FirstSpirit™
Unlock Your Content

Release Notes
FirstSpirit™ 2018-12

Status	**RELEASED**
Department	FS-Core
Copyright	2018 e-Spirit AG
File name	Releasenotes_2018_12_EN

e-Spirit AG

Stockholmer Allee 24
44269 Dortmund | Germany

T +49 231 . 477 77-0
F +49 231 . 477 77-499

info@e-Spirit.com
www.e-Spirit.com

e-Spirit

Table of contents

1 Release of Java 11 and "FirstSpirit Launcher JRE" module

Java 11 is released with FirstSpirit 2018-12, both for server and clients, the “EAP” status is terminated. On the server side, the restrictions described in the release notes for FirstSpirit 2018-10 could be removed by updating the relevant third-party components / libraries (Eclipse Jetty and ASM, see Chapter “Update of internally used software”).
On the client side, there are no known restrictions when working in FirstSpirit SiteArchitect and ServerManager when using Java 11. The display problem regarding the caret in CMS_INPUT_DOM/DOMTABLE in connection with certain fonts has been fixed with the current release (see *CORE-9193*).

To be able to use Java 11 for FirstSpirit SiteArchitect and ServerManager, the FirstSpirit Launcher must be used in conjunction with the “FirstSpirit Launcher JRE” module.
This module is also released with FirstSpirit 2018-12.
It enables customers and partners to easily switch between different Java versions for the operation of Java-based applications. The desired Java version can be selected server-wide for Microsoft Windows and macOS separately.

The included Java versions have been updated with the current release: In addition to the version delivered with the FirstSpirit Launcher (“Default”), Oracle Java 8u181, the following Java variants are currently available:

- OpenJDK 11.0.1
- Oracle Java 8u191 (EAP)

The versions in the “EAP” category are those that are planned for use in a future FirstSpirit version. These versions should initially only be used on testing systems, and potential problems should be relayed to e-Spirit in a timely fashion.

Note on Upgrade: If the module was already used in an earlier version, it must first be uninstalled on the server when upgrading to FirstSpirit 2018-12 and the module version from 2018-12 must then be reinstalled.

For further information please see also documentation for the module “FirstSpirit Launcher JRE”. The module and the documentation may be obtained from Technical Support (https://help.e-spirit.com).

2 Administration / Modules

2.1 New FirstSpirit login module with SSO via SAML 2.0 (EAP)

The FirstSpirit SAML Login Module supports “single sign-on” authentication in FirstSpirit via SAML 2.0 standard. The module is currently in the EAP phase and has so far only been used with Keycloak version 4.6.0 or higher (Keycloak homepage).

Due to the EAP status, the module is not shipped with FirstSpirit 2018-12. If there is interest in participating in the EAP phase, please contact e-Spirit Technical Support.

3 Administration

3.1 64- and 32-bit support for FirstSpirit servers

Since FirstSpirit 5.0, operation of a FirstSpirit server is only supported on 64-bit variants of operating systems. Use of a FirstSpirit server in a 32-bit environment or in 32-bit mode is not supported.

As of FirstSpirit 2018-12, the Java Service Wrapper files which enable operation of a FirstSpirit server in 32-bit mode will no longer be provided. Existing FirstSpirit installations will thus no longer receive updates for these files when the Java Service Wrapper is updated.

In order to avoid mixed operation of two Java Service Wrapper versions, it is urgently recommended to manually delete the 32-bit files from the file system under FirstSpirit 2018-12.

The files in question are listed below (~FS = installation directory of the FirstSpirit server):

- Directory `~FS/bin`
  - `wrapper-aix-ppc-32`
  - `wrapper-linux-x86-32`
  - `wrapper-macosx-universal-32`
  - `wrapper-solaris-sparc-32`
  - `wrapper-solaris-x86-32`

  - `wrapper-windows-x86-32.exe`

- Directory `~FS/server/lib` or `~FS/server/lib-isolated`
  - `libwrapper-aix-ppc-32.so`
  - `libwrapper-linux-x86-32.so`
  - `libwrapper-macosx-universal-32.jnilib`
  - `libwrapper-solaris-sparc-32.so`
  - `libwrapper-solaris-x86-32.so`
  - `wrapper-windows-x86-32.dll`

For FirstSpirit server installations which still use 32-bit, it is recommended to migrate to 64-bit in a timely manner.

3.2 Adjustment of the Technical Datasheet

As an Enterprise Content Management System, FirstSpirit is designed for use in complex IT landscapes and supports a multitude of operating systems, Java run-time environments, and databases. In order to guarantee high software quality and to ensure that FirstSpirit operates reliably in all supported configurations, regular testing is undertaken as part of the quality assurance process. The high number of supported third-party components, short release cycles, as well as combination and configuration possibilities of third-party components lead to an extremely high test complexity.

For increased planning certainty on the side of customers and partners, the “FirstSpirit Technical data sheet” lists all operating systems, Java environments, application servers, etc. as well as the specific version of each which are supported for operations with FirstSpirit. Third-party components which are being tested internally and proactively on a regular basis are categorized as **Actively supported**. Furthermore, a recommended system configuration which is also in use internally by e-Spirit is called **Reference**. As a rule, many other configurations and components are also compatible with FirstSpirit, but this cannot be ensured with regular, internal testing due to the test complexity described above. System configurations, for example, which are/were put into operation successfully by a customer or a partner or older system configurations which are no longer being tested regularly by e-Spirit thus receive the categorization **Passively supported** at this time. Should issues arise in such configurations, e-Spirit can

undertake troubleshooting measures on the basis of submitted issue reports (given a current software maintenance contract), but only in a limited time frame - this does not constitute entitlement, however. Only in the case of software configurations which are categorized as **Not supported**, e-Spirit will take no measures to resolve issues which may arise.

Due to increasing test complexity and the continuously broadening testing arena, e-Spirit plans to more strongly consider third-party components and combinations which cover as large as possible market segments in its selection in the future. Besides system configurations which are successfully put into operation by customers, planning is also to include forward-looking software and hardware trends. The goal is to test the newest version of a given third-party component for operation with FirstSpirit in order to officially approve it as quickly as possible.

In the near future, the “FirstSpirit Technical data sheet” will be revised in order to better reflect the developments described above. With the current release, this focus is already partially reflected, and the status of individual operating systems has been adjusted.

3.3 Clustering: Optimized behavior during restart of the FirstSpirit master server

In cluster operations, the behavior during restart of the FirstSpirit master server has been optimized: among others, the wait time before slave servers attempt to connect to the master server after a restart was increased from 60 to 90 seconds. If this time span is insufficient for the slave server to establish a connection to the master server, another duration (in seconds) can be configured via the parameter

```
cluster.slaveRestartDelay
```

in the file `fs-server.conf`, e.g.

```
cluster.slaveRestartDelay=120
```

The default value is 90.

Please note that the recommendations pertaining to configuration of slave servers (mount points) have been changed due to the optimizations done in this context. Please check if the configuration you use still conforms to the recommendations.

For further information about clustering, see *FirstSpirit Manual for Administrators*,

- “FirstSpirit ServerManager / Server properties / Clustering”
- “Updating the FirstSpirit Server / Updating in a cluster group”

3.4 Obtaining the status of a FirstSpirit server during start-up

During start-up of a FirstSpirit server, information about the current runlevel is now provided. These runlevels indicate if or when certain server functionality is available:

- `SHUTDOWN` (runlevel 0)
  The server is shut down.
- `IN_PROGRESS` (runlevel 20)
  The server is starting up or shutting down, no functionality is guaranteed to be available.
- `CORE_STARTED` (runlevel 40)
  Basic functionality is available, the server can be reached via the SOCKET port.
- `ROOT_WEBAPP_STARTED` (runlevel 60)
  The server can be reached via the HTTP port.
- `CORE_WEBAPPS_STARTED` (runlevel 80)
  The global web applications are available.
- `STARTED` (runlevel 100)
  All FirstSpirit functionality is available (including project-local web applications).

The runlevel is output in various places:

- according log output in the file `fs-server.log` or `fs-wrapper.log`
  Example:

```
INFO <timestamp> (de.espirit.firstspirit.server.RunLevelHolder): Reached run level:
   STARTED(100)
```

- in the file `~FS/.fs.lock`

(except for runlevel 0, in which case this file does not exist)

- using the API, via the `RunLevelAgent`
  (package `de.espirit.firstspirit.agency`, FirstSpirit Developer API)

API Example 1

```
import de.espirit.firstspirit.agency.RunLevelAgent;
runLevelAgent = context.requireSpecialist(RunLevelAgent.TYPE);
context.logInfo("current runlevel: " + runLevelAgent.getRunLevel());
```

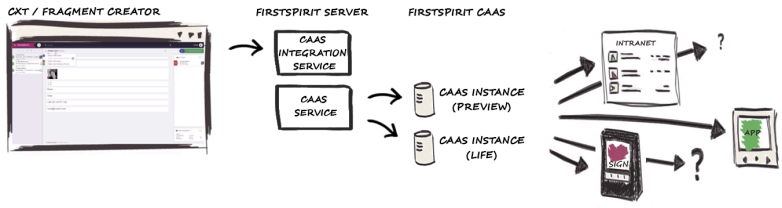
API Example 2

```
import de.espirit.firstspirit.agency.RunLevelAgent;
import de.espirit.firstspirit.server.RunLevel;
import java.time.Duration;
runLevelAgent = context.requireSpecialist(RunLevelAgent.TYPE);
runLevelAgent.waitForRunLevel(RunLevel.STARTED, Duration.ofMinutes(60));
context.logInfo("Runlevel 100 erreicht: " + runLevelAgent.getRunLevel());
```

4 FragmentCreator

4.1 Delivery of fragments and variants via FirstSpirit CaaS (Content-as-a-Service)

The module “CaaS Integration for CXT” enables delivery of a fragment project's content via the universal FirstSpirit CaaS interface so that it may be obtained by various frontends (classical web pages, mobile apps, web shops, digital signage, social media, single-page applications, portals) via pull principle.



CXT CaaS Integration

Benefits:

- The content (fragments and variants) can be reused without limit, requiring no additional implementation effort for a new output channel. The final rendering or the output channel is of no concern here. Once created, these fragments may be used in a wide range of scenarios.

- Content can be published flexibly, both on the corporate level as well as into third-party channels.

The module “CaaS Integration for CXT” contains the **service “CXT CaaS Integration Service”** which, in conjunction with the **“CaaS Service”**, facilitates event-based delivery of project contents. Configuration is done on the project level. If a fragment project is to offer its contents via FirstSpirit CaaS, the **project component “CXT CaaS Integration Configuration”** must be configured with the URL of the CaaS server and with the appropriate API key. Here, two separate CaaS instances are required, one for preview data and one for live data (see illustration).

Following this configuration, contents from the fragment project are output based on events:

- upon change: into one CaaS instance (preview)

- upon release: into another CaaS instance (live)

The exchange of fragments between FirstSpirit CXT and FirstSpirit CaaS is carried out using JSON (JavaScript Object Notation). The output channel of the project must be configured accordingly.

Example for output of fragments/variants in JSON:

```
{
   "fragmentUUid": "$CMS_VALUE(#global.page.parent.uid.replaceAll("_","-"))$",
   "documentUUid": "$CMS_VALUE(#global.page.uid.replaceAll("_","-"))$",
   "variation": $CMS_VALUE(#global.page.meta("variations").toJSON())$,
   "title": $CMS_VALUE(#global.page.meta("title").toJSON())$,
   "text": $CMS_VALUE(text.toHtml().toJSON())$,
   "url": "$CMS_VALUE(url)$"
}
```

For further information, see documentation “Content Experience Tools”.

4.2 Editing fragments and variants from within ContentCreator

Fragments and variants can now not just be referenced in ContentCreator, but also edited from within ContentCreator.

Previous behavior:

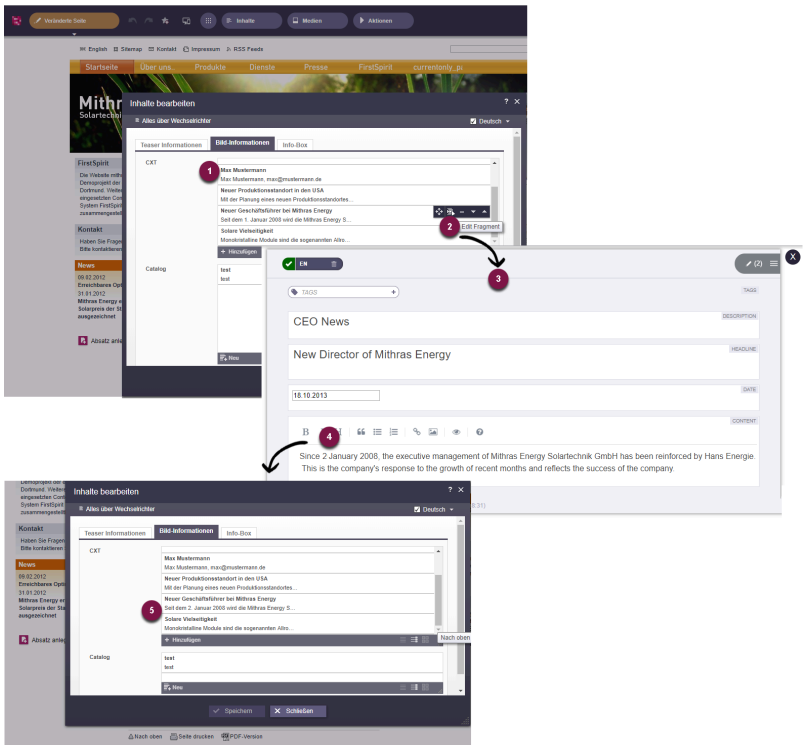
Via the “Fragment DAP” module, fragments (and variants) can be edited centrally in a fragment project and made available in various target projects (as reference). The fragments are not imported into the target projects, but referenced via the DAP access instead, meaning that physically they remain in the fragment project.

Advantages:

- no additional storage requirements for fragments used in multiple projects.
- simplified editing and management, as all content is stored in a central fragment project.

New functionality:

Fragments and variants can now also be edited from within ContentCreator. To achieve this, a properly configured `FS_INDEX` input component can be created in the project. This causes an “Edit Fragment” button to be displayed to the editor in ContentCreator at all referenced fragments. By clicking on the button, the form from the FragmentCreator is opened as a MicroApp directly within the ContentCreator and can be edited there. The editor can then close the form (FragmentCreator). All changes are directly viewable in the ContentCreator.



Initialize editing a fragment from within ContentCreator

Illustration:

- (1) Display of the referenced fragments or variants in the ContentCreator.
- (2) Insertion of the “Edit fragment” button during mouseover in the ContentCreator.
- (3) By clicking on the button, the form from the FragmentCreator opens as a MicroApp directly in the ContentCreator.
- (4) The contents of the fragment or variant can be edited. Changes are saved automatically.
- (5) The changes to the fragment or variant appear directly in the ContentCreator.

Outlook: The new **integration interface of the CXT** platform was used for this functionality. This interface provides individual functions from the CXT world in form of MicroApps which can then be used in other environments. The integration interface will be further developed in the future and can then be used, for example, to enable the processing of fragments in external applications (e.g. single-page web apps).

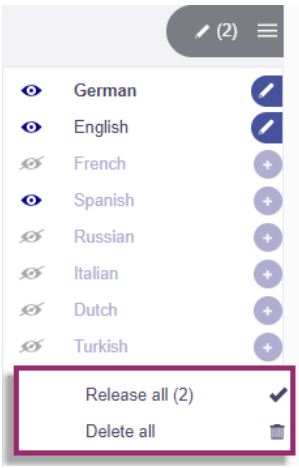
Configuration: To use the new functionality in the ContentCreator, you have to:

- Install the “Fragment DAP” module on the Server.
- Add the web component “Fragment DAP for ContentCreator” in the ServerManager under “Server Properties - Web Applications - ContentCreator”. This web application provides a configuration interface. The URL to the MicroApp must be configured here.

4.3 Releasing or deleting multiple variants of a fragment at the same time

Editors are now given the option to request release or deletion of all suitable variants of a fragment at the same time.

The according workflows can be called via the variant selection icon of a fragment.



Below the list of all available variants of the current fragment, the list now includes options to call one of the two standard workflows, “Release” and “Delete”.

The following starting conditions are possible:

- All variants of a fragment should be released:
  - clicking **Release all** starts the release process for ***all*** variants.
- All changed variants of a fragment should be released:

- clicking **Release changed** starts the release process for all ***changed*** variants.

- All variants of a fragment (and thus the fragment as a whole) should be deleted:
  - clicking **Delete all** starts the deletion process for ***all*** variants.

5 Input components

5.1 CMS_INPUT_DOM: Normalized output of block elements

For HTML output of a CMS_INPUT_DOM component, the FirstSpirit-internal document model can be “normalized” to fit the HTML document model by calling

```
$CMS_VALUE(st_dom.normalize)$
```

This causes lists and tables to no longer be nested within block elements (usually, <p>).

In the past, the first call to `.normalize` caused all subsequent DOM output to be normalized, regardless of whether `$CMS_VALUE(st_dom)$` or `$CMS_VALUE(st_dom.normalize)$` was used. This erroneous behavior has now been fixed: `.normalize` now only applies to the current call to `$CMS_VALUE(st_dom.normalize)$`. Thus, calling `.normalize` is now necessary each time the variable should be output in a normalized fashion in the template.

If `.normalize` is used in existing FirstSpirit projects, project developers should check if templates (uses of `$CMS_VALUE(st_dom)$`) have to be modified accordingly.

For further information, see *FirstSpirit Online Documentation*,

- “Template development / Template syntax / Data types / DomElement”, method `normalize`

- “Templates (basics) / Composition of templates / Format templates / Output”

5.2 FS_CATALOG: Identification of entries ("cards")

The input component FS_CATALOG enables creation of lists with sections or links. It returns a `Catalog` object. This object contains `Card` objects. Each `Card` object represents a single FS_CATALOG entry.

As of the current FirstSpirit release, a single entry (“card”) of an FS_CATALOG can now be identified using a unique ID. This ID may be accessed during generation, for instance. Such an ID may look as follows:

```
c84c8dae-512a-4142-9068-85208d4072fe
```

`Card` objects: method `getId()`

`Card` objects now provide a new method `getId()` which may be used to obtain the ID of the associated FS_CATALOG entry.

Example:

```
<p>
    $CMS_FOR(var,st_catalog)$
        ID:   $CMS_VALUE(var.getId())$
    $CMS_END_FOR$
</p>
```

Here, *var* is the variable for the (list entry) object, *st_catalog* is the variable name of the FS_CATALOG component.

`Catalog` objects: method `find(String)`

`Catalog` objects now provide a new method `find(String)` which returns a `java.util.Optional<Card>` object which contains the `Card` object with the submitted ID (if present in the catalog).

The following syntax sample outputs the contents of the input component *st_text* of a specific entry (here, the FS_CATALOG entry with the ID *7644cfe2-b117-4ec7-bc94-bb8f255d1f59*):

```
$CMS_VALUE(st_catalog.find("7644cfe2-b117-4ec7-bc94-bb8f255d1f59").get().item.st_text)$
```

System object `#card`

The system object `#card` may be used to access the entry of an FS_CATALOG input component. It is available in the (section or link) templates which are being referenced by the FS_CATALOG input component, however, only in `$CMS_VALUE(...)$` calls, not in output via `$CMS_FOR(...)$`.

In order to obtain the ID of an entry, `#card` may also be used in conjunction with `getId()` in the referenced (section or link) template.

Example:

```
$CMS_VALUE(#card.getId())$
```

For further information, see *FirstSpirit Online Documentation*,

- “Template development / Forms / Input components / CATALOG”
- “Template development / Template syntax / Data types / Card”
- “Template development / Template syntax / Data types / Catalog”
- “Template development / Template syntax / System objects / #card”

6 Module Development, Scripts, API

6.1 Check for duplicate resources

Classes and other resources are defined in FirstSpirit modules and in their components in `<resource>` entries. These reference a Jar file or a directory. These resource entries can carry information about version and “scope” (e.g. available server-wide or limited to the module) of the resource.

Duplicate resources for the same scope (i.e. web or project/server) are not permitted.

As of the current release, FirstSpirit prevents the use of duplicate resources, i.e. Jar files in identical or different versions, within the same scope. Attempts to install a module with duplicate resources will be aborted with an error message. This prevents possible subsequent erroneous behavior during use of the modules.

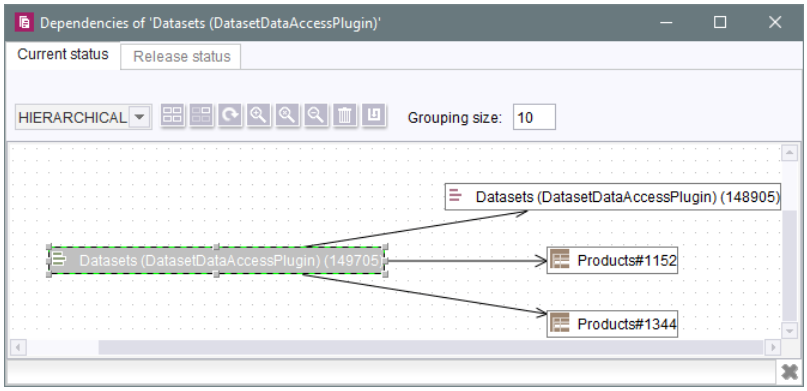
For further information, see “FirstSpirit Entwicklerhandbuch für Komponenten (Beta)”, “FirstSpirit Modul-Grundkonzeption / Modul-Bestandteile / Ressourcen” (German only).

6.2 Data Access Plugins (DAP): Advertising references

By implementing “data access plugins” (“DAP”), external sources (e.g. web services) can be connected to an FS_INDEX input component. This way, data objects from such external sources can be referenced within the index component and thus used for content purposes. Some fundamental FirstSpirit function-

ality is based on such data access plugins, e.g. the “DatasetDataAccessPlugin” used for selection of datasets from data sources of a project. Data access plugins can also be used to implement customer- and project-specific functionality.

Dependencies between objects in FirstSpirit (or to objects external to FirstSpirit) are also called “references”. The datasets selected for inclusion in an FS_INDEX component by way of the DatasetDataAccessPlugin, for example, hold “inbound” references from the FS_INDEX component. Such dependencies are visualized in the “FirstSpirit reference graph” (Ctrl + R):



In order to avoid inconsistencies, these references are checked, for example, before objects are deleted from a FirstSpirit project. Furthermore, references may be used to carry out “dependent releases”. These functionalities are provided by the FirstSpirit reference manager.

As of the current release, developers can advertise outbound references of data access plugins and thus use the functionality of FirstSpirit's reference manager.

The FirstSpirit Access API package `de.espirit.firstspirit.client.plugin.dataaccess.aspects` provides the aspect interface

```
ReferencesReporting
```

This aspect may be used by the data access plugin to submit references.

References are divided into two categories, each of which requires a separate method:

- `reportValueReferences`

is used for references to values which are selected/saved by the editor within the data access plugin, e.g. FirstSpirit objects or external elements. In the case of the DatasetDataAccess-Plugin, these are the datasets selected via FS_INDEX.

- `reportModelReferences`
  is used for references to FirstSpirit objects which are used in the configuration of the data access plugin, e.g. FirstSpirit templates. In the case of the DatasetDataAccessPlugin, this may be the table template referenced via the tag `TEMPLATE`, for example.

The interface also serves as an adapter so that only the (empty) method which is required by the data access plugin needs to be implemented.

Interface `ValueReferencesJournal` (also in the package `de.espirit.firstspirit.client.plugin.dataaccess.aspects`)
This interface is provided to submit references to values. It offers several methods for different object types and also to indicate broken references. The objects which submit the references are indicated via a unique identifier.

Interface `ModelReferencesJournal` (also in the package `de.espirit.firstspirit.client.plugin.dataaccess.aspects`)
This interface is provided to submit references to FirstSpirit objects which are used in the configuration of the data access plugin, e.g. FirstSpirit templates or media.

The data access plugin DatasetDataAccessPlugin which is provided with FirstSpirit previously used non-public API to model references. As of the current release, this data access plugin was modified to use the new, public API. If existing customer-specific implementations also use non-public API related to this topic, they should be modified accordingly as soon as possible. This affects the following interfaces:

- `ValueReferencing`
- `ModelReferencing`

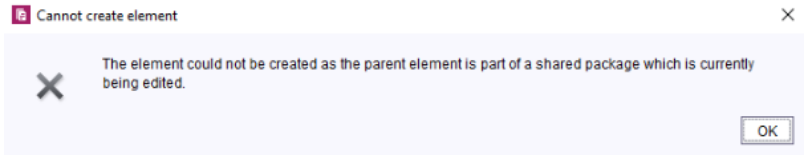
These interfaces have been marked deprecated as of FirstSpirit 2018-12 and will be removed in subsequent versions.

For further information, see also *FirstSpirit Online Documentation*, “Plug-In Development / Universal Extensions / Data Access”.

7 Modules: Corporate Content

7.1 Behavioral change for CorporateContent packages currently being edited within SiteArchitect

Editing a CorporateContent package by other users while it is being edited by an administrator could lead to problems. The functionality was improved within SiteArchitect with FirstSpirit 2018-12. If a Corporate-Content package on the master server is edited by an administrator, no further packet-changing actions can be performed by other users. An error message is displayed instead:



Error message

> ! The behavioral change only applies to actions within SiteArchitect.

8 System

8.1 Update of internally used software

With the current FirstSpirit release, the following, internally used software has been updated:

- **ASM** (Library for analyzing and modifying Java bytecode)
  Update from version 6.2 to 7.0 (official Java 11 support)
- **Eclipse Jetty** (contained in the module `fs-jetty.fsm` / “FirstSpirit Jetty Service”)
  Update to version 9.4.14

> ! As before, use of Eclipse Jetty for production purposes is not recommended!

9 Deprecations

For e-Spirit, an important goal in software development is to avoid introducing incompatibilities and migration expenditures related to updating from one FirstSpirit release to the next as much as possible or to compensate for these within the software. FirstSpirit updates should generally be deployable with little effort or able to be carried out in a fully automated fashion.

However - not least in order to ensure maintainability and to future-proof the software - e-Spirit cannot fully avoid replacing existing functionality with new mechanisms. In the future, functionality that will be removed from the software will be listed in this section, including the date at which time the functionality will be removed.

Functionality	Deprecated as of	Will be removed/ Was removed as of
FirstSpirit Developer API: de.espirit.firstspirit.agency.GroupsAgent	5.2R15	
FirstSpirit Access API: delete (de.espirit.firstspirit.access.AccessUtil)	5.2R18	
Windows installation program	2018-06	
FirstSpirit Access API: release (de.espirit.firstspirit.access.AccessUtil)	2018-06	
FirstSpirit Developer API: getLastLoginAsDate (de.espirit.firstspirit.agency.UserStatisticsAgent)	2018-07	
FirstSpirit Developer API: remainingDurationOfCurrentStageInMillis (de.espirit.firstspirit.server.MaintenanceModeInfo)	2018-07	
FirstSpirit Developer API: getStartingTimeOfStageAsDate (de.espirit.firstspirit.server.MaintenanceModeInfo)	2018-07	
FirstSpirit Access API: getSelectedWebserverConfiguration (de.espirit.firstspirit.access.serverConfiguration)	2018-10	

Functionality	Deprecated as of	Will be removed/ Was removed as of
FirstSpirit Access API: setSelectedWebserverConfiguration (de.espirit.firstspirit.access.serverConfiguration)	2018-10	
FirstSpirit Access API: getSelectedWebServer (de.espirit.firstspirit.access.project.Project)	2018-10	
FirstSpirit Access API: setSelectedWebServer (de.espirit.firstspirit.access.project.Project)	2018-10	
FirstSpirit Developer API: getLostAndFoundStoreNodes(); (de.espirit.firstspirit.feature.FeatureInstallResult)	2018-10	
FirstSpirit Developer API: getDeletedStoreNodes(); (de.espirit.firstspirit.feature.FeatureInstallResult)	2018-10	

10 Overview

ID	Description	Categories
CORE-3934	Optimized display of messages in SiteArchitect which are sent from within ServerMonitoring (“FirstSpirit / Message”).	SiteArchitect
CORE-8101	Fragments and variants can now not just be referenced in Content-Creator, but also edited from within ContentCreator. Further information can be found in chapter “FragmentCreator: Editing fragments and variants from within ContentCreator ”.	ContentCreator, FragmentCreator
CORE-8365	A new FirstSpirit login module with “single sign-on” (SSO) support via SAML 2.0 has been developed and is currently in the EAP phase. Further information can be found in chapter “Administration / Modules: New FirstSpirit login module with SSO via SAML 2.0 (EAP)”.	Modules, Server Admin-istrator

ID	Description	Categories
CORE-8509	While an administrator edits a CorporateContent package on the master server, no package-changing actions can be performed by any other user within SiteArchitect. Further information can be found in chapter “Modules: Corporate Content: Behavioral change for CorporateContent packages currently being edited within SiteArchitect”.	Corporate Content, SiteArchitect
CORE-8616	Identification of entries of an FS_CATALOG (“cards”) using a unique ID. Further information can be found in chapter “Input components: FS_CATALOG: Identification of entries ("cards")”.	Developer, FirstSpirit API, Input Components
CORE-8768	In workflows, improved mail sending to defined groups.	SiteArchitect, Workflow
CORE-8852	The module “CaaS Integration for CXT” enables delivery of a fragment project's content via the universal FirstSpirit CaaS interface. Further information can be found in chapter “FragmentCreator: Delivery of fragments and variants via FirstSpirit CaaS (Content-as-a-Service)”.	FragmentCreator, Modules
CORE-8883	During start-up of a FirstSpirit server, information about the current runlevel is now provided which may be used to react to, for example via API. Further information can be found in chapter “Administration: Obtaining the status of a FirstSpirit server during start-up”.	Developer, FirstSpirit API
CORE-9086	It is now possible to release or delete multiple variants of a fragment at the same time. Further information can be found in chapter “FragmentCreator: Releasing or deleting multiple variants of a fragment at the same time”.	FragmentCreator
CORE-9117	Improved installation of a FirstSpirit server in isolated mode on Linux.	Isolated mode (BETA), Server Administrator

ID	Description	Categories
CORE-9191	A more informative error message will now be shown if, erroneously, external users are used as technical users in schedule tasks.	FirstSpirit Administrator
CORE-9193	When using Java 11, display issues could occur related to the caret in the input components CMS_INPUT_DOM and CMS_INPUT_DOMTABLE if certain fonts were used.	Input Components, Java
CORE-9207	Improved consideration of user permissions when uploading files via a script.	Developer
CORE-9276	Optimized event handling between FirstSpirit server and web application.	ContentCreator
CORE-9297	Attempting to install a module with an incorrect `module.xml` now results in a more expressive error message.	FirstSpirit Administrator, Modules
CORE-9298	`.normalize` is now only valid until the next call of `$CMS_VALUE(st_dom)$`. Thus, calling `.normalize` is now necessary each time the variable should be output in a normalized fashion in the template. Further information can be found in chapter “Input components: CMS_INPUT_DOM: Normalized output of block elements”.	ContentCreator, Developer, Input Components, SiteArchitect
CORE-9321	In the current FirstSpirit release, updated internally used software. Further information can be found in chapter “System: Update of internally used software”.	Integrated software
CORE-9362	In cluster operations, optimized the behavior during restart of the FirstSpirit master server. Further information can be found in chapter “Administration: Clustering: Optimized behavior during restart of the FirstSpirit master server”.	Clustering
CORE-9459	Improved performance when editing SEO URLs.	SiteArchitect
CORE-9476	Optimized clean-up of data that is no longer needed when opening archive databases.	Archiving

ID	Description	Categories
CORE-9502	The method `getAdminMode()` of the interface `User` (package `de.espirit.firstspirit.access`, FirstSpirit Access API) erroneously always returned `false` for a server administrator user, even if “Administrator mode” was active in FirstSpirit SiteArchitect (menu “Project”).	Developer
CORE-9522	Erroneously, when editing a template externally, another file name extension than “txt” was used.	SiteArchitect, Template Store
CORE-9536	Optimized code completion in the definition of FS_BUTTON input components.	Developer, SiteArchitect, Template Development
CORE-9539	In rare cases, when the FirstSpirit start-up process was aborted at an inopportune time following an update, it was possible that some files were created with incomplete content. This state will now be recognized and corrected during the next start-up process.	FirstSpirit Administrator
CORE-9542	As of the current release, FirstSpirit prevents the use of duplicate resources. Further information can be found in chapter “Module Development, Scripts, API: Check for duplicate resources”.	Developer, FirstSpirit Administrator, Module development, Server Administrator, Server-Manager
CORE-9612	When a FirstSpirit server was restarted, some web applications did not work without a restart of an upstream Apache Tomcat.	FirstSpirit Administrator
CORE-9614	Improved evaluation of the design element CMS_GROUP when it is displayed in forms.	ContentCreator
CORE-9649	As of FirstSpirit 2018-12, the Java Service Wrapper files which enable operation of a FirstSpirit server in 32-bit mode will no longer be provided. Further information can be found in chapter “Administration: 64- and 32-bit support for FirstSpirit servers”.	FirstSpirit Administrator, Integrated software
CORE-9673	Improved detection of media usages in multiply nested input components.	SiteArchitect
CORE-9674	Using the FirstSpirit Jetty Web Server, malfunctioning could occur when logging on to the FirstSpirit start page in certain languages.	FirstSpirit home page, Web server

ID	Description	Categories
CORE-9675	Adjustment of the Technical Datasheet Further information can be found in chapter “Administration: Adjustment of the Technical Datasheet”.	FirstSpirit Administrator
CORE-9721	If multiple web servers were operated in parallel, certain conditions could cause a user to be redirected to the start page of the wrong web server upon logging out of ContentCreator.	ContentCreator
CORE-9722	As of the current release, developers can advertise outbound references of “data access plugins” and thus use the functionality of FirstSpirit's reference manager. Further information can be found in chapter “Module Development, Scripts, API: Data Access Plugins (DAP): Advertising references”.	Developer, FirstSpirit API, Input Components
CORE-9728	Optimized clean-up of uninstalled web applications.	Server Administrator
CORE-9741	Improved display of navigation when creating a new page.	ContentCreator
CORE-9747	In rare cases, when running on Microsoft Windows, display errors could arise on monitors with 4K resolution.	SiteArchitect
CORE-9759	The file `fs-isolated-runtime.jar` (relevant for FirstSpirit servers running in “Isolated mode”) is now rolled out to `~FS/data/fslib` (~FS = installation directory of the FirstSpirit server) when the FirstSpirit server is started.	FirstSpirit Administrator, Isolated mode (BETA)
CORE-9766	Security improvements related to object deserialization.	Module development, Security
CORE-9784	Improved problem detection when configuring a faulty module.	Isolated mode (BETA)
CORE-9831	Logging of the “FirstSpirit Jetty Server” web server has been improved.	Web server
CORE-9889	The listener in `WE_API.Common.addWorkflowTransitionListener(Common.WorkflowTransitionListener)` (Interface `Common`, Package `de.espirit.firstspirit.webedit.client.api`, FirstSpirit Developer API) was not called for some workflow switchings. This has been corrected.	ContentCreator, FirstSpirit API, Workflow

11 Categories

11.1 Archiving

ID	Description
CORE-9476	Optimized clean-up of data that is no longer needed when opening archive databases.

11.2 Clustering

ID	Description
CORE-9362	In cluster operations, optimized the behavior during restart of the FirstSpirit master server. Further information can be found in chapter “Administration: Clustering: Optimized behavior during restart of the FirstSpirit master server”.

11.3 ContentCreator

ID	Description
CORE-8101	Fragments and variants can now not just be referenced in ContentCreator, but also edited from within ContentCreator. Further information can be found in chapter “FragmentCreator: Editing fragments and variants from within ContentCreator ”.
CORE-9276	Optimized event handling between FirstSpirit server and web application.
CORE-9298	`.normalize` is now only valid until the next call of `$CMS_VALUE(st_dom)$`. Thus, calling `.normalize` is now necessary each time the variable should be output in a normalized fashion in the template. Further information can be found in chapter “Input components: CMS_INPUT_DOM: Normalized output of block elements”.
CORE-9614	Improved evaluation of the design element CMS_GROUP when it is displayed in forms.

ID	Description
CORE-9721	If multiple web servers were operated in parallel, certain conditions could cause a user to be redirected to the start page of the wrong web server upon logging out of ContentCreator.
CORE-9741	Improved display of navigation when creating a new page.
CORE-9889	The listener in `WE_API.Common.addWorkflowTransitionListener(Common.WorkflowTransitionListener)` (Interface `Common`, Package `de.espirit.firstspirit.webedit.client.api`, FirstSpirit Developer API) was not called for some workflow switchings. This has been corrected.

11.4 Corporate Content

ID	Description
CORE-8509	While an administrator edits a CorporateContent package on the master server, no package-changing actions can be performed by any other user within SiteArchitect. Further information can be found in chapter “Modules: Corporate Content: Behavioral change for CorporateContent packages currently being edited within SiteArchitect”.

11.5 Developer

ID	Description
CORE-8616	Identification of entries of an FS_CATALOG (“cards”) using a unique ID. Further information can be found in chapter “Input components: FS_CATALOG: Identification of entries ("cards")”.
CORE-8883	During start-up of a FirstSpirit server, information about the current runlevel is now provided which may be used to react to, for example via API. Further information can be found in chapter “Administration: Obtaining the status of a FirstSpirit server during start-up”.
CORE-9207	Improved consideration of user permissions when uploading files via a script.

ID	Description
CORE-9298	`.normalize` is now only valid until the next call of `$CMS_VALUE(st_dom)$`. Thus, calling `.normalize` is now necessary each time the variable should be output in a normalized fashion in the template. Further information can be found in chapter “Input components: CMS_INPUT_DOM: Normalized output of block elements”.
CORE-9502	The method `getAdminMode()` of the interface `User` (package `de.espirit.firstspirit.access`, FirstSpirit Access API) erroneously always returned `false` for a server administrator user, even if “Administrator mode” was active in FirstSpirit SiteArchitect (menu “Project”).
CORE-9536	Optimized code completion in the definition of FS_BUTTON input components.
CORE-9542	As of the current release, FirstSpirit prevents the use of duplicate resources. Further information can be found in chapter “Module Development, Scripts, API: Check for duplicate resources”.
CORE-9722	As of the current release, developers can advertise outbound references of “data access plugins” and thus use the functionality of FirstSpirit's reference manager. Further information can be found in chapter “Module Development, Scripts, API: Data Access Plugins (DAP): Advertising references”.

11.6 FirstSpirit API

ID	Description
CORE-8616	Identification of entries of an FS_CATALOG (“cards”) using a unique ID. Further information can be found in chapter “Input components: FS_CATALOG: Identification of entries ("cards")”.
CORE-8883	During start-up of a FirstSpirit server, information about the current runlevel is now provided which may be used to react to, for example via API. Further information can be found in chapter “Administration: Obtaining the status of a FirstSpirit server during start-up”.

ID	Description
CORE-9722	As of the current release, developers can advertise outbound references of “data access plugins” and thus use the functionality of FirstSpirit's reference manager. Further information can be found in chapter “Module Development, Scripts, API: Data Access Plugins (DAP): Advertising references”.
CORE-9889	The listener in `WE_API.Common.addWorkflowTransitionListener(Common.WorkflowTransitionListener)` (Interface `Common`, Package `de.espirit.firstspirit.webedit.client.api`, FirstSpirit Developer API) was not called for some workflow switchings. This has been corrected.

11.7 FirstSpirit Administrator

ID	Description
CORE-9191	A more informative error message will now be shown if, erroneously, external users are used as technical users in schedule tasks.
CORE-9297	Attempting to install a module with an incorrect `module.xml` now results in a more expressive error message.
CORE-9539	In rare cases, when the FirstSpirit start-up process was aborted at an inopportune time following an update, it was possible that some files were created with incomplete content. This state will now be recognized and corrected during the next start-up process.
CORE-9542	As of the current release, FirstSpirit prevents the use of duplicate resources. Further information can be found in chapter “Module Development, Scripts, API: Check for duplicate resources”.
CORE-9612	When a FirstSpirit server was restarted, some web applications did not work without a restart of an upstream Apache Tomcat.
CORE-9649	As of FirstSpirit 2018-12, the Java Service Wrapper files which enable operation of a FirstSpirit server in 32-bit mode will no longer be provided. Further information can be found in chapter “Administration: 64- and 32-bit support for FirstSpirit servers”.
CORE-9675	Adjustment of the Technical Datasheet

ID	Description
	Further information can be found in chapter “Administration: Adjustment of the Technical Datasheet”.
CORE-9759	The file `fs-isolated-runtime.jar` (relevant for FirstSpirit servers running in “Isolated mode”) is now rolled out to `~FS/data/fslib` (`~FS` = installation directory of the FirstSpirit server) when the FirstSpirit server is started.

11.8 FirstSpirit home page

ID	Description
CORE-9674	Using the FirstSpirit Jetty Web Server, malfunctioning could occur when logging on to the FirstSpirit start page in certain languages.

11.9 FragmentCreator

ID	Description
CORE-8101	Fragments and variants can now not just be referenced in ContentCreator, but also edited from within ContentCreator. Further information can be found in chapter “FragmentCreator: Editing fragments and variants from within ContentCreator ”.
CORE-8852	The module “CaaS Integration for CXT” enables delivery of a fragment project's content via the universal FirstSpirit CaaS interface. Further information can be found in chapter “FragmentCreator: Delivery of fragments and variants via FirstSpirit CaaS (Content-as-a-Service)”.
CORE-9086	It is now possible to release or delete multiple variants of a fragment at the same time. Further information can be found in chapter “FragmentCreator: Releasing or deleting multiple variants of a fragment at the same time”.

11.10 Input Components

ID	Description
CORE-8616	Identification of entries of an FS_CATALOG (“cards”) using a unique ID. Further information can be found in chapter “Input components: FS_CATALOG: Identification of entries ("cards")”.
CORE-9193	When using Java 11, display issues could occur related to the caret in the input components CMS_INPUT_DOM and CMS_INPUT_DOMTABLE if certain fonts were used.
CORE-9298	`.normalize` is now only valid until the next call of `$CMS_VALUE(st_dom)$`. Thus, calling `.normalize` is now necessary each time the variable should be output in a normalized fashion in the template. Further information can be found in chapter “Input components: CMS_INPUT_DOM: Normalized output of block elements”.
CORE-9722	As of the current release, developers can advertise outbound references of “data access plugins” and thus use the functionality of FirstSpirit's reference manager. Further information can be found in chapter “Module Development, Scripts, API: Data Access Plugins (DAP): Advertising references”.

11.11 Integrated software

ID	Description
CORE-9321	In the current FirstSpirit release, updated internally used software. Further information can be found in chapter “System: Update of internally used software”.
CORE-9649	As of FirstSpirit 2018-12, the Java Service Wrapper files which enable operation of a FirstSpirit server in 32-bit mode will no longer be provided. Further information can be found in chapter “Administration: 64- and 32-bit support for FirstSpirit servers”.

11.12 Isolated mode (BETA)

ID	Description
CORE-9117	Improved installation of a FirstSpirit server in isolated mode on Linux.
CORE-9759	The file `fs-isolated-runtime.jar` (relevant for FirstSpirit servers running in “Isolated mode”) is now rolled out to `~FS/data/fslib` (~FS = installation directory of the FirstSpirit server) when the FirstSpirit server is started.
CORE-9784	Improved problem detection when configuring a faulty module.

11.13 Java

ID	Description
CORE-9193	When using Java 11, display issues could occur related to the caret in the input components CMS_INPUT_DOM and CMS_INPUT_DOMTABLE if certain fonts were used.

11.14 Module development

ID	Description
CORE-9542	As of the current release, FirstSpirit prevents the use of duplicate resources. Further information can be found in chapter “Module Development, Scripts, API: Check for duplicate resources”.
CORE-9766	Security improvements related to object deserialization.

11.15 Modules

ID	Description
CORE-8365	A new FirstSpirit login module with “single sign-on” (SSO) support via SAML 2.0 has been developed and is currently in the EAP phase.

ID	Description
	Further information can be found in chapter “Administration / Modules: New FirstSpirit login module with SSO via SAML 2.0 (EAP)”.
CORE-8852	The module “CaaS Integration for CXT” enables delivery of a fragment project's content via the universal FirstSpirit CaaS interface. Further information can be found in chapter “FragmentCreator: Delivery of fragments and variants via FirstSpirit CaaS (Content-as-a-Service)”.
CORE-9297	Attempting to install a module with an incorrect `module.xml` now results in a more expressive error message.

11.16 Security

ID	Description
CORE-9766	Security improvements related to object deserialization.

11.17 Server Administrator

ID	Description
CORE-8365	A new FirstSpirit login module with “single sign-on” (SSO) support via SAML 2.0 has been developed and is currently in the EAP phase. Further information can be found in chapter “Administration / Modules: New FirstSpirit login module with SSO via SAML 2.0 (EAP)”.
CORE-9117	Improved installation of a FirstSpirit server in isolated mode on Linux.
CORE-9542	As of the current release, FirstSpirit prevents the use of duplicate resources. Further information can be found in chapter “Module Development, Scripts, API: Check for duplicate resources”.
CORE-9728	Optimized clean-up of uninstalled web applications.

11.18 ServerManager

ID	Description
CORE-9542	As of the current release, FirstSpirit prevents the use of duplicate resources. Further information can be found in chapter “Module Development, Scripts, API: Check for duplicate resources”.

11.19 SiteArchitect

ID	Description
CORE-3934	Optimized display of messages in SiteArchitect which are sent from within ServerMonitoring (“FirstSpirit / Message”).
CORE-8509	While an administrator edits a CorporateContent package on the master server, no package-changing actions can be performed by any other user within SiteArchitect. Further information can be found in chapter “Modules: Corporate Content: Behavioral change for CorporateContent packages currently being edited within SiteArchitect”.
CORE-8768	In workflows, improved mail sending to defined groups.
CORE-9298	`.normalize` is now only valid until the next call of `$CMS_VALUE(st_dom)$`. Thus, calling `.normalize` is now necessary each time the variable should be output in a normalized fashion in the template. Further information can be found in chapter “Input components: CMS_INPUT_DOM: Normalized output of block elements”.
CORE-9459	Improved performance when editing SEO URLs.
CORE-9522	Erroneously, when editing a template externally, another file name extension than “txt” was used.
CORE-9536	Optimized code completion in the definition of FS_BUTTON input components.
CORE-9673	Improved detection of media usages in multiply nested input components.
CORE-9747	In rare cases, when running on Microsoft Windows, display errors could arise on monitors with 4K resolution.

11.20 Template Development

ID	Description
CORE-9536	Optimized code completion in the definition of FS_BUTTON input components.

11.21 Template Store

ID	Description
CORE-9522	Erroneously, when editing a template externally, another file name extension than “txt” was used.

11.22 Web server

ID	Description
CORE-9674	Using the FirstSpirit Jetty Web Server, malfunctioning could occur when logging on to the FirstSpirit start page in certain languages.
CORE-9831	Logging of the “FirstSpirit Jetty Server” web server has been improved.

11.23 Workflow

ID	Description
CORE-8768	In workflows, improved mail sending to defined groups.
CORE-9889	The listener in `WE_API.Common.addWorkflowTransitionListener(Common.WorkflowTransitionListener)` (Interface `Common`, Package `de.espirit.firstspirit.webedit.client.api`, FirstSpirit Developer API) was not called for some workflow switchings. This has been corrected.