Eclipse Scout

Release Notes

Scout Team

Version 10.0

Table of Contents

About This Release

The Eclipse Scout 10.0 version was released as part of the Eclipse 2019-06 Simultaneous Release on June, 2019.

The latest version of this release is: 10.0.0.009_Simrel_2019_06.

You can see the detailed change log on GitHub.

Coming from an older Scout version? Check out the Migration Guide!

Service Releases

After the final simultaneous Eclipse release, there are no more Eclipse *service releases*. Scout 10.0 will continue to be maintained for a while and a new build may be released from time to time. Beside bug fixes, these releases may even contain some minor features. See the Simultaneous Release FAQ for details.

The following enhancements were made after the initial 10.0 release.

Simrel 2019-06 (10.0) — Release Expected June, 2019

⚠ The here described functionality has not yet been released and is part of an upcoming release.

(Section intentionally left blank for possible future release)

Obtaining the Latest Version

Runtime (Scout RT)

Scout RT artifacts are distributed via Maven:

- 10.0.0.009_Simrel_2019_06 on *Maven Central*
- 10.0.0.009_Simrel_2019_06 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>10.0.0.009_Simrel_2019_06</version>
    <type>pom</type>
    <scope>import</scope>
</dependency>
```

Eclipse IDE Tooling (Scout SDK)

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/10.0/10.0.0/009_Simrel_2019_06/

Demo Applications

The demo applications for this version can be found on the features/version/10.0.0.009_Simrel_2019_06 branch of our docs repository on GitHub.

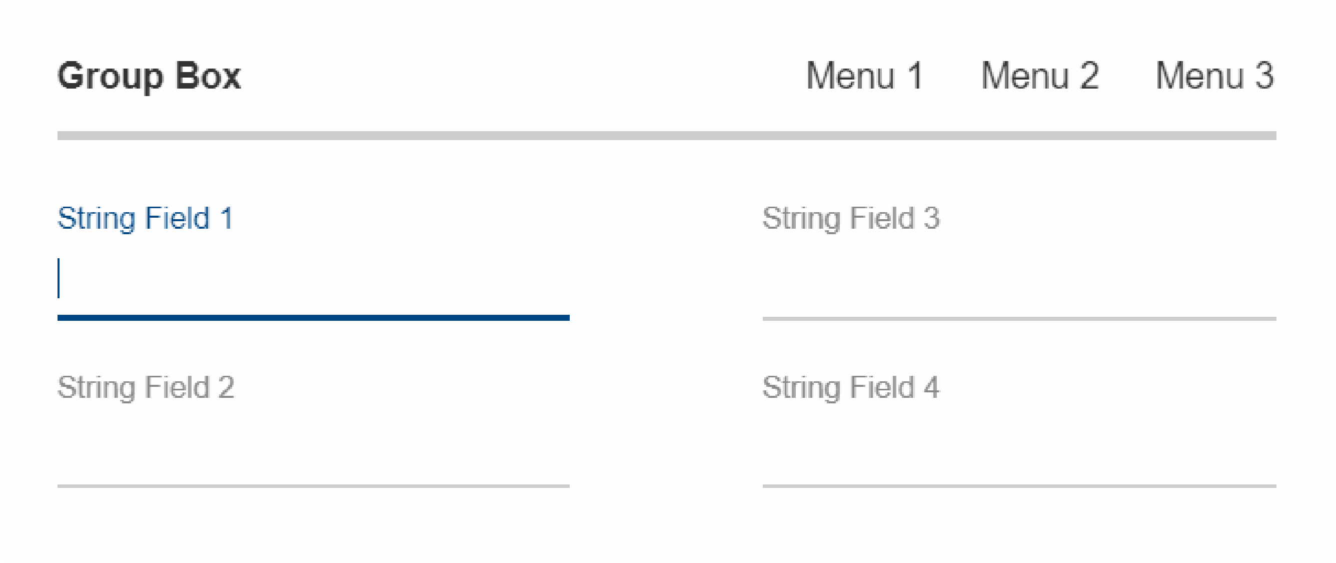
If you just want to play around with them without looking at the source code, you can always use the deployed versions:

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

Widgets

GroupBox: New MenuBar Position TITLE

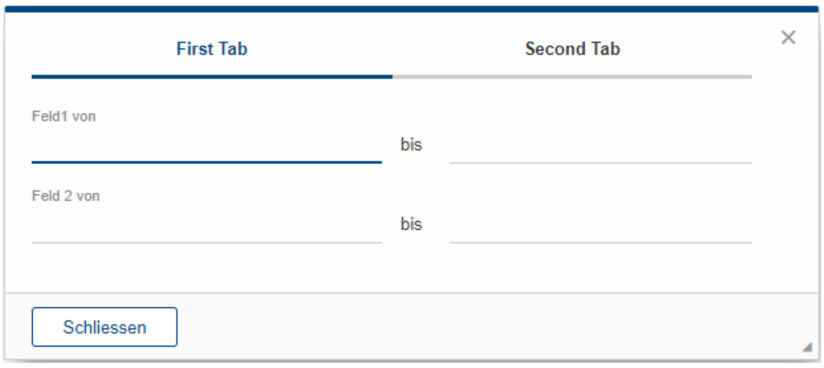
In addition to the existing menu-bar position TOP and BOTTOM, the GroupBox now supports TITLE. The menu-bar is placed in the header of the group-box right of the title-label DIV. Scout Classic: see property `menuBarPosition` in `AbstractGroupBox` and constant `IGroupBox.MENU_BAR_POSITION_TITLE`. Scout JS: see property `menuBarPosition` in `GroupBox.js` and constant `scout.GroupBox.MenuBarPosition.TITLE`.



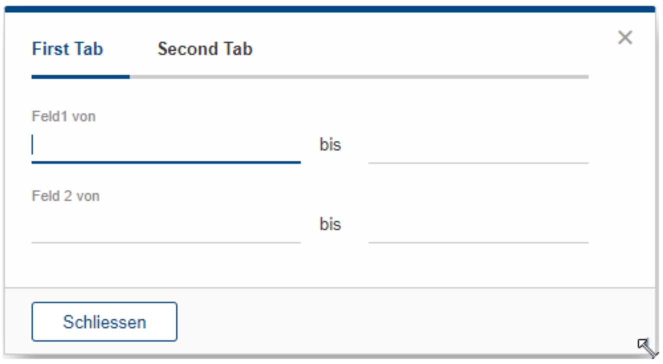
TabBox: New Tab Style

The tab box has a new property `ITabBox.PROP_TAB_AREA_STYLE` with currently two possible tab styles.

- `ITabBox.TAB_AREA_STYLE_DEFAULT`: Default appearance
- `ITabBox.TAB_AREA_STYLE_SPREAD_EVEN`: This style will spread all the tabs over the available space. When using this style, all the menu will be pushed into the ellipse menu.



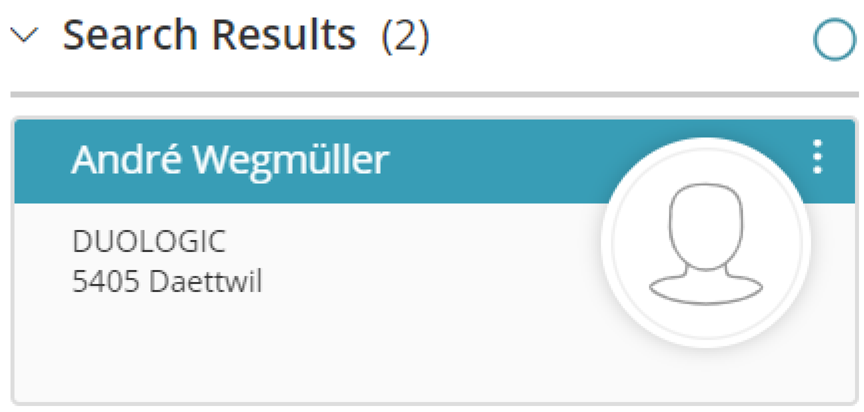
TAB_AREA_STYLE_SPREAD_EVEN



TAB_AREA_STYLE_DEFAULT

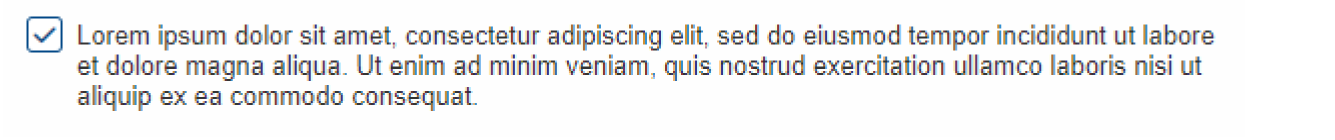
TileGrid Groups: Show Loading State per Group

Each Group in a TileGrid can now display a loading indicator in the group header individually. This is useful if each displayed group loads data from an individual data source. Scout Classic: call `AbstractGroup#setLoading(boolean)`, Scout JS: call `Group.js#setLoading(boolean)`. Note: it is still possible to set the loading state on the TileGrid, to indicate the whole grid (and not an individual group) is loading data.



RadioButton and CheckBox: Wrap Text

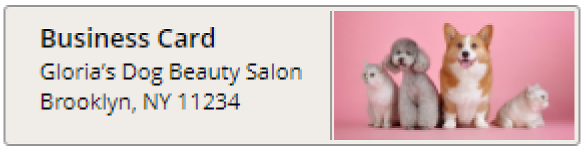
Both widgets `RadioButton` and `CheckBox` (aka BooleanField), now support the `wrapText` property. This means a radio button or a check box can have a label with a long text on multiple lines. In order to see the wrapped text, the field must have a `gridH` > 1 or set the `gridUseUiHeight` property to true.


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Button: Support for HTML and Binary Resources

The `Button` now supports the property `htmlEnabled` which allows to use HTML in the label part of the button. You can even reference binary resources in your HTML. Simply call the method `AbstractButton#setAttachments(Collection<? extends BinaryResource>)` and define a reference, say an image URL, in your label HTML:

```
<img src="binaryResource:business-card.jpg" />
```



ImageField: Support for File Upload

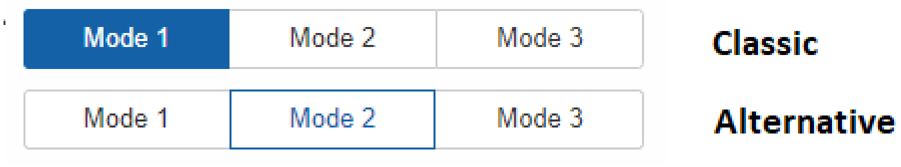
The `ImageField` has a new property `uploadEnabled`. When set to true, the field opens the native file chooser and performs a file upload.

LabelField: Support for App Links

The `LabelField` now supports app links. In order to use app links in a label field, the property `htmlEnabled` must be set to true.

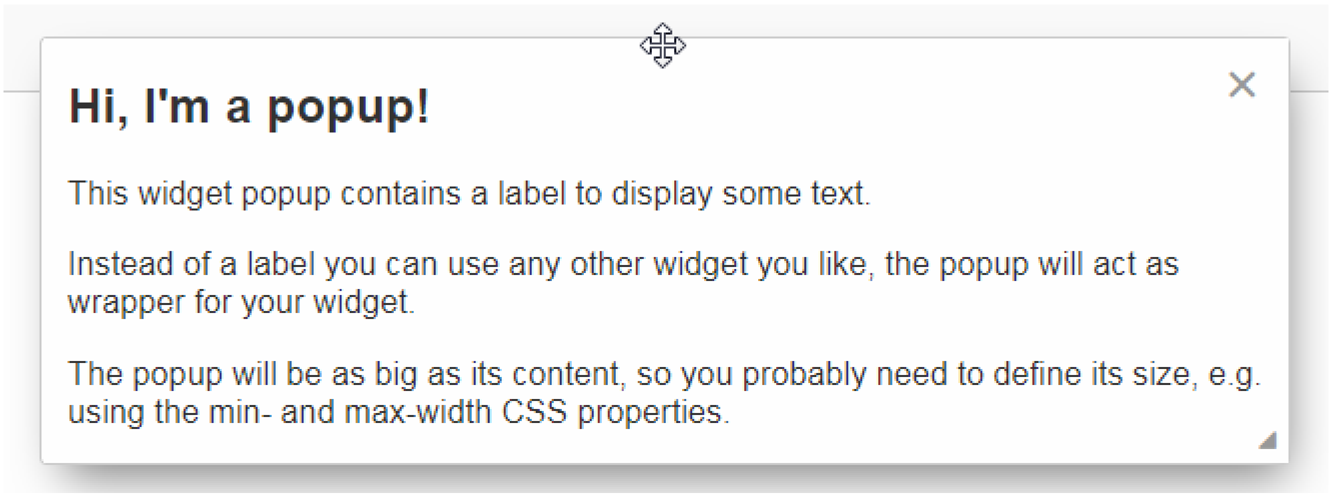
Mode Selector: Alternative Display Style

The mode selector now has a different appearance for the two field styles `IFormField.FIELD_STYLE_CLASSIC` and `IFormField.FIELD_STYLE_ALTERNATIVE`.



Popup: New Properties Closable, Movable, Resizable

The new properties allow a user to close, move or resize a popup using the typical controls he already knows by other widgets like a `Dialog`. The properties are available on the `WidgetPopup` and can even be turned on or off while the popup is open.


Hi, I'm a popup!

This widget popup contains a label to display some text.

Instead of a label you can use any other widget you like, the popup will act as wrapper for your widget.

The popup will be as big as its content, so you probably need to define its size, e.g. using the min- and max-width CSS properties.

Enabled & InheritAccessibility

The `enabled` property already existing for FormFields, Actions and other Scout elements has been moved to the top level Widget class. Therefore all widgets can now be disabled as it already was in Scout JS. Additionally Scout Classic widgets also support dynamic enabled/disabled dimensions on all widgets now as it already existed for FormFields.

The `inheritAccessibility` property that already existed for Actions (e.g. Menus) can now be used on all widgets. This is especially interesting for FormFields for which it was necessary to do that using multiple setEnabled() calls until now.

Consistent Parent

Lots of Scout widgets like `Actions`, `FormFields`, `Tiles` or `Tables` can have a parent element. Until now there have been several methods to access parent elements (e.g. `IActionNode#getParent()`, `ITile#getContainer()`, `IFormField#getParentField`, etc.) and it was only available on the specific element (e.g. for a Table). Now there is one `getParent()` method for all widgets. Furthermore visit methods have been added to traverse up the parent hierarchy.

PropertyChange Shortcut (JS)

It is now possible to listen for specific property changes rather than listening to all property change events and then checking for the right property manually. This can be done using the new notation `propertyChange:propertyName`, see the example below.

Listing 1. Listen for specific property changes

```
field.on('propertyChange:value', function(event) {
  // This listener is only executed when the 'value' property changes
  console.log('Property ' + event.propertyName + ' changed from ' + event.oldValue + '
to ' + event.newValue);
});
field.setValue('New Value.');
```

Do you want to improve this document? Have a look at the sources on GitHub.