

Developing on UDS

! INFO

- If this is the first time you are developing on the Universal Design System repository
 - You will need [Github access](#)
 - You may need to [request write access](#) for you and your team
- If you are looking to contribute to UDS, please follow the instructions to [create a new community component](#)

Before your start

- [Learn about the Universal Design System](#) to better understand the project
- [Read the developer docs](#) for more technical information
- When contributing to the repository
 - [Check the issue list](#) to see that your request doesn't already exist
 - [Raise a GitHub issue](#) to propose and initiate discussion
 - Start a discussion thread in the [#ds-support](#) Slack channel
- UDS is a monorepo, head over to the [repository on GitHub](#) to view the source code
 - Follow the instructions for local development in the main repository [README.md](#)
- The project uses [Architectural Decision Records \(ADRs\)](#) to explain the technical decisions taken to date in the creation of UDS

Set up your local development environment

Clone the repo.

```
# clone the repo
git clone git@github.com:telus/universal-design-system.git

# cd into it
cd universal-design-system
```

Make sure you're using versions of node and npm matching those given in the root package.json. Then, at the root of the repo, run:

```
npm install
```

The command will install all dependencies across all workspaces, build all packages with a build step, and setup the git hooks for development.

Start your local dev tools

Because the different packages are tightly coupled to each other, we need to watch for code changes in the various packages and rebuild local versions of the distributions so that other packages can consume these changes.

You can use `storybook` as a tool for viewing, editing, and building components in isolation in your local environment.

You can also spin up a local version of the documentation site to help make changes to the docs.

To do all this, run the following scripts from the monorepo's root directory in separate terminal windows and keep them running while you make your changes.

```
# watch for changes and rebuild packages
npm run dev

# run storybook
npm run storybook

# run documentation site
npm run docs
```

Create a local branch and make your changes

Create a local development branch off of `main` to make your changes and commit them. Before you push your branch up for review, you need to add change files that describe your changes and inform what semantic versioning bump they require. You can do this by running:

```
npm run beachball
```

Submit a PR

After pushing up your local branch, submit a PR to the `main` branch of the `universal-design-system` repo on GitHub and wait for a member of the Universal team to approve your request.

Note that packages are not automatically published to `npm` after your PR is merged. Releases are triggered manually by the Universal team, so [get in touch with us](#) if you would like to expedite a new release.

[View the full guide on how to submit a PR](#)

 [Edit this page](#)