



ARIS UPDATE COOKBOOK

ARIS 10.0 SR3
December 2017

Unchanged content since ARIS 10.0 SR2—October 2017 | It applies to version 10.0 SR3 without changes.

TABLE OF CONTENTS

1	Scope & Motivation	3
2	Preconditions for an update	4
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
3	Backup your data	5
4	Cleaning up the installation	6
4.1	Archiving and then deleting all old log files (Optional)	6
4.1.1	Archiving all log files (Optional, but recommended)	7
4.1.2	Deleting all log files	7
4.2	Selectively deleting older log files only (Optional)	7
5	Documentation and backup of all manual changes	8
6	Updating the agents on all nodes	9
6.1	Making sure that all installation files can be updated	9
6.2	Updating ARIS Agents on all nodes	9
6.3	Validating the ARIS Agent update	9
7	Prepare ARIS Cloud Controller	10
8	Prepare and set remote repository	11
8.1	Provide remote repository via web server	11
8.2	Provide remote repository via local file folder	11
9	Perform the update of all runnables on all nodes	12
10	Starting ARIS after the update	14
11	Common startup problems after the update	14
12	Updating the system database for each tenant	14
13	Enhance SAP Connectivity	15
14	Updating the ARIS online help	16
14.1	Stopping the runnables which will have their help enhanced (Optional)	16
14.2	Enhancing help	16
14.3	Names of the help artifacts	17
14.4	Executing the help enhancement	18
15	Change the agents back to local repository (optional)	19

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 (e.g., Oracle or MS SQL Server) with your ARIS installation. 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* needed if ARIS was originally installed by setup - to update such an installation, all you need to do is running 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 to be "manual", if it was created *not* by using one of the product-specific setups (e.g., 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 since been modified in a way that the update setup will refuse to update it.
- ...like to learn about what is going on under the hood of an update setup. After all, the steps described here are basically what the setup does (but naturally only for simple single-node, i.e., non-distributed installations).

2 Preconditions for an update

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

2.1 Availability of all required credentials


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

- The ARIS Agent username and password (default credentials: Clous/g3h31m) on each node that is part of the installation that should 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 that will be needed:

- Determine the currently available free disk space **f** on the machine where ARIS Agent runs
- Determine the current size **w** of the runnable working directories, which is located at (on Windows: <installDir>\server\bin\work, on 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 attempt an update!**
The update is likely to fail in such a way 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 will be 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 installed already ARIS Connect Server. In that case, the unneeded runnables could be deactivate (ECP and PostgresECP).


However, for keeping an installation consistent, also deactivated runnables have to be updated. Hence, a deactivated runnable is **not** ignored by the update process. Though, 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.

3 Backup your data

As the very first step of any update of an ARIS installation, a backup of all data has to be performed.

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 is 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 a 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 have to bear the risk for any data loss or system outage!

4 Cleaning up the installation

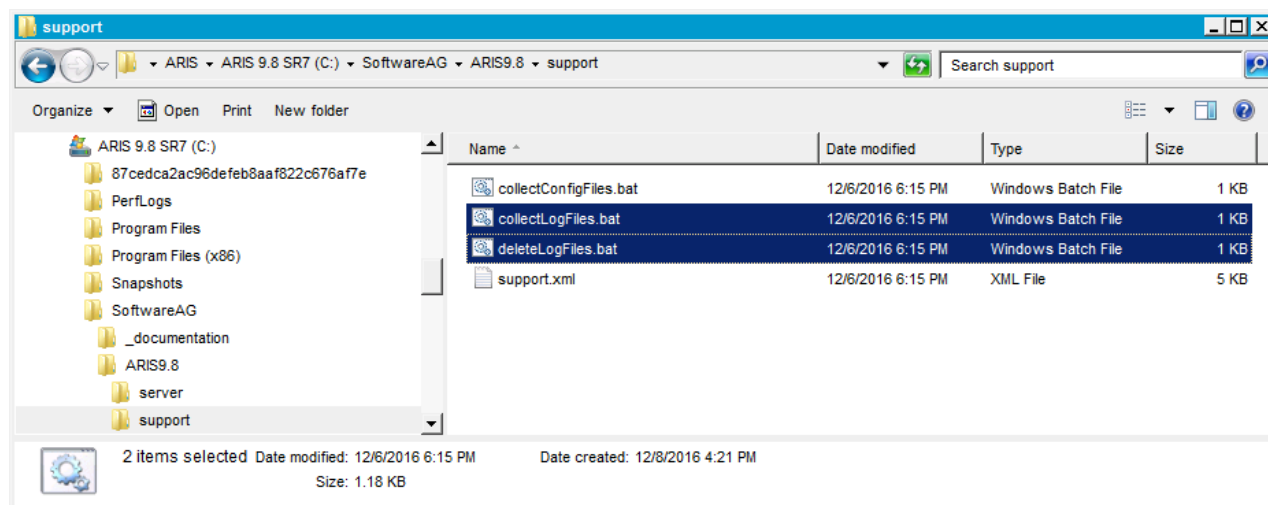
During the actual update procedure, backups of the working directories of all runnables are created in order to be able to roll back to the original version if the update fails.

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 were writing a large amount of log files which would not be cleaned automatically which has been improved considerably in more recent versions.

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

For reducing the occupied disk space you could first archive and then delete all log files of the ARIS installation. For that, two batch files are provided: "collectLogFiles" and "cleanLogFiles" (.bat on Windows, .sh on Linux). On Windows systems you can find them in the folder <installDir>\support, 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 archive "LogFiles.zip". If you want to archive all the log files before cleaning them, you should 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 has completed, copy the resulting "logfiles.zip" file to a safe place.

 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 different 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 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, those ones not done using ACC commands, that you did after your initial installation of ARIS must therefore be documented in an installation protocol. Only modifications made using ACC commands ("reconfigure", "enhance") can be deemed safe unless stated otherwise.

Naturally, any *content* that you put into ARIS (models, users, documents, collaboration comments etc.) using the UI or official interfaces *will* be preserved during an update.

Common (valid) examples of manual modifications made to an installation that will not be preserved during an update.

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 (JDBC drivers, SAP JCo adapters, SSL certificates etc.). However, for certain kinds of adaptations, it is not yet possible to perform them with ACC. Common valid cases for manual modifications 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\umcconfig.cfg
```

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

You must repeat your documented changes to those files after performing the update. Some of the changes you might have made might also no longer be relevant in the new ARIS version.

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

 Assume 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

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 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 into the ARIS installation folder
- Close all files inside the ARIS installation directory that are still open by external programs (e.g., 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 and Administration Guide" for detailed instructions on how to do the update.

i The ARIS Agent setup has to stop all runnables in order to be able 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 file generated.apptypes.cfg from the ARIS Server setup folder and overwrite the existing file in the ARIS installation folder, e.g. C:\SoftwareAG\ARIS10.0\server. Then, start ARIS Cloud Controller (ACC) with to the command-line option **-c <path to generated.apptypes.cfg>** in order 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, e.g. by creating or modifying a shortcut of 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 generated.apptypes.cfg may 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 "~". An example is explained below:

```
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 "~" will be applied 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 "!" will be applied 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**).

8 Prepare and set remote repository

On all nodes each ARIS Agent must be set NOT to use the local repository via 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 the ARIS DVD to a web or FTP server.

i 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

Point on all nodes each ARIS Agent to the repository containing the runnables of the new ARIS version to the prepared web server. This 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/10.0.1, you could use this command:

```
on all nodes set remote.repository.url="http://intranet:8080/arisrepo/10.0.1"
```

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 could copy the Setup_Data folder to a local (!) folder on all nodes and point to that location.

! Note that it really has to be physical drive on each node, because mapped network drives won't work as repository as the agent runs in the security context "system" which can only access physical drives.

Copy the Setup_Data directory from the 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.

Specify on all nodes each ARIS Agent to use a local folder by 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 of the folder the Setup_data folder is located. If you are on Windows, replace the backslashes in your path with forward slashes.

For example, if you copied the Setup_data folder to D:\ARIS\10.0.1, set the remote repository like this:

```
on all nodes set remote.repository.url=file:///d:/ARIS/10.0.1
```

9 Perform the update of all runnables on all nodes

For updating all runnables on all nodes run this ACC command:

```
on all nodes updateall
```

This will first create 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 with 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.

Note that the agent updates will NOT be rolled back, but this normal and shouldn't be a problem as the new ARIS Agent is able to work with runnables of older ARIS versions.

! The backup option could be turned off, but then a rollback won't be possible anymore. That backup data is not meant to work as a backup of your application data which you should always perform before updating!

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

i When the above ACC command was successful, the output should look similar to the following example of a three-nodes installation, of course may versions, node names, runnable instance IDs etc. vary depending on your 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 10.0.1
Runnable cloudsearch_m on node n1 will be updated to version 10.0.1
Runnable umcadmin_m on node n1 will be updated to version 10.0.1
Runnable abs_m on node n1 will be updated to version 10.0.1
Runnable apg_m on node n1 will be updated to version 10.0.1
Runnable copernicus_m on node n1 will be updated to version 10.0.1
Runnable zoo0 on node n2 will be updated to version 10.0.1
Runnable elastic_m on node n2 will be updated to version 10.0.1
Runnable couchdb_m on node n2 will be updated to version 10.0.1
Runnable adsadmin_m on node n2 will be updated to version 10.0.1
Runnable umcadmin_m on node n2 will be updated to version 10.0.1
Runnable octopus_m on node n2 will be updated to version 10.0.1
Runnable simulation_m on node n2 will be updated to version 10.0.1
Runnable loadbalancer_m on node n2 will be updated to version 10.0.1
Runnable zoo0 on node n3 will be updated to version 10.0.1
Runnable postgresECP_m on node n3 will be updated to version 10.0.1
Runnable cloudsearch_m on node n3 will be updated to version 10.0.1
Runnable couchdb_m on node n3 will be updated to version 10.0.1
Runnable adsadmin_m on node n3 will be updated to version 10.0.1
Runnable abs_m on node n3 will be updated to version 10.0.1
Runnable copernicus_m on node n3 will be updated to version 10.0.1
Runnable ecp_m on node n3 will be updated to version 10.0.1
Runnable loadbalancer_m on node n3 will be updated to version 10.0.1
Successfully updated runnable zoo0 on node n2 from version 10.0.0.2 to version
10.0.1
```

```
Successfully updated runnable elastic_m on node n2 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable couchdb_m on node n2 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable adsadmin_m on node n2 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable umcadmin_m on node n2 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable octopus_m on node n2 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable simulation_m on node n2 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable loadbalancer_m on node n2 from version 10.0.0.2
to version 10.0.1
Successfully updated runnable zoo0 on node n1 from version 10.0.0.2 to version
10.0.1
Successfully updated runnable cloudsearch_m on node n1 from version 10.0.0.2
to version 10.0.1
Successfully updated runnable umcadmin_m on node n1 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable abs_m on node n1 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable apg_m on node n1 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable copernicus_m on node n1 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable zoo0 on node n3 from version 10.0.0.2 to version
10.0.1
Successfully updated runnable postgresECP_m on node n3 from version 10.0.0.2
to version 10.0.1
Successfully updated runnable cloudsearch_m on node n3 from version 10.0.0.2
to version 10.0.1
Successfully updated runnable couchdb_m on node n3 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable adsadmin_m on node n3 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable abs_m on node n3 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable copernicus_m on node n3 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable ecp_m on node n3 from version 10.0.0.2 to
version 10.0.1
Successfully updated runnable loadbalancer_m on node n3 from version 10.0.0.2
to version 10.0.1
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 a properly start of all runnables on all nodes. Check that with this ACC command:

```
on all nodes list
```

The startup should end with all runnables in status STARTED.

11 Common startup problems after the update

If any node doesn't start up properly, you most likely missed an essential step during the update. Known problems are failed database access indicated by umcadmin instances in status FAILED and/or abs and copernicus instances stuck in STARTED.

12 Updating the system database for each tenant

i 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 fresh 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, e.g. 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 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 a customer has an issue with any of the standard reports, macros or scripts, or standard workflows of ARIS Process Governance, then the system database should be updated. In that case, it might also make sense to only perform this command or do it on one "test" tenant only and pick single report script(s) from there to be distributed 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.

Replace <abs_instanceId> with the proper ID of your ABS runnable. If you have more than one ABS instance in your installation, you only need to update the system database on any of them, though, still once for each tenant.

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

13 Enhance SAP Connectivity

The enhancement for using the ARIS SAP connectivity via SAP JCo needs to be done again after the update. Example:

```
enhance abs_<s,m,l> with webappsClasspath local file "C:/sapjco30P_11-10005326.zip"  
stop abs_<s,m,l>  
  
start abs_<s,m,l>
```

The enhancement has to be done at each ABS runnable on all nodes.

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 added files to a runnable. But updating a runnable is removing such enhancements. Hence, the help files of the new ARIS version have to be added again after the update.

i The online help files for ARIS Publisher Server are part of the runnable and don't have to be enhanced. The same holds true for and ARIS Risk and Compliance Manager up to and including version 9.8 SR1. Since 9.8 SR2 the ARCM online help has to be added in the way as for ARIS below.

14.1 Stopping the runnables which will have their help enhanced (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 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 could just stop ARIS completely:

```
on all nodes stopall
```

14.2 Enhancing help

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

```
on <nodeName> enhance <instanceId> 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

<instanceId> 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., 10.0.1.

For ABS 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. So for adding help to ABS, the structure of the enhance command is

```
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. And <lang> is the language of the help artifact.

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.

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	¹ , 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	-
Business Publisher	com.aris.documentation.publisher.publisher	-
ARCM	com.aris.documentation.arcm.arcm	-

¹ 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 command "on all nodes" can't be used for enhancements yet. So you have to execute the help enhancements for all nodes separately.

Example 1: Assuming an update to version 10.0.1, and that ABS and APG are installed on a node named n1 with instance ID abs_m and apg_m, 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 10.0.1 classifier de type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier en type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier fr type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier es type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier ja type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier pt type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier ru type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier zh type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier nl type war
on n1 enhance apg_m with help com.aris.documentation.processboardhelp 10.0.1 type war
on n2 enhance adsadmin_m with help com.aris.documentation.administration.ads 10.0.1 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.administration 10.0.1 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.tm 10.0.1 type war
on n2 enhance copernicus_m with help com.aris.documentation.connect 10.0.1 type war
```


Example 2: The same assumptions as before, but here with the additional help files for ABS (script and method help) plus additional languages (pt, ru, zh, and nl) and UMC (tenant-management help)

```
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier de type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier en type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier fr type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier es type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier ja type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier pt type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier ru type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier zh type war
on n1 enhance abs_m with help com.aris.documentation.architect 10.0.1 classifier nl type war
on n1 enhance abs_m with help com.aris.documentation.architect-scripthelp 10.0.1 type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 classifier de type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 classifier en type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 classifier fr type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 classifier es type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 classifier ja type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 classifier pt type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 classifier ru type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 classifier zh type war
on n1 enhance abs_m with help com.aris.documentation.architect-methodhelp 10.0.1 classifier nl type war
on n2 enhance adsadmin_m with help com.aris.documentation.administration.ads 10.0.1 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.administration 10.0.1 type war
on n2 enhance umcadmin_m with help com.aris.documentation.administration.tm 10.0.1 type war
on n2 enhance copernicus_m with help com.aris.documentation.connect 10.0.1 type war
on n2 enhance apg_m with help com.aris.documentation.processboardhelp 10.0.1 type war
```

15 Change the agents back to local repository (optional)

Since all runnables have been updated at this stage, the remote repository is no longer needed and it could be shut down. However, if you want to configure another instance of a runnable later, you need to keep the remote repository, otherwise, configuring a new runnable would fail, because the agent will first check the remote repository for the most recent runnable version, and use only its local repository 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 via 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.

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.

Therefore 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.

If you did not encounter any issues up to this point, you have successfully updated your installation. You should be able to use the application, find all your data as it was before, and be able to enjoy any features 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.

© 2017 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

