Dezyne VS Code Extension for Docker

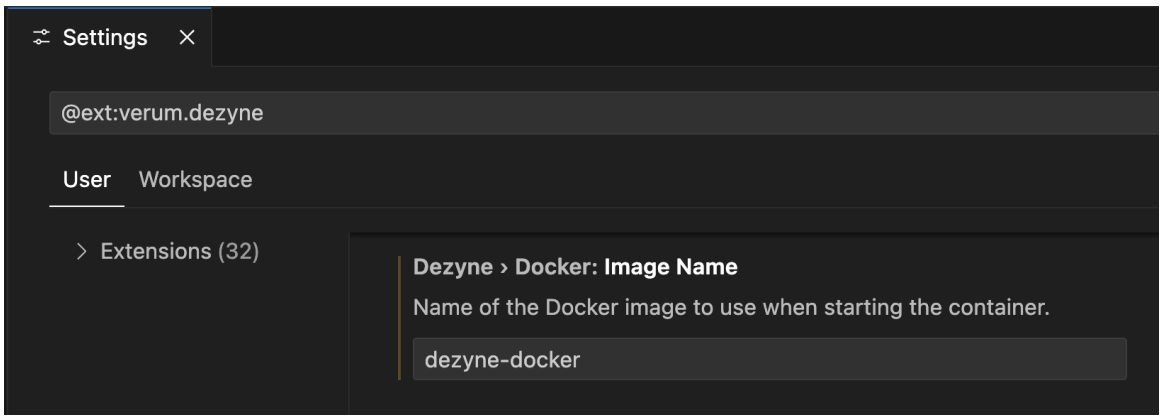
This is a custom version of Dezyne VS Code extension designed to work seamlessly with the Verum Dezyne Docker container.

Quickstart

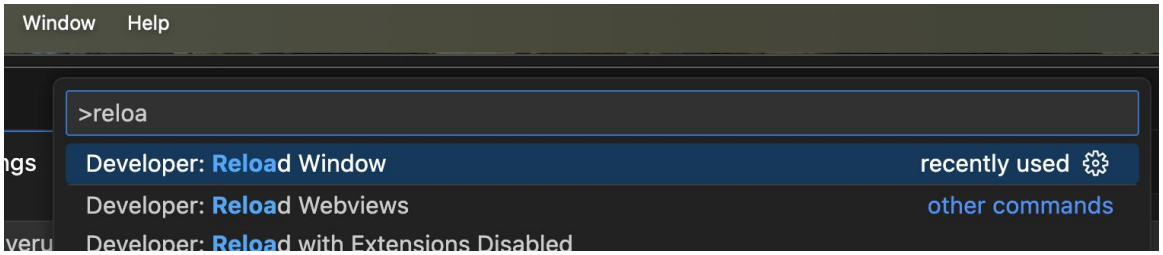
1. Install Docker Desktop
2. Load dezyne-docker.tar(docker load -i dezyne-docker.tar) or build the Docker image from Dockerfiles.zip(docker build -t dezyne-docker .)
3. Download and install the dezyne-1.6.1.vsix extension file
4. Open any .dzn file in VS Code
5. The Dezyne extension will automatically start the Docker container and map your opened directory to the container
6. Start using Dezyne!
7. Keep in mind dezyne has access only to mapped directory, so make sure .dzn files are in the active VS code directory.

Troubleshooting

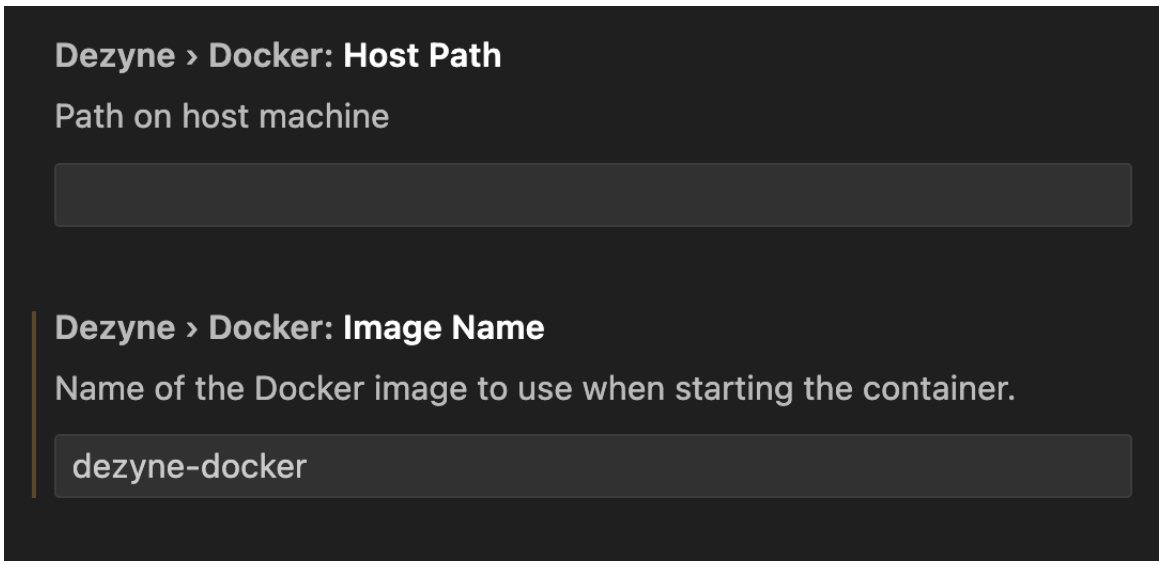
1. **Docker Image Name**: If your Docker image has a different name than "dezyne-docker", make sure to set the matching name in the Dezyne extension settings



2. **Settings Changes**: After changing any settings, reload the VS Code window (Shift+Cmd+P or Shift+Ctrl+P, then type "Reload Window")



3. **General Issues**: Reloading the window restarts extensions and the Dezyne Docker container, which can resolve many common problems
4. **Automatic Mounting**: The Dezyne extension automatically mounts the top level directory of your opened folder if both Container Path and Host Path settings are empty



5. **Path Requirements**: If custom Host and Container paths are different, the Language Server Protocol (LSP) will not work correctly, and parse error navigation will be disabled. The extension uses POSIX interfaces for both Linux and macOS, so the preferred option is to use identical absolute paths for both host and container

Docker Container Details

The Docker container includes the Dezyne toolset and uses Nginx to handle port mapping, allowing running Verum Dezyne daemon without port conflicts. The automatic volume mapping ensures your Dezyne files are accessible from both VS Code and the Docker container. In case of port change, make sure to update ports in the nginx configuration.