FirstSpirit™
Unlock Your Content

FirstSpirit™ Release Notes

FirstSpirit™ Version 5.2 Release 14

Version	**5.2.1403**
Status	**RELEASED**
Date	**2017-10-16**
Department	FS-Core
Copyright	2017 e-Spirit AG
File name	Releasenotes_5.2.1403_EN

e-Spirit AG

Stockholmer Allee 24
44269 Dortmund | Germany

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

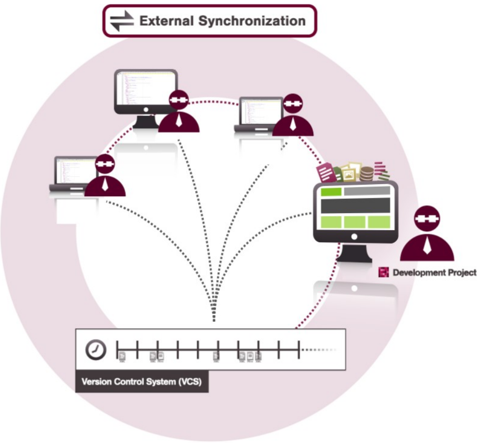
info@e-Spirit.com
www.e-Spirit.com

e-Spirit

Table of contents

1 Support for the distributed development of FirstSpirit projects (External Synchronization)

FirstSpirit projects are generally developed in a team. For distributed project development involving several team members, each team member works in their own local FirstSpirit development environment. The development processes operate at several workplaces (decentrally), at different times (asynchronously), and often at different development strands (nonlinearly). To support these projects successfully, adequate tools (such as version control systems, integrated development environments, etc.) need to be used.



FirstSpirit External Synchronization supports these concepts by providing FirstSpirit objects in a format which allows developers to export templates (and other project content):

- from a FirstSpirit project into a file system structure with a folder hierarchy and
- to import modified content from the file system structure back into a FirstSpirit project.

This export/import functionality makes it possible to:

- edit FirstSpirit objects externally,
- version FirstSpirit objects via an external version control system.

Entry points for External Synchronization are the following interfaces:

- ImportOperation
- ExportOperation

(Package: `de.espirit.firstspirit.store.access.nexport.operations`)

We recommend performing all export and import actions using **FSDevTools**. This tool uses pre-defined commands to automatically import and export data between the FirstSpirit project and the file system. The interfaces therefore do not have to be implemented individually.

Source code and documentation for the compilation, use, and functional expansion of the “FSDevTools” command line tool are published and available to download at:

- https://github.com/e-Spirit/FSDevTools

Technical requirements, configuration and concrete examples are described in the documentation *Introduction* (→*Documentation “External Synchronization”*) (.../help/odfs/edocs/sync/index.html).

Compatibility with other FirstSpirit versions: Functionality for being able to export templates and other content from a FirstSpirit project into a file system structure with folder hierarchy and thus for being able editing these externally has been already introduced with FirstSpirit version 5.1.

This functionality has been extensively enhanced and revised with the current release. For example, the export format has been changed and API methods added.

Exports which were created in earlier FirstSpirit versions using the “External Synchronization” function can still be used. Relevant changes in software behavior are described in detail in Chapter 4.

> **!** After a FirstSpirit version which still uses the old export format is upgraded or updated, the project in question must be fully exported so that the format changes take effect.
> The method `setDeleteObsoleteFiles` must be set to `true`. (This is the default setting when using FSDevTools.)

> **!** Downgrading FirstSpirit versions which use the new export format (to versions which use the old export format) is not supported. In such cases, please contact e-Spirit directly.

2 FirstSpirit documentation in HTML format

The complete FirstSpirit HTML documentation for Administrators has left the beta stage and is now generally available. The migration process for this documentation is now complete, and it is now managed entirely within a FirstSpirit project.

Another new documentation is the one for the “FirstSpirit External Synchronization” functionality (see also Chapter 1. and 4.).

The HTML documentation can be reached from the left navigation sidebar in *FirstSpirit Online Documentation* (“ODFS”).

The HTML documentation for Administrators can also be reached via the menu “Help” (menu entry “Administrators”). Context-sensitive navigation from within SiteArchitect to the documentation can be achieved by pressing F1 or by using the help button in the horizontal toolbar. The PDF version of these documentation continues to be available via ODFS in the chapter “Documentation”.

3 Oracle Java 9

FirstSpirit 5.2 is supported when used with Oracle Java 7 or Oracle Java 8 (for limitations and reference versions, see “FirstSpirit Technical Data Sheet”). Internal tests have shown that FirstSpirit is not yet functional when used with Oracle Java 9. However, it is planned that FirstSpirit will be compatible with Java 9 in the future.

Currently, starting a FirstSpirit server or FirstSpirit clients on Java 9 will be prevented. A message will be logged:

```
Running FirstSpirit using Java 9 or newer is not yet supported! Your version: 9
```

4 External Synchronization

4.1 Detection of empty directory structures (missing meta files)

Especially when using “FirstSpirit External Synchronization” in distributed development, empty directory structures (i.e. directories without meta files `FS_Files.txt`, `FS_Info.txt`, `FS_References.txt`, `StoreElement.xml`, etc.) may remain, e.g. due to moving folders in FirstSpirit. In this case, the FirstSpirit objects in the file system will no longer be recognized as such.

If these meta files are missing, an error message will be output if an export or an import is attempted, and the export/import will be prevented. If `deleteObsoleteFiles` is set to “true”, an export will only log a warning.

4.2 Export: Additional information about exported files (interface "ExportOperation.Result")

In addition to providing information about files which were created in, deleted from, or updated in the file system during an export, the interface `ExportOperation.Result` now includes newly implemented methods which provide information about moved files as well as other information (FirstSpirit Developer API, package `de.espirit.firstspirit.store.access.nexport.operations`, interface `ExportOperation.Result`).

When using **FSDevTools** (recommended best practice), this information will already be evaluated and displayed in the log.

4.3 Export: Explicitly and implicitly selected objects

Because “External Synchronization” is designed for synchronization of partial trees between projects, for each FirstSpirit object that is exported via the “External Synchronization” functionality, the following objects will also be exported:

- all parent objects of the exported object
- all child objects of the exported object

This leads to a distinction between explicitly and implicitly selected objects:

- **Explicit:** The object selected for export as well as all child objects
- **Implicit:** The chain of parent objects of the object selected for export

During an import, explicitly exported objects will be created in the target project if they are not present there and will be updated (i.e. overwritten with each object's respective state in the source project) in the target project if they are already present.

Implicitly exported objects, however, will now only be created in the target project during an import if they are not present there and will no longer be updated if they are already present. Previously, implicitly exported objects were overwritten in the target project with each object's respective state in the source project even if they were already present in the target project.

4.4 Export: Folder "Lost & Found" will no longer be exported

“Lost & Found” folders are created as needed directly underneath the root node of a store in a target project if a parent element (folder) is exported from a source project and imported (“updated”) into a target project, but the child elements of this folder are only present in the target project.

Previously, during an export, such “Lost & Found” folders were exported if they were selected for export. As of the current FirstSpirit version, these folders will by default no longer be exported.

This behavior may be controlled via the method `setExportLostAndFound(boolean enable)` of the interface `ExportOperation` (package `de.espirit.firstspirit.store.access.nexport.operations`, FirstSpirit Developer API). The default value is `false`.

4.5 General changes to the data format

In the course of export format readjustment, an export will now only create one directory named `.FirstSpirit` at the highest directory level instead of creating several directories of that name.

If exports have been made using “External Synchronization” in earlier versions of FirstSpirit, the first export with the current version of FirstSpirit will remove all existing subdirectories named `.FirstSpirit` from the external synchronization directory in the file system. Only one directory named `.FirstSpirit` will remain, at the highest directory level.

In order to create the new export format after an upgrade from an earlier version of FirstSpirit, it is stringently required to perform a new export with the option `deleteObsoleteFiles` activated. (If **FSDevTools** are used (recommended best practice), this option is activated by default.)

The directory `.FirstSpirit` at the highest directory level contains internal meta information for successful synchronization of external content with the FirstSpirit project. **When using a version control system, this directory should not be versioned!**

Furthermore, the contents of the files `Info.txt` and `Files.txt`, which also contain internal meta information, have been changed, and the files have been renamed to `FS_Info.txt` and `FS_Files.txt`, respectively. In this context, the format of the file `Entities.xml`, which contains contents of datasets, has been changed as well.

4.6 Import: Abort in case of required but missing objects

Many objects in FirstSpirit projects have dependencies to other projects, also called “references”. Among others, one goal of a successful use of “FirstSpirit External Synchronization” is to export partial trees from one project into another which will be functional in the target project. This means that the objects referenced by an exported object must be exported as well. There are two types of dependencies:

- **Required missing references:** referenced objects which are necessary in order to export a complete and functional partial tree of a project. E.g. if a page is exported, the templates upon which the page is based are required so that the page can be displayed in another FirstSpirit project following an import. The user must ensure that the required dependencies are present in the target project.

- **Optional missing references:** referenced objects which are referenced by an exported object but which are not necessary in order to export a complete and functional partial tree of a project. E.g. if a page is exported, the media objects referenced by this page are not necessary to display the page itself in the target project following an import and thus do not have to be included in the export/import.

During an import, dependencies will now be checked and the import will be aborted if this import would cause required missing dependencies in the target project. In this case, a `ValidationException` will be thrown. This exception provides information about the required missing dependencies in the target project. Until now, the import was carried out without throwing an exception.

In the course of this behavioral change, the FirstSpirit Developer API interfaces

- `FeatureError` (package `de.espirit.firstspirit.store.access.feature`)

- `SyncOperationException` (package `de.espirit.firstspirit.store.access.nexport.exceptions`)

have been extended.

4.7 Import: Deletion of objects

Deletion of objects in one project will now be correctly regarded during an import into other projects. This means that objects which have been deleted in the project of developer A will now also be deleted in the project of developer B during an import of the respective state.

If there are still required inbound references to such deleted objects, these objects will be moved to the folder “Lost & Found”. (Required references are dependent objects which are required for the functionality of an object in the target project, e.g. the page template (Templates store), which is required for the existence of a page (Page Content store), or the page (Page Content store), which is required for the existence of a page reference (Site Structure store).

Previously, objects deleted in the source project were not deleted in the target project upon import.

This behavior now also applies to the functionality “FirstSpirit ContentTransport”. Here, objects which did not have required inbound references were moved to the folder “Lost & Found”. Now, they will be deleted.

4.8 Import: Identification of new and modified datasets using API (interface "ImportOperation.Result")

As of the current version of FirstSpirit, the method `getCreatedEntities()` of the interface `ImportOperation.Result` (FirstSpirit Developer API, package `de.espirit.firstspirit.store.access.nexport.operations`) has been optimized: when importing datasets, it now only considers datasets which have been newly created since the last import. Previously, it also considered modified datasets.

Modified datasets may now be identified using the newly implemented method `getUpdatedEntities()`. Datasets which have not been modified since the last import will now not be considered by the methods `getCreatedEntities()` and `getUpdatedEntities()` during a subsequent import.

For datasets from tables with 1:N or M:N relations, the methods `getCreatedEntities()` and `getUpdatedEntities()` will always consider **both** datasets of the relation (thus, a minimum of two datasets).

When using **FSDevTools** (recommended best practice), this information will already be evaluated and displayed in the log.

5 Module Development, Scripts, API

5.1 Changes to GOM implementation class "GomButton"

In the GOM implementation class `GomButton`, the type of the attributes `onClick` and `onDrop` has been changed from `String` to `GomButtonScriptReference`. Programmatic uses of this class may require modifications to their source code in order to observe this type change.

6 System

6.1 Integrated software

Updated the integrated Chrome browser engine “JxBrowser” to version 6.15.

6.2 Updating Apache FOP

Apache FOP (“Formatting Objects Processor”) enables, for example, FirstSpirit content to be output in PDF format. A module is used to provide support for Apache FOP in FirstSpirit: *fs-fop.fsm*.
In the current version of FirstSpirit, the version of the Apache FOP used internally was updated from 1.1 to 2.2.

7 Overview

ID	Description	Categories
172092	Deletion of objects in one project will now be correctly regarded during an import into other projects. This also affects “FirstSpirit Content-Transport”. Further information can be found in chapter “External Synchronization: Import: Deletion of objects”.	Content Transport, External synchronization

ID	Description	Categories
173139	In the current version of FirstSpirit, the version of the Apache FOP used internally was updated from 1.1 to 2.2. Further information can be found in chapter “System: Updating Apache FOP”.	Integrated software
194449	During an export, the folder “Lost & Found” will no longer be exported. This behavior may be controlled via a new API method. Further information can be found in chapter “External Synchronization: Export: Folder "Lost & Found" will no longer be exported”.	External synchronization
196051 194654 194798 196699	In the course of export format readjustment, an export will now only create one directory named `.FirstSpirit` instead of creating several directories of that name. Furthermore, the contents of the files `Info.txt` and `Files.txt`, which also contain internal meta information, have been changed, and the files have been renamed to `FS_Info.txt` and `FS_Files.txt`, respectively. In this context, the format of the file `Entities.xml`, which contains contents of datasets, has been changed as well. Further information can be found in chapter “External Synchronization: General changes to the data format”.	External synchronization
196052	In addition to providing information about files which were created in, deleted from, or updated in the file system during an export, the interface `ExportOperation.Result` now includes newly implemented methods which provide information about moved files as well as other information (FirstSpirit Developer API, package `de.espirit.firstspirit.store.access.nexport.operations`, interface `ExportOperation.Result`). Further information can be found in chapter “External Synchronization: Export: Additional information about exported files (interface "ExportOperation.Result")”.	External synchronization

ID	Description	Categories
197564	During an import into a target project, implicitly exported objects will now only be created in the target project if they are not present there and will no longer be updated if they are already present in the target project. Further information can be found in chapter “External Synchronization: Export: Explicitly and implicitly selected objects”.	External synchronization
197994	If meta files are missing (empty directory structures), an error message will be output if an export or an import is attempted, and the export/import will be prevented. If `deleteObsoleteFiles` is set to “true”, an export will only log a warning. Further information can be found in chapter “External Synchronization: Detection of empty directory structures (missing meta files)”.	External synchronization
198064	Changes to the name of a server-wide action template is now taken into account during use for references within project schedules.	ServerManager
198687	When importing datasets, it is now possible to distinguish between newly created and modified datasets by using the newly implemented method `getUpdatedEntities()` in the interface `ImportOperation.Result`. Further information can be found in chapter “External Synchronization: Import: Identification of new and modified datasets using API (interface "ImportOperation.Result")”.	External synchronization
200278	“External Synchronization” no longer updates/modifies database schemata for which the options “Read-only” and “No schema sync” are set.	External synchronization
200361	During an import, dependencies will now be checked and the import will be aborted if this import would cause required missing dependencies in the target project. Further information can be found in chapter “External Synchronization: Import: Abort in case of required but missing objects”.	External synchronization

ID	Description	Categories
200507	In the GOM implementation class `GomButton`, the type of the attributes `onClick` and `onDrop` has been changed from `String` to `GomButtonScriptReference`. Programmatic uses of this class may require modifications to their source code in order to observe this type change. Further information can be found in chapter “Module Development, Scripts, API: Changes to GOM implementation class "GomButton"”.	Developer, FirstSpirit API, Input Components
200726	The evaluation of the configuration for the integrated preview has been improved.	SiteArchitect
201174	Improved deletion of elements for which edit mode is activated.	SiteArchitect
201906	Malfunctions may have occurred when changing the project description in the project properties.	ServerManager
202264	In “External Synchronization”, optimized observation of structure variables.	External synchronization
202407	Optimized evaluation of invalid references when saving datasets.	SiteArchitect
202644	Optimized synchronization of edit status between different views of the same dataset.	Content Store, SiteArchitect
202921	When trying to import a new schema into a project without specifying a suitable layer mapping, an error message is output. Additionally, the import will now also be aborted at the correct time.	External synchronization
202959	Optimized uploading of files of type `.json`.	SiteArchitect
203220	When running on the operating system macOS High Sierra (macOS version 10.13), erroneous behavior could occur during use of the integrated preview. An update of the integrated Chrome browser engine “JxBrowser” to version 6.15 corrected this issue. Further information can be found in chapter “System: Integrated software”.	Preview, SiteArchitect

8 Categories

8.1 Content Store

ID	Description
202644	Optimized synchronization of edit status between different views of the same dataset.

8.2 Content Transport

ID	Description
172092	Deletion of objects in one project will now be correctly regarded during an import into other projects. This also affects “FirstSpirit ContentTransport”. Further information can be found in chapter “External Synchronization: Import: Deletion of objects”.

8.3 Developer

ID	Description
200507	In the GOM implementation class `GomButton`, the type of the attributes `onClick` and `onDrop` has been changed from `String` to `GomButtonScriptReference`. Programmatic uses of this class may require modifications to their source code in order to observe this type change. Further information can be found in chapter “Module Development, Scripts, API: Changes to GOM implementation class "GomButton"”.

8.4 External synchronization

ID	Description
172092	Deletion of objects in one project will now be correctly regarded during an import into other projects. This also affects “FirstSpirit ContentTransport”. Further information can be found in chapter “External Synchronization: Import: Deletion of objects”.

ID	Description
194449	During an export, the folder “Lost & Found” will no longer be exported. This behavior may be controlled via a new API method. Further information can be found in chapter “External Synchronization: Export: Folder "Lost & Found" will no longer be exported”.
196051 194654 194798 196699	In the course of export format readjustment, an export will now only create one directory named `.FirstSpirit` instead of creating several directories of that name. Furthermore, the contents of the files `Info.txt` and `Files.txt`, which also contain internal meta information, have been changed, and the files have been renamed to `FS_Info.txt` and `FS_Files.txt`, respectively. In this context, the format of the file `Entities.xml`, which contains contents of datasets, has been changed as well. Further information can be found in chapter “External Synchronization: General changes to the data format”.
196052	In addition to providing information about files which were created in, deleted from, or updated in the file system during an export, the interface `ExportOperation.Result` now includes newly implemented methods which provide information about moved files as well as other information (FirstSpirit Developer API, package `de.espirit.firstspirit.store.access.nexport.operations`, interface `ExportOperation.Result`). Further information can be found in chapter “External Synchronization: Export: Additional information about exported files (interface "ExportOperation.Result")”.
197564	During an import into a target project, implicitly exported objects will now only be created in the target project if they are not present there and will no longer be updated if they are already present in the target project. Further information can be found in chapter “External Synchronization: Export: Explicitly and implicitly selected objects”.
197994	If meta files are missing (empty directory structures), an error message will be output if an export or an import is attempted, and the export/import will be prevented. If `deleteObsoleteFiles` is set to “true”, an export will only log a warning. Further information can be found in chapter “External Synchronization: Detection of empty directory structures (missing meta files)”.

ID	Description
198687	When importing datasets, it is now possible to distinguish between newly created and modified datasets by using the newly implemented method `getUpdatedEntities()` in the interface `ImportOperation.Result`. Further information can be found in chapter "External Synchronization: Import: Identification of new and modified datasets using API (interface "ImportOperation.Result")".
200278	"External Synchronization" no longer updates/modifies database schemata for which the options "Read-only" and "No schema sync" are set.
200361	During an import, dependencies will now be checked and the import will be aborted if this import would cause required missing dependencies in the target project. Further information can be found in chapter "External Synchronization: Import: Abort in case of required but missing objects".
202264	In "External Synchronization", optimized observation of structure variables.
202921	When trying to import a new schema into a project without specifying a suitable layer mapping, an error message is output. Additionally, the import will now also be aborted at the correct time.

8.5 FirstSpirit API

ID	Description
200507	In the GOM implementation class `GomButton`, the type of the attributes `onClick` and `onDrop` has been changed from `String` to `GomButtonScriptReference`. Programmatic uses of this class may require modifications to their source code in order to observe this type change. Further information can be found in chapter "Module Development, Scripts, API: Changes to GOM implementation class "GomButton"".

8.6 Input Components

ID	Description
200507	In the GOM implementation class `GomButton`, the type of the attributes `onClick` and `onDrop` has been changed from `String` to `GomButtonScriptReference`. Programmatic uses of this class may require modifications to their source code in order to observe this type change. Further information can be found in chapter “Module Development, Scripts, API: Changes to GOM implementation class "GomButton"”.

8.7 Integrated software

ID	Description
173139	In the current version of FirstSpirit, the version of the Apache FOP used internally was updated from 1.1 to 2.2. Further information can be found in chapter “System: Updating Apache FOP”.

8.8 Preview

ID	Description
203220	When running on the operating system macOS High Sierra (macOS version 10.13), erroneous behavior could occur during use of the integrated preview. An update of the integrated Chrome browser engine “JxBrowser” to version 6.15 corrected this issue. Further information can be found in chapter “System: Integrated software”.

8.9 ServerManager

ID	Description
198064	Changes to the name of a server-wide action template is now taken into account during use for references within project schedules.
201906	Malfunctions may have occurred when changing the project description in the project properties.

8.10 SiteArchitect

ID	Description
200726	The evaluation of the configuration for the integrated preview has been improved.
201174	Improved deletion of elements for which edit mode is activated.
202407	Optimized evaluation of invalid references when saving datasets.
202644	Optimized synchronization of edit status between different views of the same dataset.
202959	Optimized uploading of files of type `.json`.
203220	When running on the operating system macOS High Sierra (macOS version 10.13), erroneous behavior could occur during use of the integrated preview. An update of the integrated Chrome browser engine “JxBrowser” to version 6.15 corrected this issue. Further information can be found in chapter “System: Integrated software”.