software AG

ARIS RISK AND COMPLIANCE

DATA MIGRATION GUIDE FOR CUSTOMIZED VERSIONS

VERSION 10.0 - SERVICE RELEASE 22 AND HIGHER

APRIL 2023

Contents

1   Introduction

As of version 4.0 of ARIS Risk and Compliance the server has a migration framework for the incremental migration of data from previous versions. This document provides you with an introduction into the handling, operation and extension possibilities of this framework. It is oriented towards all developers who adapt ARIS Risk and Compliance to specific customer requirements and are responsible for data migration.

The following describes the data migration procedure for a customer-specific version of ARIS Risk and Compliance. For detailed information on data migration for a standard version, refer to **Upgrade to a new version of ARIS Risk and Compliance** (in the **Technical help** of the ARIS Risk and Compliance online help **> Upgrade** or in the **ARCM - Upgrade Guide.pdf**).

2 The migration framework of ARIS Risk and Compliance

The server has a mechanism for the incremental migration of data from previous versions. With it, data can be migrated in all database systems (Oracle, MSSQL and PostgreSQL) supported by ARIS Risk and Compliance. In addition, a portfolio of high-level API functions is available. These functions can be used within migration steps that you can write or change yourself to add tables and fields and to adapt data to new requirements.

The framework can be extended with Java classes. These must be available in text form. A compilation is not required. The internal migration logic determines the steps required to adapt data structures and data on the currently started server using an individually customizable XML migration plan.

Warning

In order to prevent data loss and irreversible changes to your data, you are recommended that you perform a full backup of all your data. To do so, use the administrative tool from your database system.

2.1 What the framework cannot do

The internal migration framework only processes data and its structures. This is sufficient for migrating the standard version of ARIS Risk and Compliance. However, it cannot adapt internal logic such as rules or workflows. This means for customer-specific adapted versions that in addition to the data migration mentioned here, the logic may have to be adapted.

3   Start the migration

From version 10.0.6, it is checked at server start whether the schema version in the connected database coincides with the current schema version of the ARCM server. If, during this check, the server detects that the server version does not coincide with the database version it tries to generate an appropriate migration strategy using **The migration plan** (Page 4). For detailed information about data migration with different database management systems, refer to **Install new version in parallel to existing installation** (in the **Technical help** of the ARIS Risk and Compliance online help **> Upgrade** or in the **ARCM - Upgrade Guide.pdf**).

Now a productive environment can be generated via a database export and import. To do so, create a new schema in the target database and generate the required tables by starting the ARIS Risk and Compliance Server. A database import can also be imported into another DBMS (database management system). It is possible, for example, to go live on an MSSQL server with a database migrated under Oracle.

4   The migration plan

The migration plan is an XML file with the name **migrationPlan.xml**. Here a version transition is assigned to a directory in which the corresponding migration logic is located. All configured version transitions (xml element migration) must be a continuous series. The series of transitions starts with the **sourceVersion** attribute of the oldest supported version and ends with the **targetVersion** attribute of the latest schema version. The **sourceVersion** attribute of a transition must match the **targetVersion** attribute of the previous transition. This does not apply to the first and last transition, since there is no predecessor and no successor.

4.1   Format

A version transition with the following attributes is defined in the **migration** tag:

- **id**: unique name of a transition in the migration plan. This id is entered into the **A_SCHEMAPROPERTY_TBL** table when the version transition is performed. Later, the migration history of the database can be tracked with this information.

- **sourceVersion**: contains the start version of the current version transition.

- **targetVersion**: contains the target version to be reached after the current version transition.

- **approach**: Indicates whether the current version transition refers to a specific approach. Up to version 10.0.10.0 a distinction exists between the risk-based approach (rba) and the control-based approach (cba).

  From version 10.0.12.0, only a unified approach is supported in ARIS Risk and Compliance. To ensure downward compatibility, the **approach** attribute must be set to **unified**, which is the only valid attribute value. See the XML example below.

- **implementation**: the folder that contains the migration logic for this version transition. The path to this folder is composed of **com.idsscheer.webapps.arcm.dl.datamigration** and the value of the implementation attribute.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<migrationPlan xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="./xsd/migrationPlan.xsd">

    <!-- Change value of attribute customizing from 'standard' to custom-schema
to run migration at custom-database. -->
    <schema customizing="standard"/>

    <migration id="10.0.10.0_RBA_to_10.0.12.0_RBA"
sourceVersion="10.0.10.0" targetVersion="10.0.12.0" approach="rba"
implementation="mig100100To100120" />
```

```
    <migration id="10.0.10.0_CBA_to_10.0.12.0_CBA"
sourceVersion="10.0.10.0" targetVersion="10.0.12.0" approach="cba"
implementation="mig100100To100120" />

    <migration id="10.0.12.0_STD_to_10.0.14.0_STD"
sourceVersion="10.0.12.0" targetVersion="10.0.14.0"
implementation="mig100120To100140" />

    <migration id="10.0.14.0_STD_to_10.0.16.0_STD"
sourceVersion="10.0.14.0" targetVersion="10.0.16.0"
implementation="mig100140To100160" />

    <migration id="10.0.16.0_STD_to_10.0.18.0_STD"
sourceVersion="10.0.16.0" targetVersion="10.0.18.0"
implementation="mig100160To100180" />

    <migration id="10.0.18.0_STD_to_10.0.20.0_STD"
sourceVersion="10.0.18.0" targetVersion="10.0.20.0"
implementation="mig100180To100200" />
```

4.2   The XML schema of the migration plan

The documented schema for the **migrationPlan.xml** file is integrated as a resource in the **arcm_datalayer_migration_migsteps-[version number].jar** library and is located in the **com.idsscheer.webapps.arcm.dl.datamigration.xsd** package. The schema allows you to use the context-sensitive auto-complete, syntax check, and help if you edit the **migrationPlan.xml** file in a development environment such as **IDEA** or **ECLIPSE**. This file cannot be changed. A modified schema cannot be processed by the migration framework.

4.3   The location of the migration plan

The migration plan is integrated in the **migrationPlan.xml** folder as a resource in the **arcm_datalayer_migration_migsteps-[version number].jar** library and is located in the **com.idsscheer.webapps.arcm.dl.datamigration** package. Changes and extensions are possible in the file **migrationPlan.xml**, as long as you generate a new library **arcm_datalayer_migration_migsteps-[version number].jar** with the modified class.

In Customizing Tool Kit (CTK) you need the sources of the standard migration in order to carry out custom adjustments. Subsequent to this, a library is created from these modified sources, which replaces the standard library. For detailed information on how to adjust migration of customer-specific databases in the CTK, refer to CTK user guide chapter **Implementing custom database migration steps**. You can download the user guide together with the CTK from the download area of ARIS Risk and Compliance in the Solution Book for partners (https://solutionbook.softwareag.com/partner/page/blueprintsolution.xhtml?guid=36768316).

5　The architecture

Java classes are saved in the source folder as migration steps (Page 4) that implement the **IMigrationStep** interface and extend the abstract **BaseMigrationStep** class. The method **::execute(...)** from **IMigrationStep** receives an instance from IMapping from the migration framework. To carry out the migration, various HighLevel functions from IMapping and IMigrationStep can now be used in the Execute method. The methods of the **IMapping** and **IMigrationStep** interfaces are stable and have Java documentation helps.

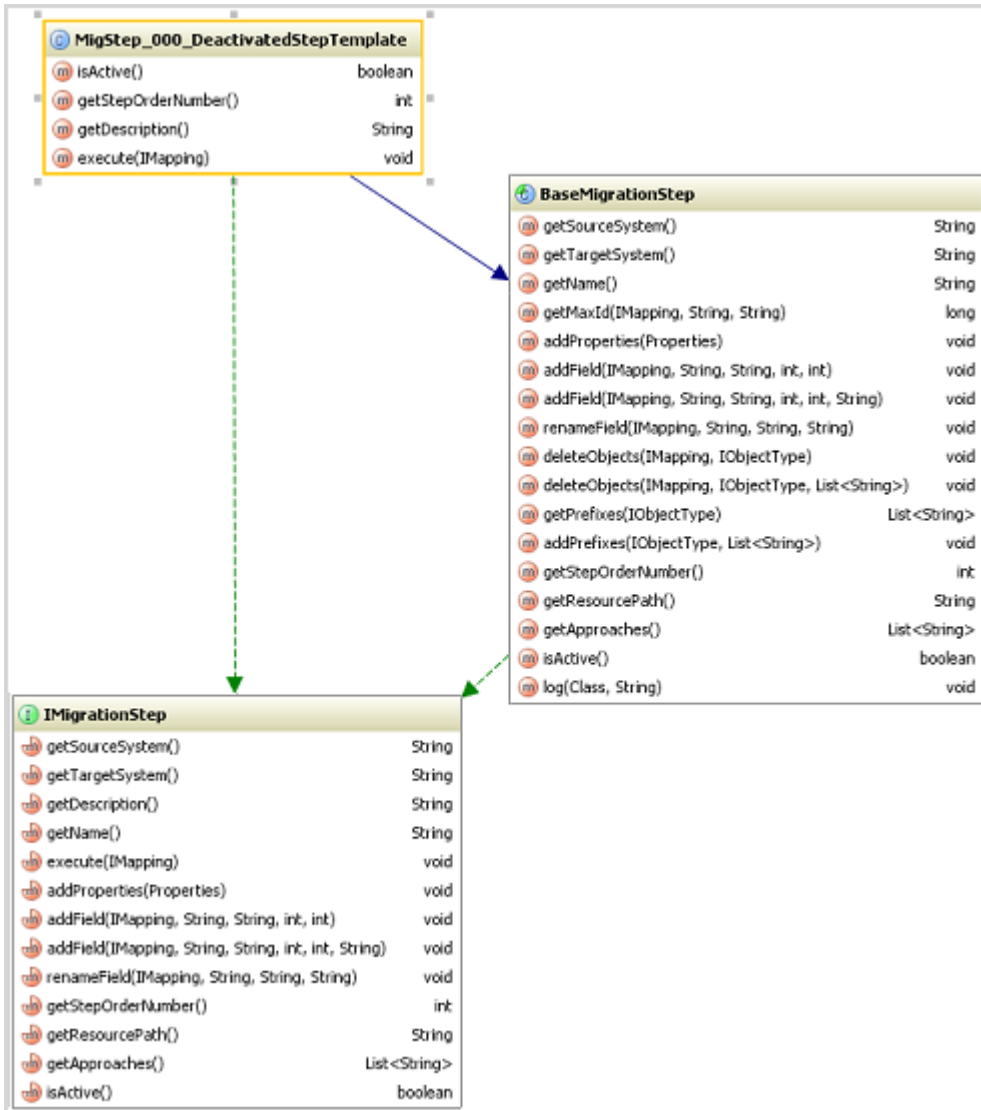| IMapping	
getConnection()	Connection
getDbmsType()	int
getDbmsTypeName()	String
genModifyTextFieldLen(String, String, int)	String
genAlterColumnName(String, String, String)	String
genAlterTableName(String, String)	String
genAddField(String, String, int, int)	String
genAddField(String, String, int, int, String)	String
genAddField(String, String, int, String)	String
genAddField(String, String, int, String, String)	String
genDeleteField(String, String)	String
migrateNumericToVarchar(String, String, String, int)	void
migrateVarcharToNumeric(String, String, String)	void
executeSQL(String)	void
createSequenceTableSQL()	String
createTableSQL(String)	String
executeQuery(String)	ResultSet
migrateClobToNumeric(String, String, String, String, Statement)	void
~~reorgSequence(Statement, String, String)~~	void
genCreateIndex(String, String, String...)	String
genAddDoubleField(String, String, int, int)	String
getErrorCodeIndexExists()	int
getErrorCodeInvalidIdentifier()	int
getDriver()	String
getUrl()	String
getUser()	String
getPwd()	String
genAddPrimaryKey(String, String, String...)	String
getDisableConstraintCall()	String
getEnableConstraintCall()	String
dropConstraints()	void
createConstraints()	void
executeScript(Statement, String, String)	void
isTableAlreadyPrepared(String)	boolean
isAttributeAlreadyPrepared(String, String)	boolean
refreshStatistic()	void
getDbSchemaVersion()	String

Figure 1: Mapping interface

Figure 2: Migration interface

5.1 The construction set

All public methods from the **IMigrationStep** and **IMapping** interfaces can be combined just like in a construction set to achieve the desired results.

5.1.1 IMigrationStep

The **migSteps** subfolder is located in the **implementation** (Page 4) folder that contains all of the logic and data regarding a version transition. This folder contains all migration steps required for version transition, which implement the **IMigrationStep** interface. The **IMigrationStep** interface provides help functions that package database-specific dependencies and that are not dependent on database-specific requirements. The following methods are not implemented by the abstract superior class **BaseMigrationStep** and must be implemented in the specific migration step.

- **IMigrationStep::getDescription() String**

  Provides the description of the step as a string.

- **IMigrationStep::execute(IMapping)**

  Carries out the step. All public methods of **IMapping** can be used.

The following methods can be overwritten in the specific migration step.

- **IMigrationStep::getStepOrderNumber()**

  Determines the order (priority) of the steps to be performed. The default implementation always returns the order number **0**. Overwrite this function and enter a number larger than **0**, according to the position at which this step should be carried out. The lower the number the higher the priority.

- **IMigrationStep::isActive() Boolean**

  Specifies whether this step should be carried out (**return true**) or not (**return false**). The default implementation returns true. The step is thus active and is carried out. Overwrite this function if you want to temporarily turn off a step during development. Some steps, such as the generation of database indices, should be deactivated as soon as a follow-up version is available. Indices should then be generated in the last version transition only.

The interface provides other useful functions for processing data and data structures. These are documented in the Java documentation.

5.1.2   IMapping

The **IMapping** interface provides help functions that package data-specific properties. With these functions you can for example, add tables or fields and process data. A few of these functions perform the relevant operations immediately and then close the required database resources automatically. A few functions also provide database resources directly. In this case, you must be sure to close these resources yourself in a Finally-block.

Close database resources such as **Connection** or **ResultSet**, which you receive from the **IMapping** interface, when they are no longer required. All methods in the **IMapping** interface affect the database connection specified in the **runtimeconfig.xml** configuration file in the **Datalayer** section. The interface provides other useful functions for processing data and data structures. These are documented in the Java documentation.

6   MigrationObject

The MigrationObject is a help structure with which to write data consistently in schema tables for ARIS Risk and Compliance. Generate and write the objects sequentially in order to prevent conflicts with the internal ID management. This help structure does not generate the tables, but rather fills in the data semantically correct.

It is possible to generate a MigrationObject with the operator **new**.

```
MigrationObject migObject = new MigrationObject("POLICYREVIEWTASK", mapping,
this, UUID.randomUUID().toString(),
OVIDFactory.getOVID(SystemGUID.INTERNAL_SYSTEM_USER.getObjID()));
```

The newly generated object provides an API that can be used to specify attributes with their values.

```
migObject.setAttribute("reviewRelevant", IMapping.TYPE_NUMBER, "0");
```

Relations can also be specified like this:

```
migObject.setRelationAttribute("POLICYREVIEWTASK", "owner_group",
IMapping.TYPE_RELATION_1_1, ownerGroupID, 5520, 0, null);
```

With the **::write()** function the object can be written in the database.

```
migObject.write();
```

In order to receive a complete overview of the API from this class, check the **javadoc** of the file **MigrationObject.java**.

7 Automatic update of the schema version

In older versions of the migration framework it was necessary to write your own MigrationStep to specify the **currentSchemaId** field in the **A_SCHEMAPROPERTY_TBL** table. This is no longer necessary with the current version of the framework. The corresponding field is now specified automatically by the framework during the data migration.

8   Partly automatic cleanup

Each migration includes the automatic migration step **MigStep_CleanupMigration**. This migration step compares the metadata of the database with the metadata of ARIS Risk and Compliance, corrects the database if possible, and if this is impossible, outputs an error message. This results in a timely warning about inconsistencies concerning the database structure. The following individual steps are performed:

- Create index, foreign and primary keys, as well as unique constraints.

- Remove dummy fields that were created during the creation of new objects.

- If required, enlarge field lengths, pre-decimal and decimal places, and output this information in the log file. However, you are recommended always specifying the field lengths properly. Data is not adjusted.

- If objects, attributes and 1:1 relationship attributes are missing an error is output.

9   Adjustments to the data migration in Customizing Tool Kit (CTK)

For detailed information on how to adjust migration of customer-specific databases in the CTK, refer to CTK user guide chapter **Implementing custom database migration steps**. You can download the user guide together with the CTK from the download area of ARIS Risk and Compliance in the Solution Book for partners (https://solutionbook.softwareag.com/partner/page/blueprintsolution.xhtml?guid=36768316).

10   Logging

Logging during migration takes place according to the settings in **log4j2.xml**. Set the **arcm** and **dl.framework** packages to **debug** before migration.

```
<Logger name=" com.idsscheer.webapps.arcm" level=" DEBUG "/>
<Logger name=" com.idsscheer.webapps.arcm.dl.framework" level=" DEBUG "/>
```

The migration output is displayed on the console and in the output file that are set in the **log4j2.xml** configuration file.

To prevent poor performance do not forget to undo these changes in the productive system.

Check the resulting log file carefully for error messages before you export the file and then import it into the productive system.

11 Technical help

USER MANUALS OF ARIS RISK AND COMPLIANCE

- The following manuals are now available in the **Technical Help of ARIS Risk and Compliance**:

    - **ARCM - Installation Guide**

    - **ARCM - Upgrade Guide**

    - **ARCM - Customizing Guide**

    - **ARCM - Data Migration Guide for Customized Versions**

    You can also find them as PDF in the documents folder on your installation media.

- The contents of the following manuals are now available in the online help of ARIS Risk and Compliance:

    - **Modeling Conventions**

    - **Semantics verifications**

    - **Administration**

- The content of **Confirmation Management Dashboard Installation** manual can now be found in the online help of ARIS under **Work with content > Confirmations > 'Confirmation management' dashboard**.

ADDITIONAL INFORMATION

For more information about ARIS, such as manuals, technical help, and compendiums, see **Additional information** in the online help of ARIS.

12 Legal information

12.1 Documentation scope

The information provided describes the settings and features as they were at the time of publishing. Since documentation and software are subject to different production cycles, the description of settings and features may differ from actual settings and features. Information about discrepancies is provided in the Release Notes that accompany the product. Please read the Release Notes and take the information into account when installing, setting up, and using the product.

If you want to install technical and/or business system functions without using the consulting services provided by Software AG, you require extensive knowledge of the system to be installed, its intended purpose, the target systems, and their various dependencies. Due to the number of platforms and interdependent hardware and software configurations, we can describe only specific installations. It is not possible to document all settings and dependencies.

When you combine various technologies, please observe the manufacturers' instructions, particularly announcements concerning releases on their Internet pages. We cannot guarantee proper functioning and installation of approved third-party systems and do not support them. Always follow the instructions provided in the installation manuals of the relevant manufacturers. If you experience difficulties, please contact the relevant manufacturer.

If you need help installing third-party systems, contact your local Software AG sales organization. Please note that this type of manufacturer-specific or customer-specific customization is not covered by the standard Software AG software maintenance agreement and can be performed only on special request and agreement.

12.2 Support

If you have any questions on specific installations that you cannot perform yourself, contact your local Software AG sales organization (https://empower.softwareag.com/Products/default.aspx). To get detailed information and support, use our Web sites.

If you have a valid support contract, you can contact **Global Support ARIS** at: **+800 ARISHELP**. If this number is not supported by your telephone provider, please refer to our Global Support Contact Directory.

For issues regarding the product documentation, you can also send a mail to documentation@softwareag.com (mailto:documentation@softwareag.com).

ARIS COMMUNITY

Find information, expert articles, issue resolution, videos, and communication with other ARIS users. If you do not yet have an account, register at ARIS Community**.**

PRODUCT DOCUMENTATION

You can find the product documentation on our documentation Web site (https://documentation.softwareag.com).

You can use numerous links on the TECHcommunity Web site. For any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory and give us a call.

TECHCOMMUNITY

On the **TECHcommunity** Web site, you can find documentation and other technical information:

- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.

- Access articles, code samples, demos, and tutorials.

- Find links to external Web sites that discuss open standards and web technology.

- Access product documentation, if you have **TECHcommunity** credentials. If you do not, you will need to register and specify **Documentation** as an area of interest.

EMPOWER (LOGIN REQUIRED)

If you have an account for **Empower**, use the following sites to find detailed information or get support:

- You can find product information on the Software AG Empower Product Support Web site.

- To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center.

- Once you have an account, you can open Support Incidents online via the eService section of Empower.

- To submit feature/enhancement requests, get information about product availability, and download products, go to Products.

FURTHER INFORMATION AND TRAININGS

Learn from your laptop computer, tablet or smartphone.