

# **ARIS**

# UPDATE COOKBOOK

Manually updating ARIS



**ARIS 9.8 SR6**

October 2016

FOR EXPERIENCED ARIS ADMINISTRATORS ONLY

# Table of contents

<b>1</b>	<b>Target group .....</b>	<b>3</b>
<b>2</b>	<b>Scope .....</b>	<b>3</b>
<b>3</b>	<b>Preconditions .....</b>	<b>3</b>
3.1	Required credentials .....	3
3.2	Available disk space .....	3
3.3	Activating deactivated runnables (for source versions 9.8.4 and later) .....	3
<b>4</b>	<b>Backup data .....</b>	<b>4</b>
<b>5</b>	<b>Cleaning up the installation .....</b>	<b>4</b>
5.1	Archiving and deleting old log files (OPTIONAL) .....	4
5.2	Deleting only selected log files (OPTIONAL) .....	4
<b>6</b>	<b>Document and backup all manual changes .....</b>	<b>5</b>
6.1	Document manual modifications of files and folders .....	5
6.2	Removing CDF of HD server.....	5
6.3	Removal of the TM-Bundle runnable.....	5
<b>7</b>	<b>Update procedure .....</b>	<b>5</b>
7.1	Updating the Oracle® schema .....	5
7.2	Updating the agents on all nodes.....	6
7.3	Start ARIS Cloud Controller .....	6
7.4	Set remote repository on all nodes.....	7
7.5	Update all runnables.....	7
7.6	Reconfigure L sizing when updating to 9.8 or later (OPTIONAL) .....	9
7.7	Start ARIS after the update .....	9
7.8	Trigger a reindexing.....	9
7.9	Update the system database (OPTIONAL) .....	10
7.10	Adding ARIS help.....	10
7.11	Reset to local repository (OPTIONAL) .....	12
7.12	Workaround for elasticsearch .....	12
7.13	Changes for LDAP since ARIS 9.8.....	12

## 1 Target group

- ! The ARIS Update Cookbook is meant for ARIS administrators who are experienced in manually installing ARIS.
- Knowing just how to install ARIS by using the provided ARIS standard setups is insufficient.

Manual installations, also known as distributed installations, are needed when one installation of ARIS Connect/Design Server has to run across several computers in order to fulfill the required performance, or a high-availability or fail-over scenario.

For every ARIS server installation Software AG strongly recommends that you request an installation service by Global Consulting Services. You can order that service from your Software AG sales representative. This is of particular importance when you intend to install ARIS as a so called multi-node installation, i.e., an installation spanning across several computers/VMs. Such a scenario requires profound knowledge of the technical ARIS infrastructure and environment. This cannot be provided in the product documentation. The required know-how can be acquired only by attending the training course "ARIS Server Installation", available via Global Education Services (<http://softwareag.com/training>).

## 2 Scope

In case you have installed ARIS 9.x via the setups provided on the ARIS DVD, there is no need to follow this document, because the setups perform all changes to properly update your ARIS installation. In fact, in such cases you don't need to manually update ARIS at all.

Hence, this document is only meant for ARIS installations that have been performed manually using ARIS Cloud Controller, or when you have changed your ARIS installation in such a way that the ARIS setups refuse to update your installation, e.g., configured an additional runnable on top of your ARIS installation via ARIS Cloud Controller (ACC).

With this document you can also understand what is going on under the hood of an ARIS setup when updating an ARIS installation, considering that a setup program can only run on a single node installation, whereas these procedures here are typically meant for multi-node or non-distributed installations.

This document covers the procedures to *manually* perform an update for these ARIS versions:

- From ARIS 9.8.x to 9.8.6
- From ARIS 9.7.x to 9.8.6
- From ARIS 9.6.x to 9.8.6

To some extent the three update procedures are redundant to each other. Depending on the configuration of your ARIS installation, some update steps might be optional, e.g., in case you are not using the built in ARIS database system but an external DBMS (Oracle or Microsoft® SQL Server®).

Such cases are indicated at the beginning of the respective section. If there is no such indication, the step is required regardless of current and planned ARIS version.

## 3 Preconditions

Before starting the update of an ARIS installation, check if the following preconditions are met.

### 3.1 Required credentials

During the update, you will in general need the following credentials:

- The username and password of ARIS Agent on each node that is part of the installation that should be updated (default: **Clous/g3h31m**)
- The password of the user **superuser** (default: **superuser**), or alternatively, the password of a user with all functional privileges for **each** tenant

### 3.2 Available disk space

For the update to be successful, sufficient disk space has to be available on each node that is part of the installation that should be updated. The following is a rough guideline on how to determine the disk space that will be needed:

- Determine the available free disk space on that computer where ARIS Agent runs
- Determine the current folder size of the runnables' working directories
  - Microsoft® Windows®: **<ARIS installation folder>\server\bin\work**
  - Linux®: **/home/ARIS/cloudagent/work**
- Make sure that the size of the determined free disk space is at least 15 GB larger than the folder size of the working directories

- ! **DO NOT ATTEMPT AN UPDATE** if there is not enough free disk space available on the target system.
- The update would fail and result in a broken installation with little chances for a repair.

### 3.3 Activating deactivated runnables (for source versions 9.8.4 and later)

Starting with ARIS 9.8 SR4 (Versions  $\geq 9.8.4$ ) it is possible to deactivate individual runnables. A deactivated runnable will be ignored by the (on all nodes) startall and the regular start command. This can be useful in cases where the full product is installed, but where only a subset of the functionality is initially used (for example, a customer might start out with only the ARIS Design Server, but might want to install the full set of ARIS Connect Server runnables, to ease a later upgrade to ARIS Connect. In this case, the customer might deactivate the ECP and PostgresECP runnables).

Still, to keep an installation consistent, deactivated runnables have to be updated with all others. Correspondingly, a deactivated runnable is not ignored by, e.g., the update or updateall commands and can be updated without problems. However, some of the steps described in this guide might require the entire installation or at least one instance of a specific application to be started for the step to be successful. So unless you fully understand the dependencies of the different runnables and which runnables are involved in which step, it is therefore highly recommended to activate all runnables before proceeding and deactivate them again later.

## 4 Backup data

**!** Updating data without having a proper backup is a risk you might not want to take on!

Although an automatic backup is performed by default by the ARIS setups, it is recommended to back up all data before starting the update process. In manually installed ARIS environments you need to run a backup manually. There are basically three ways to do a backup:

- Copy the working directories** of all runnables to a safe place (or for simplicity the whole installation folder). In addition, if you use an external DBMS (Oracle, Microsoft SQL Server) you will need to do a dump of all schemas containing ARIS data.
- Run backup tenant** for all tenants using ACC. The resulting backup contains all essential data of a tenant, with the notable exception of any portal customizations and tenant-specific portal settings.
- Use the backup feature** offered by most applications at their user interface or their respective command-line tools (backup databases via ARIS Architect or ARIS Server Administrator, backup UMC via browser or y-tenantmgmt.bat etc.). Not all applications offer such a feature.

## 5 Cleaning up the installation

During the actual update procedure, backups of the working directories of all runnables are created in order to be able to roll back to the original version if anything goes wrong.

Depending on how many files there are in the runnables' working directories, these backups will take up considerable space and it can take considerable time to create them. Particularly in environments with limited disk I/O (often the case for virtual machines or physical machines that use a SAN or NAS as their IO subsystem), the time to create these backups can significantly influence the total time for performing the update. Nonetheless, it is highly recommend to leave the automatic backup of runnable working directories enabled.

However, as a prerequisite, you can remove unnecessary files from the working directories. Particularly, prior versions of ARIS were writing a large amount of log files which would not be cleaned automatically (this has been improved considerably in more recent versions).

### 5.1 Archiving and deleting old log files (OPTIONAL)

One option is to first archive and then delete all log files of the installation. For that, there are the batch files on Windows (**collectLogFiles.bat**, **deleteLogFiles.bat**) provided in the folder **<ARIS installation folder>\support**.

On Linux systems these files have the extension .sh (instead of .bat) and they can be found in the home directory of the ARIS user that is created by the ACC Linux packages, typically **/home/ARIS**.

- collectLogFiles** collects the log files of all runnables and ARIS Agent and saves them into a single archive **LogFiles.zip** in the same folder. Depending on the size of log files accumulated in your installation, this can take some minutes and of course you need additional free space on the disk for collecting and archiving all log files.
- deleteLogFiles** removes all log files from the installation directory and frees up disk space the log files used up.

### 5.2 Deleting only selected log files (OPTIONAL)

Optionally, you could keep the most recent of your log files and just go through the various log folder of all runnables to selectively only delete the oldest and/or largest files. The following table shows the location of the log files for the different runnables.

**!** NEVER DELETE ANY LOG FILES OUTSIDE THE FOLLOWING FOLDERS!

Runnables	Folder of main log file relative to the work folder, e.g., C:\SoftwareAG\ARIS9.8\bin\work, sizing in "M"	Secondary log location
ARIS Agent	C:\SoftwareAG\ARIS9.8\server\logs\ARISCloudAgent.log	C:\SoftwareAG\ARIS9.8\server\bin\log
zoo	work_zoo_m\log\zookeeper.log	work_zoo_m\log\zookeeper.log\zookeeper.log.<index>
postgres	work_postgres_m\pgworkdata\pglog.txt	work_postgres_m\pgworkdata\pg_log\*
postgresECP	work_postgresECP_m\pgworkdata\pglog.txt	work_postgresECP_m\pgworkdata\pg_log\*
couchdb	work_couchdb_m\log\couch.log	-
cloudsearch	work_cloudsearch_m\defaultLog\log.txt	-
elastic	work_elastic_m\elasticsearch\logs\*	-
adsadmin	work_adsadmin_m\base\logs\*	-
umcadmin	work_umcadmin_m\base\logs\*	work_umcadmin_m\tools\logs\*
loadbalancer	work_loadbalancer_m\httpd\logs	work_loadbalancer_m\httpd\logs\*
abs	work_abs_m\base\webapps\abs\log\*	work_abs_m\base\logs\*
apg	work_apg_m\base\logs\age.log	work_apg_m\base\logs\*, work_apg_m\logs
copernicus	work_copernicus_m\base\logs\catalina.<date>.log	work_copernicus_m\base\logs\*
octopus	work_octopus_m\base\logs\catalina.<date>.log	work_octopus_m\base\logs\*
ecp	work_ecp_m\base\logs\catalina.<date>.log	work_ecp_m\base\logs\*
simulation	work_simulation_m\base\logs\catalina.<date>.log	work_simulation_m\base\logs\*

Table: Log folders of all runnables

## 6 Document and backup all manual changes

! An important rule of thumb is that manual changes of an installation will not survive an update!

In general, **any manual, physical modification** of an ARIS installation, in particular the manual modification or addition of files, **should be assumed to NOT be preserved during an update.**

All manual changes (i.e., those not done using ACC commands) that you did after your initial installation of ARIS 9.x must therefore be documented in an installation document. Only modifications made using ACC commands (**reconfigure**, **enhance**) can be deemed safe unless stated otherwise.

Any content that you put into ARIS (models, users, documents, collaboration comments, etc.) using the user interface or other official interfaces will be preserved during an update.

### 6.1 Document manual modifications of files and folders

After performing the update you must repeat your documented changes to those files and folders where you have made manual changes. Some of the changes you might be considering no longer relevant in the updated version.

In general it is recommended to use ACC commands (in particular **reconfigure** and **enhance**) to make configuration changes or add third party files to your installation, such as Java® DataBase Connectivity (JDBC®) drivers, SAP® JCo adapters and SSL certificates.

ARIS server products	Modifiable files/folders (example in sizing "m", path relative to <ARIS installation folder>\server\bin\work)
ARIS Connect/Designer Server	\work_abs_m\base\webapps\abs\downloadClient\config\arisloader.cfg \work_abs_m\base\webapps\abs\config\defaultServerSettings.cfg
ARIS Publisher Server	\work_businesspublisher_m\base\webapps\businesspublisher\config\webappserver.cfg \work_businesspublisher_m\base\webapps\businesspublisher\config\umcconfig.cfg \work\work_businesspublisher_m\base\webapps\businesspublisher\layouts\extensions\sap_cxn

! Any changes in the above files or added files will be lost after an update!

### 6.2 Removing CDF of HD server

The following is mandatory if your installed ARIS version is 9.8 SR3 or earlier **AND** if you manually configured the so called "heavy duty" server.

Up to ARIS 9.8 SR3 is was required to configure a CDF runnable in order to use the HD server. With ARIS 9.8 SR4 or newer that's no more needed.

However, before proceeding with the update, it's needed to remove the CDF runnables using the deconfigure command.

To remove a CDF runnable in your installation, use the deconfigure command

```
on <nodename> deconfigure cdf_<instance_size>
```

Replace <nodename> with the name of the node on which the CDF runnable is located, and replace <instance\_size> with the sizing (s, m, or l) of the CDF runnable.

When prompted confirm the command with "y". When deconfiguring via batch script add **force** in front of **deconfigure** to skip the user prompt.

Run the deconfigure command for all CDF runnables in your installation.

Example: Two CDF instances (cdf1, cdf2) on two different nodes (n1, n2) require these commands

```
on n1 deconfigure cdf1
on n2 deconfigure cdf2
```

### 6.3 Removal of the TM-Bundle runnable

The following update is mandatory when updating from any version >=98.2.0 and < 98.6.x to any version >= 98.6.x where the tenant management bundle runnable had been manually added.

When updating an installation to which the TM bundle runnable has been added, which contains the Tenant Management application, all TM bundle runnable instances have to be removed from the system using the **deconfigure** command before proceeding with the update.

From ARIS 9.8.6 onwards the Tenant Management is fully integrated in the UMC Admin runnable. Therefore, this additional runnable is no longer needed. To remove a TM bundle runnable in an installation, use the **deconfigure** command:

```
on <nodename> deconfigure <tmBundleRunnableInstanceId>
```

Replace <nodename> with the name of the node on which the TM bundle runnable is located and replace <tmBundleRunnableInstanceId> with the instance ID of the TM bundle runnable.

Confirm the to the security prompt with "y", or use the prefix keyword **force** in the command to skip that prompt, which is risky but useful if when included in a batch script where no user interaction is not feasible.

Run the deconfigure command for all TM bundle runnables in your installation. For example, if you have a TM bundle instance named "tm1" on a node named "n1" and another TM bundle instance named "tm2" on a node named "n2", you would run these commands:

```
on n1 deconfigure tm1
on n2 deconfigure tm2
```

## 7 Update procedure

### 7.1 Updating the Oracle schema

This step is only needed if you are using the external DBMS Oracle in your installation.

1) Stop all runnables on all nodes via ACC command:

```
on all nodes stopall
```

In older versions the **on all nodes stopall** command is not yet available or might not work properly due to a bug. For an installation based on such a version, you already should have stop script (consisting of individual **stop** and **wait for** commands) which you should use instead.

2) Update the database schema for each tenant in your system. To do so prepare and execute the database update script (ARIS DVD folder: **Add-ons\DatabaseScripts\Oracle\Design&ConnectServer**)

Enter the relevant information about your Oracle system in **envset.bat**. Details are available in the **ARIS Server Installation and Administration Guide.pdf** (DVD folder: **Documents\English**).

3) Identify the names of the schemas of all tenants that are used by the installation.

a) If the schemas were originally created following the recommended naming convention, the schema names will look like **aris\_<tenant\_ID>** (e.g. for tenant default: **aris\_default**)

b) If you do not know the schema name of a tenant, you can get it via an ACC command **list tenant <tenant\_ID> data for service db**. Example for tenant schema default:

```
list tenant default data for service db
```

The result of the command will show the schema name:

Key	Value
com.aris.cib.db.schema	aris_default

4) For each tenant's schema run the script **cip\_update\_schema\_for\_tenant.bat** with the schema name as parameter.

Example command for schema **aris\_default**:

```
cip_update_schema_for_tenant.bat aris_default
```

! It is important that the schemas of all existing tenants are updated this way, even if the tenants are currently not used. This includes both tenants: default and master.

5) ▲ This step is only necessary if you are using Oracle AND if your ARIS version is **exactly ARIS 9.7 GA**.

Check your ARIS version via ACC:

```
on all nodes get agent.version
```

If the version number is in the range 97.0.0.5.946097 to 97.0.0.10.958105 your ARIS version is **ARIS 9.7 GA**.

Version 97.1.0.0.981214 or higher DOESN'T need the following update.

Execute the following SQL commands using Oracle SQLPlus, SQL Explorer, or a similar tool for running SQL statements. For each tenant replace <TenantSchema> with the schema name of the respective tenant:

```
ALTER TABLE <TenantSchema>.ardb_ExplImplLocks MODIFY LOB(SVALUE) (CACHE);
ALTER TABLE <TenantSchema>.SysPendingPackets MODIFY LOB(packetData) (CACHE);
ALTER TABLE <TenantSchema>.ardb_SymbolLocks ADD (VERSION number(10,0) DEFAULT 0 not null);
ALTER TABLE <TenantSchema>.ardb_ExplImplLocks ADD (VERSION number(10,0) DEFAULT 0 not null);
ALTER TABLE <TenantSchema>.ardb_LockOwner ADD (VERSION number(10,0) DEFAULT 0 not null);
```

### 7.2 Updating the agents on all nodes

#### 7.2.1 Close open files and folders

In particular on Windows systems, if a file is still in use by a process, it cannot be replaced during the update. Therefore, it is absolutely necessary to make sure that no programs are still using any files inside the ARIS installation folder. To do so, make sure that you

- Close all ACC command line windows
- Close all Command Prompts and Windows Explorers that are pointing into the ARIS installation folder
- Close all files inside the ARIS installation folder that are still open by external programs, such as log files opened in an editor

#### 7.2.2 Update ARIS Agent

Update ARIS Agent on all nodes of the installation by running the ARIS Agent setup on Windows or using the update packages for Linux.

Refer to the **ARIS Server Installation and Administration Guide** for detailed instructions on how to do the update.

! The ARIS Agent setup has to stop all runnables in order to be able to update the Java® Runtime Environment (JRE).

#### 7.2.3 Validate ARIS Agent

After the update check the versions of all agents currently registered in ACC with this ACC command:

```
on all nodes get agent.version
```

Make sure that all lines show the ARIS version that you want to update to.

### 7.3 Start ARIS Cloud Controller

Copy the file **generated.apptypes.cfg** of the new ARIS version from an ARIS server setup folder to the ARIS installation folder and overwrite the existing file of the current ARIS version (on DVD in **Setups\Windows\ARIS\_Connect\_Server** when updating an ARIS Connect Server, or in **Setups\Windows\ARIS\_Design\_Server** when updating an ARIS Design Server).

Start the ACC batch **acc.bat** with the command line options **-c <path to generated.apptypes.cfg>** in order to provide ACC with the necessary information about runnable updates of the new ARIS version.

Make sure that ACC is started from now on with the new **generated.apptypes.cfg**.

**Example:** `acc.bat -c "C:\ARIS9.8\server\generated.apptypes.cfg"`

Add all nodes of your environment:

- Either use the command **add node** manually whenever you start ACC
- Or use a **node file** (e.g. a file containing **add node** commands) which you can pass to ACC on startup by the command line option **-n**

#### Background information for updates to ARIS version 9.8 or higher:

*The default settings for runnables provided in generated.apptypes.cfg can change from version to version. However, in versions before 9.8, changes of the settings in the new generated.apptypes.cfg (everything in the lines after "defaults") will not be applied during an update.*

For versions 9.8 and later, only a selected subset of default parameter changes will be applied. You can identify those parameters that will be changed during an update in the generated.apptypes.cfg by them being marked with either a ! or a ~, e.g., consider this "register app type" command:

```
register app type umcadmin_s com.aris.umcadmin.y-umcadmin-run-prod 98.2.0 defaults JAVA-Xmx="128m" \
~httpNumWorkerThreads=50 ~ajpNumWorkerThreads=50 CATALINA_PORT=10083 CATALINA_AJP_PORT=11083 \
JAVA_OPTS="-server" JAVA-Dcom.sun.management.jmxremote="/enabled" \
JAVA-Dcom.sun.management.jmxremote.port="9602" JAVA-Dcom.sun.management.jmxremote.authenticate="false" \
JAVA-Dcom.sun.management.jmxremote.ssl="false" ~CATALINA_AJP_packetSize=32768" \
~CATALINA_HTTP_maxHttpHeaderSize=32768 !zookeeper.session.timeout.ms="60000" \
zookeeper.connect.retry.time.ms="30000" -"JAVA-XX\:\:MaxPermSize"
```

Parameters marked with "~" (shown in blue color) will be applied during update if the parameter is not already set for the runnable (i.e., if the parameter was already set or changed by the customer, the customer's changes will be preserved).

Parameters marked with "!" (shown in red color) will be applied during an update, regardless of whether the parameter is already set or not (i.e., if the parameter was already set or changed by the customer, **the customer's changes will be overwritten(!)**).

### 7.4 Set remote repository on all nodes

Tell all agents NOT to use the local repository with the ACC command:

```
on all nodes set local.repository.only=false
```

#### 7.4.1 Provide a remote repository

There are two options to provide a remote repository:

- a) Use a Web server (similar procedure for FTP server)

Copy the DVD folder **Setups\Windows\Setup\_Data** to a physical drive inside a web or FTP server.

- ! When using Internet Information Server to serve as remote repository add the extension **.war** with **MIME Type text/plain** to avoid issues with the import of the ARIS help files

Specify for all agents the **Setup\_Data** folder in which the repository resides with all runnables of the new ARIS version:

```
on all nodes set remote.repository.url="http://<webserver>/<full_path_to_Setup_Data>"
```

Example command for a webserver **intranet** running on port 8080 and the contents of the **Setup\_data** folder available in the subdirectory **arisrepo/98.6.0**:

```
on all nodes set remote.repository.url="http://intranet:8080/arisrepo/98.6.0"
```

- b) Use a local directory on a physical drive on each node as a "simulated" remote repository

Copy the **Setup\_Data** folder to a *local physical* drive accessible on every node and point the remote.repository.url of every node to that location. Use the same directory on every node to simplify the ACC commands below.

- ! Note that it really has to be a physical drive, mapped drives as storage location for the repository can't be used, because ARIS Agent runs in the security context "SYSTEM" which can only access physical drives.

Specify for all agents the local directory as file URL:

```
on all nodes set remote.repository.url="file:///<path_to_physical_folder>"
```

Example, where backslashes of Windows are replaced by forward slashes:

```
on all nodes set remote.repository.url="file:///C:/ARIS_DVD-98.2.0.1039000"
```

### 7.5 Update all runnables

- a) When updating to a version **before 9.8** execute the ACC command

```
on all nodes updateall
```

- b) When updating to version **9.8 or newer** execute the ACC command (replace <sizing\_value> with s, m or l as used in your original installation sizing)

```
on all nodes updateall sizing <sizing_value>
```



! The old working directories will be copied into a backup directory before anything is changed. You can find these backups in **<ARIS Installation path>\server\bin\backup** (e.g., the directories in which the runnables store their data).

These backups will allow the **on all nodes updateall** command to do a rollback of the complete update in case of an error so that the version is again consistent by restoring all runnables to their previous version, instead of having a mixture of runnables with previous and new version that will most likely not work.

If you have a mixture of runnables of different sizings, you can do the update manually for each runnable using the command

```
on <nodeName> update <instance_ID> sizing <sizing_value>
```

To reduce the amount of effort and individual commands needed, identify which sizing is the least common and update the runnables of this sizing individually with the "update" command.

For example, if you have originally installed all runnables using the M sizing except two ABS runnables (on nodes n1 and n2) which you installed using the I sizing, you would first update the ABS runnables with the I sizing, using the commands

```
on n1 update abs_i sizing i
on n2 update abs_i sizing i
```

Finally, you would update all other runnables using the m sizing with the regular "updateall" command (already done updates are automatically ignored):

```
on all nodes updateall sizing m
```

Note that ARIS Agent updates will NOT be rolled back, but this normally shouldn't be a problem, the new agent should be able to work with the old runnables.

The backup of the working directories takes some time and consumes disk space. It can be turned off, but it is not recommended to do so because then the rollback mechanism will not work anymore.

! The automatic backup mechanism and the rollback that depends on the backups is just a safety net. It is not meant to replace a backup of your application data which you should always do before an update!

If you attempt an update without backing up your data, any system outages and data loss will be your own responsibility.

**Example:** If **on all nodes updateall** was successful, the output of ACC should look similar to the following:

```
ACC+ >on all nodes updateall sizing m
WARNING! The updateall command will update ALL currently not running instances on all nodes.
Are you sure? (Y/N)
y
Runnable zoo0 on node n1 will be updated to version 98.2.0.
Runnable cloudsearch_m on node n1 will be updated to version 98.2.0.
Runnable umcadmin_m on node n1 will be updated to version 98.2.0.
Runnable abs_m on node n1 will be updated to version 98.2.0.
Runnable apg_m on node n1 will be updated to version 98.2.0.
Runnable copernicus_m on node n1 will be updated to version 98.2.0.
Runnable zoo0 on node n2 will be updated to version 98.2.0.
Runnable elastic_m on node n2 will be updated to version 98.2.0.
Runnable couchdb_m on node n2 will be updated to version 98.2.0.
Runnable adsadmin_m on node n2 will be updated to version 98.2.0.
Runnable umcadmin_m on node n2 will be updated to version 98.2.0.
Runnable octopus_m on node n2 will be updated to version 98.2.0.
Runnable simulation_m on node n2 will be updated to version 98.2.0.
Runnable loadbalancer_m on node n2 will be updated to version 98.2.0.
Runnable zoo0 on node n3 will be updated to version 98.2.0.
Runnable postgresECP_m on node n3 will be updated to version 98.2.0.
Runnable cloudsearch_m on node n3 will be updated to version 98.2.0.
Runnable couchdb_m on node n3 will be updated to version 98.2.0.
Runnable adsadmin_m on node n3 will be updated to version 98.2.0.
Runnable abs_m on node n3 will be updated to version 98.2.0.
Runnable copernicus_m on node n3 will be updated to version 98.2.0.
Runnable ec_p_m on node n3 will be updated to version 98.2.0.
Runnable loadbalancer_m on node n3 will be updated to version 98.2.0.
Successfully updated runnable zoo0 on node n2 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable elastic_m on node n2 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable couchdb_m on node n2 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable adsadmin_m on node n2 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable umcadmin_m on node n2 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable octopus_m on node n2 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable simulation_m on node n2 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable loadbalancer_m on node n2 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable zoo0 on node n1 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable cloudsearch_m on node n1 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable umcadmin_m on node n1 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable abs_m on node n1 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable apg_m on node n1 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable copernicus_m on node n1 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable zoo0 on node n3 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable postgresECP_m on node n3 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable cloudsearch_m on node n3 from version 96.0.3.3 to version 98.2.0.
Successfully updated runnable couchdb_m on node n3 from version 96.0.3.3 to version 98.2.0.
```



Successfully updated runnable adsadmin\_m on node n3 from version 96.0.3.3 to version 98.2.0.  
 Successfully updated runnable abs\_m on node n3 from version 96.0.3.3 to version 98.2.0.  
 Successfully updated runnable copernicus\_m on node n3 from version 96.0.3.3 to version 98.2.0.  
 Successfully updated runnable ecp\_m on node n3 from version 96.0.3.3 to version 98.2.0.  
 Successfully updated runnable loadbalancer\_m on node n3 from version 96.0.3.3 to version 98.2.0.  
 23 of 23 runnables updated.

## 7.6 Reconfigure L sizing when updating to 9.8 or later (OPTIONAL)

If the default configuration of a runnable has changed from one version to another, the number of worker threads for the ABS runnable in L sizing has been increased from 200 (<9.8) to 400 (≥9.8).

Even if you updated using **updateall** and specified the sizing, this change will not be applied automatically, as this parameter is often customized and we do not want to overwrite such customizations here.

Therefore, you need to check if you made a modification to the configure parameters **httpNumWorkerThreads** and **ajpNumWorkerThreads** of the ABS runnable. You can check the current setting of these configure parameters with the ACC command:

```
show instance <abs_instance_ID>
```

If you find the current value of these two parameters to be larger than 400, leave them as they are and skip the reconfigure command. Otherwise increase the values of these parameters using ACC:

```
reconfigure abs_l +httpNumWorkerThreads="400" +ajpNumWorkerThreads="400"
```

## 7.7 Start ARIS after the update

Start all runnables on all nodes in the proper order to verify the updated ARIS installation:

```
on all nodes startall
```

When the command completes, check that all runnables on all nodes are running before you continue:

```
on all nodes list
```

If not all runnables successfully started up, you most likely missed an essential step during the update. Typically, if instances of umcadmin\_, abs\_ or copernicus\_ go to status **FAILED** or remain stuck in **STARTING**, there is most likely a problem with the database access. Make sure that you really updated the schemas of all tenants.

## 7.8 Trigger a reindexing

! This step is

- **mandatory** when updating from a version 9.5.x or 9.6.x to 9.8.x
- **optional, but recommended**, when updating from a version 9.7.x or 9.8.x to 9.8.6

This can be done by invoking the ACC command **updatesystem** on any abs\_ runnable. The command requires username, password, and tenant ID of a user that has the privilege **Database administrator** or Server administrator or is the superuser.

```
on <nodeName> invoke updatesystem on <abs_instance_ID> tenant.name="<tenant_ID>" \
tenant.user.name="<username>" tenant.user.pwd="<password>"
```

Note that while you are asked to specify credentials of a specific user on a specific tenant, the command will perform the re-indexing of the modeling data for all tenants.

Replace **nodeName** with the name of a node on which there is an ABS instance, replace **abs\_instance\_ID** with the instance ID of that ABS instance. And replace **tenant\_ID** with the tenant name you want to use for the operation. Put the values for these three parameters in quotes to avoid any problems with special characters.

**Example:** abs\_m runs on node n1, user **system** with password **1234ArisAdmin** on the tenant **mycompany** starts the reindexing:

```
on n1 invoke updatesystem on abs_m tenant.name="mycompany" tenant.user.name="system" \
tenant.user.pwd="1234ArisAdmin"
```

! Make sure no one is using ARIS while updatesystem is running!

While **updatesystem** is running with users accessing ARIS might experience all sorts of apparent data inconsistencies. It is therefore highly recommended to not allow users to access the system during that time. The simplest way to prevent user access is to just stop the loadbalancer runnable.

Optional: Reindexing the modeling data of selected tenants only

In case you have a system with many tenants and lots of modeling data, the **updatesystem** command might take some time. As mentioned above, during that time, it is not advised to let users access the system.

To minimize outage, you can alternatively perform the reindexing selectively for specific tenants using the ACC command **maintainAll**. Once **maintainAll** has completed for a tenant, users of that tenant can start modeling right away:

```
on <nodeName> invoke maintainAll on <abs_instance_ID> tenant.name="<tenant_ID>" \
tenant.user.name="<username>" tenant.user.pwd="<password>"
```

Note that here, **tenant\_ID** not only chooses the tenant of the user that is used to execute the command, but here it also chooses which tenant is being reindexed. So, if you have different tenants with different credentials for your administrative users, such as **system** or **superuser**, you will have to specify the password of that user for the respective tenant.

Each tenant has its own system database for keeping method customizing, user written reports etc., which are **never** shared across tenants.

With each ARIS version the default content of the system database is updated. Updating the system database will provide that newer content, done via the ACC command **updatesystemdb**.

Before ARIS 9.8 **updatesystemdb** was an optional step. It will only update filters, reports and templates that are shipped with the standard product. Usually customers have their own filters and templates and do not need updates for those which are shipped with the standard product.

If you are expecting to receive a fixed standard report, macro or script by the new ARIS version, then **updatesystemdb** should be performed.

In such a case, it might also make sense to perform this command on a "test" tenant only, in order to pick certain updated report script etc. from there and import them manually to other tenants.

This example runs **updatesystemdb** for three tenants (default, tenant1, tenant2). The tenant **master** is not needed to be updated:

```
invoke updatesystemdb on <abs_instance_ID> tenant.name=default tenant.user.name=superuser tenant.user.pwd="<passwd_default>"
invoke updatesystemdb on <abs_instance_ID> tenant.name=<tenant1> tenant.user.name=superuser tenant.user.pwd="<passwd_tenant1>"
invoke updatesystemdb on <abs_instance_ID> tenant.name=<tenant2> tenant.user.name=superuser tenant.user.pwd="<passwd_tenant2>"
```

Replace **abs\_instance\_ID** with the proper ID of your ABS runnable. If you have more than one ABS instance in your installation, you only need to run **updatesystemdb** on ONE of them, but still once for each tenant whose system database you want to update. Replace **passwd \*** with the password of the **superuser** of the respective tenant.

! If you have multiple nodes, you might need to prefix the invoke command with  
• on <nodename> (or use **set current node**)

! If you do not update the systemdb for all tenants, the system will work but you will not have  
• the updates & fixes for SAG standard reports, checks etc.

## 7.10 Adding ARIS help

In newer ARIS versions, the ARIS help files are not part of the runnables but are so-called **enhancements**, e.g., files that are added to a runnable after the runnable itself has been installed/configured. An update of a runnable will delete these help enhancements, therefore the updated help files of the new version have to be added again after the update.

! On a pure Publisher installation, the online help files are part of the runnable and don't need to be enhanced.  
• The same holds true for ARCM installations up to and including version 9.8 SR1. For 9.8 SR2 or above, the ARCM help has to be added in the way described here as well.

### 7.10.1 Stopping the runnables (OPTIONAL)

Ensure that on all nodes all runnables of the following types are in the state "stopped" (if they are present on your installation):

ABS, Octopus, ADSAdmin, UMCAdmin, Copernicus, APG

You can stop all runnables with a specific instance ID over all nodes using the command

```
on all nodes stop <runnable_instance_ID_*>
```

Example:

```
on all nodes stop abs_m
```

You would stop all runnables whose instance ID is **abs\_m** on all nodes currently known to the ACC. This will obviously only work if you gave all runnables of the same type the same instance ID on each node where you originally decided to configure them.

Alternatively, you can just stop the entire installation using the command

```
on all nodes stopall
```

! It is not strictly necessary to stop the runnables to which the help will be enhanced. All help content is just static files that will  
• become available the moment the enhance command completed successfully.

### 7.10.2 Enhancing help

The basic structure of an enhance command for adding the help to a runnable of the types APG, UMCAdmin, ADSAdmin, Octopus and Copernicus is

```
on <nodeName> enhance <instance_ID> with help <helpArtifact> <version> type war
```

where the placeholders in angled brackets have the following meaning

**nodeName** is the logical name of the node, e.g. **n1**

**instance\_ID** is the instance ID of the runnable you want to add the help to, e.g. **abs\_m**

**helpArtifact** is the technical name of the actual set of help files, e.g. com.aris.documentation.architect. The technical name will be different for the help of different runnables.

**version** is the version you are updating to, e.g., 98.4.0

For ABS, some parts of the online help have been separated into dedicated files for the different languages. The language can be specified with an additional option, the "classifier" in the enhance command. So for adding help to ABS, the structure of the enhance command is

```
on <nodeName> enhance <instance_ID> with help <helpArtifact> <version> classifier <lang> type war
```

The placeholders **nodeName**, **instance\_ID**, **helpArtifact**, and **version** have the same meaning as above.

**lang** is the language of the help artifact. Currently, there are ABS help files for five different languages: **en** (English), **de** (German), **fr** (French), **es** (Spanish), **ja** (Japanese), **pt** (Portuguese)

You can simply add all languages using several enhance commands, or you can pick only those languages that you need, depending on your user base.

The name of the help artifacts of the different runnables are the following:

Runnables	Help artifacts	Languages
ABS	com.aris.documentation.architect	de, en, fr, es, ja, pt (>= 9.8.2), nl+ru+zh (>= 9.8.6)
	com.aris.documentation.architect-scripthelp (since 9.8.2)	-
	com.aris.documentation.architect-methodhelp (since 9.8.2)	independent *, de, en, fr, es, ja, pt+nl+ru+zh (>= 9.8.6)
Octopus	com.aris.documentation.connect	-
ADSAdmin	com.aris.documentation.administration.ads	-
UMCAdmin	com.aris.documentation.administration.administration	-
	com.aris.documentation.administration.tm (since 9.8 SR6)	-
Copernicus	com.aris.documentation.connect	-
APG	com.aris.documentation.processboard	-

\* The methodhelp consists of a common part (an artifact without language classifier and parts for the different languages. The common part should always be enhanced, while the language specific parts can be chosen depending on the languages required by one's user base.

### 7.10.3 Executing the help enhancement

To build the necessary enhance commands, take the basic structure of the enhance commands as given above, and put in the nodename, instance ID, artifact name, version and—in the case of ABS only—the language.

Unfortunately the command **on all nodes** cannot yet be used for enhancements. So you have to execute the help enhancements for all nodes (and in the case of ABS for all languages) separately.

Example, assuming an update to version 98.6.0, assuming that ABS and APG are installed on a node named **n1** with instance ID **abs\_m** and **apg\_m**, respectively, and assuming that the other runnables that need their help enhanced are configured on a node **n2**:

```
on n1 enhance abs_m with help com.aris.documentation.architect 98.6.0 classifier de type war
on n1 enhance abs_m with help com.aris.documentation.architect 98.6.0 classifier en type war
on n1 enhance abs_m with help com.aris.documentation.architect 98.6.0 classifier fr type war
on n1 enhance abs_m with help com.aris.documentation.architect 98.6.0 classifier es type war
on n1 enhance abs_m with help com.aris.documentation.architect 98.6.0 classifier ja type war
on n1 enhance abs_m with help com.aris.documentation.architect 98.6.0 classifier pt type war
on n1 enhance abs_m with help com.aris.documentation.architect 98.6.0 classifier ru type war
on n1 enhance abs_m with help com.aris.documentation.architect 98.6.0 classifier zh type war
on n1 enhance abs_m with help com.aris.documentation.architect 98.6.0 classifier nl type war
on n1 enhance abs_m with help com.aris.documentation.architect-scripthelp 98.6.0 type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 classifier de type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 classifier en type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 classifier es type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 classifier fr type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 classifier ja type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 classifier pt type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 classifier ru type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 classifier zh type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 98.6.0 classifier nl type war
on n2 enhance octopus_m with help com.aris.documentation.connect 98.6.0 type war
on n2 enhance adsadmin_m with help com.aris.documentation.administration.ads 98.6.0 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.administration 98.6.0 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.tm 98.6.0 type war
on n2 enhance copernicus_m with help com.aris.documentation.connect 98.6.0 type war
on n1 enhance apg_m with help com.aris.documentation.processboard 98.6.0 type war
```

### 7.11 Reset to local repository (OPTIONAL)

Since now all runnables have been updated from the remote repository, we no longer need the remote repository and it can in theory be shut down. However, if you did that and you would want to configure another instance of a runnable, this would fail (as the agent will, by default, always look first into the remote repository and only use the version of the runnable that is in its local repository if the file in the remote repository is the same). If you want to be able to configure additional runnables, you can now switch the agents again to use only the runnables already contained in their local repository.

```
on all nodes set local.repository.only=true
```

**!** Note that you can only configure additional instances of those runnable types that are already in the respective agent's local repository in the respective version. So if you have configured, e.g., an ABS runnable on a node n1, after applying the setting above, you can configure an additional ABS runnable on the same node (or configure it again if you deconfigured it). However, you will not be able to configure an additional ABS runnable on, e.g., node n2, if there hasn't been an instance of the ABS runnable on that node yet.

So it is recommended to leave the **local.repository.only** setting set to **false** if you can just leave the remote repository in place (i.e., if the server that you used will be running anyway and if disk space doesn't matter). That way, you can configure additional instances of any runnable type on any node.

Only if you really need to take down the remote repository after the update has completed, you should apply the setting above.

► **YOU ARE DONE with the update when updating ARIS 9.8.2 to 9.8.6.** When updating from prior ARIS versions continue with the procedure.

### 7.12 Workaround for elasticsearch

When updating from any version before 9.8.x to version 9.8.0, you need to execute the following command for each elasticsearch instance in your installation (i.e., to 9.8 GA version or SR1 or any hotfix based on these versions, not when updating to 9.8.2.x or later):

```
on <nodeName> reconfigure <elasticsearch_ID> plugin.apptypes=elasticsearch
```

### 7.13 Changes for LDAP since ARIS 9.8

If you use an LDAP system with ARIS and have been using SSL to access the LDAP system from ARIS, you might have to adapt your LDAP configuration slightly:

In prior versions, it was possible to use various combinations of different SSL-related settings and still have SSL working fine. To enable new features, these settings now have to be handled consistently.

In version 9.8, UMC offers user-friendly forms to maintain the following LDAP settings.

#### 7.13.1 Using SSL/TLS

When using SSL/TLS you can either specify the LDAP server URL (UMC parameter `com.aris.umdldap.url`) with scheme **ldap**, and set the SSL setting (`com.aris.umdldap.ssl`) to **true** and the SSL mode (`com.aris.umdldap.ssl.mode`) to **ssl**, like in this example:

```
com.aris.umdldap.url = ldap://myLDAPserver.domain:636
com.aris.umdldap.ssl = true
com.aris.umdldap.ssl.mode = ssl
```

Alternatively, you can use the LDAP server URL (`com.aris.umdldap.url`) with scheme **ldaps** and set SSL (`com.aris.umdldap.ssl`) to **false**, like in this example:

```
com.aris.umdldap.url = ldaps://myldap.domain:636
com.aris.umdldap.ssl = false
```

#### 7.13.2 Using STARTTLS

When using STARTTLS the only valid setting in versions since 9.8 is shown in this example:

```
com.aris.umdldap.url = ldap://myLDAPserver.domain:389
com.aris.umdldap.ssl = true
com.aris.umdldap.ssl.mode = starttls
```

► **YOU ARE DONE with the update when updating ARIS 9.6.x or 9.7.x to 9.8.6.**

## ABOUT SOFTWARE AG

Software AG offers the world's first Digital Business Platform. Recognized as a leader by the industry's top analyst firms, Software AG helps you combine existing systems on premises and in the cloud into a single platform to optimize your business and delight your customers. With Software AG, you can rapidly build and deploy digital business applications to exploit real-time market opportunities. Get maximum value from big data, make better decisions with streaming analytics, achieve more with the Internet of Things, and respond faster to shifting regulations and threats with intelligent governance, risk and compliance. The world's top brands trust Software AG to help them rapidly innovate, differentiate and win in the digital world. Learn more at [www.SoftwareAG.com](http://www.SoftwareAG.com).

© 2016 Software AG. All rights reserved. Software AG and all Software AG products are either trademarks or registered trademarks of Software AG. Other product and company names mentioned herein may be the trademarks of their respective owners.

