YUM Package Manager Cheatsheet

A comprehensive cheat sheet for the YUM package manager, covering essential commands, options, and troubleshooting tips for managing software packages on RPM-based Linux distributions.

Basic Commands

Package Information

`yum info <package>`	Displays detailed information about a specific package, including its version, release, size, and description. **Example:** `yum info httpd`
`yum list installed`	Lists all installed packages on the system. **Example:** `yum list installed`
`yum list available`	Lists all packages available for installation from the configured repositories. **Example:** `yum list available`
`yum list updates`	Lists available updates for installed packages. **Example:** `yum list updates`
`yum search <keyword>`	Searches for packages based on a keyword in the package name or description. **Example:** `yum search security`
`yum provides <file>`	Finds which package provides a specific file. **Example:** `yum provides /usr/bin/ls`

Package Management

`yum install <package>`	Installs the specified package and its dependencies. **Example:** `yum install vim`
`yum update <package>`	Updates the specified package to the latest available version. **Example:** `yum update firefox`
`yum update`	Updates all installed packages to the latest available versions. **Example:** `yum update`
`yum remove <package>`	Removes the specified package. **Example:** `yum remove thunderbird`
`yum autoremove`	Removes orphaned dependencies that are no longer required. **Example:** `yum autoremove`
`yum history`	Displays the history of YUM transactions. **Example:** `yum history`

Repository Management

Repository Configuration

Repository configuration files are located in `/etc/yum.repos.d/`. Each `.repo` file defines a repository.

Example:
`/etc/yum.repos.d/epel.repo`

Key attributes in a `.repo` file:

- `[repositoryid]` - Unique ID for the repository.
- `name` - Human-readable name for the repository.
- `baseurl` - URL where package data is located.
- `enabled` - Whether the repository is enabled (1) or disabled (0).
- `gpgcheck` - Whether GPG signature checking is enabled (1) or disabled (0).
- `gpgkey` - URL of the GPG key for verifying packages.

Repository Commands

`yum repolist`	Lists all enabled repositories. **Example:** `yum repolist`
`yum repoinfo <repo>`	Displays information about a specific repository. **Example:** `yum repoinfo epel`
`yum --enablerepo=<repo> install <package>`	Enables a specific repository for a single command. **Example:** `yum --enablerepo=epel install htop`
`yum --disablerepo= <repo> update`	Disables a specific repository for a single command. **Example:** `yum --disablerepo=epel update`

Managing Repository Priorities

You can set priorities for repositories to prefer packages from certain repositories over others. This is done by installing the `yum-plugin-priorities` package and adding a `priority` option to the `.repo` file.

Example:
`priority=1` (Higher priority)

Advanced Usage

Package Groups

`yum groupinstall <group>`	Installs a group of related packages. **Example:** `yum groupinstall "Development Tools"`
`yum groupupdate <group>`	Updates packages within a group. **Example:** `yum groupupdate "Web Server"`
`yum grouplist`	Lists available package groups. **Example:** `yum grouplist`
`yum groupremove <group>`	Removes a group of packages. **Example:** `yum groupremove "Legacy UNIX Compatibility"`

Downgrading Packages

`yum downgrade <package>`	Downgrades a package to the previous version. **Example:** `yum downgrade httpd`
`package-cleanup --oldkernels --count=2`	This command, often used in conjunction with YUM, removes old kernel packages, keeping only the specified number of latest kernels. This helps to free up disk space. **Example:** Keeping the two latest kernel versions: `package-cleanup --oldkernels --count=2`

Other Useful Options

`yum clean all` - Clears all cached package data.
`yum clean packages` - Cleans only package files.
`yum clean headers` - Cleans only package headers.

`yum check` - Checks the RPM database for problems.

`yum reinstall <package>` - Reinstalls the specified package.

Example:
`yum reinstall bash`

Troubleshooting

Common Issues

Problem: Package not found. **Solution:** Ensure the repository containing the package is enabled. Use `yum repolist` to check and `--enablerepo` to enable it for the install command.
Problem: Dependency issues. **Solution:** YUM usually resolves dependencies automatically. If issues persist, try `yum install yum-utils` followed by `package-cleanup --dupes` and `package-cleanup --problems`.
Problem: GPG key errors. **Solution:** Import the correct GPG key for the repository. Find the key URL in the repo file and use `rpm --import <keyurl>`.

YUM History

`yum history list`	Lists all YUM transactions with their IDs. **Example:** `yum history list`
`yum history info <id>`	Shows detailed information about a specific transaction. **Example:** `yum history info 5`
`yum history undo <id>`	Attempts to undo a specific transaction. **Example:** `yum history undo 7`
`yum history redo <id>`	Attempts to redo a specific transaction. **Example:** `yum history redo 8`