

Adabas for Linux, UNIX and Windows

Adabas High Availability

Version 7.0.1

October 2022

This document applies to Adabas for Linux, UNIX and Windows Version 7.0.1 and all subsequent releases.

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

Copyright © 1987-2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: ADAOS-HAM-701-20220622

Table of Contents

Preface	v
1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 Introduction to ADAHAM	5
3 Using ADAHAM	7
Initializing ADAHAM	8
Requesting Access	8
Notify Access	8
Synchronizing Loop	8
Validating Status	9
Releasing Access	10

Preface

This document provides information about the Adabas High Availability tool ADAHAM.

The Adabas High Availability document is organized as follows:

<i>Introduction to ADAHAM</i>	Provides a general description of Adabas High Availability, the environment and the components.
<i>Using ADAHAM</i>	Provides a description of how to use the functionality of ADAHAM.

1 About this Documentation

- Document Conventions 2
- Online Information and Support 2
- Data Protection 3

Document Conventions

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

Online Information and Support

Product Documentation

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

In addition, you can also access the cloud product documentation via <https://www.software-ag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://tech-community.softwareag.com>. From here you can, for example:

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

Product Support

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

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

Data Protection

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

2 Introduction to ADAHAM

The Adabas High Availability Tool ADAHAM is designed to prevent access conflicts to Adabas containers. On each host where it is installed, ADAHAM is used to synchronize concurrent access to the Adabas containers, and then it starts the Adabas nucleus. The synchronization is performed with a file called *ADAHA.{DBID}*, which is located in the same directory as the database configuration. The contents of the file are continuously refreshed by an additional ADAHAM process that works on running Adabas nucleus instance. A time stamp and all relevant information about the running nucleus are written to the file *ADAHA.{DBID}*. If the host is down, the time stamp is used to notify other instances of ADAHAM that might be waiting.

ADAHAM performs the following steps:

- Checks whether the current local nucleus is in a consistent and healthy state;
- Corrects, shuts down and/or cleans up the Adabas nucleus inter-process resources if the nucleus is not working correctly;
- Checks whether another Adabas nucleus/ADAHAM pair is running on another node.

ADAHAM is *not* a daemon; it must be started in a loop so that it can check the status of the Adabas nucleus and refresh the information in the file *ADAHA.{DBID}*.

ADAHAM and the associated scripts are designed to run in any distributed environment, such as a cloud or Docker environment. The script *distributedCheck.sh* provides the entry point to the functionality of ADAHAM. Initially, the scripts are created for Adabas in a Docker environment. The corresponding handling is used in the Docker scripts that are provided in each Adabas installation in the directory *SAG directory/Adabas/docker*.



Note: ADAHAM is supported only on Linux.

3 Using ADAHAM

- Initializing ADAHAM 8
- Requesting Access 8
- Notify Access 8
- Synchronizing Loop 8
- Validating Status 9
- Releasing Access 10

Initializing ADAHAM

In order to perform failsafe tests on the Adabas nucleus, ADAHAM sends a special Adabas call that tests all of the relevant stages within the nucleus. This Adabas call needs an Adabas file that contains test records. To create the FDT for this file, the environment variable ADAHAM must be set to the Adabas file number, and the following initialization command must be issued:

```
distributedCheck.sh initialize
```

Requesting Access

In order to obtain exclusive access to the Adabas containers, the virtual instance of ADAHAM first needs to get information about their state, and then request access. This process includes unique tests, and locking the containers for registration. Access is requested with the following command:

```
distributedCheck.sh state
```

If another instance is already using the Adabas containers, the command will loop and wait until the other Adabas nucleus is no longer active. This call blocks and waits until either access is granted, or an error status is returned.

Notify Access

In the initial stage, the node attempts to get offline access to ADAHAM. The node wants to access ADAHAM, but the nucleus has not yet been started. Use the following command:

```
distributedCheck.sh notify
```

Synchronizing Loop

While the Adabas nucleus is active, the current state of the nucleus needs to be tracked asynchronously in a loop. Information about the current state is written to the file *ADAHA.{DBID}*. The corresponding script will return when the nucleus goes offline again (either as a result of being shut down correctly, or because of a nucleus crash). Use the following command to start the synchronization:

```
distributedCheck.sh sync
```

If the local Adabas nucleus is active, this command will update the current status of the nucleus. If the nucleus has failed or is inactive, the corresponding record in the file *ADAHA.{DBID}* will be updated.

Validating Status

The virtual instance can acquire the status of the file *ADAHA.{DBID}* in order to perform health checks, or to wait and track a running Adabas nucleus. Use the following command to validate the status:

```
distributedCheck.sh status
```

The corresponding exit code shows the status (see table below).

Exit Code	Meaning
0	Adabas is running correctly
148	Adabas is offline
101	Init of HA
102	Adabas nucleus during AUTORESTART
103	Adabas nucleus in startup phase
105	Shutdown requested
106	Cancel requested
107	Adabas nucleus during shutdown phase
108	Adabas nucleus shutdown done, clean-up phase
110	Local ADAHAM working
111	Another ADAHAM requesting unique access to the Adabas containers
200	Registered ADAHAM or Adabas nucleus no longer working, time update fails
201	Adabas nucleus process no longer exists
250	Deadlock found
251	Structure level mismatch

The exit code is 0 if the local Adabas nucleus is running correctly.

An exit code of between 100 and 150 can be returned if the Adabas nucleus is in a controlled startup or shutdown phase, or if it is running on a remote host. If an exit code in this range is returned, the Adabas nucleus is in a valid state, and no further action is required.

If an exit code >200 is returned, ADAHAM will force a shutdown of the Adabas nucleus and clean up the nucleus resources. After this, another instance of ADAHAM can start up the Adabas nucleus.

Releasing Access

The ADAHAM file *ADAHA.{DBID}* needs to be released during shutdown. Use the following command to release ADAHAM access to the local Adabas nucleus:

```
distributedCheck.sh release
```