log and getting useful information out of it.

Reading the log

The **Application log** tab is a running record of the program's activity. Each entry has four columns:

- **Time** - when the entry was recorded, in local time.
- **Level** - how important it is, shown with an icon.
- **Category** - which part of the program produced it, such as a server connection or an action.
- **Message** - what happened.



What the levels mean

The level tells you whether an entry is routine or something to act on. From least to most serious:

- **Debug** - fine-grained detail, useful when you are tracing a problem and rarely otherwise.
- **Info** - normal activity, such as a successful connection or a completed action.
- **Warning** - something unexpected that the program handled, worth a look but not a failure.
- **Error** - an operation that failed, such as a server it could not reach.
- **Fatal** - a serious failure that stops part of the program from working.

When you are troubleshooting, read the **Error** and **Warning** entries first: they point straight at what went wrong. The **Info** entries around them give the context, showing what the program was doing at the time. **Debug** entries add the most detail and are most useful when you are reproducing a problem on purpose.

The log keeps the most recent entries in memory, so it shows current activity rather than a complete history. To capture what you see for later, filter and export it.

For the full reference, see the application log tab and log levels in the handbook.

Filtering and exporting

A busy log scrolls quickly. The filter strip above the list narrows it to what you are looking for, and the right-click menu gets the result out of the program.

Narrow the log

Use the controls at the top of the **Application log** tab together; they combine, so each one you add tightens the result:

- **Level toggles** - turn off the levels you do not care about. Hiding **Debug** and **Info** leaves just the **Warning**, **Error**, and **Fatal** entries, which is usually what you want when something has gone wrong.
- **Category** - pick a category to focus on one part of the program, such as server connections.
- **Search text** - type to keep only entries whose message contains that text, like a server name.

Clear the filters to see everything again.

Copy and export

Right-click the list to copy or save entries:

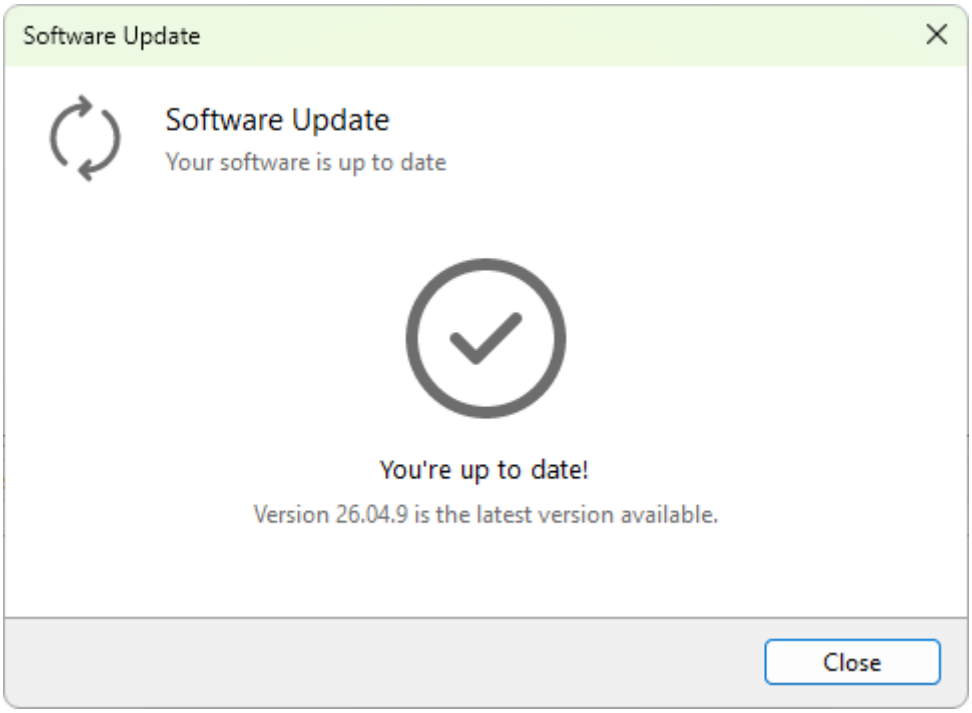
- **Copy** the selected rows to paste into an email or a ticket.
- **Export to file...** to save the visible entries. Choose **CSV** to open in a spreadsheet, **JSON** for processing, or **Text** for a plain log.

When you contact support, the most useful attachment is an export filtered to the **Warning**, **Error**, and **Fatal** entries from around the time the problem happened. That gives the smallest file that still shows what went wrong.

The handbook covers this in filtering and searching log entries and copying and exporting log entries.

Updates and licensing

The last part of running Terminal Services Manager well is keeping it current and licensed. Updates bring fixes and new features, and registering your license removes the evaluation reminders and sets the program up for ongoing use.

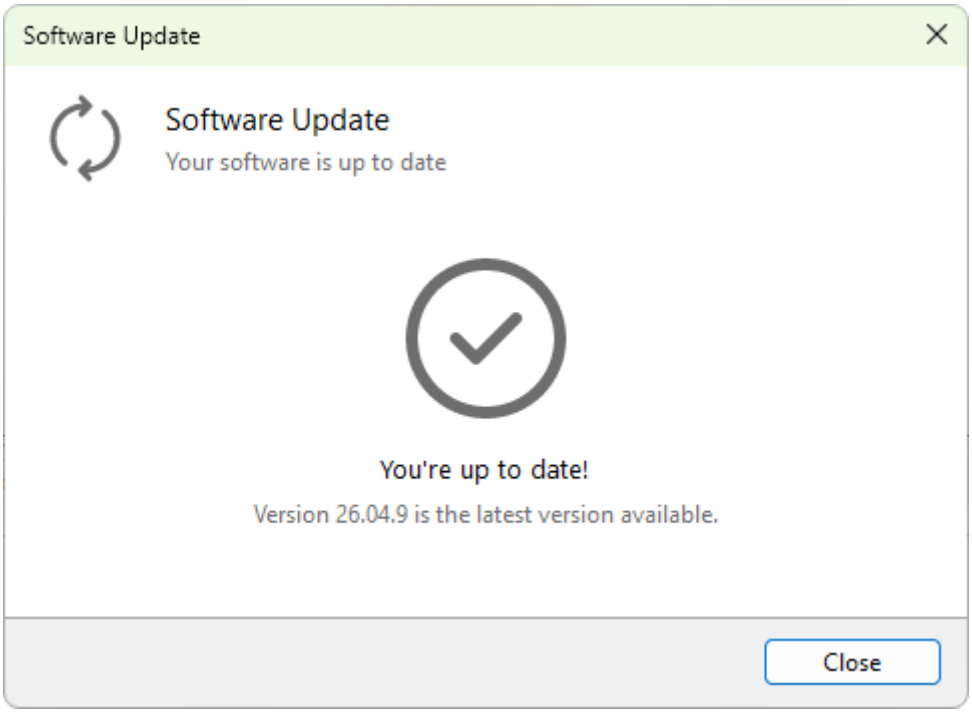


Keeping up to date

Keeping the program current means you get fixes and new features as they ship. Terminal Services Manager checks for updates on its own and lets you check on demand.

Check on demand

Choose **Help > Check for updates...** at any time. The **Software Update** dialog tells you whether you are up to date or a newer version is available. When one is, choose **Download & Install** and the program downloads the new version and installs it for you.



Automatic checks

Terminal Services Manager also checks periodically and lets you know when a new version is ready, so you do not have to remember to look. You can adjust how it behaves in **Preferences**, under software updates.

Rolling out to several machines

If you run Terminal Services Manager on more than one workstation, you do not have to update each one by hand. Install or upgrade from the command line with the Windows Package Manager, `winget upgrade tsmanager`, or run the installer with its silent switches from your own deployment tooling. The installing Terminal Services Manager page in the handbook lists the silent switches, and installing with winget covers the package manager.

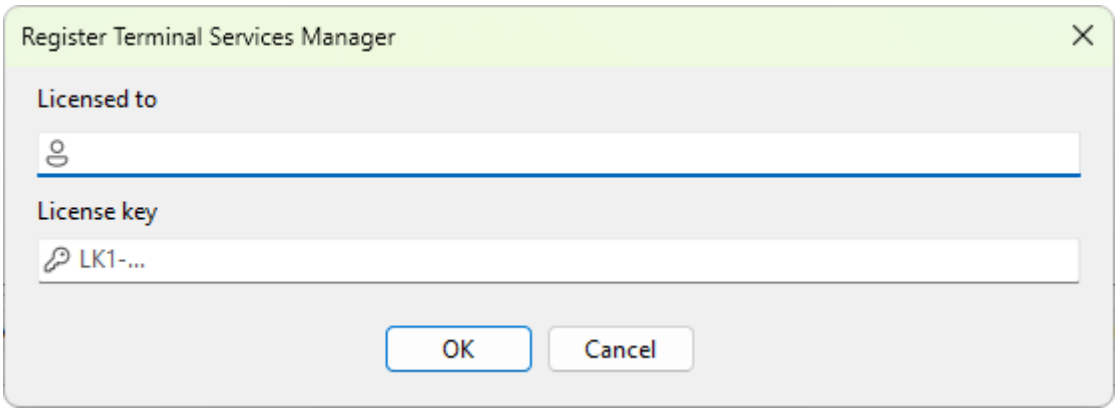
Once the program is current, make sure it is registered.

Registering your license

Registering Terminal Services Manager removes the evaluation reminders and sets it up for ongoing use. You enter your license once on each machine.

Enter your license key

1. Choose **Help > Enter registration code...**.
2. In **Licensed to**, type the name your license was issued to.
3. In **License key**, paste the key. It begins with `LK1-TSM-`.
4. Click **OK**.



The name and key are checked together, so both must match what was issued. When they do, the dialog closes and the program is registered. You can confirm it under **Help > About**, which then shows the name the program is licensed to, and the evaluation reminders stop.

If you do not have a key yet

Your license key is in the email you received when you purchased, and in your account on the LizardSystems customer portal. If you cannot find it, the support team can resend it.

For the same steps in the handbook, see registering your license.