Rolling back versions

> CAUTION
>
> UDS is served via NPM packages, depending on how your product has the **package.json** configured, updates and new UDS releases might affect and create unwanted inconsistencies for specific use cases.

Our releases follow the semantic versioning specification to control the release of new features and fixes in the components.

Keeping all the packages updated is the best way to access everything UDS has to offer and solve existing bugs, however a new version of a UDS package might break your application for multiple reasons (use case was not covered, dependency with external packages, environment specifics, etc.)

Before upgrading a UDS package dependency

Before upgrading to a new version of a UDS package, please assess the changelogs to see if there are any changes that may be impacting your application.

After reviewing the changes, please test any component and utility you have via regression testing.

If you identify any unusual behavior, please reach out in #DS-support in slack or review the contribution guide for requests and contributions.

What to do if an issue is identified?

> CAUTION
>
> If an issue is identified, communicate it to the core team and roll back to the previous version while a fix is being worked on. Once the core team has a proper understanding of the problem and context, the issue and action will be communicated on the #DS-support channel providing a timeline and potential release date of a new version containing the fix.

How to upgrade a UDS Package dependency?

1. Ensure your dependency covers the new released version
2. Upgrade the packages consumed. Upgrade guide
3. Test the new package in your acceptance environment

How to Downgrade a UDS Package dependency?

To downgrade an npm package, run the `npm install <package>@<version>` command. You need to provide the version you want to install with the `@<version>` syntax.

For example, UDS releases version 1.22 of the `components-base` package and that introduces breaking changes from version 1.21.

When you need to downgrade `components-base` from version 1.22 to version 1.21, use the command below:

```
npm install @telus-uds/components-base@1.21.0
```

npm will overwrite the package already installed in your node_modules/ folder.

To install an exact version, you need to explicitly define the major, minor, and patch version like this:

```
npm install @telus-uds/components-base@1.21.0
```

The above command installs `@telus-uds/components-base` exactly version 1.21.0.

Alternatively, you should be able to set a specific version of you dependency in `package.json` and just run `npm install`.

You can also do the same with globally installed packages. Just add the -g option to the npm install command as follows:

```
npm install -g @telus-uds/components-base@1.21.0
```

The above command will overwrite the corresponding package installed at the global level.

Edit this page