



# **ARIS** UPDATE COOKBOOK

**ARIS 10.0 SR5**  
July 2018

## CONTENTS

<b>1</b>	<b>Scope &amp; Motivation</b>	<b>3</b>
<b>2</b>	<b>Preconditions for an update</b>	<b>4</b>
2.1	Availability of all required credentials	4
2.2	Available disk space	4
2.3	Activating deactivated runnables (for source versions 9.8.4 and later)	4
2.4	Check vm.max_map_count	5
<b>3</b>	<b>Backup your data</b>	<b>6</b>
<b>4</b>	<b>Cleaning up the installation</b>	<b>7</b>
4.1	Archiving and then deleting all old log files (Optional)	7
4.1.1	Archiving all log files (Optional, but recommended)	8
4.1.2	Deleting all log files	8
4.2	Selectively deleting older log files only (Optional)	8
<b>5</b>	<b>Documentation and backup of all manual changes</b>	<b>9</b>
<b>6</b>	<b>Updating the agents on all nodes</b>	<b>10</b>
6.1	Making sure that all installation files can be updated	10
6.2	Updating ARIS Agents on all nodes	10
6.3	Validating the ARIS Agent update	10
<b>7</b>	<b>Prepare ARIS Cloud Controller</b>	<b>11</b>
<b>8</b>	<b>Prepare and set remote repository</b>	<b>12</b>
8.1	Provide remote repository via web server	12
8.2	Provide remote repository via local file folder	12
<b>9</b>	<b>Perform the update of all runnables on all nodes</b>	<b>13</b>
<b>10</b>	<b>Starting ARIS after the update</b>	<b>15</b>
<b>11</b>	<b>Common startup problems after the update</b>	<b>15</b>
<b>12</b>	<b>Updating the system database for each tenant</b>	<b>15</b>
<b>13</b>	<b>Updating the demo databases for each tenant</b>	<b>16</b>
<b>14</b>	<b>Updating the ARIS online help</b>	<b>17</b>
14.1	Stopping the runnables which will have their help enhanced	17
14.2	Enhancing help	17
14.3	Names of the help artifacts	18
14.4	Executing the help enhancement	19
<b>15</b>	<b>Configure the agents on all nodes to use the local repository (optional)</b>	<b>20</b>
<b>16</b>	<b>Update completed</b>	<b>20</b>

## 1 Scope & Motivation

This document describes the steps necessary to **manually** perform an update of any ARIS 10.0 installation to any 10.0 SR.

Depending on the actual version being updated from (the **source** version) or being updated to (the **target** version) some of the steps described below might not be needed or be optional. Similarly, some steps might depend on whether or not you are using an external database system with your ARIS installation, for example, Oracle or Microsoft SQL Server. In these cases, this will be indicated at the beginning of the respective section. If there is no such indication, the step is required regardless of actual source or target version.

It is important to point out that the steps described in this document are **not** required if ARIS was originally installed via setup - to update such an installation, all you need to do is run the ARIS setup of the new version to update your installation.

The following steps described in this document are only necessary if you...

- ...want to update an ARIS installation that was originally installed manually via provisioning (either on a single node or on multiple nodes). We consider an installation as **manual** if it was **not** created with one of the product-specific setups, for example, the ARIS Connect Server setup for a single-node installation, but by manually installing (empty) ARIS Agent(s) on the machine(s) and then using ARIS Cloud Controller (ACC) to install or "configure" the individual components (so called **runnables**) that comprise an ARIS system on these agent(s).
- ...want to update a setup-based ARIS installation that has been changed so that the update setup refuses to update.
- ...want to learn about what goes on under the hood of an update setup. After all, the steps described here are basically what the setup does (but of course only for simple single node, that is, not distributed installations).

## 2 Preconditions for an update

Before even attempting to update an ARIS installation, you must check if the following preconditions are met.

### 2.1 Availability of all required credentials


In general, you need the following credentials during the update.

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

### 2.2 Available disk space

For a successful update, sufficient free disk space must be available on each node that is part of the update installation. The following is a rough guideline on how to determine the disk space required:

- Determine the currently available free disk space **f** on the machine where the ARIS Agent runs.
- Determine the current size **w** of the runnable working directories located in (under Windows: <installDir>\server\bin\work, under Linux: /home/ARIS/cloudagent/work).
- Make sure that the condition **f > w + 100 GB** is fulfilled.

 If there is not enough disk space available on the system, **do not try an update!**  
The update will probably fail so that the installation cannot be repaired!

### 2.3 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 is ignored by the startup process of the runnables. 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 for an easier upgrade to ARIS Connect, they have already installed ARIS Connect Server. In that case, the unneeded runnables can be deactivated (ECP and PostgresECP).

To keep an installation consistent, however, deactivated runnables must also be updated. Hence, a deactivated runnable is **not** ignored by the update process. 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.

## 2.4 Check `vm.max_map_count`

From ARIS 10 SR4, the **elastic** runnable (elastic search) requires more virtual memory on your Linux operating system in order to prevent indices from causing out-of-memory exceptions. On the machine where the **elastic** runnable is installed, check which value the **vm.max\_map\_count** system parameter is currently set to.

For this, use the **sysctl vm.max\_map\_count** command. If the value is lower than 262144, change it to 262144.

To change the value for this session only, use the following command (root access permissions required):

```
sudo sysctl -w vm.max_map_count=262144
```

This command changes the value for the current session only. After a reboot, this change is undone.

To change the value permanently, add this line to the **sysctl.conf** configuration file that is available in the ARIS installation directory on your machine:


```
vm.max_map_count=262144
```

### 3 Backup your data

As the very first step of any update of an ARIS installation, all data have to be backed up.

There are basically three ways to do a backup:

- Creating a tenant backup for all tenants using ARIS Cloud Controller. This contains all essential data of a tenant, with the notable exception of any portal customizations and tenant-specific portal settings.
- Copying the working directories of all runnables to a safe place, or even the whole ARIS installation folder. In addition, if an external DBMS is used (Oracle or MS SQL Server) dumps of all schemas containing ARIS data are needed.
- Using the backup functionality offered by most applications via their UI (e.g., backup database in ABS, UMC's "backup data" etc.), or their command-line tools (ABS: arisadm, UMC: y-tenantmgmt). Please note that not all applications offer such functionality, but depending on your usage of ARIS, this might be all you need.

 If you run the update without having current and complete data backups, you bear the risk for any data loss or system outage!

## 4 Cleaning up the installation

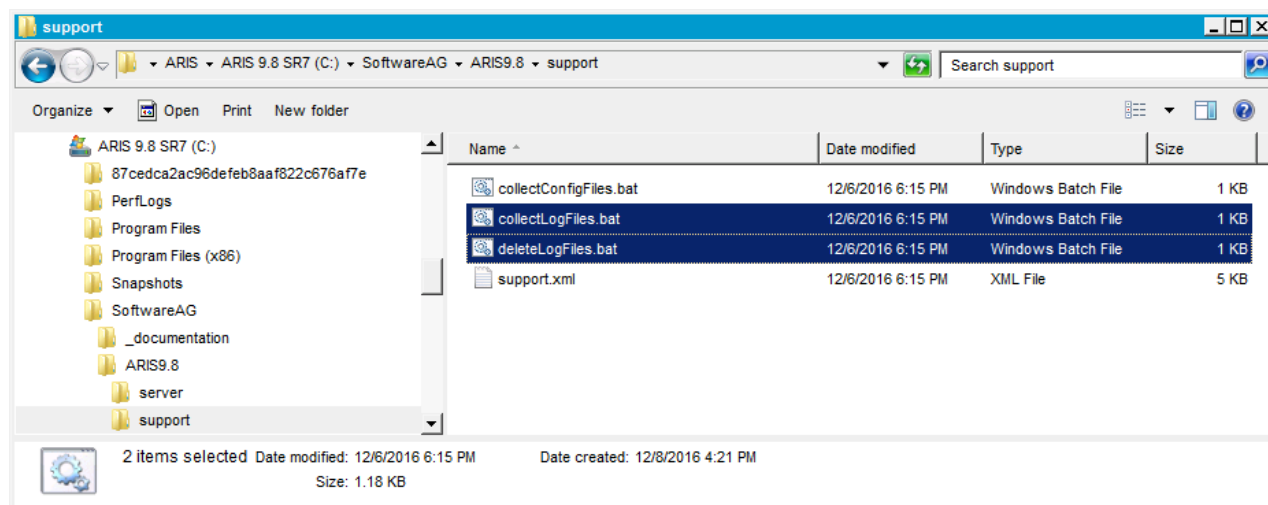
During the actual update procedure, backups of the working directories of all runnables are created in order to be able to roll back to the original version if the update fails.

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, which is often the case for virtual machines or physical machines using 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 preparation step, you can remove unnecessary files from the working directories. In particular, older versions of ARIS wrote a large number of log files that would not be cleaned automatically. This has been improved considerably in more recent versions.

### 4.1 Archiving and then deleting all old log files (Optional)

To reduce the used disk space, you can first archive and then delete all log files of the ARIS installation. Two batch files are provided for this: **collectLogFiles** and **cleanLogFiles** (**.bat** on Windows, **.sh** on Linux). On Windows systems you can find them in the **<installDir>\support** folder, as shown in the screenshot below:



On Linux systems you can find them in the **home** directory of the ARIS user that is created by the **ARIS Cloud Controller Linux** packages:

```


/home/&ARIS# ls -ls
total 36
4 -rwxr--r-- 1 aris root 321 Jun 30 19:45 21-aris.conf
4 -rwxr--r-- 1 aris root 182 Jun 30 19:45 aris
8 -rwxr--r-- 1 aris root 4406 Jun 30 19:45 ArisCloudagent
4 drwxr-xr-x 9 aris root 4096 Jul 1 10:45 cloudagent
4 -rwxr-xr-x 1 aris root 594 Jun 30 19:45 collectLogFiles.sh
4 -rwxr-xr-x 1 aris root 501 Jun 30 19:45 deleteLogFiles.sh
4 drwxr-xr-x 5 aris root 4096 Jul 1 10:32 jre
4 -rwxr--r-- 1 aris root 2461 Jun 30 19:45 support.xml
/home/&ARIS#

```

### 4.1.1 Archiving all log files (Optional, but recommended)

The **collectLogFiles** utility collects all log files of all runnables and the agent and puts them into a single **LogFiles.zip** archive. If you want to archive all the log files before cleaning them, run **collectLogFiles** first. Depending on the amount of log files accumulated in your installation, this can take quite a while and of course you need additional free space on the disk for the log file archive.

Once **collectLogFiles** is complete, copy the resulting **logfiles.zip** file to a safe location.

 If you keep the archive with the log files on the same drive/partition where ARIS is installed, make sure that there is **still** enough free disk space as described in the **Preconditions** section in this document.

### 4.1.2 Deleting all log files

After you successfully archived the log files, run the **cleanLogFiles** utility to remove them from the installation directory to free up disk space.

## 4.2 Selectively deleting older log files only (Optional)

If you want to keep the most recent of your log files, you can also manually go through the various log directories of the different runnables and selectively only delete the oldest and/or largest files. The following table shows the location of the log files for the various runnables.

 Do **not** delete any files outside the following directories!

Component name	Path to main log file/directory (relative to component's working dir, secondary log locations in parentheses)
cloud agent	<installDir>\logs\ARISCloudAgent.log (<installDir>\bin\log)
zoo	log\zookeeper.log (log\zookeeper.log.<index>)
postgres	pgworkdata\pg_log.txt (pgworkdata\pg_log\*)
postgresECP	pgworkdata\pg_log.txt (pgworkdata\pg_log\*)
couchdb	log\couch.log
cloudsearch	defaultLog\log.txt
elastic	elasticsearch\logs\<hostName>.log
adsadmin	defaultLog\ads.log (defaultLog\*, base\logs\*)
umcadmin	defaultLog\umc.log (defaultLog\*, base\logs\*)
loadbalancer	httpd\logs\error.log (httpd\logs\*)
abs	base\webapps\abs\log\* (base\logs\*)
apg	base\logs\age.log (base\logs\*)
copernicus	base\logs\catalina.<date>.log, base\logs\system.out (base\logs\*)
octopus	base\logs\catalina.<date>.log, base\logs\system.out (base\logs\*)
ecp	base\logs\catalina.<date>.log, base\logs\log.txt (base\logs\*)
simulation	base\logs\catalina.<date>.log, base\logs\log.txt (base\logs\*)
dashbording	base\logs\*.log, um\presto\data\*.log, um\plugins\log\*.log



## 5 Documentation and backup of all manual changes

 Manual changes to an installation will not survive an update.

In general, **any manual, physical modification** of an ARIS installation, in particular manual modification or addition of files, **are not preserved during an update**. All manual changes not done using ACC commands after the initial installation of ARIS must therefore be documented in an installation protocol. Only modifications made using ACC commands ("**reconfigure**", "**enhance**") are safe unless stated otherwise.

Of course, all **content** that you enter into ARIS (models, users, documents, collaboration comments etc.) via the UI or official interfaces **is** preserved during an update.

In general, you are recommended to use ACC commands (in particular **reconfigure** and **enhance**) to make configuration changes or add third party files to your installation (JDBC drivers, SAP JCo adapters, SSL certificates etc.). However certain types of modification cannot yet be performed with ACC. Common valid cases for manual modification are usually related to certain configuration files, in particular these files:

- ARIS Connect / Design Server:


```
<installDir>\server\bin\work\work_abs_<sizing>\base\webapps\abs\downloadClient\config\arisloader.cfg
```

```
<installDir>\server\bin\work\work_abs_<sizing>\base\webapps\abs\config\defaultServerSettings.cfg
```

- ARIS Publisher Server:


```
<installDir>\server\bin\work\work_businesspublisher_<sizing>\base\webapps\businesspublisher\config\webappserver.cfg
```

```
<installDir>\server\bin\work\work_businesspublisher_<sizing>\base\webapps\businesspublisher\config\umccconfig.cfg
```

 Any changes to the above files will be lost after an update!

You must repeat your documented changes to those files after performing the update. Note that some of those changes might also no longer be relevant in the new ARIS version.

Another common case of manual modification are the files of the ARIS Publisher layouts.

 Expect any files copied manually into an installation will be gone after an update.

Therefore, archive these files and document where they need to be copied back to after the update.

## 6 Updating the agents on all nodes

### 6.1 Making sure that all installation files can be updated

Especially on Windows systems, a file that is still being used by a process cannot be replaced during the update. Therefore, it is absolutely necessary to make sure that no program is still using any files or folders inside the ARIS installation directory.


Therefore, make sure that you have done the following:

- Stop and close all ACC command-line prompts.
- Exit all Windows command-line prompts that access a folder inside the ARIS installation directory.
- Close all Windows Explorers that point to the ARIS installation folder.
- Close all files inside the ARIS installation directory that are still open in external programs (for example, log files opened by a text editor).

### 6.2 Updating ARIS Agents on all nodes

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

Refer to the **ARIS Server Installation Guides** for detailed instructions on how to run the update ARIS Agent.

 The ARIS Agent setup has to stop all runnables in order to update the Java runtime environment.

### 6.3 Validating the ARIS Agent update

After performing the update, run the following ACC command to check the new ARIS version number of all agents currently registered in ACC:

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

Make sure that all lines show the same version that you wanted to update to.

## 7 Prepare ARIS Cloud Controller

Copy the **generated.apptypes.cfg** file from the ARIS Server setup folder and overwrite the existing file in the ARIS installation folder, for example, C:\SoftwareAG\ARIS10.0\server. Then start ARIS Cloud Controller (ACC) with the command-line option **-c <path to generated.apptypes.cfg>** to provide ACC with the necessary information about the runnable of the new ARIS version. Make sure that ACC is started from now on with the new **generated.apptypes.cfg** file, for example, by creating or modifying a shortcut to ARIS Cloud Controller.

Example:

```
acc.bat -c "C:\SoftwareAG\ARIS10.0\server\generated.apptypes.cfg"
```

Add all nodes of your distributed ARIS installation:

- Either use the **add node** command manually whenever you start ACC
- Or use a node file containing all **add node** commands which you pass as command-line option **-n** to ACC

### Background information for updating runnable parameters

The default settings for runnables provided in the **generated.apptypes.cfg** file may change from version to version.

However, in versions prior to 9.8, changes to the settings in the new **generated.apptypes.cfg** file (everything in the lines after **defaults**) will not be applied during an update.

As of version 9.8, only a selected subset of the default parameter changes is applied. You can identify the parameters in the **generated.apptypes.cfg** file that **are changed** during the update. They are marked with **!** or **~**. For example:

```
register app type umcadmin_s com.aris.umcadmin.y-umcadmin-run-prod 98.1.0
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 **~** are applied if the parameter is **NOT** already set for the runnable. Parameters set or changed by the customer are retained.

Parameters marked with **!** are applied regardless of whether the parameter is already set or not.

**Parameters that the customer has set or changed are overwritten.**


## 8 Prepare and set remote repository

On all nodes, each ARIS Agent must be set so that it does **NOT** use the local repository. Issue this ACC command:

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

### 8.1 Provide remote repository via web server

Copy the Setup\_Data folder from the ARIS DVD to 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

On all nodes, point each ARIS Agent to the prepared web server that holds the repository containing the runnables of the new ARIS version. The following command points to a web server:


```
on all nodes set \
remote.repository.url="http://<your_webserver>/<full_path_to_Setup_Data_folder>"
```

For example, if your web server's hostname is **intranet**, is serving http requests on port 8080 (instead of the default port 80), and the contents of the **Setup\_data** folder have been copied into the folder **arisrepo/10.0.5**, you could use this command:

```
on all nodes set remote.repository.url=http://intranet:8080/arisrepo/10.0.5
```

### 8.2 Provide remote repository via local file folder

If you do not have a web or FTP server available in your environment, and do not want to install one, you can copy the **Setup\_Data** folder to a local (!) folder on all nodes and point to that location.

 Note that each location must be a physical drive because mapped network drives do not function as repositories, as the agent runs in the security context system that can only access physical drives.

Copy the **Setup\_Data** directory from the ARIS DVD to a physical drive on each node. Use the same folder on every node to simplify the procedure, as this allows you to use the **on all nodes** command.

On all nodes, specify that each ARIS Agent uses a local folder by using a file URL in the **remote.repository.url** ACC command set.

To use a local folder using a file URL in the ACC command **set remote.repository.url**:

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

Replace **<physicalPath>** with the path to the folder that contains the **Setup\_data**. If you copy the physical path from your Windows Explorer, make sure to replace the backslashes with forward slashes. The URL must only contain forward slashes.

For example, if you copied the **Setup\_data** folder to **D:\ARIS\10.0.5**, set the remote repository like this:

```
on all nodes set remote.repository.url=file:///d:/ARIS/10.0.5
```

## 9 Perform the update of all runnables on all nodes

To update all runnables on all nodes, run this ACC command:

```
on all nodes updateall
```

This creates a backup of all working folders before any change is made. The backups are stored in folder **<ARIS Installation path>\server\bin\backup**.

**i** These backups provide the option of a rollback to the former ARIS version in case the update fails, instead of having a mixture of runnables from the previous and new version that will most likely not work. The backup of the working folders takes some time and consumes a significant amount of disk space. These backups cannot be used as a backup of your application data. Always backup application data before updating!

Note that ARIS Agent updates are **NOT** rolled back, but this is normal and should not be a problem, as the new ARIS Agent can work with runnables of older ARIS versions.

**w** You can turn the backup option off, but then a rollback is no longer possible.

If you attempt an update without backing up your data, you run the risk of system outages and/or data loss, for which Software AG cannot assume responsibility.

**w** The updateall command first checks whether enough free disc space is available. If not, a message like this will appear.

On node localhost 20,93 GB of usable space is available for back up, downloaded artifacts, working directory, plugins and unpacked runnables, but 72,67 GB must be available.

Please make sure that sufficient free disc space is available.

**i** When the above ACC command was successful, the output looks similar to the following example of a three-node installation. Of course, versions, node names, runnable instance IDs, etc. can vary depending on the environment:

```
ACC+ >on all nodes updateall
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 10.0.5
Runnable cloudsearch_m on node n1 will be updated to version 10.0.5
Runnable umcadmin_m on node n1 will be updated to version 10.0.5
Runnable abs_m on node n1 will be updated to version 10.0.5
Runnable apg_m on node n1 will be updated to version 10.0.5
Runnable copernicus_m on node n1 will be updated to version 10.0.5
Runnable zoo0 on node n2 will be updated to version 10.0.5
Runnable elastic_m on node n2 will be updated to version 10.0.5
Runnable couchdb_m on node n2 will be updated to version 10.0.5
Runnable adsadmin_m on node n2 will be updated to version 10.0.5
Runnable umcadmin_m on node n2 will be updated to version 10.0.5
Runnable octopus_m on node n2 will be updated to version 10.0.5
Runnable simulation_m on node n2 will be updated to version 10.0.5
Runnable loadbalancer_m on node n2 will be updated to version 10.0.5
Runnable zoo0 on node n3 will be updated to version 10.0.5
```

```
Runnable postgresECP_m on node n3 will be updated to version 10.0.5
Runnable cloudsearch_m on node n3 will be updated to version 10.0.5
Runnable couchdb_m on node n3 will be updated to version 10.0.5
Runnable adsadmin_m on node n3 will be updated to version 10.0.510.0.5
Runnable abs_m on node n3 will be updated to version 10.0.5
Runnable copernicus_m on node n3 will be updated to version 10.0.5
Runnable ecp_m on node n3 will be updated to version 10.0.5
Runnable loadbalancer_m on node n3 will be updated to version 10.0.5
Successfully updated runnable zoo0 on node n2 from version 10.0.0.2 to version
10.0.5
Successfully updated runnable elastic_m on node n2 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable couchdb_m on node n2 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable adsadmin_m on node n2 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable umcadmin_m on node n2 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable octopus_m on node n2 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable simulation_m on node n2 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable loadbalancer_m on node n2 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable zoo0 on node n1 from version 10.0.0.2 to version
10.0.5
Successfully updated runnable cloudsearch_m on node n1 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable umcadmin_m on node n1 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable abs_m on node n1 from version 10.0.0.2 to version
10.0.5
Successfully updated runnable apg_m on node n1 from version 10.0.0.2 to version
10.0.5
Successfully updated runnable copernicus_m on node n1 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable zoo0 on node n3 from version 10.0.0.2 to version
10.0.5
Successfully updated runnable postgresECP_m on node n3 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable cloudsearch_m on node n3 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable couchdb_m on node n3 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable adsadmin_m on node n3 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable abs_m on node n3 from version 10.0.0.2 to version
10.0.5
Successfully updated runnable copernicus_m on node n3 from version 10.0.0.2 to
version 10.0.5
Successfully updated runnable ecp_m on node n3 from version 10.0.0.2 to version
10.0.5
Successfully updated runnable loadbalancer_m on node n3 from version 10.0.0.2 to
version 10.0.5
23 of 23 runnables updated.
```

## 10 Starting ARIS after the update

For the next steps, the system needs to be up and running. Start ARIS on all nodes with this ACC command:

```
on all nodes startall
```

Wait for the command to complete to ensure all runnables have started properly on all nodes. Check this with this ACC command:

```
on all nodes list
```

This should show all runnables in status **STARTED**.

## 11 Common startup problems after the update

If any node does not start up properly, you may have missed an essential step during the update. Known problems are failed database accesses, indicated by umcadmin instances with the status **FAILED**, and/or **abs** and **copernicus** instances stuck in **STARTED**.

## 12 Updating the system database for each tenant

### What is the **system database**?

The system database contains the ARIS method (all model/object/connection/symbol types, filters, macros, reports, and similar content) that is not directly coupled to a specific ARIS user database.

In a new ARIS installation, the system database contains only predefined ARIS content provided by Software AG. ARIS users can customize the system database and add new content to it, for example, by defining custom method extensions, writing their own reports, etc.

Each tenant in your system has its own system database. And each ARIS tenant has its own method customizing, additional reports, etc. which are not shared with other tenants.

Since the default content of the system database shipped together with ARIS is also often updated as part of a new ARIS version, the new version of the system database can be applied to an existing installation during an update. This is what the **updatesystemdb** step described here will do.

Updating the system database was an optional step before ARIS 9.8. It only updated filters, reports, and templates shipped with the standard product. Usually, customers have their own filters and templates and do not need updates for those that ship with the standard product.

If a customer has an issue with any of the standard reports, macros or scripts, or standard workflows of ARIS Process Governance, the system database should be updated. In that case, it might also make sense to execute this command only for a “test” tenant and select individual report script(s) from there for distribution to other tenants.

Run the ACC command **updatesystemdb** for all your tenants (except for the tenant **master**) beginning with the tenant **default**:

```
invoke updatesystemdb on <abs_instanceId> tenant.name=default \  
tenant.user.name=superuser tenant.user.pwd="<superuser password in default>"
```

```
invoke updatesystemdb on <abs_instanceId> tenant.name=<tenant1> \  
tenant.user.name=superuser tenant.user.pwd="<superuser password in tenant1>"
```

```
invoke updatesystemdb on <abs_instanceId> tenant.name=<tenant2> \  
tenant.user.name=superuser tenant.user.pwd="<superuser password in tenant2>"
```

If you have multiple nodes, you might need to prefix the invoke command with **on <nodename>**.

Alternatively use **set current node** to switch to those nodes with ABS runnables.

If you have more than one ABS instance in your installation, you only need to update the system database once per tenant on each instance.

If you do not update the system database for all tenants, the system will work but you will not have the updates and fixes for SAG standard reports, checks, etc.

## 13 Updating the demo databases for each tenant

Demo databases are provided on the ARIS DVD with predefined contents for demonstration purposes. With each new ARIS version, those databases come with updates and corrections which are required to properly demonstrate new ARIS functionalities.

The demo databases can be found in the "<ARIS DVD>\Content\Databases" folder:

- ARIS Connect Governance Models.adb
- Governance Automation Models.adb
- Performance.Ready - light.adb
- United Motor Group.adb

In new ARIS installations, you can rely on the auto-import mechanism to import the latest "United Motor Group" demo database. However, the existing demo database is not overwritten when ARIS is updated. This database must therefore be updated manually.

If you are not sure whether users have made changes to any of the demo databases in the ARIS installation, rename the old databases and import the new demo databases.



## 14 Updating the ARIS online help

In newer ARIS versions, the **online help** files are not part of the runnables, but need to be updated via so-called **enhancements** which add files to a runnable. But updating a runnable removes such enhancements. Hence, the help files of the new ARIS version must be added again after the update.

**i** The online help files for ARIS Publisher Server are part of the runnable and do not have to be enhanced.

The same applies to ARIS Risk and Compliance Manager up to and including version 9.8 SR1. Since version 9.8 SR2 the ARCM online help must be added as with ARIS.

### 14.1 Stopping the runnables which will have their help enhanced

Ensure that on all nodes all runnables of the following types are in the state **STOPPED** (if they are available on your system):

ABS, Octopus, ADSAdmin, UMCAdmin, Copernicus, APG

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

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

Example for stopping all ABS runnables on all nodes:

```
on all nodes stop abs_m
```

Obviously, the above command only works if all ABS runnables have identical names on all nodes. Alternatively, you can also stop ARIS completely:

```
on all nodes stopall
```

### 14.2 Enhancing help

The basic structure of an enhancement command for adding the help to runnables of types **apg**, **umcadmin**, **adsadmin**, **octopus**, and **copernicus** is as follows:

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

where:

**<nodeName>** is the logical name of the node, for example, n1

**<instanceId>** is the instance ID of the runnable you want to add the help to, for example, "octopus\_m"

**<helpArtifact>** is the technical name of the actual set of help files, for example, **com.aris.documentation.architect**.

The technical name can be different for the help of different runnables.

**<version>** is the version you are updating to, for example, 10.0.5.

For the **abs** runnable, some parts of the online help have been separated into dedicated files for different languages. The language can be specified with an additional option, the **<classifier>** in the enhance command. For adding help to abs, the structure of the enhancement command is therefore:

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

The placeholders **<nodeName>**, **<instanceId>**, **<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 available: en (English), de (German), fr (French), es (Spanish), and ja (Japanese).

You can easily add all languages using multiple enhancement commands or just select the languages you need depending on your user base.

### 14.3 Names of the help artifacts

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

Runnable	Help artifacts (since version)	Languages
abs	com.aris.documentation.architect	de, en, fr, es, ja, pt, ru, zh, nl
	com.aris.documentation.architect-scripthelp	-
	com.aris.documentation.architect-methodhelp	<sup>-1</sup> , de, en, fr, es, ja pt, ru, zh, nl
adsadmin	com.aris.documentation.administration.ads	-
umcadmin	com.aris.documentation.administration.administration	-
	com.aris.documentation.administration.tm	-
copernicus	com.aris.documentation.connect	-
apg	com.aris.documentation.processboard	-
businesspublisher	com.aris.documentation.publisher.publisher	-
arcm	com.aris.documentation.arcm.arcm	-

<sup>1</sup> 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.

## 14.4 Executing the help enhancement

For adding all necessary enhance commands, take the basic structure of the enhance commands shown above and replace the parameters **<nodename>**, **<instance ID>**, **<artifact name>**, **<version>**, and for ABS only the parameter **<lang>** accordingly.

The **on all nodes** command cannot yet be used for enhancements. Therefore, you must execute the help enhancements for all nodes separately.

**Example 1:** Assume that you want to update to version 10.0.5 and that ABS and APG are installed on node **n1** with instance IDs **abs\_m** and **apg\_m**, while the other runnables for which you want to enhance help are configured on node **n2**:

```
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier de type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier en type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier fr type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier es type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier ja type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier pt type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier ru type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier zh type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier nl type war
on n1 enhance apg_m with help com.aris.documentation.processboardhelp 10.0.5 type war
on n2 enhance adsadmin_m with help com.aris.documentation.administration.ads 10.0.5 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.administration 10.0.5 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.tm 10.0.5 type war
on n2 enhance copernicus_m with help com.aris.documentation.connect 10.0.5 type war
```


**Example 2:** The assumptions as in example 1 apply, supplemented by the additional help files for ABS (script and method help) and by the languages **pt**, **ru**, **zh**, and **nl** and UMC (tenant-management help):

```
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier de type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier en type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier fr type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier es type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier ja type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier pt type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier ru type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier zh type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.5 classifier nl type war
on n1 enhance abs_m with help com.aris.documentation.architect-scripthelp 10.0.5 type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 classifier de type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 classifier en type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 classifier fr type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 classifier es type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 classifier ja type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 classifier pt type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 classifier ru type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 classifier zh type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.5 classifier nl type war
on n2 enhance adsadmin_m with help com.aris.documentation.administration.ads 10.0.5 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.administration 10.0.5 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.tm 10.0.5 type war
on n2 enhance copernicus_m with help com.aris.documentation.connect 10.0.5 type war
on n2 enhance apg_m with help com.aris.documentation.processboardhelp 10.0.5 type war
```

## 15 Configure the agents on all nodes to use the local repository (optional)

Since all runnables have been updated at this stage, the remote repository is no longer needed and it can be shut down. However, if you want to configure another instance of a runnable later, you must keep the remote repository. Otherwise, configuring a new runnable will fail, because the agent first checks the remote repository for the most recent runnable version, and uses its local repository only if the file in the remote repository is of the same version. If you want to be able to configure additional runnables, you can now switch the agents to again only use the runnables already contained in their local repository using this ACC command:

```
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.

For example, if you have configured, 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 cannot configure an additional ABS runnable that runs on node **n2**, for example, if no instance of the ABS runnable exists on this node.

It is therefore recommended that you leave the **local.repository.only** parameter set to **false** if you can simply leave the remote repository in place, that is, if the server you are using is running and the disk space is irrelevant. In this way, you can configure additional instances of any runnable type on any node. Only if you really need to shut down the remote repository after the update has completed should you use the above setting.

## 16 Update completed

If no problems have occurred up to this point, you have successfully updated your installation. You can use the application, find all your data as before and use all functions and bugfixes of the new version.

---

### ABOUT SOFTWARE AG

The digital transformation is changing enterprise IT landscapes from inflexible application silos to modern software platform-driven IT architectures which deliver the openness, speed and agility needed to enable the digital real-time enterprise. Software AG offers the first end-to-end Digital Business Platform, based on open standards, with integration, process management, in-memory data, adaptive application development, real-time analytics and enterprise architecture management as core building blocks. The modular platform allows users to develop the next generation of application systems to build their digital future, today. With over 45 years of customer-centric innovation, Software AG is ranked as a leader in many innovative and digital technology categories. Learn more at [www.SoftwareAG.com](http://www.SoftwareAG.com).

© 2018 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