

## Release Notes FirstSpirit 4.2.453

Previously released versions: 4.2.38, 4.2.206, 4.2.219, 4.2.223, 4.2.432, 4.2.437, 4.2.446

### Error in the Oracle / Sun JDK versions 6 update 24, 25 and 26

FirstSpirit is released for the main version of a JDK, e.g. Oracle Java 6 (JDK 1.6) or IBM JDK 1.5.

Regular updates of JDK are issued as part of the debugging in JDK; these updates are checked within the scope of quality assurance to ensure they are compatible with FirstSpirit.

**In very rare cases, incompatibilities can occur due to errors in JDK, which cannot be handled in programming terms. This is not a FirstSpirit error!**

Due to an error in the Oracle (formerly Sun) JDK versions 6 update 24, 25 and 26, the virtual machine (VM) can crash.

**For this reason, the above versions should not be used in productive use!**

### FirstSpirit API: Deprecation of metadata methods

FirstSpirit provides developers with API functions, in order to access functions and values.

The **getMeta()** and **setMeta(Data data)** methods of the **IDProvider** interface (**de.espirit.firstspirit.access.store.IDProvider**, Access-API) can be used to access the metadata of an element or to change it.

In FirstSpirit Version 4.2.450 and higher, both methods are marked as being "deprecated", i.e. the methods can be omitted in a subsequent version of FirstSpirit.

Instead of these two methods, the new replacement methods introduced **getMetaFormData()** and **setMetaFormData(FormData formData)** of the **IDProvider** interface should be used.

### FirstSpirit API: ValueEngineer and InitialPreset

Module and component development with FirstSpirit is described in the Developer manual for components for FirstSpirit Version 4.2R4 (only German).

Chapter 3.14.7 (ValueEngineer - Handling the values of a SwingGadget) describes the handling of **SwingGadget** values with the **ValueEngineer** interface (**de.espirit.firstspirit.client.access.editor.ValueEngineer**, Developer-API).

In FirstSpirit Version 4.2.452 and higher, the additional aspect `InitialValueProviding` (`de.espirit.firstspirit.client.access.editor.InitialValueProviding`, Developer-API) was introduced for the **ValueEngineer** interface, in order to provide an initial value.

To use this aspect, the GOM object, which is related to the **ValueEngineer**, must define or overwrite the **getPreset()** method of the **GomFormElement** interface (`de.espirit.firstspirit.access.store.templatestore.gom.GomFormElement`, Access-API).

The **InitialPreset** class (`de.espirit.firstspirit.access.store.templatestore.gom.InitialPreset`, Developer-API) must be used as the covariant return value for the method:

```
public InitialPreset getPreset() {  
    return (InitialPreset) super.getPreset();  
}
```

In addition, the **setPreset()** method must be implemented:

```
public void setPreset(final InitialPreset preset) {  
    super.setPreset(preset);  
}
```

### Package pool: Menu order in the target project

Content packages can be subscribed in a project. Apart from pages and sections, a content package can also include elements from the Structure Store.

On updating a subscription, new elements are sorted according to defined rules (see FirstSpirit Release Notes - FirstSpirit Version 4.2 Release 4, Chapter 7.4 - FirstSpirit PackagePool, page 166 ff.).

Although corresponding resorting is required in many cases, in several cases the order of the master project is to be retained.

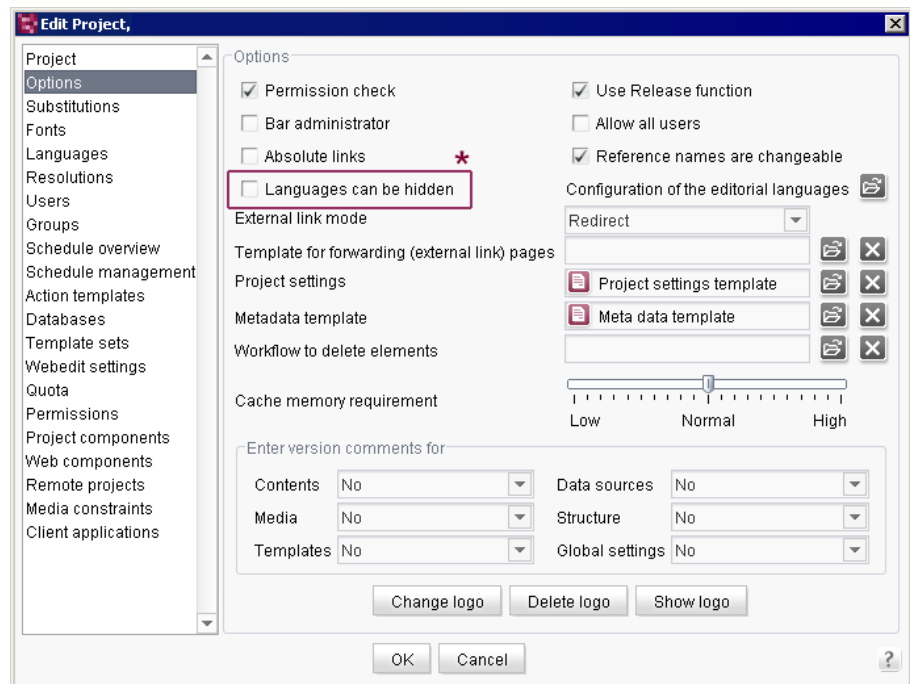
To this end, the "Changeable" option of the subscription is now evaluated during the update.

If this option is enabled (content is changeable), the sorting continues to be made according to the above-mentioned rules.

However, if the option is disabled (content is unchangeable), the order of the master project is decisive for the sorting of elements.

## New function: "Hide languages"

Worldwide operating companies usually publish their websites in many different languages. Because of this large number of languages, it is usually difficult for the individual editors to identify the language in which the content is to be entered. To increase the working efficiency of the individual editors, in FirstSpirit Version 4.2.453 and higher it is possible to hide



Option "Languages can be hidden"

individual languages for specific users. This means that the editor can only see the languages that are relevant for them.

The new function can be set up in the Server and project configuration. In the properties of a project, the function can be enabled in the "Options" area, with the help of the "Languages can be hidden" option.

If this function has been enabled, the additional subpoint "Visible project languages" is available in JavaClient, under the "View" menu item.

If the editor hides a language, the corresponding language tab is no longer displayed, however, the content for this language continues to be loaded.

To use this function in a project, the project must be technically prepared for its used!

**For example, the forms must not contain any mandatory fields ("allowEmpty" parameter), because otherwise errors can occur in the hidden language channel, when the fields are validated. As a result, changes in a language cannot be saved.**

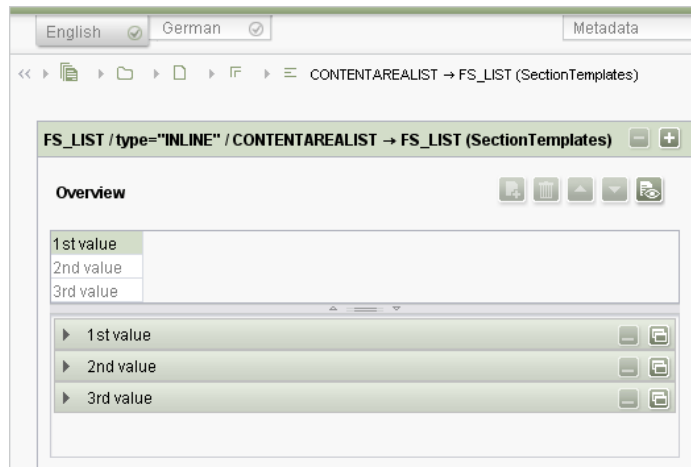
**In several cases, the hiding of a language does not take effect until JavaClient has been restarted!**

**If Edit mode is activated, a language tab is not hidden until after Edit mode has been exited and the element has been updated (<F5>).**

## FS\_LIST input component: "rows" parameter

The input component FS\_LIST can be designed in a very flexible way. For example, an overview (<ADD **component="overview"...**>) can be defined for the types DATABASE, INLINE and SERVICE.

If the label of an entry cannot be fully displayed in the overview, a scrollbar appears. In this case, the last line of the overview can be partially or completely overlapped by the scrollbar. The entry can be displayed by enlarging the workspace or by moving the horizontal separator bar.



Input component FS\_LIST

## Using Content Highlighting and the CMS\_GROUP design element

In several rare cases, it is possible that when Content Highlighting is used in conjunction with the CMS\_GROUP design element, the selected element is not surrounded with a border. If the element is selected again in this case, it is surrounded with a border.

## CMS\_INPUT\_TEXT input component: Missing cursor in Edit mode

If the input component CMS\_INPUT\_TEXT is selected (left mouse button) in View mode, unfortunately, after Edit mode is activated (<CTRL>+<E>), the cursor is not displayed in the input component. The cursor is displayed again if the user re-selects the input component.