WinGet Windows package manager

START HERE

`winget search <text>`
Find a package and its ID

`winget show <id>`
Read details first

`winget install --id <id> -e`
Install one exact package

`winget list`
See what is installed

`winget upgrade`
See what can be upgraded

`winget upgrade --all`
Upgrade everything

`winget uninstall <id>`
Remove a package

`winget export -o <file>`
Save your app list

Source winget, msstore → *search* → **Manifest** ID, versions, installers → *install* → **Installer** msi, exe, msix, zip → *runs* → **Installed app** Apps & features

CONCEPTS

A query matches name, ID and moniker as a case-insensitive substring. <id> below works as a plain query; add --id and -e when several packages match.

list and upgrade also see apps installed by other means (Apps & features), not only those winget installed.

FIND PACKAGES

`winget search <text>`
Search all sources

`winget search --id <text>`
Match the package ID only

`winget search --cmd <command>`
Find the package providing a command

`winget search <text> -s msstore`
Search only the Microsoft Store

`winget show <id>`
Details, homepage, installer URL

`winget show <id> --versions`
List available versions

INSTALL

`winget install <query>`
Install the single matching package
Several matches print a list; refine the query.

`winget install --id <id> -e`
Install by exact ID

`winget install <id1> <id2> <id3>`
Install several packages in one go

`winget install <id> -v <version>`
Install a specific version

`winget install <id> -s winget`
Pick the source when IDs clash

`winget install <id> --scope user`
Install for current user only
Values: user or machine. Not every installer honours it.

`winget install <id> -h`
Silent install, no installer UI

INSTALL (cont.)

`winget install <id> -i`
Show the installer's own wizard

`winget install <id> --override "<args>"`
Replace installer arguments

`winget install <storeid> -s msstore`
Install a Store app by its ID

`winget download <id>`
Download the installer only
Saved to your Downloads folder unless -d <dir> is given.

INSTALLED AND UPGRADES

`winget list`
List installed apps

`winget list <query>`
Filter installed apps

`winget list --upgrade-available`
Only apps with an update

`winget upgrade`
Show available upgrades

`winget upgrade <id>`
Upgrade one package

`winget upgrade --all`
Upgrade every package

`winget upgrade --all --include-unknown`
Also apps with unknown version
Some apps report no version; they are skipped without -u.

`winget upgrade --all --include-pinned`
Also apps with a non-blocking pin

`winget upgrade --all -h`
Upgrade everything silently

`winget upgrade <id> --uninstall-previous`
Remove the old version afterwards

UNINSTALL AND REPAIR

`winget uninstall <query>`
Uninstall a matching app

`winget uninstall --id <id> -e`
Uninstall by exact ID

`winget uninstall <id> -s winget`
Avoid the Store agreement prompt

UNINSTALL AND REPAIR (cont.)

`winget repair <id>`
Repair a broken install

PINS

`winget pin add <id>`
Skip in upgrade --all

`winget pin add <id> --blocking`
Block every upgrade

`winget pin add <id> -v 1.2.*`
Gate to a version range
Wildcard only as the last version part.

`winget pin list`
Show current pins

`winget pin remove <id>`
Remove a pin

A plain pin still allows winget upgrade <id>. Blocking and gating pins need --force or removal. Apps may still self-update.

EXPORT, IMPORT, CONFIGURE

`winget export -o <file.json>`
Write installed apps to JSON

`winget export -o <file.json> --include-versions`
Record exact versions too

`winget import -i <file.json>`
Install everything in the file

`winget import -i <file.json> --ignore-unavailable`
Skip packages that are not found

`winget import -i <file.json> --no-upgrade`
Leave installed apps alone

`winget configure -f <file.winget>`
Apply a configuration file
Review the file first: it can run any DSC resource.

`winget configure test -f <file>`
Check the system against it

SOURCES

`winget source list`
Show configured sources

`winget source update`
Refresh all source indexes

SOURCES (cont.)

`winget source add -n <name> -a <url>`
Add a source (admin)

`winget source reset --force`
Restore the default sources

Defaults: winget (community repository), msstore (Microsoft Store) and winget-font. winget-font is explicit, so target it with -s winget-font.

SETTINGS AND DIAGNOSTICS

`winget --version`
Print the client version

`winget --info`
Version, paths, group policies

`winget settings`
Open settings.json in an editor

`--verbose-logs`
Any command: write a verbose log

`--disable-interactivity`
Any command: never prompt

`--accept-source-agreements`
Accept source terms without a prompt

`--accept-package-agreements`
Accept package licences without a prompt

RECIPES

Unattended upgrade of everything

```
winget source update
winget upgrade --all --silent `
  --include-unknown `
  --accept-package-agreements `
  --accept-source-agreements
```

Move apps to a new PC

```
winget export -o apps.json
# on the new machine
winget import -i apps.json `
  --ignore-unavailable `
  --accept-package-agreements
```

Script a dev box

```
winget install -h `
  Git.Git `
  Microsoft.PowerShell `
  Microsoft.VisualStudioCode
# scripts: --disable-interactivity
```

Hold one app back

```
winget pin add <id> --blocking
winget upgrade --all  # skips it
winget pin list
winget pin remove <id>
```