



White Paper

Fabasoft Integration for CMIS

Fabasoft Folio 2022

Copyright © Fabasoft R&D GmbH, Linz, Austria, 2022.

All rights reserved. All hardware and software names used are registered trade names and/or registered trademarks of the respective manufacturers.

No rights to our software or our professional services, or results of our professional services, or other protected rights can be based on the handing over and presentation of these documents.

Contents

1 Introduction	4
2 Software Requirements	4
3 Installation of the Fabasoft Integration for CMIS	4
4 Configuration of the Fabasoft Integration for CMIS	4
4.1 Configuration of Java Virtual Machine Options	5
5 Configuration of CMIS Compliant Products	5
6 Conformance	5
6.1 Data Model	5
6.2 Services	6
6.2.1 Repository Services	6
6.2.2 Navigation Services	6
6.2.3 Object Services	6
6.2.4 Multi-Filing Services	6
6.3 Restful AtomPub Binding	6
6.4 Web Service Binding	7
7 For Developers	7
7.1 Available Repositories	7
7.2 The “aspect” Parameter	7
7.2.1 aspect=true	9
7.2.2 Update of Properties	9
7.2.3 SOAP Message Example	9
7.3 User-Defined Forms	10
7.4 Quota	10
7.5 Synchronized Collection	11
7.6 Additional Properties	11
7.7 Providing Documents	11
7.8 Performance	12
7.9 Additional Arguments	12

1 Introduction

The Fabasoft Integration for CMIS complies with the CMIS specification version 1.0. It meets all of the MUST or REQUIRED level requirements defined within. This document describes the installation and configuration of the Fabasoft Integration for CMIS.

2 Software Requirements

System environment: All information contained in this document implicitly assumes a Microsoft Windows environment or Linux environment.

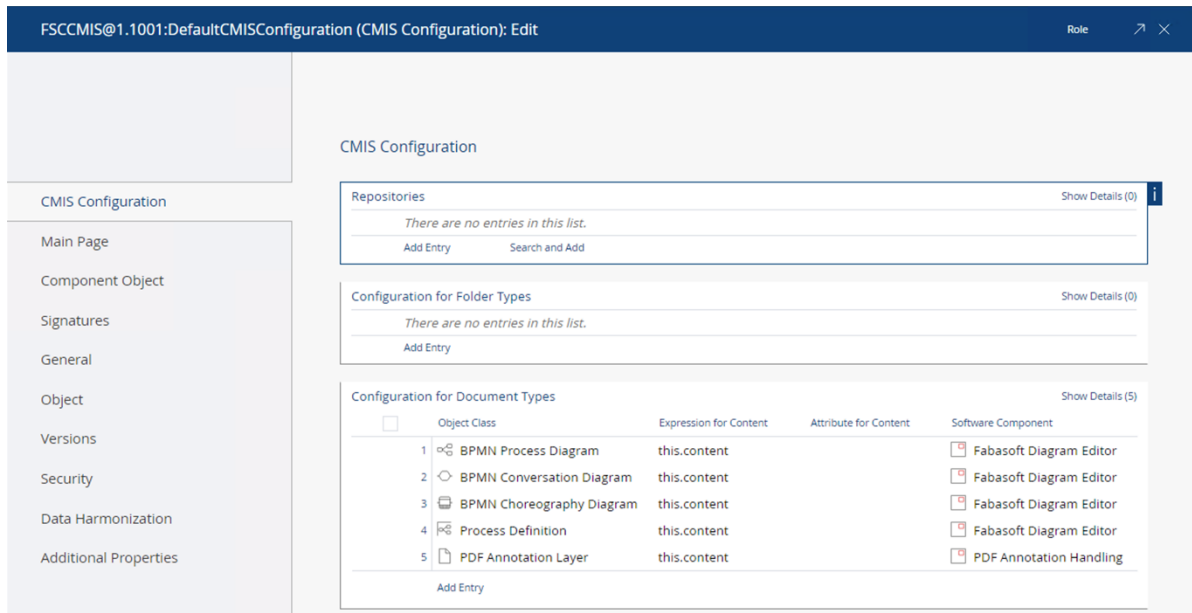
Supported platforms: For detailed information on supported operating systems and software see the software product information on the Fabasoft distribution media.

3 Installation of the Fabasoft Integration for CMIS

The Fabasoft Integration for CMIS will be installed automatically during the installation of the Fabasoft Folio domain. There are no further steps required.

4 Configuration of the Fabasoft Integration for CMIS

The *Fabasoft Integration for CMIS* (FSCCMIS@1.1001) software component provides the *Configuration for CMIS* (FSCCMIS@1.1001:DefaultCMISConfiguration) configuration object. The following fields are available:



The screenshot shows the configuration interface for CMIS Configuration. The title bar reads "FSCCMIS@1.1001:DefaultCMISConfiguration (CMIS Configuration): Edit". The interface is divided into a left sidebar and a main content area. The sidebar contains a list of navigation items: CMIS Configuration, Main Page, Component Object, Signatures, General, Object, Versions, Security, Data Harmonization, and Additional Properties. The main content area is titled "CMIS Configuration" and contains three sections:

- Repositories:** A section with the text "There are no entries in this list." and buttons for "Add Entry" and "Search and Add". A "Show Details (0)" link is also present.
- Configuration for Folder Types:** A section with the text "There are no entries in this list." and an "Add Entry" button. A "Show Details (0)" link is also present.
- Configuration for Document Types:** A table with columns for "Object Class", "Expression for Content", "Attribute for Content", and "Software Component". The table contains five entries:

	Object Class	Expression for Content	Attribute for Content	Software Component
1	BPMN Process Diagram	this.content		Fabasoft Diagram Editor
2	BPMN Conversation Diagram	this.content		Fabasoft Diagram Editor
3	BPMN Choreography Diagram	this.content		Fabasoft Diagram Editor
4	Process Definition	this.content		Fabasoft Diagram Editor
5	PDF Annotation Layer	this.content		PDF Annotation Handling

An "Add Entry" button is located at the bottom of the table. A "Show Details (5)" link is also present.

- **Repositories**

The desk of the user is a repository by default. Additional repositories can be defined in this field. By default, only instances of the object class *Folder* can be specified. Instances of further object classes can be used once they are configured in the *Configuration for Folder Types* section.

- *Configuration for Folder Types*
To use instances of object classes as CMIS folders the following parameters have to be defined for the desired object class:
 - *Expression for Child Objects*
Define an expression which returns a list of child objects. This expression will be used when reading the child elements of an instance of the configured object class.
 - *Attribute for Child Objects*
Define a property of type `COOSYSTEM@1.1:OBJECTLIST` which will be used when adding new child elements or deleting existing elements.
- *Configuration for Document Types*
By default only instances of `COOSYSTEM@1.1:ContentObject` or derived classes can be used as CMIS documents. To use instances of other object classes you need to define the following parameters for the desired object class:
 - *Expression for Content*
Define an expression which returns a content of type `COOSYSTEM@1.1:Content` which will be used when reading the content of an instance of the configured object class.
 - *Attribute for Content*
Define a property of type `COOSYSTEM@1.1:Content` which will be used when modifying the content of instances of the configured object class.

4.1 Configuration of Java Virtual Machine Options

It may be necessary to adapt the Java Virtual Machine options on the Fabasoft Folio web server to handle large files without running out of memory. Therefore, it is possible to change the provided heap space by creating the environment variable `COOJAVA_JVMOPTIONS`. Set this variable to `-Xms128m -Xmx512m` for a minimum of 128 MB and a maximum of 512 MB.

5 Configuration of CMIS Compliant Products

Repository access is possible via two different CMIS protocol bindings:

- RESTful AtomPub binding
- Web Services binding

Both bindings can be used by setting an endpoint with the following structure:

```
https://<webservice>[<port>]/<vdir>/cmis
```

Example: `https://at.cloud.fabasoft.com/folio/cmis`

6 Conformance

The Fabasoft Integration for CMIS conforms to the CMIS specification and meets all of the MUST and REQUIRED level requirements defined within this specification. Supported requirements are listed in the next chapters.

6.1 Data Model

- The repository identity must uniquely identify an available repository at the service endpoint.
- A CMIS repository must have these base types:

- o cmis:document object-type
- o cmis:folder object-type
- The root folder is a special folder insofar as it cannot be created, deleted, or moved using CMIS services.

6.2 Services

6.2.1 Repository Services

- getRepositories
- getRepositoryInfo
- getTypeChildren
- getTypeDescendants
- getTypeDefinition

6.2.2 Navigation Services

- getChildren
- getFolderParent
- getObjectParents

6.2.3 Object Services

- createDocument
- createDocumentFromSource
- createFolder
- getAllowableActions
- getObject
- getProperties
- getObjectByPath
- getContentStream
- updateProperties
- moveObject
- deleteObject
- deleteTree
- deleteContentStream

6.2.4 Multi-Filing Services

- addObjectToFolder
- removeObjectFromFolder

6.3 Restful AtomPub Binding

- Implementations of CMIS must be compliant with Atom (RFC4287) and Atom Publishing Protocol (RFC5023).

- Exceptions must be mapped to the appropriate HTTP status code defined in the CMIS domain model.
- CMIS link relations must be included if the client can perform the operation.
- Any feed must be a valid Atom Feed document.
- Repositories must provide URI templates.

6.4 Web Service Binding

- A CMIS web services binding must comply with WS-I Basic Profile 1.1 and Basic Security Profile 1.0.

7 For Developers

If you want to write your own CMIS client, profound knowledge about the CMIS standard is mandatory. As a starting point the OpenCMIS client examples may be helpful:

<https://chemistry.apache.org/java/examples/>

Note: Apache Chemistry does not support aspects.

7.1 Available Repositories

The available repositories can be configured the following way:

- The desk of the user is a repository by default.
- The repositories defined in the CMIS configuration (see chapter 4 “Configuration of the Fabasoft Integration for CMIS”).
- The repositories returned by the use case `FSCCMIS@1.1001:CMISFilterRepositories`.

Syntax

```

override CMISFilterRepositories{
  variant User {
    impl = expression {
      RepositoryFilterType repo1 = coort.CreateAggregate(#RepositoryFilterType);
      repo1.repositorybaseurl = "https://s1.fabasoft.com/fsc/cmisis";
      repo1.repositoryid = "root";
      repo1.repositoryname = "S1";

      RepositoryFilterType repo2 = coort.CreateAggregate(#RepositoryFilterType);
      repo2.repositorybaseurl = "https://s2.fabasoft.com/fsc/cmisis";
      repo2.repositoryid = "root";
      repo2.repositoryname = "S2";

      repositories += repo1;
      repositories += repo2;
    }
  }
}

```

7.2 The “aspect” Parameter

The aspect parameter can be used to retrieve all properties of an object that are defined on a specified form and that contain a value. Additionally, the standard properties `atom:icon`, `cmis:baseTypeId` and `cmis:createdBy` are available.

Usage:

To get a list of all child objects following URL can be used:

```
https://<webserver>/<vdir>/cmis/<repository>/<object>/children?aspect=<action>
```

To get the values of all available properties (that contain a value) the following URL can be used:

```
https://<webserver>/<vdir>/cmis/<repository>/<object>/entry?aspect=<action>
```

The value of the `aspect` parameter is matched with the value of the *Action* property. The *Action* property can be defined on the desired object class, on the "User Interface" tab in the *Forms* property. The matching action determines the used form. The form contains form pages. All properties of a basic type (string, integer, date, contents and so on) that are assigned to the form pages and that contain a value are included in the response XML. Properties that contain no value are not included in the XML due to performance reasons.

In the following example properties of the form that matches the `COOATTREDIT@1.1:EditObjectAttributes` action are returned.

Example

```
http://localhost/fsc/cmis/COO.1.3285.1.1346/COO.1.3285.4.945/entry?aspect=COOATTREDIT_1_1_EditObjectAttributes
```

In the following example the properties `objname` and `objversmaxnr` of the object `COO.1.3285.4.945` are set.

Example

```
curl.exe -X PUT -u "user":"passwd"  
"http://localhost/fsc/cmis/COO.1.3285.1.1346/COO.1.3285.4.945/entry?aspect=COOATTREDIT_1_1_EditObjectAttributes" -d @changeprops.xml
```

changeprops.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:cmisra="http://docs.oasis-open.org/ns/cmis/restatom/200908/" xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/">  
  <cmisra:object>  
    <cmis:properties>  
      <cmis:propertyString propertyDefinitionId="COOSYSTEM@1.1:objname">  
        <cmis:value>My object name</cmis:value>  
      </cmis:propertyString>  
      <cmis:propertyInteger propertyDefinitionId="COOSYSTEM@1.1:objversmaxnr">  
        <cmis:value>3</cmis:value>  
      </cmis:propertyInteger>  
    </cmis:properties>  
  </cmisra:object>  
</entry>
```

In the following example a Microsoft Word document is created directly on the desk. The name "My document" and the subject "My subject" are set.

Example

```
curl.exe -X POST -u "user":"passwd"  
"http://localhost/fsc/cmis/COO.1.3285.1.1346/COO.1.3285.1.1346/children?aspect=COOATTREDIT_1_1_EditObjectAttributes" -d @createdoc.xml
```

createdoc.xml


```
<?xml version="1.0" encoding="utf-8"?>
<entry xmlns="http://www.w3.org/2005/Atom" xmlns:cmisra="http://docs.oasis-
open.org/ns/cmris/restatom/200908/" xmlns:cmis="http://docs.oasis-
open.org/ns/cmris/core/200908/">
  <cmisra:object>
    <cmis:properties>
      <cmis:propertyString propertyDefinitionId="cmis:name">
        <cmis:value>My document</cmis:value>
      </cmis:propertyString>
      <cmis:propertyId propertyDefinitionId="cmis:objectTypeId">
        <cmis:value>COOMSOFFICE@1.1:WinWordObject</cmis:value>
      </cmis:propertyId>
      <cmis:propertyId propertyDefinitionId="cmis:baseTypeId">
        <cmis:value>cmis:document</cmis:value>
      </cmis:propertyId>
      <cmis:propertyString propertyDefinitionId="COOSYSTEM@1.1:objsubject">
        <cmis:value>My subject</cmis:value>
      </cmis:propertyString>
    </cmis:properties>
  </cmisra:object>
</entry>
```

In the following example an object should be created in a folder that consists of several object lists. The document should be created in the folder COO.1.3285.4.934 in the object list COO.1.1.1.1907.

Example

```
curl.exe -X POST -u "user":"passwd"
"http://localhost/fsc/cmris/COO.1.3285.1.1346/COO.1.3285.4.934$COO.1.1.1.1907/
children?aspect=COOATTREDIT_1_1_EditObjectAttributes" -d @createdoc.xml
```

7.2.1 aspect=true

The `aspect=true` parameter is only for internal use. It reduces the amount of data that is transferred to the client (the response XML contains only the most important information). This setting does not conform to the CMIS specification.

Note: The URL pointing to the PDF overview is included for objects without content.

7.2.2 Update of Properties

The CMIS service "updateProperties" can be used to modify properties that are defined in the aspect.

7.2.3 SOAP Message Example

The following example shows how to use the aspect parameter in a SOAP message.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://docs.oasis-open.org/ns/cmris/messaging/200908/"
xmlns:fsc="http://www.fabasoft.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getChildren>
      <ns:repositoryId>COO.1.506.1.215</ns:repositoryId>
      <ns:folderId>COO.1.506.1.345</ns:folderId>
      <ns:extension>
        <fsc:aspect>true</fsc:aspect>
      </ns:extension>
    </ns:getChildren>
  </soapenv:Body>
</soapenv:Envelope>
```

```
</ns:getChildren>
</soapenv:Body>
</soapenv:Envelope>
```

7.3 User-Defined Forms

The properties of user-defined forms can be accessed like any other property.

When you create a “Container With User Data” or “Room With User Data” containing one list or several lists on the same form page the first list can be retrieved the following way.

Example

```
curl.exe -u "user":"passwd" "http://localhost/fsc/cmisis/root/COO.1.506.3.4498/children"
```

If several lists on different form pages are available the \$ operator can be used to access each list.

Example

Provides an overview of all available lists:

```
curl.exe -u "user":"passwd" "http://localhost/fsc/cmisis/root/COO.1.506.3.4539/children"
```

As a result, the following entries are returned:

```
<id>cmisrep:entry:COO.1.506.3.4539$COO.1.506.1.8467</id>
<id>cmisrep:entry:COO.1.506.3.4539$COO.1.506.1.8469</id>
<id>cmisrep:entry:COO.1.506.3.4539$COO.1.506.1.8471</id>
```

The following way one of these lists can be accessed:

```
curl.exe -u "user":"passwd" "http://localhost/fsc/cmisis/root/COO.1.506.3.4539$COO.1.506.1.8469/children"
```

Note: Depending on the used shell, the \$ sign may need to be escaped (\\$).

In the following example, properties of the form that matches the COOATTREDIT@1.1:EditObjectAttributes action are returned. The properties of the user-defined form are included.

Example

```
curl.exe -u "user":"passwd" "http://localhost/fsc/cmisis/root/COO.1.506.3.4534/entry?aspect=COOATTREDIT_1_1_EditObjectAttributes"
```

7.4 Quota

If a user quota is defined for a Fabasoft Folio Domain, the data is also available in the response XML of the following call:

```
https://<webserver>[<port>]/<vdir>/cmis
```

- `fsc:maxquota`
Maximum available storage space (in MB) for the user.
- `fsc:curquota`
Currently used storage space (in KB) by the user.
- `fsc:maxobjcount`
Maximum available number of objects for the user.

- `fsc:curobjcount`
Number of objects created by the user.
- `fsc:maxcontsize`
Maximum size (in bytes) of a single document.

Example:

```
<fsc:properties>
  <cmis:property propertyDefinitionId="fsc:maxquota">
    <cmis:value>10240</cmis:value>
  </cmis:property>
  <cmis:property propertyDefinitionId="fsc:maxobjcount">
    <cmis:value>10000</cmis:value>
  </cmis:property>
  <cmis:property propertyDefinitionId="fsc:curobjcount">
    <cmis:value>115</cmis:value>
  </cmis:property>
  <cmis:property propertyDefinitionId="fsc:curquota">
    <cmis:value>1701483</cmis:value>
  </cmis:property>
  <cmis:property propertyDefinitionId="fsc:maxcontsize">
    <cmis:value>2147483647</cmis:value>
  </cmis:property>
</fsc:properties>
```

7.5 Synchronized Collection

The following URL can be used to get a collection of all objects that are marked as synchronized by a user:

`https://<webserver>[<port>]/<vdir>/cmis/<repository>/-/synchronize`

7.6 Additional Properties

Following properties are available:

- The property `fsc:objTeamRoom` returns the Teamroom (COO address) of a document or folder.
- The property `fsc:objDeleted` returns `true` if the object is in the wastebasket.
- The property `fsc:canSetContentStream` determines whether the object can be modified.
- The property `fsc:contentStreamEncryption` determines the encryption method of the object.

The properties `canSetContentStream` and `contentStreamEncryption` are only included in the response XML if the object is modifiable or encrypted.

Example:

```
<cmis:propertyBoolean propertyDefinitionId="fsc:canSetContentStream">
  <cmis:value>true</cmis:value>
</cmis:propertyBoolean>

<cmis:propertyString propertyDefinitionId="fsc:contentStreamEncryption">
  <cmis:value>AES 256 CBC</cmis:value>
</cmis:propertyString>
```

7.7 Providing Documents

The available document types are defined in the *Web Service Configuration*:

- *Object Classes for Newly Created Content Objects*
Objects of object classes in this list are included.
- *Object Classes Allowed in Web Folders*
Objects of object classes in this list are included. If *Show PDF Summary* is set to “Yes” a PDF overview is generated. Additionally, for objects not in this list (and that are not excluded) a PDF overview is generated.
- *Object Classes Not Allowed in Web Folders*
Objects of object classes in this list are excluded.

7.8 Performance

Please be aware that some operations like getting the folder tree of a large structure may exceed the system resources.

7.9 Additional Arguments

The following additional arguments are available:

Parameter `needChoices`

To reduce the amount of data generated choices are no longer provided by default, starting with version 2019. If these additional choices are needed the URL parameter `needChoices=true` may be added.