Adabas Client for Java

Frequently Asked Questions

Version 3.0.0

December 2024

ADABAS & NATURAL

This document applies to Adabas Client for Java Version 3.0.0 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: ACJ-FAQ-300-20241221

Table of Contents

1 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

Product Training

You can find helpful product training material on our Learning Portal at **https://learn.softwareag.com**.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at **https://techcommunity.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://containers.softwareag.com/products** and discover additional Software GmbH resources.

Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software GmbH products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Frequently Asked Questions

Adabas Data Designer

Why is the Finish button greyed out (inactive) when I create a map definition using the mis.fdt example file?

The file *mis.fdt* in the directory *demodb* does not contain any long names for fields, and consequently the button **Finish** is not activated.

Attempting to execute one of the query mapping java example files returns the message "EmployeeMap does not exist or is not unique" - what other restrictions are known?

The prerequisites for using the Adabas Data Designer are described in the section *Adabas Client for Java - Concepts and Use Cases*: the database should be running and a metadata file already generated. Execute *GenerateExampleMappings* in Eclipse, and check for the following messages:

```
sample EmployeeMap for Adabas demo file 11 defined
sample VehicleMap for Adabas demo file 12 defined.
```

Now try to execute the query mapping java example file again.

Error Messages and Codes

Where can I find information about error messages and codes?

A list of ACJ messages and codes is included in the `QueryExceptionInfo` class. For further information, please refer to the javadoc package provided with the product.

What causes the message "log4j:WARN No appenders could be found for logger."

Sometimes three warnings can be observed on the console:

```
log4j:WARN No appenders could be found for logger.
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
```

They are caused by a missing "log4j.properties" configuration file. You must ensure that this file is a part of the class path.

Logging

How do I start logging ?

Any errors that might occur are logged by default. If the log4j logging facility is used (which is the default in Adabas Client for Java), a log4j.log file will be written.In order t o increase the log granularity, you can define the logging category in the log4j.properties file, e.g. to "DEBUG". There are several example configuration files provided with the product that define the logging category as "INFO".

How do I log a message and/or an exception?

As mentioned in the Release Notes for this version, the logging mechanism is based on the Simple Logging Facade for Java framework (for further information, please refer to http://www.slf4j.org/manual.html).

... SLF4J does not rely on any special class loader machinery. In fact, each SLF4J binding is hardwired at compile time to use one and only one specific logging framework. For example, the slf4j-log4j12-1.7.12.jar binding is bound at compile time to use log4j. In your code, in addition to slf4j-api-1.7.12.jar, you simply drop one and only one binding of your choice onto the appropriate class path location. Do not place more than one binding on your class path. ...

All necessary jar files are part of the lib folder provided to use log4j (for further information, please refer to **http://wiki.apache.org/logging/log4j1**) as underlying logging framework. An example property file *log4j.properties* is contained in the "classic" examples folder.

How do I activate logging in one of the examples provided?

Import the slf4j Logger and LoggerFactory in your application :

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

Create a final static logger and define a name for the logger :

```
final static Logger logger = LoggerFactory.getLogger(<your-classname>.class);
```

Then log your messages with different priorities, for example INFO, WARN, ERROR :

```
...

logger.info("Filenumber given : {} ", fileNumber);

...

catch (QueryException e) {
   logger.error("Catch Exception - reason : {}", e.getMessage());

...
```

In order to set logging level INFO and output to FILE, adapt the log4j.properties file accordingly :

```
log4j.rootCategory=INFO, LOGFILE

# File
log4j.appender.LOGFILE=org.apache.log4j.RollingFileAppender
log4j.appender.LOGFILE.File=log4j.log
```