

Eclipse Scout

Release Notes

Scout Team

Version 22.0

Table of Contents

About This Release	2
Service Releases	2
Obtaining the Latest Version	2
Demo Applications	3
Mobile Improvements	4
Default Detail Form and Tile Overview	4
More Information in Outline	5
New Style for Views	6
Fullscreen Dialogs	7
Wizard: Improved Progress Field	7
New MobilePopup	7
New Tablet Transformation: Auto Hide Navigation	7
New ESLint Settings	8
Custom Webpack Config	9
Automatic Script Chunks	10
New Widgets	11
Breadcrumb Bar Field	11
Chart	11
Widget Enhancements	15
Table: New Property 'compact'	15
Menu: New Property 'textPosition'	15
Form	15
Widget	16
TabBox: Support for LabelVisible	16
Maven master	17



Looking for something else? Visit <https://eclipsescout.github.io> for all Scout related documentation.

About This Release

The latest version of this release is: 11.0-SNAPSHOT.

You can see the [detailed change log](#) on GitHub.

Coming from an older Scout version? Check out the [Migration Guide!](#)

Service Releases

Scout 11.0 will continue to be maintained for a while and new builds may be released from time to time. Beside bug fixes, these releases may even contain some minor features.

Obtaining the Latest Version

Scout Runtime for Java

Scout Runtime artifacts for Java are distributed using Maven Central:

- [11.0-SNAPSHOT](#) on *Maven Central*
- [11.0-SNAPSHOT](#) on *mvnrepository.com*

Usage example in the parent POM of your Scout application:

```
<dependency>
  <groupId>org.eclipse.scout.rt</groupId>
  <artifactId>org.eclipse.scout.rt</artifactId>
  <version>11.0-SNAPSHOT</version>
  <type>pom</type>
  <scope>import</scope>
</dependency>
```

Scout Runtime for JavaScript

Scout Runtime artifacts for JavaScript are distributed using npm:

- [Scout Core Runtime](#)
- [All official Scout JavaScript packages](#)

IDE Tooling (Scout SDK)

Starting with Scout 11 the IDE Tooling requires at least Java 11 to run. With the help of the SDK plugins you can still develop applications running with Java 8, but the IDE itself requires Java 11 or newer.

Scout officially supports [IntelliJ IDEA](#) and [Eclipse for Scout Developers](#).

Eclipse

You can download the complete Eclipse IDE with Scout SDK included (EPP) here:

[Eclipse for Scout Developers](#)

To install the Scout SDK into your existing Eclipse IDE, use this update site:

<http://download.eclipse.org/scout/releases/11.0/>

IntelliJ IDEA

You can download the Scout plugin for IntelliJ IDEA from the [JetBrains Plugin Repository](#) or you can use the plugins client built into IntelliJ IDEA. Please refer to the [IntelliJ Help](#) on how to install and manage plugins.

Demo Applications

The demo applications for this version can be found on the [features/version/11.0-SNAPSHOT](#) branch of our docs repository on GitHub.

If you just want to play around with them without looking at the source code, you can always use the deployed versions:

- <https://scout.bsi-software.com/contacts/>
- <https://scout.bsi-software.com/widgets/>
- <https://scout.bsi-software.com/jswidgets/>

Mobile Improvements

Default Detail Form and Tile Overview

When no page is selected, the desktop client shows the default detail form, or if not available, the tile outline overview. This feature was disabled on mobile, until now. So if you startup your mobile Scout Classic app you will probably notice the new look. If you don't like it you can always disable it by not creating a default detail form (`Outline#createDefaultDetailForm`) or by setting `Outline#getConfiguredOutlineOverviewVisible` to false.



More Information in Outline

Until now, the outline in mobile mode displayed the same summary cell as in desktop mode. Even though you could already create a custom summary cell for mobile, doing so is time-consuming. This is why we decided to implement a different strategy for mobile to compute a better summary cell. It is enabled by default so that every table page will benefit right away.



Since this new strategy is based on the new feature introduced for tables ([Table: New Property 'compact'](#)), it can be easily customized in the same way. If you don't like it you can return to the previous strategy by overriding `AbstractTable#createSummaryCellBuilder`, but why should you ;-)?

New Style for Views

The device transformer now adjusts the style of a view so that it matches the style of a dialog. Also, the view tabs won't be visible anymore. This should give a more consistent look and should make it easier to use.

Person

Alice



First name

Alice

Last name

Date of birth

11/26/1865

Gender

☐ Male

☒ Female

OK

Cancel

If you don't like that behavior, you can disable it in general or for specific forms:

```
BEANS.get(IDeviceTransformationService.class).excludeFormTransformation(this,  
MobileDeviceTransformation.USE_DIALOG_STYLE_FOR_VIEW);
```


Fullscreen Dialogs

The device transformer now sets the property `maximized` of dialogs to true so that every dialog will be displayed in full screen. See also [Support Maximized](#).

If you don't like that behavior, you can disable it in general or for specific forms:

```
BEANS.get(IDeviceTransformationService.class).excludeFormTransformation(this,
MobileDeviceTransformation.MAXIMIZE_DIALOG);
```

Wizard: Improved Progress Field

The progress visualization and step activation has been improved for smaller screens.

New MobilePopup

MobilePopup was only available on Scout JS but is now available for Scout Classic as well. A mobile popup slides in from the bottom. This is it, not more. It has already been used for search or tool forms on mobile.

New Tablet Transformation: Auto Hide Navigation

The navigation is now automatically hidden when a view is opened and shown again when the last view is closed. This was introduced because the space is very limited and the navigation mostly not needed when a view is open.

If you don't like the change you can simply disable it by disabling the transformation `TabletDeviceTransformation#AUTO_HIDE_NAVIGATION`. Check out the mobile chapter in the [Tech Doc](#) on how to do it.

New ESLint Settings

The eslint rules used by Scout have been adjusted to encourage the use of some great ES6 features. The following rules have been turned on: `no-var`, `prefer-arrow-callback`, `prefer-rest-params`, `prefer-spread`.

If you have a dependency to `@eclipse-scout/eslint-config` and you update to the latest version, your project will use the new settings, too. This also means your JavaScript code may report some new warnings. Have a look at the [Migration Guide](#) to learn how to handle them (e.g. migrating code or disabling the rules).

Custom Webpack Config

If you have a lot of themes, working on a less file can be frustrating since every theme will be built. To save time, it is now possible to control which themes should be built. To do so, create a `webpack.config.custom.js` file (which can be added to `.gitignore` because not every developer has the same preferences). Add the following content:

```
const baseConfig = require('./webpack.config');
module.exports = (env, args) => {
  args.themes = args.themes || ['default'];
  return baseConfig(env, args);
};
```

To make this work your base webpack config needs to register the themes using the function `baseConfig.addThemes`:

```
baseConfig.addThemes(config.entry, {
  themes: args.themes,
  availableThemes: ['default', 'dark'],
  generator: name => [`yourapp-theme${name}`, `./src/main/js/yourapp-
theme${name}.less`]
});
```

Automatic Script Chunks

Before Scout 11 script (JavaScript and CSS) chunks have been declared explicitly. While you have full control on the output using this explicit definition, it is often not optimal concerning performance and might require modifications when a new module or library is added to the project. Therefore starting with version 11 Scout additionally supports automatic chunk creation for your script assets.

When using this new feature (which is the default configuration now), chunks are automatically created based on size and code usage (imports). Therefore, it is no longer necessary to list every chunk manually in the html file. Instead, Scout automatically inserts the necessary script tags according to the chunks created.

Have a look at the [Migration Guide](#) to learn how to use automatic script chunks.

New Widgets

Breadcrumb Bar Field

The new breadcrumb bar widget allows to display a hierarchical structure in a horizontal style. Each breadcrumb item is clickable. The breadcrumb can automatically shrink if there is not enough place available.

[Storyboard](#) > [Folder](#) > **Child Folder**

Chart

The new chart widget allows visualizing data in several ways. It can be used inside a chart field, a chart tile or as a plain chart. The supported chart types include bar (vertical and horizontal), line, bar and line combined, pie, doughnut, bubble, polar area, radar, fulfillment, salesfunnel, speedo and venn. All charts support different color schemes and themes.

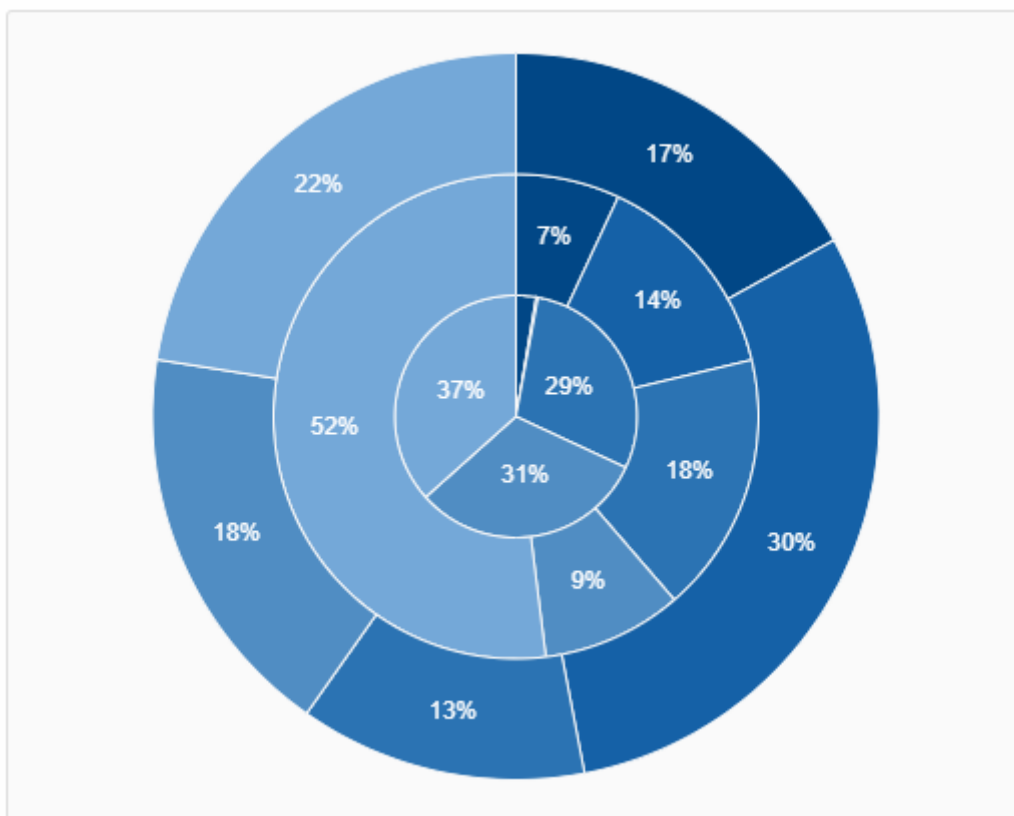


Figure 1. Pie chart in a tile using the default color scheme.



Figure 2. Combined bar line chart in a tile using the alternative inverted color scheme.

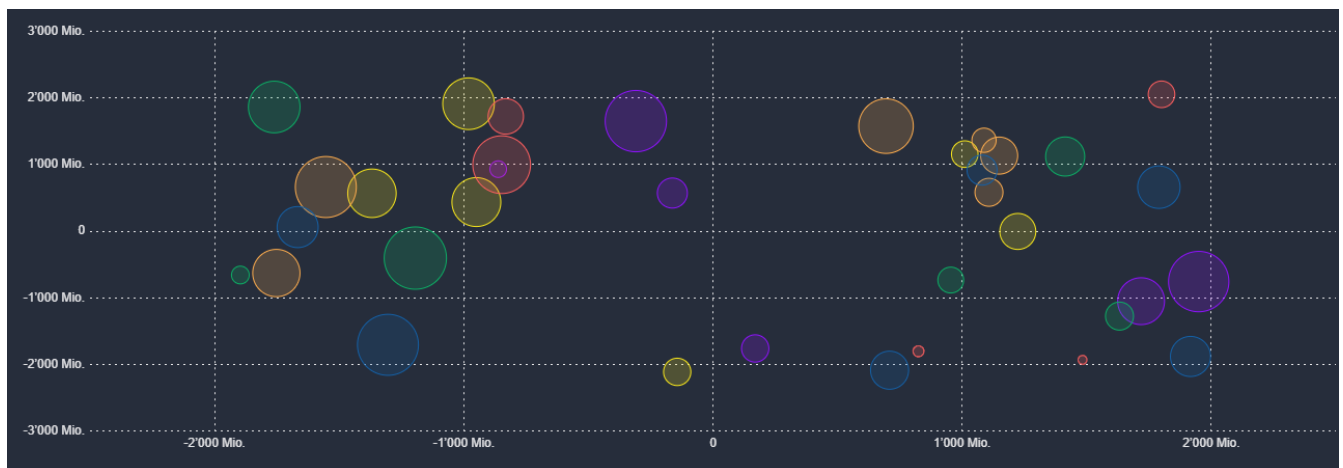


Figure 3. Bubble chart using the rainbow color scheme and the dark theme.

There are numerous styling options, e.g. tension and fill for line and radar charts, stacked bar charts or a transparent and a checkable mode.

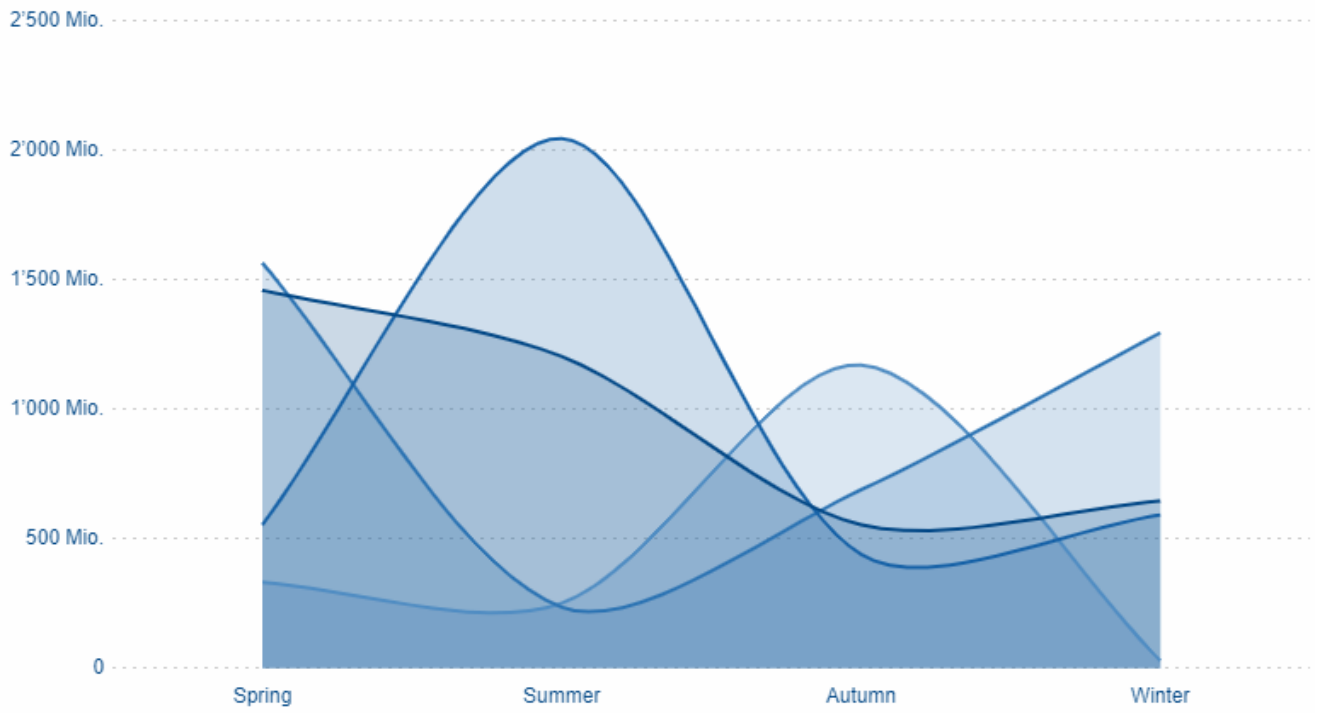


Figure 4. Tension and fill for line charts.

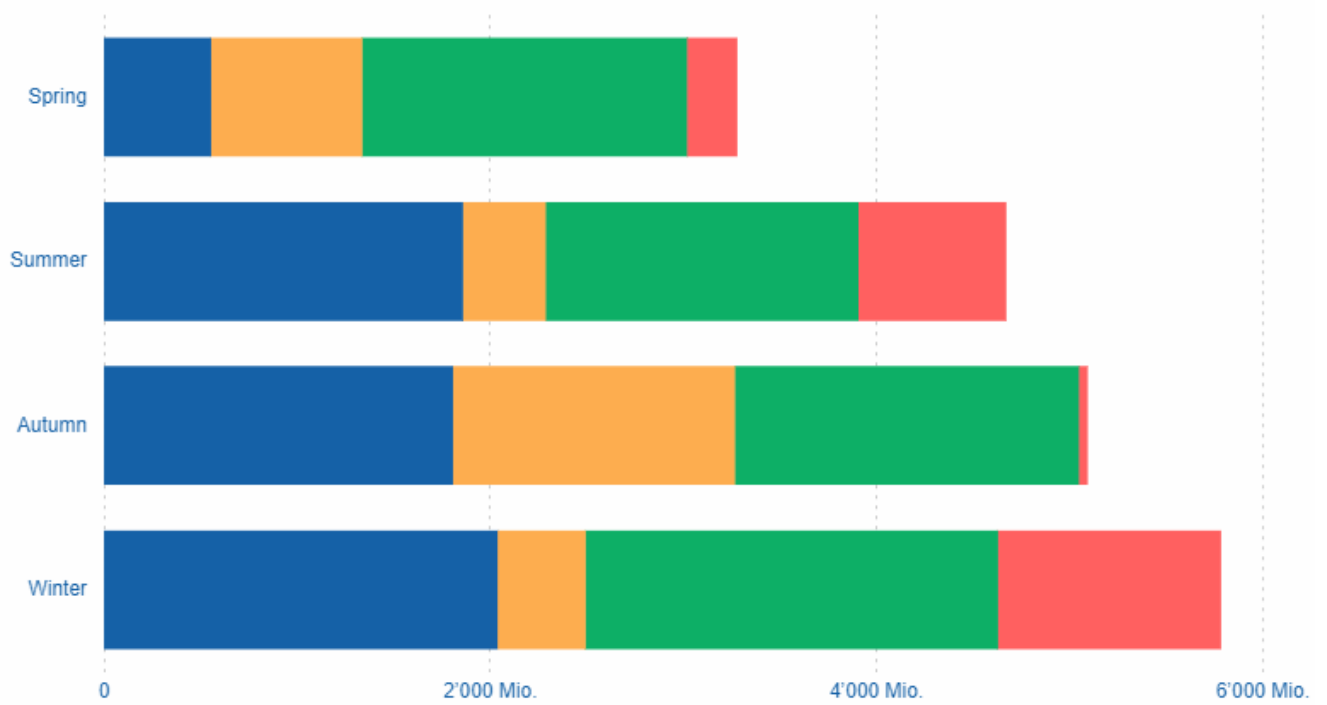


Figure 5. Stacked horizontal bar chart.

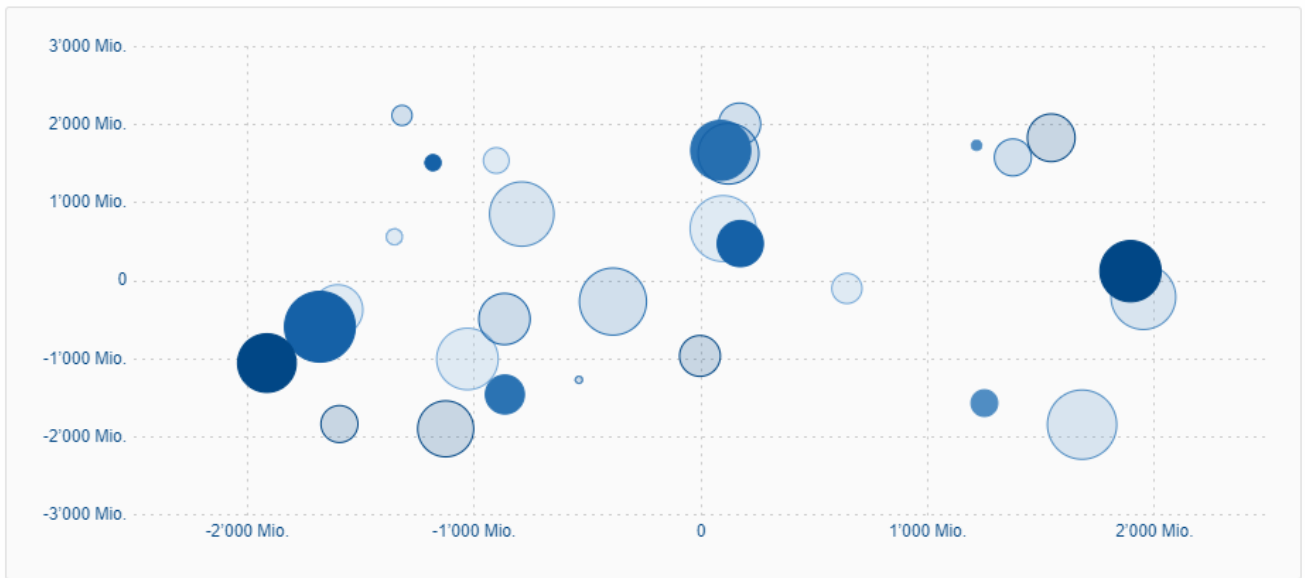


Figure 6. Checkable bubble chart.

For more information about the chart widget see [Technical Guide for Scout JS](#).

Widget Enhancements

Table: New Property 'compact'

It is now possible to set a table into compact mode. This will hide all columns and insert a new one containing the content of every other column. The content will be displayed vertically which means the cells of a row will be put below each other. This new style is perfect if the width of the table is limited (e.g. on a mobile phone).

The table uses a so called `ITableCompactHandler` which does the conversion. If you don't like how the content is arranged you can easily control the conversion by adjusting the compact handler or by writing your own one.

String Column	Date Column	Number Column	Smart Column	Boolean Column	Icon Column	Html Column
Row #0	09/25/2020	0	Albanian	<input checked="" type="checkbox"/>	☆	App Link
Row #1	09/24/2020	1	Albanian (Albania)	<input type="checkbox"/>	👤	App Link
Row #2	09/24/2020	2	Arabic	<input checked="" type="checkbox"/>	📁	App Link
Add row						



Row #0 09/25/2020 Number Column: 0 Smart Column: Albanian Boolean Column: X More
Row #1 09/24/2020 Number Column: 1 Smart Column: Albanian (Albania) Icon Column: 👤 Html Column: App Link
Row #2 09/24/2020 Number Column: 2 Smart Column: Arabic
Add row

Menu: New Property 'textPosition'

With this new property, the text can be positioned on the bottom of the icon



First menu



Second menu

Form

Support Maximized

The property `maximized` is an old property but had no effect with the current UI. With Scout 11 the property will finally be interpreted. Setting it to true on a dialog will make it use the whole screen.

New Property 'headerVisible'

With this new property you can manually control whether the header should be visible. Until now it depended on the display hint property. This could be useful when embedding the form in a popup or another widget.

Widget

ExecFocusIn, ExecFocusOut

Every widget now supports focus events. You can either use a `WidgetListener` or implement `execFocusIn` and `execFocusOut`. The currently focused element is available using `IDesktop#getFocusedElement()`.

Important: the focus tracking needs to be explicitly activated before it can be used. Just call `IDesktop#setTrackFocus(true)` to do so. And remember to deactivate it again once you don't need it anymore to not produce unnecessary requests.

New Method 'reveal'

If a widget in a scrollable container is out of view you can bring it back by using the new `reveal` method.

On JS side this method was already available, now it is available for Scout Classic apps as well.

TabBox: Support for LabelVisible

Setting the property `labelVisible` to false will now hide the tab area. Until now, the property had no effect for tab boxes.

Maven master

Some maven profiles have been removed from the maven master (e.g. `org.eclipse.scout:maven_rt_plugin_config-master`) (since 3.14.1). These profiles have been used for release management. No Scout build jobs do still use git commands through maven. Instead, any git command is issued directly (usually from a jenkins pipeline). Therefore, we cannot support these profiles anymore. If any of these profiles or configurations are still in use in your project, you may copy them from an older maven master version into your own parent maven pom.

Removed profiles

- `release.setversion`
- `release.checkin`
- `release.tag`

Removed related properties

- `master_release_pushChanges`
- `master_release_tagName`
- `master_release_checkinMessage`

Also, the default plugin configuration of `org.apache.maven.plugins:maven-release-plugin` was removed.

The following script is a template which can be used to change the project versions for a new release build without the removed profiles and direct git commands.

Listing 1. Shell script to set project version to a new version for release builds

```
# set version of maven project modules
mvn -f ${mavenParentPom} org.codehaus.mojo:versions-maven-plugin:set
-DnewVersion=${newVersion} -DgenerateBackupPoms=false
# set version of npm project packages
mvn -f ${mavenParentPom} process-sources -N -P npm-install-node,npm-install-
workspace,npm-deploy -Dmaster_node_dir=${NODE_BIN} -Dmaster_npm_release_build=true
-Dmaster_release_newVersion=${newVersion}
# add changed files to git changes
git add "**/pom.xml" "**/package.json"
```



Do you want to improve this document? Have a look at the [sources](#) on GitHub.