Adabas for Linux and Cloud

Adabas-to-Adabas (A2A) Replication

Version 7.3.0

August 2026

ADABAS & NATURAL

This document applies to Adabas for Linux and Cloud Version 7.3.0 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 © 1987-2026 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH 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 GmbH and/or its subsidiaries is located at https://softwareag.com/licenses.

Use of this software is subject to adherence to Software GmbH's licensing conditions and terms. These terms are part of the product documentation, located at https://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 GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at https://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 GmbH.

Document ID: ADAOS-A2A-730-20260811

Table of Contents

Adabas-to Adabas (A2A) Replication v
1 About this Documentation 1
Document Conventions 2
Online Information and Support 2
Data Protection 3
2 Adabas-to-Adabas (A2A) Replication 5
A2A Architecture 6
Utility Functionality for A2A Replication 8
Replication Target File 9
Referential Integrity (RI) 10
Data Integrity 11
Handling of Adabas Response Codes 12
Utilities and Replication 13
Overview Replication Status 14

Adabas-to Adabas (A2A) Replication

This document contains information about Adabas-to-Adabas (A2A) replication.

The following topic is covered:

- *Adabas-to-Adabas (A2A) Replication*

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 *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	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**.

Product Training

You can find helpful product training material on our Learning Portal at **https://learn.softwareag.com**.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at **https://techcommunity.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 GmbH 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 GmbH resources.

Product Support

Support for Software GmbH 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 Adabas-to-Adabas (A2A) Replication

- A2A Architecture 6
- Utility Functionality for A2A Replication 8
- Replication Target File 9
- Referential Integrity (RI) 10
- Data Integrity 11
- Handling of Adabas Response Codes 12
- Utilities and Replication 13
- Overview Replication Status 14

Adabas-to-Adabas (A2A) replication is a component of the Event Replicator for Adabas Open Systems. This means that you must install the Event Replicator for Adabas Open Systems before you can use the A2A replication. The A2A replication also requires a special Adabas license file; A2A replication is only possible after you have activated this license file.

Please refer to the documentation of the Event Replicator for Adabas Open Systems for further information. In order to support the A2A replication, some additional functionality has been incorporated into Adabas; this additional functionality is described in the following.

A2A Architecture

While the replication to other replication targets is performed by the Event Replicator without the necessity of special code in the Adabas nucleus, the A2A replication is performed directly in the Adabas nucleus by additional code. Three system files (and one LOB file) are used for metadata and for intermediate storage of the transactions to be replicated. Unlike the replication execution, the administration of the A2A replication is a part of the Event Replicator administration; in order to let Adabas know the definitions made in the Event Replicator interface, the Event Replicator administration calls an Adabas-A2A-internal replication administration interface.

Replication metadata file
The replication metadata file contains information about the source, target and status of currently defined replications. It contains one record for each replication source file of an A2A replication.

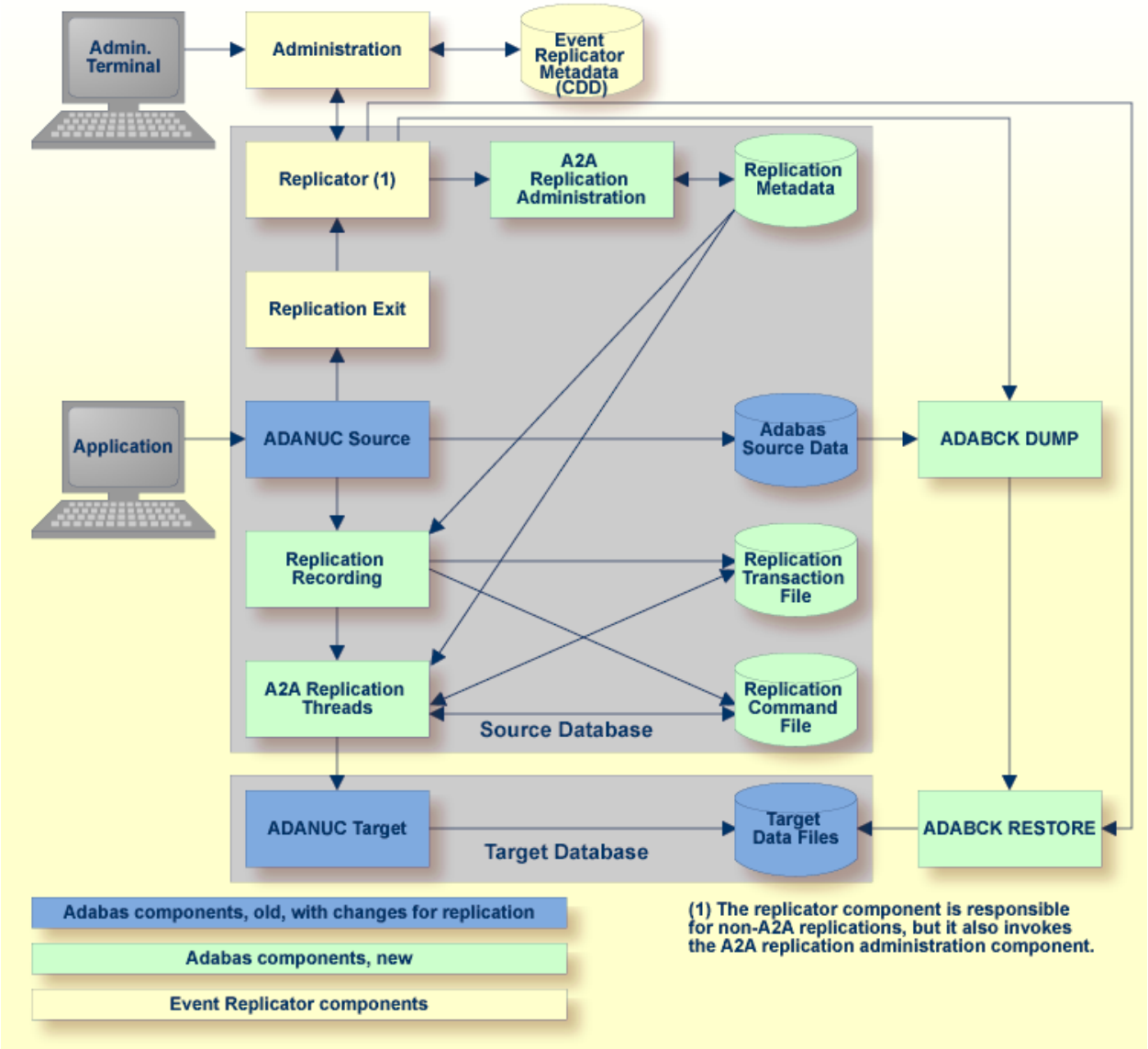
Replication command file
The replication command file contains records that describe database modification commands (A1, E1, N1, N2) not yet replicated to all target files.

Replication transaction file
This is for the intermediate storage of transactions not yet replicated to all target files. The replication transaction file contains one record for each transaction until this transaction has been replicated to all target databases.

Replication LOB file
This is the LOB file of the replication command file. The buffers of an Adabas command can become quite large, especially if you use the ACBX interface. For this reason, the buffers for the commands stored in the replication command file are stored in LOB fields.

Note: LOB files require a DATA block size of 32 KB. Therefore, you must add a DATA container extent with 32 KB block size if you want to start using A2A replication, and if all of your other DATA container extents have a smaller block size.

The following diagram shows the architecture of the A2A replication:



Replication Administration

The Event Replicator provides an administration tool which writes information to the Event Replicator metadata.

The administration tool communicates with the replicator component, which is not only responsible for the Adabas-to-relational and Adabas-to-JMS replication, but also invokes the A2A replication administration component, which is part of Adabas. The A2A replication administration stores the replication metadata in the Adabas replication metadata system file. Having the replication metadata within Adabas enhances the performance of the A2A replication.

For the connection between Adabas and the Event Replicator, a special kind of user exit, the replication exit is required. The replication exit starts the replicator component.

Initial State Processing

In order to start live data event replication, you must first copy the data to be replicated into the target database. It is important that this copy contains a consistent state of the files to be replicated (all users at ET status) and also that the replication guarantees that the recording of the transactions to be replicated starts at exactly the time when the files have this consistent state. In the Event Replicator, this step is called the *initial state*.

When you execute *deploy* in the Event Replicator administration, this initial state processing is started by internally using the backup utility ADABCK. ADABCK functionality has been extended in order to guarantee the consistency between the replication source and target files.

Note: If the target database is too small for the replication target files, ADABCK automatically increases the size of target database. For more information, please refer to *Utilities*, *ADABCK, Functional Overview*.

Replication Recording

During the initial state processing, the recording of the database modification commands to be replicated is started: the replication recording component writes the transactions and commands to be replicated to the replication transaction file and to the replication command file.

Active Replication

The replicator reads the transaction records in the correct chronological sequence, then it reads the corresponding command records in the right sequence. This guarantees that the commands are replicated in the correct sequence. The A2A replicator threads apply the commands to the target database. They try to do this in parallel, but some internal rules governing the order in which the commands are to be executed are necessary in order to guarantee the consistency between source and target files. When all replication data have been processed, the replicator threads wait for notification from the replication recording that a new transaction has been committed.

Utility Functionality for A2A Replication

Some new functionality has been added to some of the Adabas utilities in order to support A2A replication. The following contains a short description of these functions; please refer to the *Utilities* documentation for further details.

ADABCK REPLICATION

ADABCK is used for the initial state processing of the A2A replication, i.e. copying an initial state of a replication to the target database. Some extensions have been implemented in ADABCK for A2A replication in order to prevent transactions getting lost when the replication is started and also to mark the target files as replication target files. These extensions are called by the Event Replicator administration and are, therefore, not directly visible to the users. For this reason these extensions are not documented in the *Utilities* section.

ADADBM REMOVE_REPLICATION

If a database is no longer intended to be a replication source for A2A replication, the function ADADBM REMOVE_REPLICATION removes the replication system files and the database is reverted back to a normal database that is not A2A replication-enabled.

ADADBM REPLICATION_FILES

The new function ADADBM REPLICATION_FILES defines the new system files required for the A2A replication. This function must be executed before you can start with A2A replication.

ADADBM RESET_REPLICATION

Target files for an A2A replication cannot be updated by normal Adabas users. The new function ADADBM RESET_REPLICATION must be executed if you want to use these files as normal files again.

ADAOPR DISPLAY=REPLICATION

The new function DISPLAY=REPLICATION gives you an overview of the currently-defined replications and their status.

Replication Target File

A replication target file (replication target flag set) can only be updated by specially-marked replication calls from an Adabas replicator. Normal insert and update Adabas calls (A1, E1, N1, and N2) from other users will be rejected with response 17 (invalid file number) and subcode value=2 (unauthorized system file access).

However, replication target files can be converted back to normal files by using the new ADADBM function RESET_REPLICATION. In this case, the standard Adabas processing for data fields and files is restored, all Adabas commands are permitted and the standard SY option handling is enabled (see notes below).

Notes:

1. The replication target flag can only be reestablished by a renewed initial state processing.
2. The current status of the replication target flag can be checked using ADAREP FILES.
3. If the replication target flag is reset in the FCB, subsequent replication commands will be rejected for this file.
4. In a replication target file, system generated fields (SY option set) get the same values in the target record as in the source record.

Caution: The current version of A2A replication does not support the replication of utility operations, for example ADADBM REFRESH. For this reason, utility operations are permitted on replication target files. It is the responsibility of the DBA to ensure that utility operations are performed in sync on both the source and the target file, and only when the replications are up-to-date. If utility operations are only performed on the replication target file but not on the source file (or vice versa), inconsistencies between the source and target file can occur, and the replication can get errors. ADAINV is an exception to this behaviour - adding or removing descriptors does not change the data that is stored in a record, and can, therefore, be done without the risk of any inconsistencies or errors.

Referential Integrity (RI)

If referential integrity (RI) is defined, a database modification command can implicitly trigger other database modification commands (for example, via cascaded delete). In order to prevent these implicitly-triggered commands getting lost during Adabas-to-Adabas replication, the following rules apply for handling RI in A2A replication:

- If a file to be replicated contains the foreign key of an RI constraint, the primary file must also be replicated.
- It is possible to replicate only the primary file of an RI constraint, but if you subsequently want to add the file that contains the foreign key, you must include the primary file in a subsequent initial state processing; otherwise the RI constraint will not be defined in the target database.

Data Integrity

Initial State Processing during Parallel Updates

The functionality of ADABCK has been extended to enable the initial state processing. In order to keep the source and target file in sync, the ADABCK ET synchronization performed at the end of a backup is used. Normally this is used to guarantee a consistent state of the Adabas files on the backup: during the ET synchronization, only active transactions may be terminated, and any Adabas commands initiating a new transaction are put into a wait status until the ET synchronization has finished. At the end of the ET synchronization, when all open transactions are finished and the source data files are still locked, the replication status of the replications concerned is set to Recording in the replication metadata file in order to indicate that updates for the files to be replicated must be recorded from this point onwards. This guarantees that exactly those updates based on the saved state of the files are recorded for replication.

Data Integrity during Database Crashes

An important requirement for a replication solution is that data integrity must be maintained even if the source database or the target database crashes.

Source Database Crash

The source Adabas database is also used as intermediate storage of the update transaction; the replication data is stored in the same user transaction, that has to be replicated. This automatically guarantees that the storage of the replication data is also committed if an update transaction is committed, and that the storage of the replication data is also rolled back if an update transaction is rolled back. This guarantees that the replicator can process all committed transactions, and that no transactions are lost.

On the replicator side, it may occur that the source nucleus crashes during a transaction for the updates in the replication target file. Following the auto restart of the source database, a new OP command for the target database is performed by the replication database, which backs out the open transaction in the target database. This makes it possible to continue the Adabas session of the previous source nucleus session, and this means that an open transaction from the previous source nucleus session is rolled back.

Target Database Crash

If a target database crashes, it is possible that the replication database does not know if a transaction was committed in the target database or not. In order to guarantee the consistency of the target database, ET data are written by the replication database (please refer to section *Ababas Basics, Database Design, Transaction Recovery* for further details).

Handling of Adabas Response Codes

When the Adabas commands are replicated, non-zero Adabas response codes can occur on the target database. Some of them can be handled by the replication database itself and do not need any intervention, whereas others will require the intervention of the DBA.

Response Code 9

In order to avoid a timeout in the target database it is useful to define large timeout values in the target database. The replicator database provides such an OP command for the target database. It is also important that the WORK for the target database is large enough so that no WORK overflow is caused by the replication.

- Initially,the replication nucleus tries to repeat the transaction.
- If the response code 9 still occurs, the replication nucleus writes an appropriate message to its nucleus log and sets the replication exit status to recording. Then the DBA can act on the basis of what caused the error, for example, he can increase the WORK size of the target database and then reset the status to active using the Replication Administration tool.

Response Code 148

Response code 148 (Adabas nucleus not active) indicates that the target database is not available. An appropriate error message is printed in the replicator nucleus log, and the target database is polled with OP commands by the replicator until it is available again. Then the current transaction is restarted.

Response Codes from Net-Work

- Response code 149 indicates a communication error.
- Response code 224 indicates a command timeout by Net-Work.

An appropriate error message is printed in the replicator nucleus log, and an OP command is issued for the target database. Then the current transaction is restarted. If the error occurs again, the status of the replication is set to recording.

Switching the Status Recording following Target Nucleus Response Codes

There are several response codes that can occur when the target database is available, but it does not make sense to continue the replication because interaction by the DBA is required. For these response codes, a BT and CL command are issued by the replication database and the status is changed to recording. The response codes concerned are listed below:

- 47 - Maximum for NISNHQ parameter exceeded
- 48 - Requested database operation not allowed
- 49 - Compressed record too long for DATA container
- 72 - User queue overflow
- 77 - No space available in database container
- 79 - Hyperexit not available
- 114 - File refresh via E1 failed
- 162 - Buffer pool overflow
- 245 - Pending utility entries in UCB
- 255 - Client-server communication error

After the problems have been resolved, the DBA should reset the status to active again.

Handling unexpected Response Codes from the Target Database

Other Adabas response codes will result in setting the replication status to error (together with an appropriate error message in the nucleus log). For the target database the replicator issues a BT and a CL command.

It is the responsibility of the DBA to solve the problem and then execute another “initial state processing”.

Utilities and Replication

Utility operations are currently not replicated to the target database. It is, therefore, very dangerous to use certain utility functions with replication source files, this applies to the following functions in particular:

- ADABCK OVERLAY
- ADADBM functions
  - ADD_FIELD
  - CHANGE

  - CHANGE_FIELDS
  - DEFINE_REFINT
  - DELETE and recreate with ADABCK, ADAFDU or ADAORD
  - DROP_FIELDS
  - DROP_LOBFILE
  - DROP_REFINT
  - REFRESH
  - RENUMBER and recreate with ADABCK, ADAFDU or ADAORD
- ADAFDU ADD_LOBFILE
- ADAMUP
- ADAREC REGENERATE

If these utility functions are used for a source file of a replication, it is the responsibility of the user to ensure that the source and target file of the replication are kept in sync:

- Before the utility is started for the source file, there must be no pending update operations for that file.
- The same utility function must also be performed for the target file.
- There must be no new update operations replicated to the target file before the utility function has completed for the target database.

If these rules are violated, the replication may get errors, and the replicated data may be incorrect.

Overview Replication Status

The replication administration tool controls the correct sequence of the “initial state processing” via the replication status. After a successful initialisation, the A2A replication is in status active and ready for replication.

Status	Meaning
Inactive	Currently no data are replicated to the target file, and no activities have been made to initiate the replication.
Prepare	This status indicates that it is planned to perform the initial state processing for the replication. This status is the prerequisite for creating a backup of files to be replicated via ADABCK with parameter REPLICATION.
Initialization	This status indicates that ADABCK with parameter REPLICATION is running and creates a backup containing the initial state of files to be replicated.

Status	Meaning
Recording	Adabas is currently recording the update transactions within the replication command file and the replication transaction file, but currently does not replicate the update operations to the target database. During a “recording” phase, the replication transaction and the corresponding meta data are temporarily stored in the new replication system files. Depending on the data volume involved, this may also lead to a space problem in the source database.
Active	The replication is active; all modifications of the source file are replicated to the target file.
Error	An unexpected error occurred during replication. In order to continue replication, a new initial state processing is required.

The replication status of all replications defined in the source database can be displayed with ADAOPR DISPLAY=REPLICATION.

If an unexpected response code occurs during replication to the target database, an appropriate comment is displayed.