Eclipse Scout

Release Notes

Scout Team

Version 9.0

Table of Contents

About This Release

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

The Eclipse Scout 9.0 version is planned to be part of the Eclipse 2019-03 Simultaneous Release (release schedule). The release is scheduled for March 2019.

The latest version of this release is: (not yet released)

You can see the detailed change log on GitHub.

Service Releases

After Eclipse Photon, there no longer are Eclipse *service releases* (see the Simultaneous Release Cycle FAQ for details).

There may be additional maintenance builds of Scout 9.0, but no plans have been finalized yet. Beside bugfixes, these maintenance builds may even contain some minor features. See the following notes for details.

Simrel 2018-06 (9.0.x) Release Expected in June 2018

(Section intentionally left blank for possible future release)

Obtaining the Latest Version

Runtime (Scout RT)

Scout RT artifacts are distributed via Maven:

- 8.0.0.021_Simrel_2018_09 on *Maven Central*
- 8.0.0.021_Simrel_2018_09 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>8.0.0.021_Simrel_2018_09</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 Photon

To install the Scout SDK into your existing Eclipse IDE, use this update site: http://download.eclipse.org/scout/releases/8.0/8.0.0/018_Simrel_2018_09_M3/

Demo Applications

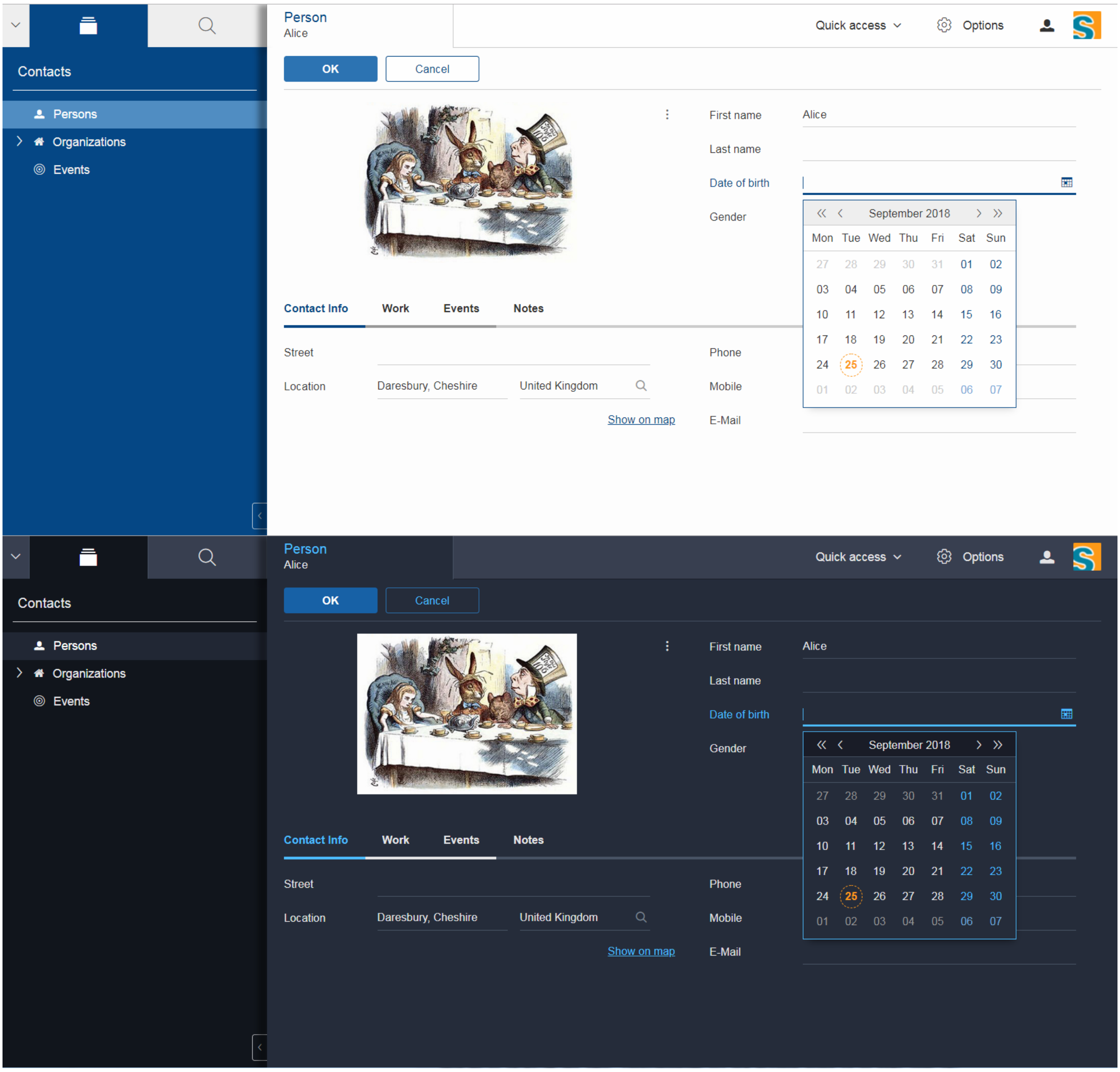
The demo applications for this version can be found on the features/version/8.0.0.021_Simrel_2018_09 branch of our docs repository on GitHub.

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/

Dark Theme

Enter the dark side… and use the new dark theme of Scout!



First name: Alice
Last name:
Date of birth:
Gender:
Street:
Location: Daresbury, Cheshire | United Kingdom
Phone:
Mobile:
E-Mail:

September 2018

Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	31	01	02
03	04	05	06	07	08	09
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
01	02	03	04	05	06	07

Figure 1. Dark Theme

Scout now provides a dark theme in addition to the default theme. You can either activate it by default by setting the property `scout.ui.theme` to `dark` in the `config.properties`, or let the user choose what he likes more.

Disabling Close- & Cancel-Buttons

Until Scout 8.0 a Close- or Cancel-Button ignored the enabled-granted property to ensure the form can be closed even if the full form has been disabled using `setEnabledGranted(false)`. This was confusing because the same convenience was not available for all other enabled dimensions.

Since Scout 9.0 Close- and Cancel-Buttons can be disabled like any other form field. But one special handling is still present: The method `isEnabledIncludingParents` ignores the enabled state of the parents and always returns the state of the button only.

So if a Form or GroupBox is disabled using `setEnabled(false)` or `setEnabledGranted(false)` or any other dimension, the full form gets disabled except the Close- and Cancel-Buttons. As soon as the button is disabled explicitly (e.g. by calling `setEnabled(false)` on the button itself or by propagating to the button using `setEnabled(false, false, true)` on a parent composite) it will be disabled and the form cannot be closed anymore.

New Servlet Filters to create a Scout run context

Before Scout 9 the `ServerRunContextFilter` was used to create Scout server contexts for REST APIs. This filter used a user based TTL cache that was not bound to the HTTP session.

Starting with Scout 9 there are two new filters available:

- `HttpRunContextFilter`: Creates a Scout run-context without HTTP- and server sessions for stateless REST backends. It supports subject, correlationId, locale, transaction, etc.
- `HttpServerRunContextFilter`: Creates a Scout server-run-context that additionally has a user-agent and an optional Scout server session.

Improved Scrollbar Usability

The layout structure of the scrollbar comes now with an additional div, and the positioning of the scollbar uses now padding instead of margin.

With this change, the usability of the scout scrollbar has improved. The thumb is now easier to catch, especially when positioned at the very edge of the screen.

Design change for WizardProgressField

The wizard progress has a new design.

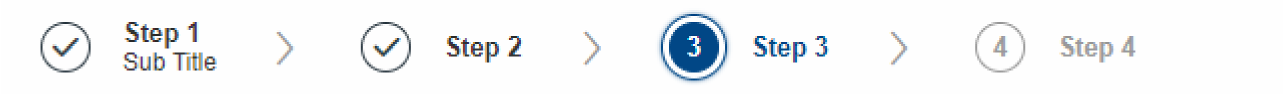


Figure 2. Wizard Progress

Wizard steps can now be marked as finished, in this case they will be displayed with a check mark icon in the wizard progress.

New widget Mode Selector

The widget "ModeSelector" was added. It has similar functionality as the RadioButtonGroup but with another design.

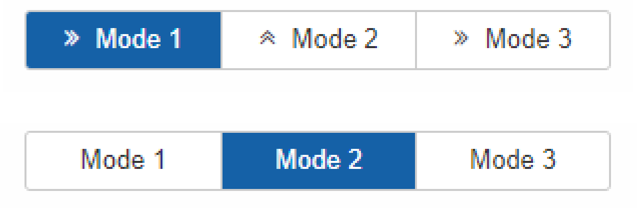


Figure 3. Mode Selector

Improvements for Pages in Scout JS applications

The API to work with Pages (`PageWithTable`, `PageWithNodes`) has been improved. It is now possible to declare child pages in the static JSON model of outlines and the table within a `PageWithTable` has a default reload handler installed.

Now the method `_loadTableData` (which is responsible for fetching data for a `PageWithTable`) also gets an optional argument `searchFilter` holding the exported data of the first form that is attached to the table using a `FormTableControl` (typically the SearchForm). This makes it easier to use the values from a search form by e.g. passing them to a REST backend to limit the results returned from the server.

Finally the `TreeNode` (and therefore all pages because they are tree nodes) get a method `_jsonModel` to declare the static JSON model that belongs to that tree node or page. This works the same way as with all other widgets now.

New event 'lookupCallDone'

All fields having lookup calls (ListBox, RadioButtonGroup, SmartField, TagField) now fire a new event 'lookupCallDone' always when a lookup call has been executed and the result was processed by the field.

Property lookup order changed

The Scout properties are now resolved in a slightly different order (Bug 541099). The environment variables are now resolved *before* the `config.properties` file.

1. System properties
2. Environment variables
3. Config properties file
4. Default value of property

Using environment variables, it is now possible to override values in the configuration file, as is already possible using system properties (`-D` flags on JVM command line). This change should simplify the usage of Scout in environments where the application should be static (example: Kubernetes, Docker), but still allow a degree of flexibility.

Since environment variables are not allowed to contain dots/periods (`.`), the new lookup also searches for an equivalent environment variable by replacing periods with underscores (`_`) and converting the property to uppercase.

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