Watchexec Cheatsheet

A comprehensive cheat sheet for Watchexec, a tool that executes commands on file changes. Includes installation, basic usage, advanced options, and practical examples.

Installation & Basic Usage

Installation

Install Watchexec using your preferred package manager.

macOS:

```
brew install watchexec
```

Rust (Cargo):

```
cargo install watchexec
```

Linux/Windows:
Download pre-built binaries from the GitHub releases page.

Basic Usage

The most basic use case: running a command when any file in the current directory changes.

Example:

```
watchexec echo 'File changed!'
```

This will print 'File changed!' to the console every time a file in the current directory is modified.

Specifying the command to run

```
watchexec npm test
```

Runs `npm test` whenever a file changes in the current directory.

Watching Specific Files/Directories

You can tell Watchexec to only watch specific files or directories using the `--watch` or `-w` flag.

Example:

```
watchexec -w src -w tests npm test
```

This command watches the `src` and `tests` directories, and runs `npm test` when any file within those directories changes.

Advanced Options & Configuration

Filtering File Changes

Watchexec allows you to filter which file changes trigger a command execution using extensions or ignore patterns.

Example (Extensions):

```
watchexec --exts js,jsx,ts,tsx npm run
build
```

This command only triggers the `npm run build` command when a file with a `.js`, `.jsx`, `.ts`, or `.tsx` extension is modified.

Example (Ignore Patterns):

```
watchexec --ignore 'node_modules/*' --
ignore 'dist/*' npm run dev
```

This command ignores changes within the `node_modules` and `dist` directories, preventing unnecessary command executions.

Controlling Command Execution

You can control how Watchexec executes your commands using flags like `--restart` and `--debounce`.

Example (Restart):

```
watchexec --restart npm start
```

If `npm start` is already running, Watchexec will restart the process on file change. If not specified, watchexec will run the command in parallel which might not be desirable.

Example (Debounce):

```
watchexec --debounce 500 npm run test
```

This command debounces the execution of `npm run test` by 500 milliseconds. If multiple file changes occur within that time, the command will only be executed once after the last change.

Signal Handling

Use `--signal` or `-s` to specify which signal should be used to terminate the previously running process.

Example:

```
watchexec --signal SIGTERM npm start
```

This will send `SIGTERM` to the `npm start` process before restarting it.

Practical Examples

Web Development Workflow

Automatically rebuild and reload your web application on code changes.

Example (React):

```
watchexec -w src --exts
js,jsx,ts,tsx,css,scss npm run start
```

Watches the `src` directory for changes in JavaScript, TypeScript, CSS, and SCSS files, then restarts the development server.

Backend Development Workflow

Automatically restart your backend server on code changes.

Example (Node.js):

```
watchexec -w src --exts js,ts --restart
node server.js
```

Watches the `src` directory for changes in JavaScript and TypeScript files, then restarts the Node.js server.

Running Tests

Automatically run your tests when source files change.

Example (Jest):

```
watchexec -w src -w test --exts
js,jsx,ts,tsx npm test
```

Watches the `src` and `test` directories for changes in JavaScript and TypeScript files, then runs the Jest test suite.

Troubleshooting & Tips

Common Issues

Too many restarts:
If your command is triggered too frequently, use the `--debounce` flag to limit the execution rate.

Ignoring files:
Ensure your `--ignore` patterns are correctly specified. Use glob patterns for directories (e.g., `node_modules/*`).

Tips and Tricks

Combine with other tools:
Watchexec works well with other build tools and task runners like Make, Gulp, or Webpack.

Use environment variables:
Pass environment variables to your command using the standard syntax (e.g., `NODE_ENV=development watchexec npm start`).

Configuration Files

For complex configurations, consider using a Watchexec configuration file (e.g., `watchexec.toml`).

Example `watchexec.toml`:

```
[commands]
  main = "npm run dev"

[watch]
  paths = ["src", "public"]
  extensions = ["js", "jsx", "ts",
"tsx", "html", "css", "scss"]

[ignore]
  globs = ["node_modules/*", "dist/*"]
```

Run with `watchexec --config watchexec.toml`.