

# **Adabas Encryption for z/OS**

## **Introduction to Adabas Encryption for z/OS**

Version 8.6.1

September 2025

This document applies to Adabas Encryption for z/OS Version 8.6.1 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 © 2021-2025 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: EZ-INTRO-861-20250929EN**

## Table of Contents

Introduction to Adabas Encryption .....	v
1 About this Documentation .....	1
Document Conventions .....	2
Online Information and Support .....	2
Data Protection .....	3
2 Encryption Basics .....	5
3 IBM Z® Support for Encryption .....	7
4 Adabas Encryption for z/OS .....	9
Main Features of Adabas Encryption .....	10
Compatibility with Existing Security Tools .....	11
5 Encryption of Sequential Datasets .....	13
6 Encryption of Container Datasets .....	15



---

# Introduction to Adabas Encryption

---

Adabas Encryption for z/OS is a selectable unit of Adabas that encrypts the database container datasets (ASSO, DATA, WORK, and so on). This document describes its purpose and basic functioning.

<a href="#">Encryption Basics</a>
<a href="#">IBM Z® Support for Encryption</a>
<a href="#">Adabas Encryption for z/OS</a>
<a href="#">Encryption of Sequential Datasets</a>
<a href="#">Encryption of Adabas Database Container Datasets</a>

---

# 1

## About this Documentation

---

■ Document Conventions .....	2
■ Online Information and Support .....	2
■ Data Protection .....	3

## Document Conventions

---

Convention	Description
<b>Bold</b>	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>.

### Product Training

You can find helpful product training material on our Learning Portal at <https://learn.software-ag.com>.

### Tech Community

You can collaborate with Software GmbH 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 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 Encryption Basics

---

Various organizations, such as those running Adabas on IBM z/OS, have generated decades' worth of valuable information, including financial data, personal files, and proprietary business material. These organizations have a critical need to protect this data against ever-increasing cybercrime, and to comply with growing regulations (HIPAA, SOX, PCI DSS, GDPR and others).

Encryption is an effective way to protect data against unauthorized inspection or undetected manipulation. Encryption uses a secret key to transform readable data ("plaintext") into unreadable, meaningless noise ("ciphertext") that can only be understood using the correct secret encryption key. Done properly, encryption makes it practically impossible for someone without the secret key to tell the difference between encrypted data and purely random data.

Datasets on z/OS are customarily protected using an access control system like RACF®, ACF2® or TopSecret®. These systems prevent unauthorized accesses to protected datasets for all access paths under their control. They are not effective on access paths outside their control, though, such as when a dataset has been copied to another system or resides on shared DASD and is accessed from another system.

Encryption protects datasets wherever they reside. Encrypted datasets can be read in plaintext only on systems that have been provisioned with the required encryption keys. On those systems, the encryption keys can (and should) be protected from unauthorized use via the access control system.

*Data-in-flight* is data that is flowing across a public network (such as the internet) or a private network (such as a corporate LAN). *Data-at-rest* is data stored in datasets that are written to storage devices such as disk or tape, and data-at-rest persists even when the associated application is no longer running. Organizations that encrypt their data, especially data-at-rest, must carefully manage all their encryption keys across the entire key life cycle: from the creation of a key, through its use, and to its retirement and eventual destruction. Sharing a key with an unauthorized person means losing the protection provided by the encryption. Losing a key means losing the data it protects. Encryption key management is a challenging topic on its own and is *not* done by Adabas.



## 3 IBM Z® Support for Encryption

---

Adabas Encryption is based on z/OS Dataset Encryption, which provides for the encryption of datasets on DASD. z/OS Dataset Encryption was originally introduced (in 2017) for VSAM extended format datasets and extended format sequential datasets and later (in 2020) extended to basic format and large format sequential datasets, as well as datasets that are read and written directly using Execute Channel Program (EXCP, a low-level I/O interface).

z/OS Dataset Encryption encrypts datasets on DASD configured as 3390-type devices. Encrypted datasets must be managed by DFSMS.

A Dataset is either fully encrypted or fully unencrypted. The same encryption attributes, that is, the encryption key, encryption algorithm and associated parameters, apply to the entire dataset. For the encryption, DFSMS uses AES with 256-bit keys in XTS mode.

An existing dataset cannot be changed from “unencrypted” to “encrypted,” or vice versa. Rather, a new dataset must be created with the desired encryption attributes, and the data must be migrated from the existing dataset to the new one. The same applies if a dataset encrypted with one key must be encrypted with another key or must not be encrypted anymore.

Encryption keys and their associated encryption algorithms and parameters are defined in the Integrated Cryptographic Service Facility (ICSF). Encryption keys are associated with and identified by key labels, so that they can be referred to outside ICSF. A key label is a name for an encryption key and its associated parameters.

When a new dataset is created, the specification of a key label decides whether and how the dataset will be encrypted. There are three ways to specify a key label. In the order of precedence, these are:

1. In the RACF profile for the dataset: The key label can be specified in the DFP segment of the profile. It applies to all new datasets with names that match the profile.

2. In the JCL: The key label can be specified in the DSKEYLBL parameter. This takes effect if no key label is derived from the RACF profile for the dataset. Similar parameters exist for equivalent ways to create a dataset via IDCAMS DEFINE, dynamic allocation or ISPF.
3. In the SMS policy: The key label can be specified in the data class applicable to the dataset. This takes effect if no key label is derived from the RACF profile or specified in the JCL.



**Caution:** Establishing dataset encryption in a production environment is a complex undertaking. Mistakes can lead to the loss of data or to an inadvertent failure to protect data!

For a fuller introduction, see the IBM Redbook “Getting Started with z/OS Data Set Encryption” from June 2018. While this book predates the encryption of basic format and large format sequential datasets, and of datasets accessed directly using EXCP, their concepts are very similar.

For definitive specifications, see the IBM manual “DFSMS Using Data Sets,” Chapter 5, “Protecting data sets,” Section “Data set encryption.”

z/OS Dataset Encryption is a prerequisite for using Adabas Encryption. Once it has been set up and configured for any kinds of datasets on DASD, you are ready to start using Adabas Encryption for encrypting your Adabas databases.

# 4

## Adabas Encryption for z/OS

---

■ Main Features of Adabas Encryption .....	10
■ Compatibility with Existing Security Tools .....	11

Adabas Encryption for z/OS is a new, out-of-the-box, selectable unit of Adabas that encrypts database container datasets (ASSO, DATA, WORK, and so on) at the dataset level. It leverages the security and performance benefits of hardware-based encryption and the key management facilities provided by the IBM Z platform.

Encryption of datasets on DASD is an effective way to strengthen their security:

- Separate one user's authorization to access sensitive data in datasets from another user's authorization to manage those datasets. Dataset management includes tasks like backing up, migrating, or replicating datasets.
- Protect datasets on shared DASD against accesses from other systems that can have different access control rules (in RACF, for instance). This is important, for example, when the datasets reside in a storage area network (SAN).
- Ensure dataset security across all storage tiers, from a single DASD device up to cloud storage.

The use of Adabas Encryption is transparent to existing Adabas application programs. No application changes are required for installing and using Adabas Encryption.

Adabas Encryption encrypts datasets on DASD (namely, the database container datasets). It does *not* encrypt datasets on tape (for example, save tapes) or data in memory (for example, the Adabas buffer pool).

## Main Features of Adabas Encryption

---

Adabas Encryption provides for the following:

- *Integration into z/OS Dataset Encryption* —

Leveraging z/OS Dataset Encryption for the creation and use of encrypted databases and for high-performance, hardware-based encryption and decryption operations (using CPACF).

- *Integration into enterprise key management* —

Managing the Adabas encryption keys the same way as the other encryption keys used in the organization.

- *Transparent access to encrypted databases for application programs* —

No changes or adjustments to application programs for working with encrypted databases.

- *Full support of encrypted databases by the Adabas nucleus and utilities* —

No functional restrictions for the application programming interface or administrative functions.

- *Support for encrypting databases initially and for encryption key rotation* —

Migration to encrypted databases with little downtime.



■ *zIIP support* —

Offloading of encryption and decryption operations to zIIP processors.

## Compatibility with Existing Security Tools

---

*Adabas Encryption* applies to the database container datasets on DASD (data-at-rest). To secure the traffic flowing to and from Adabas (that is, Adabas calls) through your network (data-in-flight), use *Encryption for Entire Net-Work*. Together, the two keep your Adabas data encrypted whenever it is not being actively used by Adabas or your application programs.

Encrypting your database keeps your data secret when it is accessed directly, without going through Adabas. To protect your data when it is accessed through Adabas, you need access control. This can be established using Adabas System Authorization Facility (SAF) Security. With Adabas SAF Security, you can do the following:

- Define resource profiles that establish and enforce rules defining which Adabas users have permission to perform which operations (for example, search, read, and update) on which Adabas resources (for example, databases, files).
- Maintain and administer these rules in the same access control system (RACF, ACF2, TopSecret) that you use for the other resources (for example, datasets, encryption keys) in your z/OS system.

Adabas Encryption and Adabas SAF Security complement each other to allow only those accesses to your Adabas data that are permitted by your access control system, and to prevent all other accesses. They make your access control system the central point for governing your Adabas security.



## 5 Encryption of Sequential Datasets

---

The Adabas nucleus and utilities use the following sequential datasets:

Dataset type	Associated DD name(s)
Job input parameters	DDCARD, DDKARTE  Often provided via the JES spool (for example, 'DD *')
Job output protocol	DDPRINT, DDDRUCK  Often directed to the JES spool (for example, 'DD SYSOUT=*')
Payload input data	For example: DDEBAND for ADALOD LOAD DDREST1-8 for ADASAV RESTORE
Payload output data	For example: DDSAVE1-8 for ADASAV SAVE DDSIAUS1-2 for ADARES PLCOPY

Adabas reads and writes these sequential datasets using standard system interfaces (BSAM or QSAM). They may be encrypted if the encryption (for output) and decryption (for input) takes place at that system interface level or below.

Adabas itself is not involved in the encryption and decryption of sequential datasets. It supports the use of encrypted sequential datasets for nucleus and utility executions with or without the add-on Adabas Encryption.

With z/OS Dataset Encryption, encrypted basic format and large format sequential datasets on DASD need 8 bytes more space per block than unencrypted sequential datasets. In Adabas, the calculation of the default block and record sizes (BLKSIZE & LRECL) for sequential output datasets on DASD has been adjusted to take this extra space into account. This avoids the wasting of DASD space.



**Note:** Zaps AO852011, AO851020, AO843024, and AO842016 adjust the default block size calculation for extended format and encrypted sequential datasets. They are recommended for all installations.

## 6 Encryption of Container Datasets

---

The Adabas nucleus and utilities work with the following “database container datasets”:

Container	Associated DD name(s)
ASSO	DDASSOR1 ... DDASSOR $n$ , where $1 \leq n \leq 9$ DDASSO10 ... DDASSO $nn$ , where $10 \leq nn \leq 99$
DATA	DDDATAR1 ... DDDATAR $n$ , where $1 \leq n \leq 9$ DDDATA10 ... DDDATA $nn$ , where $10 \leq nn \leq 99$
WORK	DDWORKR1, DDWORKR4
CLOG	DDCLOGR1 ... DDCLOGR $n$ , where $1 \leq n \leq 8$
PLOG	DDPLOGR1 ... DDPLOGR $n$ , where $1 \leq n \leq 8$
RLOG	DDRLOGR1
DSIM	DDDSIMR1
SORT	DDSORTR1, DDSORTR2
TEMP	DDTEMPR1

When the selectable unit Adabas Encryption is used, any of the above container datasets can be encrypted.

The decision whether to encrypt a dataset or not is made at the dataset level when the dataset is created. One dataset can be encrypted while another dataset is left unencrypted.



**Note:** It is possible to encrypt, say, DDASSOR1 and DDDATAR1, and to leave DDASSOR2 and DDDATAR2 unencrypted, but such a setup has certain ramifications and is not recommended. Refer to the section *Operations > Encrypting Only Parts of ASSO and DATA*.

A dataset can be changed from unencrypted to encrypted only by creating it anew and, if necessary, migrating the data it contains. The following table describes the state that each database container dataset must be in before it can be replaced by a new dataset:

Container	Change state
ASSO	<p>DDASSOR<math>n</math> and DDASSO<math>nn</math> datasets must be created from new.</p> <p>Data in the database must be migrated to the new datasets, usually via an ADASAV SAVE and ADASAV RESTORE or RESTONL.</p> <p>Refer to the section <i>Operations &gt; Migrating to an Encrypted Database</i> for information on the methods that can be used to migrate an existing database to a new, encrypted one, including methods with brief Adabas downtime.</p>
DATA	<p>DDDATAR<math>n</math> and DDDATA<math>nn</math> datasets must be created from new.</p> <p>Data in the database must be migrated to the new datasets, usually via an ADASAV SAVE and ADASAV RESTORE or RESTONL.</p> <p>Refer to the section <i>Operations &gt; Migrating to an Encrypted Database</i> for information on the methods that can be used to migrate an existing database to a new, encrypted one, including methods with brief Adabas downtime.</p>
WORK	<p>DDWORKR1 can be changed after the Adabas nucleus has been shut down normally (that is, without a pending autorestart). In a cluster, this can be done separately for each nucleus.</p> <p>DDWORKR4 (used when Adabas is running with DTP=RM in conjunction with the Adabas Transaction Manager) can be changed after the nucleus has shut down normally, if there are no unresolved distributed transactions involving this database. In a cluster, all nuclei share the same DDWORKR4 dataset and must all be shut down.</p>
PLOG	DDPLOGR $n$ can be changed after the nucleus has been shut down normally and all outstanding PLCOPY operations have completed. In a cluster, this can be separately for each nucleus.
CLOG	DDCLOGR $n$ can be changed after the nucleus has been shut down normally and all outstanding CLCOPY operations have completed. In a cluster, this can be separately for each nucleus.
RLOG	<p>DDRLOGR1 (used by the Adabas Recovery Aid) can be changed after the nucleus has been shut down normally and when no utility is running. ADARAI PREPARE must be run with the new RLOG.</p> <p>The contents of the old RLOG cannot be migrated to the new one. In a cluster, all nuclei share the same DDRLOGR1 and must all be shut down.</p>
DSIM	DDDSIMR1 (used by the Adabas Delta Save Facility) can be changed when it contains no data from an online save that is still to be included in an outstanding ADASAV MERGE operation.
SORT	DDSORTR1 and DDSORTR2 can be changed between any utility operations that use them.
TEMP	DDTEMPR1 can be changed between any utility operations that use it.

Refer to the section *Operations > Migrating to an Encrypted Database* for information on the methods that can be used to migrate an existing database to a new, encrypted one.

Once a database container dataset is encrypted, Adabas Encryption must be used for all nucleus and utility jobs working with that dataset. Without Adabas Encryption, the nucleus and utilities cannot read encrypted container datasets.

Adabas reads and writes database container datasets using EXCP, which is a low-level I/O interface. If a container dataset is encrypted, Adabas uses a z/OS service to encrypt the data before it is written to disk and to decrypt the data after it has been read from disk. The actual encryption, decryption and key management are performed by z/OS. The invocation and control of the encryption and decryption operations are performed by Adabas.

Adabas encrypts the blocks in encrypted container datasets already during formatting (ADAFRM ASSOFRM, DATAFRM, and so on) and again during each write with new data. Different blocks with the same plaintext data before encryption (for example, binary zeros) have different ciphertext data after encryption.

