Fabasoft

White Paper

Fabasoft Integration for Fabasoft Mindbreeze Enterprise

Fabasoft Folio 2026 Update Rollup 2

Copyright © Fabasoft R&D GmbH, Linz, Austria, 2026. 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 4

3.1 Fabasoft Mindbreeze Enterprise 4

3.2 Fabasoft Integration for Fabasoft Mindbreeze Enterprise 4

4 Configuration 11

4.1 Fabasoft Mindbreeze Enterprise Configuration 11

4.2 Indexing Service 14

4.3 Filter Service 16

4.4 OAuth Authentication 17

4.5 Fabasoft Folio Stores 19

4.6 Indexing of Archived Contents and Objects 19

4.7 Modification of the timeout properties 19

4.8 Indexing via UNC Paths 20

4.9 Optimized Full-Text Queries 20

4.10 ACL Check Optimization 20

4.11 Custom Forms for Hit Display/Preview 21

4.12 Custom Context Menus in the Mindbreeze Web Client 21

4.13 Mindbreeze Web Client Language 21

4.14 Displaying Search Results in the Embedded Fabasoft Mindbreeze Enterprise Web Client 22

4.15 Mindbreeze Web Client Authentication 22

5 Configuration of an Automated Task 23

6 Indexing und Searching 23

6.1 Manual Indexing 23

6.2 Rebuilding an Index 24

6.2.1 Online Rebuild 24

6.2.2 Index-Based Recovery 24

6.3 Re-Indexing 25

6.4 Search 25

6.4.1 Searching in Metadata 25

6.4.2 Search in Contents 25

6.4.3 Quick Search 25
7 Scripts for Indexing 25
7.1 Script “Build Index 4 Threads” 26
7.2 Script “Recover Index 4 Threads” 26
7.3 Script “Delta Indexing” 27
7.4 Script “Build Index 4 Threads” (Limit by Date) 29
7.5 Script “Recover Contents 4 Threads” (New Filter) 30
7.6 Script “Recover Objects 4 Threads” (New Property) 30
8 Indexing Performance 31

1 Introduction

This document describes the installation and configuration of Fabasoft Integration for Fabasoft Mindbreeze Enterprise. The integration enables the use of Fabasoft Mindbreeze Enterprise as search tool in Fabasoft products.

2 Software Requirements

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

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

Descriptions in this document are based on following software:

- Fabasoft Mindbreeze Enterprise

For the integration, the Mindbreeze SDK 2025 is used. According to the Mindbreeze software product information, versions can be used, that are supported by this SDK.

3 Installation

3.1 Fabasoft Mindbreeze Enterprise

For the integration of Fabasoft Mindbreeze Enterprise the installation of the third-party product Fabasoft Mindbreeze Enterprise is required.

1. Install Fabasoft Mindbreeze Enterprise as described in the manual of the third-party product.
2. In a Microsoft Windows system environment the server nodes of Fabasoft Mindbreeze Enterprise that are configured as search service have to be run as a dedicated user.

Note: Following indexing service capabilities are required: `aclreferences`, `startandcountinlegacyxml` and `fsc:aclobjects`.

3.2 Fabasoft Integration for Fabasoft Mindbreeze Enterprise

The web service based integration communicates with an existing or dedicated Fabasoft Folio web service for processing Fabasoft Mindbreeze Enterprise search requests.

To install the web service based integration, perform the following steps:

1. Create index and filter services in Fabasoft Mindbreeze Enterprise (Advanced / Indices, Advanced / Filters).
2. Configure the integration in Fabasoft Folio as outlined in chapter 4 . Be sure to configure a *User with Access to Integration Web Service*.
3. Prepare a Fabasoft Mindbreeze Enterprise plugin ZIP archive that contains the following files:

Note: If the installation uses a different category to isolate multiple independent Fabasoft Folio installations that share a Mindbreeze Enterprise installation, the name of the ZIP archive and the IDs of plugins (Plugin/id) must not match that of other Fabasoft Folio installations. Additionally, the default category "Fabasoft Components" must be changed to the configured category in the plugin resources.

- `plugins.xml` (plugin description)
  A Fabasoft Mindbreeze Enterprise plugins XML file that describes the web service based integration. In the XML, the element `location` defines the URL of the target web service:
  `<location>https://<hostname>/<vdir>/mesapi</location>`
  Credentials required for connecting to the URL must be registered in Fabasoft Mindbreeze Enterprise (Advanced / Network / Credentials) and assigned to a new endpoint (Advanced / Network / Endpoints) corresponding to the web service.
  The plugin `SessionValidationService` is not required in case of a Fabasoft Cloud installation.

```
<?xml version="1.0"?>
<Plugins>
  <version>
  </version>
  <id>Fabasoft Components</id>
  <plugins>
    <Plugin>
      <kind>RESOURCE</kind>
      <extension>
        <Extension>
          <point>com.mindbreeze.datasource.CategoryDescriptor</point>
        </Extension>
      </extension>
      <resource_path>categorydescriptor.xml</resource_path>
    </Plugin>
    <Plugin>
      <kind>RESOURCE</kind>
      <extension>
        <Extension>
          <point>com.mindbreeze.datasource.CategoryIcon</point>
        </Extension>
      </extension>
      <resource_path>categoryicon.png</resource_path>
    </Plugin>
    <Plugin>
      <kind>CODE</kind>
      <id>Fabasoft Components</id>
      <extension>
        <Extension>
          <point>mindbreeze.security.PrincipalResolutionService</point>
        </Extension>
      </extension>
      <code>
        <Code>
          <platform>WEBSERVICE</platform>
          <location></location>
          <os>ANY_OS</os>
          <cpu>ANY_CPU</cpu>
          <exectype>SANDBOX</exectype>
```

```
      <properties>
        <KeyValuePair>
          <key>com.mindbreeze.datasource.category</key>
          <value>Fabasoft Components</value>
        </KeyValuePair>
        <KeyValuePair>
          <key>com.mindbreeze.datasource.categoryinstance</key>
          <value>Default</value>
        </KeyValuePair>
      </properties>
    </Code>
  </code>
  <version/>
  <properties>
    <KeyValuePair>
      <key>com.mindbreeze.datasource.category</key>
      <value>Fabasoft Components</value>
    </KeyValuePair>
    <KeyValuePair>
      <key>com.mindbreeze.datasource.categoryinstance</key>
      <value>Default</value>
    </KeyValuePair>
  </properties>
  <config_option/>
</Plugin>
<Plugin>
  <kind>CODE</kind>
  <id>Fabasoft Components</id>
  <extension>
    <Extension>
      <point>mindbreeze.query.security.AuthorizationService</point>
    </Extension>
  </extension>
  <code>
    <Code>
      <platform>WEBSERVICE</platform>
      <location></location>
      <os>ANY_OS</os>
      <cpu>ANY_CPU</cpu>
      <exectype>SANDBOX</exectype>
      <properties>
        <KeyValuePair>
          <key>com.mindbreeze.datasource.category</key>
          <value>Fabasoft Components</value>
        </KeyValuePair>
        <KeyValuePair>
          <key>com.mindbreeze.datasource.categoryinstance</key>
          <value>Default</value>
        </KeyValuePair>
      </properties>
    </Code>
  </code>
  <version/>
  <properties>
    <KeyValuePair>
      <key>com.mindbreeze.datasource.category</key>
      <value>Fabasoft Components</value>
    </KeyValuePair>
    <KeyValuePair>
```

```
      <key>com.mindbreeze.datasource.categoryinstance</key>
      <value>Default</value>
    </KeyValuePair>
  </properties>
  <config_option/>
</Plugin>
<Plugin>
  <kind>CODE</kind>
  <id>Fabasoft Components</id>
  <extension>
    <Extension>
      <point>mindbreeze.query.context.ContextualizationService</point>
    </Extension>
  </extension>
  <code>
    <Code>
      <platform>WEBSERVICE</platform>
      <location></location>
      <os>ANY_OS</os>
      <cpu>ANY_CPU</cpu>
      <exectype>SANDBOX</exectype>
      <properties>
        <KeyValuePair>
          <key>com.mindbreeze.datasource.category</key>
          <value>Fabasoft Components</value>
        </KeyValuePair>
        <KeyValuePair>
          <key>com.mindbreeze.datasource.categoryinstance</key>
          <value>Default</value>
        </KeyValuePair>
      </properties>
    </Code>
  </code>
  <version/>
  <properties>
    <KeyValuePair>
      <key>com.mindbreeze.datasource.category</key>
      <value>Fabasoft Components</value>
    </KeyValuePair>
    <KeyValuePair>
      <key>com.mindbreeze.datasource.categoryinstance</key>
      <value>Default</value>
    </KeyValuePair>
  </properties>
  <config_option/>
</Plugin>
<Plugin>
  <kind>CODE</kind>
  <id>Fabasoft Components</id>
  <extension>
    <Extension>
      <point>mes.ipc.query.datasource.SearchInResolutionService</point>
    </Extension>
  </extension>
  <code>
    <Code>
      <platform>WEBSERVICE</platform>
      <location></location>
      <os>ANY_OS</os>
```

```
        <cpu>ANY_CPU</cpu>
        <exectype>SANDBOX</exectype>
        <properties>
          <KeyValuePair>
            <key>com.mindbreeze.datasource.category</key>
            <value>Fabasoft Components</value>
          </KeyValuePair>
          <KeyValuePair>
            <key>com.mindbreeze.datasource.categoryinstance</key>
            <value>Default</value>
          </KeyValuePair>
        </properties>
      </Code>
    </code>
    <version/>
    <properties>
      <KeyValuePair>
        <key>com.mindbreeze.datasource.category</key>
        <value>Fabasoft Components</value>
      </KeyValuePair>
      <KeyValuePair>
        <key>com.mindbreeze.datasource.categoryinstance</key>
        <value>Default</value>
      </KeyValuePair>
    </properties>
    <config_option/>
  </Plugin>
  <Plugin>
    <kind>CODE</kind>
    <id>SessionAuthenticateService</id>
    <extension>
      <Extension>
        <point>mindbreeze.query.client.SessionAuthenticationService</point>
      </Extension>
    </extension>
    <code>
      <Code>
        <platform>WEBSERVICE</platform>
        <location></location>
        <os>ANY_OS</os>
        <cpu>ANY_CPU</cpu>
        <exectype>SANDBOX</exectype>
        <properties/>
      </Code>
    </code>
    <version/>
    <properties/>
    <config_option/>
  </Plugin>
  <Plugin>
    <kind>CODE</kind>
    <id>SessionNotificationService</id>
    <extension>
      <Extension>
        <point>mindbreeze.query.client.SessionNotificationService</point>
      </Extension>
    </extension>
    <code>
      <Code>
```

```
      <platform>WEBSERVICE</platform>
      <location></location>
      <os>ANY_OS</os>
      <cpu>ANY_CPU</cpu>
      <exectype>SANDBOX</exectype>
      <properties/>
    </Code>
  </code>
  <version/>
  <properties>
    <KeyValuePair>
      <key>BEFORE_START</key>
      <value>true</value>
    </KeyValuePair>
    <KeyValuePair>
      <key>AFTER_START</key>
      <value>true</value>
    </KeyValuePair>
    <KeyValuePair>
      <key>BEFORE_END</key>
      <value>true</value>
    </KeyValuePair>
    <KeyValuePair>
      <key>AFTER_END</key>
      <value>true</value>
    </KeyValuePair>
  </properties>
  <config_option/>
</Plugin>
<Plugin>
  <kind>CODE</kind>
  <id>SessionTransformationService</id>
  <extension>
    <Extension>
      <point>mindbreeze.query.client.SessionTransformationService</point>
    </Extension>
  </extension>
  <code>
    <Code>
      <platform>WEBSERVICE</platform>
      <location></location>
      <os>ANY_OS</os>
      <cpu>ANY_CPU</cpu>
      <exectype>SANDBOX</exectype>
      <properties/>
    </Code>
  </code>
  <version/>
  <properties/>
  <config_option/>
</Plugin>
<Plugin>
  <kind>CODE</kind>
  <id>SessionValidationService</id>
  <extension>
    <Extension>
      <point>mindbreeze.query.client.SessionValidationService</point>
    </Extension>
  </extension>
```

```
      <code>
        <Code>
          <platform>WEBSERVICE</platform>
          <location></location>
          <os>ANY_OS</os>
          <cpu>ANY_CPU</cpu>
          <exectype>SANDBOX</exectype>
          <properties/>
        </Code>
      </code>
      <version/>
      <properties/>
      <config_option/>
    </Plugin>
  </plugins>
</Plugins>
```

- `categorydescriptor.xml` (category descriptor)
  A Fabasoft Mindbreeze Enterprise category descriptor XML file that controls Fabasoft Mindbreeze Enterprise web browser client behavior. For an example, see:

```
<?xml version="1.0" encoding="UTF-8"?>
<category id="Fabasoft Components">
  <name>Fabasoft Folio</name>
  <metadata>
    <metadatum id="objcreatedby" selectable="true">
      <name xml:lang="en">Created by</name>
      <name xml:lang="de">Erzeugt von</name>
    </metadatum>
    <metadatum id="objchangedby" selectable="true">
      <name xml:lang="en">Changed by</name>
      <name xml:lang="de">Geändert von</name>
    </metadatum>
    <metadatum id="objsubject" selectable="false">
      <name xml:lang="en">Subject</name>
      <name xml:lang="de">Betreff</name>
    </metadatum>
    <metadatum id="documentnumber" selectable="false">
      <name xml:lang="en">Document Number</name>
      <name xml:lang="de">Dokumentnummer</name>
    </metadatum>
    <metadatum id="surname" selectable="false">
      <name xml:lang="en">Surname</name>
      <name xml:lang="de">Nachname</name>
    </metadatum>
    <metadatum id="firstname" selectable="false">
      <name xml:lang="en">First Name</name>
      <name xml:lang="de">Vorname</name>
    </metadatum>
    <metadatum id="orgname" selectable="false">
      <name xml:lang="en">Organization</name>
      <name xml:lang="de">Organisation</name>
    </metadatum>
    <metadatum id="street" selectable="false">
      <name xml:lang="en">Street</name>
      <name xml:lang="de">Straße</name>
    </metadatum>
    <metadatum id="zipcode" selectable="false">
      <name xml:lang="en">Zip Code</name>
```

```
        <name xml:lang="de">PLZ</name>
      </metadatum>
      <metadatum id="city" selectable="false">
        <name xml:lang="en">City</name>
        <name xml:lang="de">Ort</name>
      </metadatum>
      <metadatum id="objclass" selectable="false" aggregatable="true"
visible="true">
        <name xml:lang="de">Objektklasse</name>
        <name xml:lang="en">Object Class</name>
      </metadatum>
    </metadata>
</category>
```

- `categoryicon.png` (category icon)
  A 16x16 PNG image file that represents the Fabasoft Folio installation in the Fabasoft Mindbreeze Enterprise web browser client.

4. Install the plugin in Fabasoft Mindbreeze Enterprise (Advanced / Plugins).

4 Configuration

4.1 Fabasoft Mindbreeze Enterprise Configuration

The `FSCMINDBREEZE@1.1001:MindbreezeDefaultConfig` configuration is used to define properties of Fabasoft Folio objects that should be indexed.

1. In the "Domain Administration", click the "Configuration Objects" tab.
2. Open the configuration object `FSCMINDBREEZE@1.1001:MindbreezeDefaultConfig` in edit mode.
3. By default, the configuration provides entries for several object classes and the associated properties. Additionally, an entry for all objects with contents is provided.
   Note: The file extensions of the contents to be indexed must be provided in the "Filter Service" object.

Following fields are available:

- *Enable Metadata Full-Text Search*
  If this field is set to "Yes", all metadata full-text queries are handled by Fabasoft Mindbreeze Enterprise. Otherwise the Fabasoft Folio COO Service handles the metadata full-text queries. For more information on the metadata full-text search see chapter 6.4.1 "Searching in Metadata".
- *Generate External References*
  If this field is set to "Yes", external references will be created automatically. For example, the external reference for the property `COOSYSTEM@1.1:objsubject` will be `_objsubject`. External references of properties not from `COOSYSTEM@1.1` contain the corresponding software component (e.g. `FSCFOLIO@1.1001:addrsurname` will be `_FSCFOLIO_1_1001_addrsurname`).
  Automatically created external references may be overwritten by explicitly defined external references (*Mappings for Metadata* > *Property Description* > *External Reference*).

Note: If you change this field the Fabasoft Mindbreeze Enterprise index must be rebuilt (see chapter 6.2 "Rebuilding an Index").

- *Mappings for Metadata*
  In this field the object classes are defined that should be indexed. Only the properties provided in the mappings are included in the index.
  Note: Objects are only added to the Fabasoft Mindbreeze Enterprise index if this property contains an entry for the respective object classes (or base classes). Properties omitted in this mapping will not be searchable using Fabasoft Mindbreeze Enterprise. Additionally, due to security reasons it is not possible to specify properties protected by special access types.
  - *Object Class*
    In this field you can define the object class containing the properties that should be indexed.
  - *Abstract*
    If an object class is defined as "abstract" the properties of the object class apply to all object classes that are configured in the Fabasoft Mindbreeze Enterprise configuration.
  - *Property Description*
    In this field you have to define the properties of the object class selected in the *Object Class* field that you want to be included in the Fabasoft Mindbreeze Enterprise index.
    - *Property Path*
      The property or path to the property to be indexed (e.g. the *Surname* field of a person).
    - *External Reference*
      External references are keys for the metadata properties. (e.g. the query `firstname:Max` returns all persons with the first name "Max").
      The external reference "mes:date" causes the value of the given date/time property to be used as the primary date of an indexed object or content (taken into account during hit ranking and displayed in the hit display of Mindbreeze Enterprise Search). The default primary date is the modification date of the object or content.
      Note: If multiple definitions exist for an external reference, the first definition that results in an indexable value is used. The processing order is undefined.
      Note: An external reference has to be defined if *As List*, *Narrow* or *Widen* is set to "Yes".
    - *As List*
      If set to „No", objects and object lists are indexed as strings, which means that no context icons or context menus are made available for these fields in the Fabasoft Mindbreeze Enterprise client.
    - *Narrow*
      If set to "Yes", a context menu item is displayed in the Fabasoft Mindbreeze Enterprise client to allow you to narrow the search result list down to search results referencing the same object in this property. This setting is only evaluated when the *As List* field is set to "Yes".
    - *Widen*
      If set to "Yes", a context menu item is displayed in the Fabasoft Mindbreeze Enterprise client to allow you to widen your search to include search results referencing the same object in this property. This setting is only evaluated when the *As List* field is set to "Yes".

- *Aggregatable (Available as Filter in Facets)*
  If set to "Yes", the property represented by *Property Path* is indexed with the aggregatable flag (corresponding to the category descriptor flag) so that it's available for filtering in facets.
- *Do Not Consider Access Types for Search*
  If set to "Yes", the property represented by *Property Path* is searchable despite configured access types in *Access Types for Reading Property*.

- *Software* Component
  In this field you have to provide a software component because otherwise the settings are lost after an update of the Fabasoft Folio Domain.
  Note: Make sure that the used software component contains *Integration for Fabasoft Mindbreeze Enterprise* (`FSCMINDBREEZE@1.1001`) in the *Extended Components* field.

- *Mappings for Contents*
  In this field you can define which content properties of an object class should be indexed.
- *Object Classes Not Searchable*
  In this field you can define object classes which should not be indexed if the base class is defined for indexing.
- *Use Domain Constraints*
  If set to "Yes", you can limit the search in the Fabasoft Mindbreeze Enterprise client to a certain Fabasoft Folio Domain or Fabasoft Folio Tenant.
- *Use Scope Constraints*
  If set to "Yes", you can limit the search in the Fabasoft Mindbreeze Enterprise client to a certain area.
- *Predefined Constraints*
  In this field you can provide predefined constraints for filtering search results using the Fabasoft Mindbreeze Enterprise client.
  Example: You want to define a predefined constraint for limiting search results to instances of typical Microsoft Office object class*es*. To implement this predefined constraint, add a new entry to the *Predefined Constraints* field and select an appropriate component object (e.g. `COOMSOFFICE@1.1:CategoryMSOffice` an instance of class `COOTC@1.1001:AppCategory`) in the *Object for Name and Icon* field. In the *Constraint* field, define the object classes for that constraint "`categoryclass:COO.1.1.1.1264 categoryclass:COO.1.1.1.1265 categoryclass:COO.1.1.1.1266 …`", which will be passed on to Fabasoft Mindbreeze Enterprise for filtering the search result list. Thus, if the *Microsoft Office* predefined constraint is selected in the Fabasoft Mindbreeze Enterprise client, the search result list will be restricted to instances of these object classes (which are the address of object classes like *Microsoft Word Document, Microsoft Excel Worksheet, Microsoft PowerPoint Presentation, etc.*).

  - *Object for Name and Icon*
    The object referenced in this field determines the name and the icon displayed for the predefined constraint in the Fabasoft Mindbreeze Enterprise client.
  - *Constraint*
    In this field, you have to define a constraint for filtering the search result list. Refer to the "Fabasoft Mindbreeze Enterprise Guide" for further information on the syntax of constraints.

- *Default*
  This field determines whether the predefined constraint is active or inactive by default.
- *Software* Component
  In this field you have to provide a software component because otherwise the settings are lost after an update of the Fabasoft Folio Domain.

- *User With Access to Integration Web Service*
  In this field you can define which user has access to the integration web service (`/mesapi`) required for the web service based integration. If not set, the integration web service is inaccessible.
- *Root and Intermediate Certificate Authorities (X.509 Bundle)*
  In this field you can define the root and intermediate certificates (X.509 CA bundle) used for kernel-based indexing and searching as well as FilterService-based content conversions, categorization and associated Keycloak requests.
- *Category*
  This field can be used to define a new category (default: "Fabasoft Components") to isolate multiple independent Fabasoft Folio installations that share a Mindbreeze Enterprise installation. The category is used to associate indexed objects with a category descriptor and plugins and is passed to Mindbreeze Enterprise as a constraint in queries.
  Note: Changes require reindexing (a new index should be created) and reinstallation of plugins using the new category value.
  Note: This field overrides categories configured for Indexing Service objects.

Note: More than one configuration object can exist in a Fabasoft Folio Domain. However, only one configuration object may be active at the same time. The active configuration object is determined in the "Current Domain", on the "Components Configuration" tab, in the *FSC Mindbreeze Configuration* field.

4.2 Indexing Service

To configure the settings for a specific indexing service, proceed as outlined and keep in mind that your configuration will apply to that specific indexing service only:

1. In the "Domain Administration", under "Domain Objects", click the "Services" tab.
2. Open the "Indexing Service" object in edit mode.
3. Following fields are available:

- *Category (Obsolete)*
  This field can be used to define a new category (default: "Fabasoft Components") to isolate multiple independent Fabasoft Folio installations that share a Mindbreeze Enterprise installation. The category is used to associate indexed objects with a category descriptor and plugins and is passed to Mindbreeze Enterprise as a constraint in queries.
  Note: Changes require reindexing (a new index should be created) and reinstallation of plugins using the new category value.
  Note: This field is obsolete. Configure the category in a per-domain basis using the Fabasoft Mindbreeze Configuration.
- *Indexing Service URL List*
  Use this field to define the URL (including port) of the indexing service of Fabasoft

Mindbreeze Enterprise. If multiple URLs are specified, the data will be evenly distributed across the specified indexing services.
Note: By default, Fabasoft Mindbreeze Enterprise assigns port 23100 to the first indexing service.

- *Search Service URL List*
  Use this field to define the URL (including port) of the search service of Fabasoft Mindbreeze Enterprise. Multiple search services may be used.
  Note: By default, Fabasoft Mindbreeze Enterprise uses the HTTPS protocol and port 23300 for the first search service.
- *Chunk Size*
  This field contains the number of objects, to be used in chunks within one transaction. The default chunk size is 100.
- *Start Point*
  This field specifies which objects should be indexed from the date/time they are last changed at.
- *End Point*
  This field specifies which objects should be indexed until the date/time they are last changed at.
- *Number of Threads*
  This field allows you to specify the number of concurrent threads to use.
- *Start Point of Removal of Deleted Objects*
  This field allows you to specify which objects should be deleted from the index based on the date/time they were deleted. This feature requires auditing of the action "`COOSYSTEM@1.1: ObjectDelete`" and the use of a dedicated audit database. In case a dedicated new index was created, this date must be set to the start date for the new index, so that no recently deleted objects are left over.
- *Disable Removal of Deleted Objects from the Index*
  If this field is set to "Yes", no objects will be deleted from the index.
- *Alternate Condition*
  This field allows you to define your own condition for indexing of objects.
- *Ignore Default Scope*
  If this field is set to "Yes", the indexing will process objects across all Fabasoft Folio COO Stores for all objects that match the *Alternate Condition*, even if this indexing service is not defined on the Fabasoft Folio COO Stores.
- *Force Index Update With Alternate Condition*
  If this field is set to "Yes", the indexing will re-index all objects that match the *Alternate Condition*, even if objects have already been indexed. This functionality is useful for applying configuration changes without full re-indexing.
- *File for External Index State*
  Use this field to specify a file, where information on indexed objects is stored. By default, the index state is stored in Fabasoft Folio. By storing the index state in an external file, initial indexing into a new index based on a new indexing service object can be executed in parallel to delta indexing into an existing index. If a file is specified, *Ignore Default Scope* must be set because the alternative Indexing Service object cannot be assigned to

a COO Store without removing the current Indexing Service object. Additional restrictions can be specified in *Alternate Condition*.
Once completed, the new Indexing Service can be switched to by disabling delta indexing for the current Indexing Service, executing an index recovery which transfers the index state to Fabasoft Folio, and assigning the new Indexing Service to COO Stores. After removing the file from the Indexing Service, delta indexing can be re-enabled.

- *Directory for Log Files*
  Use this field to specify a folder, where the log files will be stored.
- *Filter Services*
  This field allows you to specify the filter services to be used by the Fabasoft Mindbreeze Enterprise indexing service.
- *Client Certificate*
  For HTTPS encryption a PKCS12 client authentication certificate with embedded private key is required. The CA certificate (cacert.pem), used to generate the client certificate, has to be imported in the Fabasoft Mindbreeze Enterprise client (tab "Certificates"). These certificates are generated automatically by the Fabasoft Folio Setup and are stored automatically within the appropriate place. The certificates are stored in a Microsoft Windows environment in `C:\Program Files\Fabasoft\Components\MindbreezeIntegration` and in a Linux environment in `/opt/fabasoft/share/mes/certificate/`.
- *Private Key*
  This field contains the path to the generated private key (in a Microsoft Windows environment for example `C:\Program Files\Fabasoft\Components\MindbreezeIntegration\request.key` and in a Linux environment `/opt/fabasoft/share/mes/certificate/request.key`). This file has to be readable for every Fabasoft Folio Kernel, whenever a query is passed to Fabasoft Mindbreeze Enterprise. After the installation of the *Fabasoft Integration for Fabasoft Mindbreeze Enterprise* the private key is stored in the path specified. If an installation is split across multiple computers, the private key has to be stored on all servers hosting a Fabasoft Folio Web Service.
- *Passphrase of Private Key*
  In this field the passphrase of the private key has to be entered. (The path to the text file containing the passphrase is in a Microsoft Windows environment `C:\Program Files\Fabasoft\Components\MindbreezeIntegration\passwd` and in a Linux environment `/opt/fabasoft/share/mes/certificate/passwd`).
- *OAuth Service*
  This field specifies the OAuth Service object that should be used to authenticate Folio to Mindbreeze.

4.3 Filter Service

A filter service object can be configured in multiple index service objects. Follow these steps to configure a filter service:

1. In the “Domain Administration”, under “Domain Objects”, click the “Services” tab.
2. Open the “Filter Service” object in edit mode.

Following fields are available:

- *Filter Service URL List*
  This field contains one or more URLs (including port information) representing Fabasoft Mindbreeze Enterprise filter services. During indexing, local Fabasoft Mindbreeze Enterprise filter services (identification is based on the host name) are used in favor of remote filter services to eliminate unnecessary network traffic.
  Note: By default, the first Fabasoft Mindbreeze Enterprise filter service of a Fabasoft Mindbreeze Enterprise installation is reachable via port 23400.
- *File Extensions*
  This field contains file extensions of all file types that should be indexed. For example, specifying "doc" causes Microsoft Office Word objects to be indexed. If no extensions are specified, all file types are indexed. A filter service with a matching file extension takes precedence over a filter service without files extensions.
- *OAuth Service*
  This field specifies the OAuth Service object that should be used to authenticate Folio to Mindbreeze.

4.4 OAuth Authentication

If the authentication from Folio to Mindbreeze should be done via OAuth, an OAuth Service object needs to be created for each different OAuth configuration. When using Mindbreeze InSpire the service URLs additionally need to be changed based on the following pattern:
`https://mindbreeze.fabasoft.com:<Keycloak port>/realm/master/api/v1/<service>/<service port>`

- `<Keycloak port>`
  The port on which the service is running.
- `<service>`
  The Mindbreeze service (index or query).
- `<service port>`
  The port on which the Mindbreeze service listens (e.g. 23100 or 23300).

1. Create a new client on the Keycloak service and set the fields on the "Settings" tab as follows and click "Save" afterwards.
   - Client ID: *give the client a unique ID (e.g. fabasoft-folio)*
   - Name: *give the client a name (e.g. Fabasoft Folio)*
   - Description: *optional description of the client*
   - Enabled: ON
   - Consent Required: OFF
   - Login Theme: *empty*
   - Client Protocol: openid-connect
   - Access Type: confidential
   - Standard Flow Enabled: OFF
   - Implicit Flow Enabled: OFF

- Direct Access Grants Enabled: OFF
- Service Accounts Enabled: ON
- Authorization Enabled: OFF
- Root URL: *empty*
- Base URL: *empty*
- Admin URL: *empty*
- Advanced Settings
  - Access Token Lifespan: *it is recommended to set this to 10 minutes*

2. On the "Mappers" tab create a new mapper and set the fields as follows:
   - Protocol: openid-connect
   - Name: realm roles
   - Mapper Type: User Realm Role
   - Realm Role prefix: *empty*
   - Multivalued: ON
   - Token Claim Name: roles
   - Claim JSON Type: String
   - Add to ID token: OFF
   - Add to access token: ON
   - Add to userinfo: OFF
3. On the "Service Account Roles" tab add "InSpire Index Writer" and "InSpire Application Impersonation" to "Assigned Roles" and remove "offline_access".
4. An OAuth Service object can be created in the "Domain Administration", under "Domain Objects", click the "Services" tab.
   Provide the client ID of the Keycloak client in the attribute *Client ID* (`COOSYSTEM@1.1:oatclientid`).
   If the same Keycloak service should be used for all associated service URLs the Keycloak Token Endpoint needs to be provided in the attribute *Authentication Service* (`COOSYSTEM@1.1:oatsrvurl`). Otherwise the OAuth Token Endpoint is derived from the service URL using the pattern: `https://<serviceurl-authority>/auth/realms/master/protocol/openid-connect/token`.
   The Indexing and Filter Service Objects then need to reference the OAuth Service object in the attribute *OAuth Service* (`COOSYSTEM@1.1:oauthservice`).
5. When creating a new OAuth Service object, a new public key pair is created and stored on the object. The *Public Key* (`COOSYSTEM@1.1:oatpublickey`) needs to be imported into the Keycloak service. To do so set "Client Authenticator" to "Signed JWT" under the "Credentials" tab on the Keycloak client and import the public key PEM by clicking "Import Certificate", choose "Public Key PEM" as the "Archive Format" and select the PEM file in "Select file".

When using Mindbreeze InSpire, connect to the Mindbreeze Configuration interface and navigate to the "Filters" tab. Expand the filter service, check "Advanced Settings" and configure "Base Configuration" as follows:

- Destination Pattern
  `https://mindbreeze.fabasoft.com:<Keycloakport>/realm/<realm>/api/v1/index/(\d+)(/.*)?`
- Destination Replacement
  `http://localhost:\1\2`

4.5 Fabasoft Folio Stores

A Fabasoft Mindbreeze Enterprise filter service object must be configured in Fabasoft Folio COO Stores to include associated Fabasoft Folio objects for indexing.

1. In the "Domain Administration", under "Domain Objects", click the "Stores" tab.
2. Open the Fabasoft Folio COO Store in edit mode.
3. Set the desired Fabasoft Mindbreeze Enterprise indexing service object in the *Indexing Service* field.

Note: Archived Fabasoft Folio objects remain associated to the original Fabasoft Folio COO Store. During indexing, archived objects are read from the archive and sent to Fabasoft Mindbreeze Enterprise. Consequently, archived objects in Fabasoft software products can be searched via Fabasoft Mindbreeze Enterprise.

4.6 Indexing of Archived Contents and Objects

Indexing archived contents and objects stored in an archival system via Fabasoft iArchive requires no additional configuration. The Fabasoft Folio Kernel handles these automatically by reading them from the archive and sending them to Fabasoft Mindbreeze Enterprise.

4.7 Modification of the timeout properties

Within a full-text search the configured limit of hits and the maximal search time are sent to Fabasoft Mindbreeze Enterprise. The default values are 10000 for the maximum hit count and 5 seconds for the maximum search time.

These default values can be changed in the Microsoft Windows Registry on the Fabasoft Folio Web Server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Fabasoft\FscWeb\Modules\1_FSC]
@="//?/C:/Program Files/Fabasoft/Components/Web/1_FSC/ASP/content/bin/fscvext.dll"

 [HKEY_LOCAL_MACHINE\SOFTWARE\Fabasoft\FscWeb\Modules\1_FSC\FTSCOUNT]
@="5000"

[HKEY_LOCAL_MACHINE\SOFTWARE\Fabasoft\FscWeb\Modules\1_FSC\FTSTIMEOUT]
@="20"
```

In a Linux environment an environment variable can be set for these counters.

```
echo -n 5000 > /var/opt/fabasoft/instances/WebService_<ID>/env/FTSCOUNT

echo -n 20 > /var/opt/fabasoft/instances/WebService_<ID>/env/FTSTIMEOUT
```

These counters can also be modified in each query by inserting the designated property:

```
<Search Term> count:1000 timeout:15
```

The maximum hit count of a quick search is always 100. You can only change the hit count with inserting the property `count:<value>`

4.8 Indexing via UNC Paths

If you want to index via UNC paths, follow the following steps:

1. In the "Domain Administration", under "Domain Objects", click the "Services" tab.
2. Open the MMC service in the edit mode.
3. Open the detail view of the *MMC Service Areas*.

If a UNC path (Universal Naming Convention) is referenced in the field *Path to Directory on Server*, the path will be sent to Fabasoft Mindbreeze Enterprise. During the indexing process the Fabasoft Mindbreeze Enterprise filter service indexes the contents directly from the MMC area via these UNC paths. This results in an increased performance.

Note: Each service user, who runs an indexing service, needs access rights to read from the MMC area. In a Linux environment the share has to be mounted.

4.9 Optimized Full-Text Queries

If an additional search criterion, that has not been indexed, is defined beside the full-text search, there is a chance to reduce the time of the query in the database. If the *Enable Optimized Full-Text Queries* field is set to „Yes", the result of Fabasoft Mindbreeze Enterprise will be taken and filtered afterwards in the database. If the *Enable Optimized Full-Text Queries* field is set to „No", the search request will be first filtered in the database and afterwards searched with Fabasoft Mindbreeze Enterprise.

4.10 ACL Check Optimization

The Fabasoft Integration for Fabasoft Mindbreeze Enterprise optimizes ACL checks during query processing by providing Fabasoft Mindbreeze Enterprise with security context data as part of the indexing process. The data is derived from security-relevant properties in Fabasoft Folio (e.g. ACL, Referenced Object) and allows Mindbreeze to filter results before passing the final result to the Fabasoft Folio ACL check. This optimization can dramatically improve search performance because the number of objects that need to be checked via the Fabasoft Folio ACL check decrease.

In order to take full advantage of this optimization, ACLs containing property paths across object boundaries must be configured along with a list of associated object classes that represent object containers. An ACL or the property "Referenced Object" must not reference objects that are outside of a given object hierarchy identified by a specific object class. Hierarchies of that object class must contain object references of all objects that are part of the hierarchy. The hierarchy must be traversable via the property "`COOSYSTEM@1.1:objchildren`" or any other property with "`COOSYSTEM@1.1:attrischild`" set. If all of these conditions are met, ACLs and object classes of containers can be configured to optimize access checks. Otherwise, these objects cannot be pre-filtered before the final ACL check via Fabasoft Folio.

To configure the list of ACLs and object classes, proceed as follows:

1. In the “Domain Administration”, click the “Configuration Objects” tab.
2. Open the configuration object `FSCCONFIG@1.1001:Configuration` in edit mode.
3. Click the “Search” tab.
4. In the field “ACLs for Evaluation of Security Context Across Object Boundaries”, create a new entry with the desired ACL and software component.
5. In the field “Superordinated Objects for Security Context Updates”, create a new entry with the desired object class and software component.

Alternatively, you can configure these using the following customization points in a DUCX project:

- `FSCCONFIG@1.1001:SearchSecACLs`
- `FSCCONFIG@1.1001:SearchSecContainers`

4.11 Custom Forms for Hit Display/Preview

By default, the Fabasoft Mindbreeze Enterprise Web Client displays indexed properties for each hit and as part of the preview based on the default form in Fabasoft Folio.

The list of properties can be customized for the hit display and preview on a per object class basis by configuring custom forms containing desired properties. The following actions are available for configuring custom forms:

- `FSCMINDBREEZE@1.1001:ViewHit` (Hit Display)
- `FSCMINDBREEZE@1.1001:ViewPreview` (Preview)

4.12 Custom Context Menus in the Mindbreeze Web Client

By default the Fabasoft Mindbreeze Enterprise Web Client is using the "container independent menu" as defined on the according object class. To overwrite this default behavior to use custom context menus there are two new properties for each object class (`FSCMINDBREEZE@1.1001:classmbctxmenu` and `FSCMINDBREEZE@1.1001:classmbctxmenuex`). Using these properties the context menus can be defined analog to the existing configuration possibilities (e.g. the container independent context menu for search results).

Furthermore the transaction variable `TV_ISMESCONTEXT` will be set during context menu evaluation. This transaction variable can be used to define dynamic context menus using `FSCMINDBREEZE:1.1001:classmbzmenuex` and an "User Interface Scoping Rule".

4.13 Mindbreeze Web Client Language

By default, the Fabasoft Mindbreeze Web Client displays language-specific strings depending on the web browser language. If a user environment specifies a language, information provided by Fabasoft Folio is still solely based on the web browser language to present a consistent user interface. However, so-called preferred languages (`FSCCONFIG@1.1001:cfgbrowserlanguages`) can be configured in the Fabasoft Folio Administration Configuration to customize that behavior:

1. In the “Domain Administration”, click the “Configuration Objects” tab.

2. Open the "Administration" configuration object in edit mode.
3. In the field "Preferred Languages When Using the Web Browser Language", configure one or more preferred languages. If a preferred language represents a subordinate language of the web browser language, that language is used instead.

4.14 Displaying Search Results in the Embedded Fabasoft Mindbreeze Enterprise Web Client

By default, the search result of a quick search is displayed as drop-down list. If the search results should be displayed in the embedded Fabasoft Mindbreeze Enterprise Web Client (after pressing "Enter"), following configuration steps are necessary.

1. Edit the "Mindbreeze Configuration".
2. Set *Show Quick Search Result in Overlay* to "Yes".
3. Define the *Web Client Service URL (Embedded Search)* of the Fabasoft Mindbreeze Enterprise Web Client.
   For example: `https://mindbreeze.fabasoft.com:23350`
   Note: Make sure that `<web client service URL>/stylesheets/embedded.css` and `<web client service URL>/script-bundle/embedded.js` is reachable.

If the JavaScript file cannot be accessed, the search result is displayed as the default drop-down list. In case of problems:

- Check whether *Web Client Service URL (Embedded Search)* is defined correctly.
- Check whether SAML is used as authentication mechanism.
- Check whether the file paths of the two files exist on the Fabasoft Mindbreeze Enterprise Web Server.

If everything works fine the virtual application `FSCVENV@1.1001:MindbreezeSearchOverlayApp` is opened as overlay after pressing "Enter". The action `FSCMINDBREEZE@1.1001:GetFTSQueryConstraints` is used to evaluate the query. The virtual application `FSCVENV@1.1001:ShareMindbreezeSearchResultApp` is used to share the results in the object list.

4.15 Mindbreeze Web Client Authentication

To enable Single Sign-On for Fabasoft Folio and the Mindbreeze Web Client, you can use one of the following authentication mechanisms:

- Kerberos
- SAML
- Client Certificates
- Fabasoft Folio Cookie-Based Authentication (Kerberos, LDAP, SAML)

In all cases, Mindbreeze must be configured to map the authentication identity of the respective authentication mechanism to a user name that corresponds to the *userlogname* in Fabasoft Folio.

In case of Fabasoft Folio Cookie-Based Authentication, a login to Fabasoft Folio must precede the use of the Mindbreeze Web Client. The initial access triggers a Fabasoft Folio request to verify that the Fabasoft Folio session cookie passed to the integration is valid. The following prerequisites must be met for this scenario to work:

- Use Fabasoft Folio Cookie-Based Authentication (Kerberos, LDAP, SAML).
- Ensure that the Fabasoft Folio session cookie is accessible to the Mindbreeze Enterprise Web Client by changing the VAPP configuration to set a cookie domain and/or .to issue the cookie for / rather than the web service virtual directory.
- Enable the Session Authentication plugin provided by the integration for the Mindbreeze Client Service used in this scenario.
- If either service is running behind a load balancer, ensure that the load balancer provides an *X-Forwarded-For* HTTP header. Otherwise, the actual client IP address is not known to Fabasoft Folio or the integration and the cookie validation will fail due to an IP mismatch (the session cookie is bound to the client IP address).

5 Configuration of an Automated Task

If the Fabasoft Folio AT Services was denoted on the "Review Configuration" tab of the setup during the installation of the *Fabasoft Integration for Fabasoft Mindbreeze Enterprise*, an automated task for indexing is created automatically.

To create an automated task manually, do the following:

1. In the "Domain Administration", under "User Objects", click the "List of Automated Tasks" tab.
2. Create a new "Automated Task" and edit the properties.
3. In the *Tasks* field, create a new "Automated Task" and edit the properties.
4. Add in the *Object* field your Fabasoft Mindbreeze Enterprise indexing service and in the *Action* field the action "`COOSYSTEM@1.1:BuildIndex`".
5. Click "Next" to save the configuration.

Note: If the "Automated Task" has been created during the setup, it is disabled by default and can be changed to the state "Waiting" to activate the "Automated Task".

6 Indexing und Searching

6.1 Manual Indexing

1. In the "Domain Administration", under "Domain Objects", click the "Services" tab.
2. Open the context menu of the indexing service for Fabasoft Mindbreeze Enterprise and click "Build/Update Index". All objects defined in the Fabasoft Mindbreeze Enterprise configuration will be indexed. If the index already exists, each object gets checked, if it has been changed after the last indexing. Changed objects are updated.

6.2 Rebuilding an Index

If you want to rebuild an index, e.g. because of a restore, it is necessary to restore the database values with the `RecoverIndex()` action. This can be done with the context menu entry "Recover Index State" on the indexing service in the "Services" tab. Delete the old index in the file system.

6.2.1 Online Rebuild

Rebuilding an index usually involves downtime because the new index cannot be built without disabling the old index for delta indexing.

The Fabasoft Integration for Fabasoft Mindbreeze Enterprise provides an alternative that is based on an external index state stored in a file system. The following steps can be executed to rebuild an index with minimal update delay:

1. Create a new *Indexing Service* object with a configuration for the new Fabasoft Mindbreeze Enterprise Indexing Service.
2. Enable *Ignore Default Scope* in the Indexing service to allow indexing objects without being required to assign the indexing service to COO Stores. By default, all COO Stores are included after enabling the option. For additional restrictions, configure *Alternate Condition.*
3. Configure *File for External Index State* to enable the external storage of the index state created during the indexing process.
4. Start the indexing process using the new Indexing Service. Re-indexing using the old indexing service can remain active throughout the duration of the operation.
   If indexing was successful, the start time of the indexing process is stored as a new *Start Point* for the next indexing run. A tolerance of one hour will be used to cover objects which are created or changed via long-running transactions.
5. Restart the indexing process using the new indexing service once or several times in a row to minimize the number of unprocessed changes.
6. Once completed, trigger re-indexing using the old indexing service one more time and disable automatic re-indexing.
7. Start an index recovery using the new Index Service to transfer the index state to Fabasoft Folio. The process recovers all objects changed after *Start Point* and objects that have not been indexed since index creation.
8. Move the indexing service URLs from the new indexing service object to the old indexing service object. The new index is now online. Re-enable re-indexing.

6.2.2 Index-Based Recovery

Index Recovery using the `IndexRecovery()` action resets the index state stored in Fabasoft Folio comprised of an index date per Fabasoft Folio object and an index date per Fabasoft Folio content. In some scenarios, it is useful to be able to reset the index state.based on the actual state information stored in a Fabasoft Mindbreeze Enterprise index rather than specifying a single recovery date.

By setting the parameter `loadstate` when executing the `IndexRecovery()` action, the recovery process reads information on indexed objects and contents from Fabasoft Mindbreeze Enterprise and resets the index state in Fabasoft Folio accordingly. The index state of objects not stored in the index is reset automatically based on the configured scope (included COO Stores and alternative condition).

6.3 Re-Indexing

During re-indexing new objects and objects that have been changed since the last indexing will be indexed. If the ACL of indexed objects has changed, the affected objects will be updated during a re-indexing.

Note: If users get permissions on an object due to an ACL change, those users will not find the corresponding objects until a re-indexing.

Specifically for PostgreSQL: After indexing, where many objects have been changed (e.g. initial indexing, ACL changes, migrations and so on), a `CLUSTER <table name>;` should be conducted on the tables `cooobject` and `fsclogcontentmappinglist` for performance reasons.

6.4 Search

6.4.1 Searching in Metadata

Select in a string property the command *Full-Text Query*, if you want to send a metadata full-text query to Fabasoft Mindbreeze Enterprise.

Note: This command is also available when the property is not indexed. In this case the search request is sent to the database.

6.4.2 Search in Contents

If you make a full-text search request in a content property, the metadata is also searched.

6.4.3 Quick Search

If you use quick search with Fabasoft Mindbreeze Enterprise in Fabasoft Folio, the search request will search in metadata and contents from each object that can be inserted in the selected field.

7 Scripts for Indexing

Although there are a number of built-in actions for indexing, the following scripts may be used. The scripts are listed as examples.

Note:

- `<COOADDRESS>` is a placeholder for the address of the indexing service object. You have to adapt this value for your installation.
- For log files you have to consider the path depending on your installation (Microsoft Windows or Linux)

- Every MMC area without logging has to be closed (you cannot delta index these areas). To index these areas, you have to set the indexing parameter "CONTENT IS NOT NULL" (e.g. „meth.SetParameter2(5, "STRING", "CONTENT IS NOT NULL");“)

7.1 Script “Build Index 4 Threads”

```
//LANGUAGE="JavaScript"

/*========================================================================
  Copyright (c) Fabasoft R&D GmbH, A-4020 Linz, 1995-2009.

  Alle Rechte vorbehalten. Alle verwendeten Hard- und Softwarenamen sind
  Handelsnamen und/oder Marken der jeweiligen Hersteller.

  Der Nutzer des Computerprogramms anerkennt, dass der oben stehende
  Copyright-Vermerk im Sinn des Welturheberrechtsabkommens an der vom
  Urheber festgelegten Stelle in der Funktion des Computerprogramms
  angebracht bleibt, um den Vorbehalt des Urheberrechtes genuegend zum
  Ausdruck zu bringen. Dieser Urheberrechtsvermerk darf weder vom Kunden,
  Nutzer und/oder von Dritten entfernt, veraendert oder disloziert werden.
  ========================================================================*/

// This script can be used to create an initial index.
// All objects, configured in the Mindbreeze Configuration object are
// sent to Fabasoft Mindbreeze Enterprise.

// Get the index service by object address.

var indexservice = coort.GetObject("<COOADDRESS>");

// Define Logfile.
// Directory "C:\Logs" must exist.

var datetime = coort.Parse(
  cootx,
  "STRING(coort.GetCurrentDateTime(this))",
  2).Evaluate3(cootx, null, indexservice).getItem(0);

datetime = datetime.replace(new RegExp(/:/g), "-");
var logfilename = "C:\\Logs\\Index-" + datetime + ".log";

cootx.SetVariable2("COOSYSTEM@1.1", 2, "BOOLEAN", true);
cootx.SetVariable2("COOSYSTEM@1.1", 14, "STRING", logfilename);

// Set indexing parameters.
// Objects are processed in chunks of 400 objects, 4 threads are used.

var meth = indexservice.GetMethod(cootx, "BuildIndex");
meth.SetParameter2(1, "INTEGER", 400); // chunksize:   400
meth.SetParameter2(4, "INTEGER", 4);   // threadcount: 4

try {
  indexservice.CallMethod(cootx, meth);
}
catch(e) {
  coort.Trace(e);
}
```

7.2 Script “Recover Index 4 Threads”

```
//LANGUAGE="JavaScript"

/*========================================================================
  Copyright (c) Fabasoft R&D GmbH, A-4020 Linz, 1995-2009.
```

```
  Alle Rechte vorbehalten. Alle verwendeten Hard- und Softwarenamen sind
  Handelsnamen und/oder Marken der jeweiligen Hersteller.

  Der Nutzer des Computerprogramms anerkennt, dass der oben stehende
  Copyright-Vermerk im Sinn des Welturheberrechtsabkommens an der vom
  Urheber festgelegten Stelle in der Funktion des Computerprogramms
  angebracht bleibt, um den Vorbehalt des Urheberrechtes genuegend zum
  Ausdruck zu bringen. Dieser Urheberrechtsvermerk darf weder vom Kunden,
  Nutzer und/oder von Dritten entfernt, veraendert oder disloziert werden.
  ============================================================================*/

// This script can be used to recover an index by a specific time.
// All objects, indexed after the specified date, are marked for re-indexing.

// Get the index service by object address.

var indexservice = coort.GetObject("<COOADDRESS>");

// Define Logfile.
// Directory "C:\Logs" must exist.

var datetime = coort.Parse(
  cootx,
  "STRING(coort.GetCurrentDateTime(this))",
  2).Evaluate3(cootx, null, indexservice).getItem(0);

datetime = datetime.replace(new RegExp(/:/g), "-");
var logfilename = "C:\\Logs\\RecoverIndex-" + datetime + ".log";

cootx.SetVariable2("COOSYSTEM@1.1", 2, "BOOLEAN", true);
cootx.SetVariable2("COOSYSTEM@1.1", 14, "STRING", logfilename);

// Mark all objects indexed after "2005-01-01 00:00:00" for reindexing.
// Objects are processed in chunks of 1000 objects, 4 threads are used.

var meth = indexservice.GetMethod(cootx, "RecoverIndex");

meth.SetParameter(1, "DATETIME", "2005-01-01 00:00:00");
meth.SetParameter(2, "INTEGER", 1000);
meth.SetParameter(5, "INTEGER", 4);

try {
  indexservice.CallMethod(cootx, meth);
}
catch(e) {
  coort.Trace(e);
}
```

7.3 Script “Delta Indexing”

```
//LANGUAGE="JavaScript"

/*============================================================================
  Copyright (c) Fabasoft R&D GmbH, A-4020 Linz, 1995-2009.

  Alle Rechte vorbehalten. Alle verwendeten Hard- und Softwarenamen sind
  Handelsnamen und/oder Marken der jeweiligen Hersteller.

  Der Nutzer des Computerprogramms anerkennt, dass der oben stehende
  Copyright-Vermerk im Sinn des Welturheberrechtsabkommens an der vom
  Urheber festgelegten Stelle in der Funktion des Computerprogramms
  angebracht bleibt, um den Vorbehalt des Urheberrechtes genuegend zum
  Ausdruck zu bringen. Dieser Urheberrechtsvermerk darf weder vom Kunden,
```

```
  Nutzer und/oder von Dritten entfernt, veraendert oder disloziert werden.
  ============================================================================*/

// This script can be used to re-index objects.
// Less objects are processed in common, so less locks will be visible to users.
// To specify the date of the last complete index run improves performance.

var currdate = new Date();

// Define Logfile.
// Directory "C:\Logs" must exist.

var logfilename = "C:\\Logs\\Re-Index-" + FormatDate(currdate, true) + ".log";

// Get the index service by object address.

var indexservice = coort.GetObject("<COOADDRESS>");

// Correct date according to your desired delta indexing interval.

var hoursToSubtract = 1;
// Amount of hours considered to set date of last index run.

var fullIndexHour = 23;
// Specifiy the hour to start a complete index run (till last complete index run).

var lastIndex = "2009-00-00 00:00:00";
// Date of last complete index run.

cootx.SetVariable2("COOSYSTEM@1.1", 2, "BOOLEAN", true);
cootx.SetVariable2("COOSYSTEM@1.1", 14, "STRING", logfilename);

var hour = currdate.getHours();

if (hour == fullIndexHour){
  indexFrom = lastIndex;
}
else {
  currdate.setHours(hour - hoursToSubtract);
  indexFrom = FormatDate(currdate, false);
}

// Set indexing parameters.
// Objects are processed in chunks of 20 objects, 4 threads are used.

coort.Trace("IndexFrom: " + indexFrom);

var meth = indexservice.GetMethod(cootx, "BuildIndex");

meth.SetParameter2(1, "INTEGER", 20);
meth.SetParameter2(2, "DATETIME", indexFrom);
meth.SetParameter2(4, "INTEGER", 4);

try {
  indexservice.CallMethod(cootx, meth);
}
catch(e) {
  coort.Trace(e);
}

function FormatInt(input){
  return (input < 10) ? "0" + input : input;
}

function FormatDate(input, filename){
  return input.getFullYear() + "-" + FormatInt(input.getMonth()+1) + "-" +
  FormatInt(input.getDate()) + ((filename==true) ? "_" : " ") +
  FormatInt(input.getHours()) + ((filename==true) ? "_" : ":") +
```

```
  FormatInt(input.getMinutes()) + ((filename==true) ? "_" : ":") +
  FormatInt(input.getSeconds())
}
```

7.4 Script “Build Index 4 Threads” (Limit by Date)

```
//LANGUAGE="JavaScript"
/*=============================================================================
  Copyright (c) Fabasoft R&D GmbH, A-4020 Linz, 1995-2009.

  Alle Rechte vorbehalten. Alle verwendeten Hard- und Softwarenamen sind
  Handelsnamen und/oder Marken der jeweiligen Hersteller.

  Der Nutzer des Computerprogramms anerkennt, dass der oben stehende
  Copyright-Vermerk im Sinn des Welturheberrechtsabkommens an der vom
  Urheber festgelegten Stelle in der Funktion des Computerprogramms
  angebracht bleibt, um den Vorbehalt des Urheberrechtes genuegend zum
  Ausdruck zu bringen. Dieser Urheberrechtsvermerk darf weder vom Kunden,
  Nutzer und/oder von Dritten entfernt, veraendert oder disloziert werden.
  ============================================================================*/

// This script can be used to create an initial index.
// The date limit enables a separation objects to index.
// This can be used to start multiple independent indexing runs
// or to index objects in parallel on multiple servers.
// All objects, configured in the Mindbreeze Configuration object are
// sent to Fabasoft Mindbreeze Enterprise.

// Get the index service by object address.

var indexservice = coort.GetObject("<COOADDRESS>");

// Define Logfile.
// Directory "C:\Logs" must exist.

var datetime = coort.Parse(
  cootx,
  "STRING(coort.GetCurrentDateTime(this))",
  2).Evaluate3(cootx, null, indexservice).getItem(0);

datetime = datetime.replace(new RegExp(/:/g), "-");
var logfilename = "C:\\Logs\\BuildIndex-" + datetime + ".log";

cootx.SetVariable2("COOSYSTEM@1.1", 2, "BOOLEAN", true);
cootx.SetVariable2("COOSYSTEM@1.1", 14, "STRING", logfilename);

// Set indexing parameters.
// Objects are processed in chunks of 400 objects, 4 threads are used.

var meth = indexservice.GetMethod(cootx, "BuildIndex");

meth.SetParameter2(1, "INTEGER", 400);
meth.SetParameter2(2, "DATETIME", "<date from (objchangedat)>");
meth.SetParameter2(3, "DATETIME", "<date till (objchangedat)>");
meth.SetParameter2(4, "INTEGER", 4);

try {
  indexservice.CallMethod(cootx, meth);
}
catch(e) {
  coort.Trace(e);
}
```

7.5 Script "Recover Contents 4 Threads" (New Filter)

```
//LANGUAGE="JavaScript"

/*=============================================================================
  Copyright (c) Fabasoft R&D GmbH, A-4020 Linz, 1995-2009.

  Alle Rechte vorbehalten. Alle verwendeten Hard- und Softwarenamen sind
  Handelsnamen und/oder Marken der jeweiligen Hersteller.

  Der Nutzer des Computerprogramms anerkennt, dass der oben stehende
  Copyright-Vermerk im Sinn des Welturheberrechtsabkommens an der vom
  Urheber festgelegten Stelle in der Funktion des Computerprogramms
  angebracht bleibt, um den Vorbehalt des Urheberrechtes genuegend zum
  Ausdruck zu bringen. Dieser Urheberrechtsvermerk darf weder vom Kunden,
  Nutzer und/oder von Dritten entfernt, veraendert oder disloziert werden.
  ============================================================================*/

// If the list of file extensions to index is modified after a successful
// indexing run, this script has to be executed first.
// Afterwards file extension can be added in the filter service and the
// indexing can be restarted.

// Get the index service by object address.

var indexservice = coort.GetObject("<COOADDRESS>");

// Define Logfile.
// Directory "C:\Logs" must exist.

var datetime = coort.Parse(
  cootx,
  "STRING(coort.GetCurrentDateTime(this))",
  2).Evaluate3(cootx, null, indexservice).getItem(0);

datetime = datetime.replace(new RegExp(/:/g), "-");
var logfilename = "C:\\Logs\\AddFilterExtensions-" + datetime + ".log";

cootx.SetVariable2("COOSYSTEM@1.1", 2, "BOOLEAN", true);
cootx.SetVariable2("COOSYSTEM@1.1", 14, "STRING", logfilename);

// Mark all objects indexed after "2000-01-01 00:00:00" for re-indexing.
// Objects are processed in chunks of 1000 objects, 4 threads are used.

var meth = indexservice.GetMethod(cootx, "RecoverIndex");

meth.SetParameter2(1, "DATETIME", "2000-01-01 00:00:00");
meth.SetParameter2(2, "INTEGER", 1000);
meth.SetParameter2(4, "STRING",
  ".objlogcontmap.objlogmapindexedat=1900-01-01 00:00:00");
meth.SetParameter2(5, "INTEGER", 4);

try {
  indexservice.CallMethod(cootx, meth);
}
catch(e) {
  coort.Trace(e);
}
```

7.6 Script "Recover Objects 4 Threads" (New Property)

```
//LANGUAGE="JavaScript"

/*=============================================================================
  Copyright (c) Fabasoft R&D GmbH, A-4020 Linz, 1995-2009.
```

```
  Alle Rechte vorbehalten. Alle verwendeten Hard- und Softwarenamen sind
  Handelsnamen und/oder Marken der jeweiligen Hersteller.

  Der Nutzer des Computerprogramms anerkennt, dass der oben stehende
  Copyright-Vermerk im Sinn des Welturheberrechtsabkommens an der vom
  Urheber festgelegten Stelle in der Funktion des Computerprogramms
  angebracht bleibt, um den Vorbehalt des Urheberrechtes genuegend zum
  Ausdruck zu bringen. Dieser Urheberrechtsvermerk darf weder vom Kunden,
  Nutzer und/oder von Dritten entfernt, veraendert oder disloziert werden.
  ============================================================================*/

// Occasionally it can be required to reset objects of a specific class after
// an index run, to re-index these objects (e.g. adding a new property to index).

// Get the index service by object address.

var indexservice = coort.GetObject("<COOADDRESS>");

// Define Logfile.
// Directory "C:\Logs" must exist.

var datetime = coort.Parse(
  cootx,
  "STRING(coort.GetCurrentDateTime(this))",
  2).Evaluate3(cootx, null, indexservice).getItem(0);

datetime = datetime.replace(new RegExp(/:/g), "-");
var logfilename = "C:\\Logs\\RecoverObjects-" + datetime + ".log";

cootx.SetVariable2("COOSYSTEM@1.1", 2, "BOOLEAN", true);
cootx.SetVariable2("COOSYSTEM@1.1", 14, "STRING", logfilename);

// Mark all objects indexed after "2000-01-01 00:00:00" for re-indexing.
// Objects are processed in chunks of 1000 objects, 4 threads are used.

var meth = indexservice.GetMethod(cootx, "RecoverIndex");

meth.SetParameter2(1, "DATETIME", "2005-01-01 00:00:00");
meth.SetParameter2(2, "INTEGER", 1000);
meth.SetParameter2(3, "OBJECT", "<COOADDRESS of object class >");
meth.SetParameter2(5, "INTEGER", 4);

try {
  indexservice.CallMethod(cootx, meth);
}
catch(e) {
  coort.Trace(e);
}
```

8 Indexing Performance

The indexing duration is highly dependent on the number of objects which should be processed. Therefore, the workload can be chunked and divided by the number of threads using the indexing service configuration. However, as the loading of the objects from the COO service has a major impact on the overall performance, this part of the indexing can be tuned using the following environment variables. For details about how to set these please refer to the white paper “Fabasoft Folio Environment Variables”.

- `INDEXLOADCOMPLETE`
  If enabled (default), the objects are loaded using `LoadAllAttributes`. Otherwise the objects are loaded without the property `COOSYSTEM@1.1:objversions`.
- `INDEXLOADTHREADS`
  The number of threads, which are used for loading the objects. Please consider the COO service load and worker threads when raising this value. By default, only one thread is used.
- `INDEXPRELOADLIMIT`
  The maximum number of objects, which are loaded in the background while processing the current chunk. Its granularity is the chunk size divided by the number of load threads. By default, one chunk is preloaded in the background. As preloading objects highly affects the memory usage, please consider limiting the kernel object cache size using the variable `CACHESIZE`.