Fedora

- Bazzite
- Paketmanager in Fedora DNF

Bazzite

Update über das Terminal

1. **Öffne das Terminal**: Du kannst das Terminal entweder über das Menü oder durch Drücken von `Ctrl + Alt + T` öffnen.
2. **Führe den Update-Befehl aus**: Gib den folgenden Befehl ein und drücke `Enter`:

```
sudo ujust update
```

- Dies aktualisiert die Liste der verfügbaren Pakete.
- **Führe das Upgrade durch**: Nach dem Update-Befehl, um die installierten Pakete zu aktualisieren, benutze:

```
sudo ujust upgrade
```

- Bestätige den Vorgang, wenn du dazu aufgefordert wirst.
- **Neustart**: Nach dem Abschluss des Updates ist es meist eine gute Idee, das System neu zu starten, um sicherzustellen, dass alle Änderungen wirksam werden.

```
cat /etc/os-release
```

Anzeige der Installierten Version von bazzite

Typische Befehle für Bazzite Linux (Fedora Silverblue-basiert)

- `rpm-ostree install <paket>` – Pakete systemweit hinzufügen (Transaktion via rpm-ostree).
- `rpm-ostree uninstall <paket>` – Installierte Pakete wieder entfernen.
- `rpm-ostree upgrade` – Das gesamte System auf die neueste OS-Version aktualisieren.
- `rpm-ostree rollback` – Zum vorherigen System-Snapshot zurückkehren.
- `rpm-ostree status` – Aktuellen und vorherigen Deployments anzeigen.
- `rpm-ostree rebase <ref>` – Auf einen anderen OS-Stream (z. B. `fedora/38/x86_64/silverblue`) umschalten.
- `flatpak install <remote> <app>` – Anwendungen per Flatpak hinzufügen.
- `flatpak uninstall <app>` – Flatpak-Anwendungen entfernen.
- `flatpak update` – Alle installierten Flatpaks aktualisieren.
- `toolbox create` – Eine mutable Toolbox-Umgebung (Podman-Container) anlegen.
- `toolbox enter` – In die erstellte Toolbox-Shell wechseln.
- `podman run ...` – Container direkt mit Podman starten (z. B. für Entwicklungsumgebungen).
- `dnf install <paket>` – Nur innerhalb einer Toolbox/Container nutzbar, um klassische RPM-Pakete zu installieren.

- `systemctl reboot` / `systemctl poweroff` – System neu starten bzw. herunterfahren.

Paketmanager in Fedora DNF

Using the DNF software package manager

DNF is a software package manager that installs, updates, and removes packages on Fedora and is the successor to YUM (Yellow-Dog Updater Modified). DNF makes it easy to maintain packages by automatically checking for dependencies and determines the actions required to install packages. This method eliminates the need to manually install or update the package, and its dependencies, using the `rpm` command. DNF is now the default software package management tool in Fedora.

Usage

`dnf` can be used exactly as `yum` to search, install or remove packages.

To search the repositories for a package type:

```
# dnf search packagename
```

To install the package:

```
# dnf install packagename
```

To remove a package:

```
# dnf remove packagename
```

Other common DNF commands include:

- `autoremove` - removes packages installed as dependencies that are no longer required by currently installed programs.
- `check-update` - checks for updates, but does not download or install the packages.
- `downgrade` - reverts to the previous version of a package.
- `info` - provides basic information about the package including name, version, release, and description.
- `reinstall` - reinstalls the currently installed package.
- `upgrade` - checks the repositories for newer packages and updates them.
- `exclude` - exclude a package from the transaction.

For more DNF commands refer to the man pages by typing `man dnf` at the command-line, or DNF Read The Docs

Automatic Updates

The `dnf-automatic` package is a component that allows automatic download and installation of updates. It can automatically monitor and report, via e-mail, the availability of updates or send a log about downloaded packages and installed updates.

For more information, refer to the Read the Docs: DNF-Automatic page.

System Upgrades

To update your Fedora Linux release from the command-line do:

```
sudo dnf upgrade --refresh
```

- and reboot your computer.
  Important: Do not skip this step. System updates are required to receive signing keys of higher-versioned releases, and they often fix problems related to the upgrade process.
- Download the updated packages:

```
sudo dnf system-upgrade download --releasever=43
```

Change the `--releasever=` number if you want to upgrade to a different release. Most people will want to upgrade to the latest stable release, which is `43`, but in some cases, such as when you're currently running an older release than `42`, you may want to upgrade just to Fedora Linux `42`. System upgrade is only officially supported and tested over 2 releases at most (e.g. from `41` to `43`). If you need to upgrade over more releases, it is recommended to do it in several smaller steps (read more).

Language Support Using DNF

DNF can be used to install or remove Language Support. A detailed description with a list of available languages can be found on Language Support Using Dnf page.

Plugins

The core DNF functionality can be extended with plugins. There are officially supported Core DNF plugins and also third-party Extras DNF Plugins. To install them, run

```
# dnf install dnf-plugins-core-PLUGIN_NAME
```

or

```
# dnf install dnf-plugins-extras-PLUGIN_NAME
```

Excluding Packages From Transactions

Sometimes it is useful to ignore specific packages from transactions, such as updates. One such case, for example, could be when an update includes a regression or a bug. DNF allows you to exclude a package from the transaction:

- using the command line

```
sudo dnf upgrade --exclude=packagename
```

- using its configuration files

You can add a line to `/etc/dnf/dnf.conf` to exclude packages:

```
excludepkgs=packagename
```

This can also be added to the specific repository configuration files in `/etc/yum.repos.d/`. Globs may be used here to list multiple packages, and each specification must be separated by a comma. If you have used this configuration, you can disable it in individual DNF commands using using the `--disableexcludes` command line switch.

If you use a GUI update application which does not allow you to specify packages to exclude when they run, this method can be used.

Using the DNF Versionlock plugin

You can also use the DNF `versionlock` plugin to limit the packages that are included in a transaction. It allows you to list what versions of particular packages should be considered in a transaction. All other versions of the specified packages will be ignored. The plugin is part of `dnf-plugins-core` package and can be installed using the command below:

```
sudo dnf install 'dnf-command(versionlock)'
```

To lock the currently installed version of a package, use:

```
sudo dnf versionlock add package
```

To remove the version lock, use:

```
sudo dnf versionlock delete package
```

The `list` command can be used to list all locked packages, while the `clear` command will delete all locked entries.

References

1. DNF Command Reference
2. DNF wiki
3. DNF VersionLock