GBS Retirement Manager 3.1

Importing Files into GBS Workflow Manager

Document Version 1.0 – 2019-03-21

Contents

1 Introduction

GBS Retirement Manager provides an interface to transfer documents from IBM Notes databases to GBS Workflow Manager applications.

With the Retirement Manager, you can export documents from a Notes database to a local folder. All documents from a specific Notes database will be stored inside of an own local folder. The content of each document will be printed to a PDF file. Most of the document fields will be stored in an XML file, which have the same name as the PDF file (except the file type). The name of the PDF and the corresponding XML file will be the document Unique ID of the Notes document.

After the export, you can import the documents into an Workflow Manager application. For this, you have to use the import tool. The import tool will create a new application for each local folder, i.e. for each Notes database. After that, a new document will be created in the Workflow Manager application for each PDF file inside of the local folder. The PDF document will be attached to that document. For each item of the XML file, the import tool will create an own field in the Workflow Manager document.

If the exported document contains a field `form`, the import tool looks to a form with the same name in the Workflow Manager application. If such a form exists, then this form will be used to create the document in the Workflow Manager application. If the form does not exist, the default form will be used. The default form can be specified in the configuration file (refer to 3.1).

If an Workflow Manager application for a local folder exists and it is not configured that an existing application should be used, this local folder will be skipped and the import tool uses the next local folder and creates an Workflow Manager application for this folder.

2 Preconditions

Before using the import tool, make sure that the following preconditions are fulfilled:

- You have exported documents from one or more Notes databases inside of at least one local folder. This local folder is the source folder.
- You have access to the source folder.
- You have access to an Workflow Manager domain and have Administrator access to it.
- You have access to the import tool **workflowmanager-interface.jar**. It is located in the `%root application%` folder.
- You have installed Java JRE 1.8 on the computer on which the import tool should run.

3 Components

3.1 Configuration File

The configuration file defines global settings for the import process.

Filename: **config.xml**
Location: This file must be located in the data folder which will be defined with the parameter `-Ddata`.

Items are:

`<name>`: A name for the project configuration.

Section "`<appdesigner>`":

`<apiUrl>`: Path to the Workflow Manager API on the Workflow Manager server.

`<domain>`: Name of the domain which should be used for the import process.

`<user>`: Login name which will be used for the import process; can be replaced by the command line parameter.

`<password>`: Password which will be used for the import process; can be replaced by the command line parameter.

Section "`<importTool>`":

`<useExistingApplications>`: Flag to define whether documents in an existing application should be overwritten. This is the case if this flag has the value "1".

`<defaultFormAlias>`: Alias name of the default form which should be used for the import process; all documents will be created based on this form if the document does not contain a `form` field or the form defined in the `form` field does not exist in the application.

`<templateFile>`: Path to a template of an application design. This design will be imported to the new application.

<composeApplicationName>: Parameter for composing an application name from all passed arguments. Allowed arguments are `s`, `t`, `p`, `n`, `r` mapped to following *databaseinfo.xml* elements respectively:

```
s = </server>

t = </title>

p = </filepath>

n = </filename>

r = <replicaid>
```

For example, you can compose an application name from *title* + *filepath* + *filename* using "tpn". To separate name elements, "-" will be used. If an application with the same name exists, a new application will not be created. By using a "p" parameter, a "\" char in the path will be replaced by a "_" char.

Structure:

```
<project_configuration>
        <name>…name…</name>
        <!-- required -->
        <appdesigner>
              <apiUrl>…path…<apiUrl>
              <domain>…domain…</domain>
              <user>…username…</user>
              <password>…password…</password>
        </appdesigner>
        <importTool>
              <useExitingApplications>…</useExistingApplications>
              <defaultFormA>…name…</defaultFormAlias>
              <importApplicationFile>… file …</importApplicationFile>
              <composeApplicationName> … type …
  </composeApplicationName>
        </importTool>
        <meta_data_extension>xml</meta_data_extension>
        <binary_extensions>pdf</binary_extensions>
</project_configuration>
```

3.2 Project File

The project file defines settings for each Notes database (folder). It will be created by the Retirement Manager export process.

Filename: **databaseinfo.xml**
Location: This file must be located inside of each folder.

Structure:

```
<database_info>

    <server>…server…</server>
    <title>…title… </title>
    <filepath>…filepath…</filepath>
    <filename>…filename…</filename>
    <replica_id>…replica id…</replica_id>

</database_info>
```

3.3 Working File

This file contains structured information for every imported document. Only exceptions if some error is caused by some import failures, will be written to the working file. At the end of the import of a document, a status (e.g. ‘Finished’, ‘Error’) will be written.

Default filename: **workingFile.xml**
Location: This file will be stored in the folder which will be defined with the parameter `-Ddata`.

3.4 Import Log File

Each action during the import process will be logged into the log files. Every error will be added to the log files. Two types of log files exist:

- A global log file

  The global log file contains all entries for all imported folder during the import process.

  Filename: **rmLog.log**
  Location: This file will be stored in the folder which will be defined with the parameter `-Ddata`.

- A single log file for each folder (Notes database): Contains only entries for the specific imported folder.

  Filename: **<Replica id of the Lotus Notes database>.log**
  Location: This file will be stored in the folder of the Notes database.

4 Parameters

The import tool can be used from the command line.

Command:

```
java <params> -jar workflowmanager-interface.jar
```

Parameters:

-Ddata

Path to the folder which contains files to import. The folder must contain subfolders for each database.

If this parameter exists, then a *rmLog.log* file will be created in that folder.

If this parameter is not provided, a *rmLog.log* file will be created in the `%root application%` folder.

Example: `-Ddata=C:\GBS\RetirementManager\domino_export`

-DuserName

(Optional) Username of the user which should log in to use the Workflow Manager.

If this parameter does not exist, user property from *config.xml* file is going to be used.

Example: `-DuserName=Arno Rautmann`

-Dpassword

(Optional) Password of the user which should log in to use the Workflow Manager.

If this parameter does not exist, password property from *config.xml* file is going to be used.

Example: `-Dpassword=password`

5 Example

The following configuration file and command show you how to use the Retirement Manager Import tool.

Configuration file (*config.xml*)

```
<project_configuration>
<name>Sample</name>
<appdesigner>
	<apiUrl>https://tecone-test.abc.com/api/appdesigner/</apiUrl>
	<domain>ABC</domain>
	<user>Will Harber</user>
	<password>password</password>
</appdesigner>
<importTool>
	<useExistingApplications>1</useExistingApplications>
	<defaultFormAlias>RetirementManagerImport</defaultFormAlias>
	<importApplicationFile>d:\temp\install\RetirementManager\Template-GBS-
	RetirementManager.zip</importApplicationFile>
	<composeApplicationName>t</composeApplicationName>
</importTool>
<meta_data_extension>xml</meta_data_extension>
<binary_extensions>pdf</binary_extensions>
</project_configuration>
```

Command

```
java -Ddata=D:\GBS\RetirementManager\domino_export -jar workflowmanager-
interface.jar
```

6 Instructions

1. Open the command line interface on your computer and navigate to the `%root application%` folder, in which the import tool exists.

   Example: `C:\GBS\RetirementManager\`

2. Enter the command specified in section 5 and click on ENTER.
3. Follow the process statements in the command line.
4. If the process was interrupted, look into the global log file to get the reason for this. Afterwards, restart the process.

GBS Europa GmbH is unable to guarantee, either explicitly or tacitly, the quality, execution, standardization or suitability for a specific purpose.

The product descriptions are general and descriptive in nature. They can be interpreted neither as a promise of specific properties nor as a declaration of guarantee or warrantee. The specifications and design of our products can be changed at any times without prior notice, especially to keep pace with technical developments.

Web site: www.gbs.com
Email address: info@de.gbs.com
Locations: www.gbs.com/en/locations

GBS
A BULPROS COMPANY