stellar

EXCHANGE DATABASE CORRUPTION

Troubleshooting Guide

Get in-depth information on Exchange database corruption, including the types of corruption, root causes, troubleshooting steps, and more.

CONTENTS

INTRODUCTION

Exchange Server database corruption can happen due to hardware failure, storage disk issues, missing or corrupted database objects, misconfigured antivirus, etc. It leads to situations such as Exchange Server databases being unable to mount and not being accessible, causing an outage if backups or high-availability setup like DAG are not available to support a resilient setup. A prolonged Exchange Server outage can hamper business communication and other email-based transactions, as the users cannot access their mailboxes or public folders. Exchange Server database corruption can lead to severe repercussions if it is not resolved quickly and effectively.

Some statistics highlight the need for advanced troubleshooting know-how for Exchange Server database corruption scenarios. According to HG Insights, approx. 0.85 million organizations in the world are using Microsoft Exchange Server. As per Statista, 43% of the total Exchange Server mailboxes worldwide are deployed on-premises in 2020. This data indicates the potential email database corruption issues these organizations may face (at least once) in the lifetime of their on-premises Exchange Server. Therefore, understanding how to troubleshoot Exchange database corruption can provide an effective method to restore email connectivity as quickly as possible.

This Guide shares an in-depth knowledge of Exchange Server database-related issues, including Exchange database corruption, causes, types and scenarios, troubleshooting steps, and Exchange Server database recovery methods. The information covered in this Guide applies to Exchange Server 2013, 2016, and 2019.

WHAT IS EXCHANGE SERVER DATABASE CORRUPTION?

Microsoft Exchange Server uses an EDB file (.EDB file format) to store, update, and sometimes archive the users' mailbox data along with public folders. This EDB file, also called the Exchange Server database file, is formatted as a B-tree data structure that keeps the data in a sorted form. Any anomaly or defect in the EDB file data structure or missing database objects like transaction logs is generally referred to as Exchange database corruption. There could be several reasons for an EDB file corruption, based on Physical and Logical root causes, resulting in various outage scenarios described later in this Guide.

WHAT CAUSES EXCHANGE DATABASE CORRUPTION?

As understood earlier, a missing database object or defective database structure is the primary cause of corruption. Following are the three primary reasons behind these causes explaining why an Exchange database is corrupted:

1. SYSTEM FAILURE AND CRASH

The server machine, like another computer system, comprises hardware components such as hard drives, motherboard, and power supply units. These components can deteriorate and fail anytime, leading to issues like hard drive bad sectors or system crash.

"Dirty shutdown" is a typical culmination of a system crash. It happens when the transactions are only partially committed to the database file due to a system crash disrupting the normal process. Consequently, the Exchange database is unable to mount due to a missing or altered log file. Dirty shutdown can cause a prolonged outage if the log file is unavailable to replay the transactions and there is no backup database.

2. STORAGE CAPACITY ISSUES

A large EDB file is prone to corruption due to the continual storage of additional mailbox data in the same file. EDB file has a limited storage capacity, as specified in the registry, exceeding which can corrupt the Exchange database file. Apart from the database increasing in size, you must consider the transaction logs, as these can take considerable space if a regular backup is not taken. A successful backup method will purge the logs, so if your logs are not being purged, you need to check the backup software for any incompatibility or issues.

3. ANTIVIRUS CONFIGURATION ISSUES

Running an improperly configured antivirus on Microsoft Exchange Server can result in problems, as the program can lock or quarantine a database or transaction log file, thereby hampering the Exchange Server from updating it.

Antivirus misconfiguration can result in significant failure scenarios, including EDB file corruption, DAG mailbox replication issues, etc. Therefore, it is crucial to exclude these specific files to avoid their scanning by the antivirus program. The Guide shares more information on antivirus related issues in the troubleshooting section later.

TYPES OF EXCHANGE DATABASE CORRUPTION

Exchange Server database files can get corrupted at different levels, namely the file system, application, and database. These are broadly categorized into the Physical and Logical types, as follows:

1. **Physical corruption** occurs at the 'file system' level due to a hardware problem in the storage drive, motherboard, etc., and it involves a malfunctioning or crashed system. System failure and storage capacity issues, discussed in the preceding section, correspond to hardware-related issues and physical corruption. Physical corruption cannot replicate with the Exchange replication process.

2. **Logical corruption** happens at the 'application' or 'database' level. For instance, an Information Store-related issue is an app-level corruption, and an error in the Extensible Storage Engine (ESE) is a database-level corruption. However, the ESE or JET Blue error codes can also indicate page or file system corruption, i.e., 'physical' issues. Logical corruption can replicate, unlike physical corruption.

ESE OR JET ERROR CODES INDICATING CORRUPTION

Extensible Storage Engine (ESE) is a Microsoft storage technology that enables applications like Exchange Server and Active Directory to store and retrieve data from tables. Applications using the ESE technology, earlier known as Joint Engine Technology (JET), use specific error codes to indicate the issues related to the database, input/output operations, etc. The following table summarizes ESE error codes for Exchange database corruption and other issues:

#	Error Code	Error Type	Description
1.	JET_errLogFileCorrupt-501	Logical	The error indicates corruption in the transaction log file.
2.	JET_errLogWriteFail-510	Logical	This error code indicates a failure writing to the transaction log file.
3.	JET_errDatabaseDirtyShutdown-550	Logical	This error indicates that the database is in a dirty shutdown state.
4.	JET_errReadVerifyFailure-1018	Physical	This is a page-level corruption
5.	JET_errPageNotInitialized-1019	Logical	This error appears when there is a blank or uninitialized page in the database due to lost page flush*.
6.	-1021 JET_errDiskReadVerificationFailure	Physical	This error indicates that a disk returned a bad checksum error.
7.	JET_errDiskIO-1022	Physical	This error code indicates a generic disk I/O problem.
8.	JET_errAttachedDatabaseMismatch -1216	Logical	This error appears when an outstanding database attachment is found; however, the database is missing or does not match the attachment details.

EXCHANGE OUTAGE - DATABASE CORRUPTION SCENARIOS

This section outlines Exchange Server outage scenarios involving database corruption. Downtime can happen due to a dismounted database, which you cannot mount for reasons like loss of backend storage access, missing or corrupted transaction logs (dirty shutdown state), logical corruption in the database file, or inadequate disk space. Such downtimes can turn into prolonged outages in the absence of a backup or automated recovery mechanism like Database Availability Group (DAG) to support failover. The following are major Exchange Server database dismount scenarios:

1. **Information Store is unable to start:** This situation arrives due to a disrupted communication between Exchange Server and Active Directory. It can also happen when the antivirus software or firewall locks a crucial Exchange file or blocks RPC communications. An incorrectly configured antivirus software can quarantine database files or logs, hampering the recovery mechanisms like DAG replication, backup recovery, etc. Notably, antivirus tools can also corrupt the Exchange database file while attempting to access the EDB file.

2. **Database in dirty shutdown state:** The "dirty shutdown" state arrives due to sudden dismounting of the Exchange database before the transactions (i.e., dirty pages) are committed to the Information Store. It is impossible to mount the database in a dirty shutdown state without the missing transaction logs and backup database. The ultimate solution in this situation is to recover the Exchange database file and restore it to a consistent state to allow mounting. This task can have a varying complexity depending on the database state.

3. **Insufficient disk space:** This situation involves rapid growth in the mailbox database and transaction log size, exhausting the disk space. Usually, the disk space is emptied after log truncation with every full or incremental backup. However, sometimes the database growth rate may be higher than log truncation, causing disk space shortage and database dismounting. You cannot mount the database until the disk space issue is resolved.

4. **Intermittent dismounting:** In this scenario, a database dismounts due to logical corruption, which can happen due to reasons like failure in writing to the transaction log file, blank or uninitialized database page, outstanding database attachment, issues in the transaction log file, etc. Here, fixing the database corruption solves the problem; however, it would require a proper diagnosis of the issue and might be daunting to resolve.

TROUBLESHOOTING EXCHANGE DATABASE CORRUPTION

Navigating through the different Exchange Server database dismount scenarios can be daunting, but it is crucial for troubleshooting the underlying corruption issues efficiently and effectively. The below decision tree* lays down the Exchange database dismount scenarios, including those having corruption.

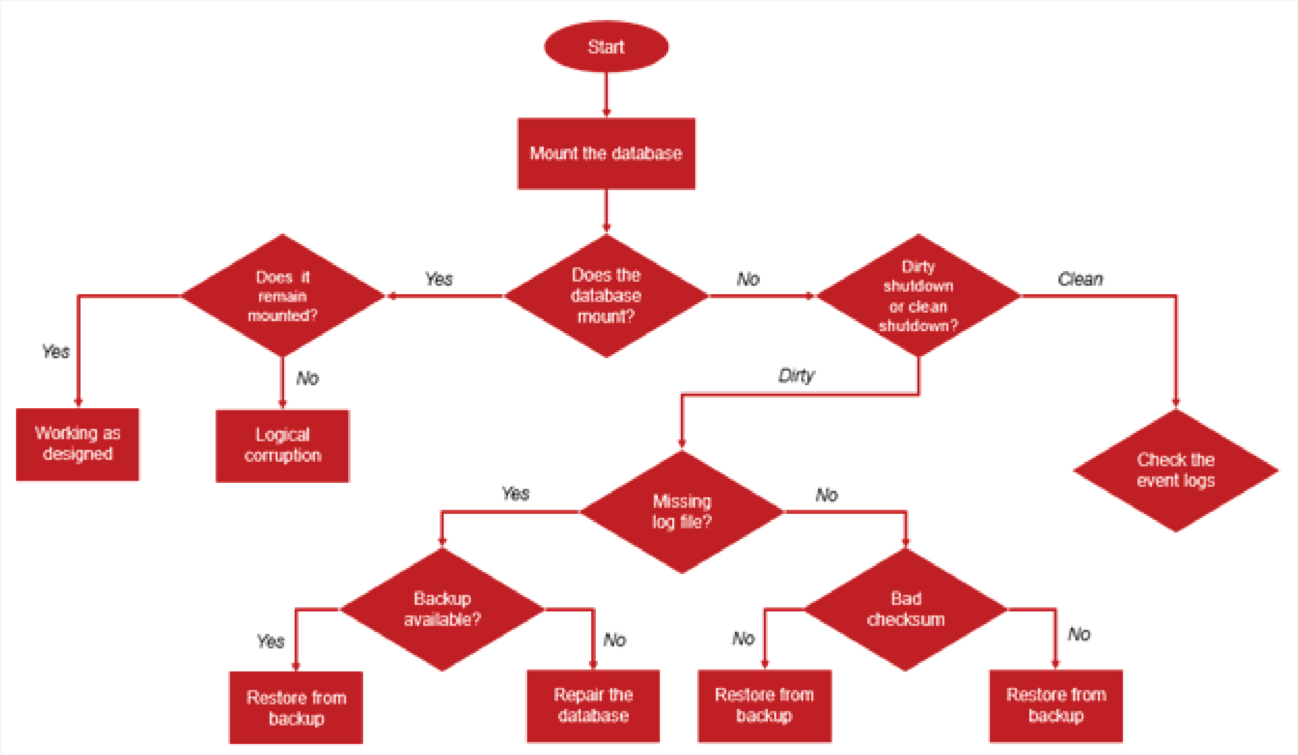


Fig 1: Email Usage (Year-wise)

**Cunningham, Paul. Higginbotham, Andrew. Exchange Server Troubleshooting Companion. Retrieved from https://gallery.technet.microsoft.com/office/Exchange-Server-Troubleshoo-db3a0dab*

Following are the steps to troubleshoot Exchange Server database corruption, starting with checking the specific error codes and their root causes, as follows:

Step 1: Check the JET error code

Check the ESE or JET error message, if available, to determine if the database corruption is of the physical or logical type and what does it indicate. Revisit the table of JET error codes, presented earlier in this Guide, for quick reference. For example, error codes 1018 and 1022 indicate physical corruption, and error codes 1019 and 1026 point to logical corruption.

Step 2: Determine the root cause

Identify the “root cause” of the JET error code after identifying the type of database corruption. The below table maps the ESE or JET error codes to their root causes.

#	JET Error Code	Cause
1.	JET_errLogFileCorrupt-501	JET error 501 happens due to hardware lost flush or I/O corruption.
2.	JET_errLogWriteFail-510	JET error 510 appears when a controller or hard disk fails to respond to disk commands. It may also occur due to improperly configured antivirus which locks the AD log files.
3.	JET_errDatabaseDirtyShutdown-550	JET error 550 appears due to sudden shutdown of the system due to which the transactions are not committed to the Information Store.
4.	JET_errReadVerifyFailure-1018	JET error 1018 appears due to hardware failure.
5.	JET_errPageNotInitialized-1019	JET error 1019 appears because of invalid pointers and linkages between the pages.
6.	-1021 JET_errDiskReadVerificationFailure	JET error 1021 may appear due to bad blocks on the hard disk.
7.	JET_errDiskIO-1022	JET error 1022 occurs when the operating system is unable to access the disk. This may happen due to an issue in the hardware or device drivers.
8.	JET_errAttachedDatabaseMismatch -1216	JET error 1216 occurs due to modified logs or lost I/O flush when the system is shutdown, leading to sudden stopping of a storage group.

Step 3: Fix the root cause of Exchange Server database corruption

After understanding the database error and the causes, the next step is to fix them. The following are the recovery steps for the specific ESE or JET errors:

1. JET_errLogFileCorrupt -501:

Resolve this error by restoring the logically corrupted database from a backup or reinstalling the domain controller.

2. JET_errLogWriteFail -510:

Fix this error as follows:

a) Restore the database access by restarting the server.

b) Replace the controller or disk

c) For software-related issues, check whether the antivirus program locks the Active Directory log files (or even a database file).

If so, add the files to the antivirus exclusion list as per the recommended exclusions for Windows antivirus programs on Exchange servers.

d) Check whether a third-party application or service is affecting Exchange Server functioning, as follows:

- Press Windows + R
- Enter MSCONFIG and click OK
- On the Services tab, select Hide all Microsoft Services
- Uncheck the box for third-party services
- Disable all enabled startup items
- Restart the server

3. JET_errDatabaseDirtyShutdown -550

This error message indicates dirty shutdown, one of the common Exchange Server downtime scenarios, often involving database and transaction log file corruption. This Guide comprises a dedicated section, “Troubleshooting dirty shutdown issues.”

4. JET_errReadVerifyFailure -1018

Error code 1018 indicates page-level corruption (i.e., physical corruption) in the database due to hardware failure. Here are the ways to resolve this Checksum error:

- Assess the hardware, including the motherboard, controller, cables, physical disks, etc., against the OEM recommended specifications and check for any indication of failure like the hardware monitoring system.
- Update the firmware or check whether the error appeared due to a recent update.
- You can attempt offline defragmentation to resolve index consistency related issues. How ever, ensure that you have a backup copy of the database before performing this operation. **Please read this blog post with detailed instructions on** how to defragment an Exchange mailbox database using PowerShell.
- Lastly, you can restore the database from a backup if the above steps did not resolve JET error 1018.

5: JET_errPageNotInitialized -1019

This error is like JET_errReadVerifyFailure -1018, but it occurs due to a lost flush page. You may resolve the error using the offline defrag method mentioned earlier if caused due to an index consistency issue.

6: -1021 JET_errDiskReadVerificationFailure

This is a disk-level bad checksum error occurring due to bad blocks on the hard disk. Running a disk check may resolve this problem by storing the data on the healthy blocks.

TROUBLESHOOTING EXCHANGE SERVER DIRTY SHUTDOWN

This section focuses on troubleshooting Exchange dirty shutdown error indicated as JET_errDatabaseDirtyShutdown -550. We begin by determining the database state and trying to fix the corruption to allow successful mounting. Here are the steps you need to follow:

Step 1: Check the database state

If the Exchange database is dismounted and you are unable to mount it, the immediate action should be to check the database shutdown state to ascertain whether the transactions in the memory were committed to the Exchange Information Store during the shutdown process. Another way to determine the cause of the database not mounting is to view the event viewer or try to mount the database and see the error prompt.

You can use the Eseutil /mh command on the dismounted database to check its shutdown state, essentially, the “State” and “Log required” attributes.

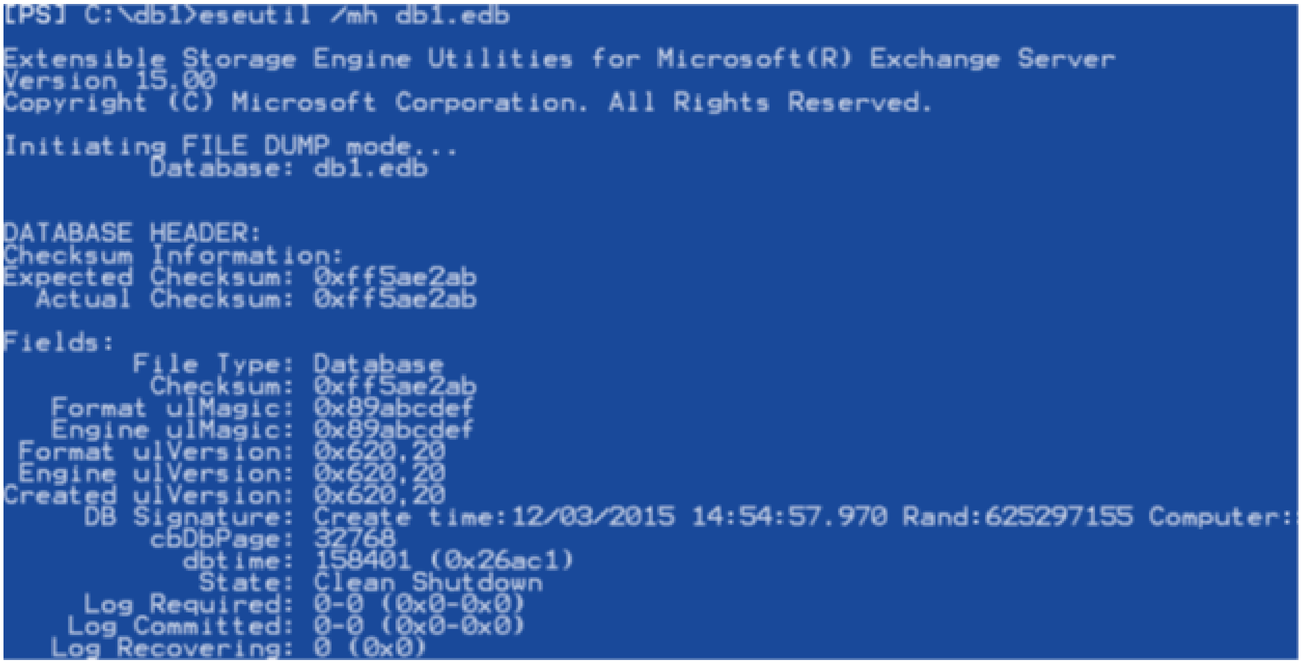
- State attribute: it displays the database shutdown state, i.e., clean or dirty.
- Log required attribute: it displays the transaction log files needed to replay the changes in the database tables.

The below example illustrates the use of the Eseutil /mh command on a dismounted database named db1, validating its clean shutdown state.

```
C:\db1>eseutil /mh db1.edb
```

You will notice that the State field in the below screen displays “Clean Shutdown” and “Log required” field shows 0-0, indicating no transaction logs are needed to mount the database.

Therefore, based on the above decision tree, you need to check the event logs to know the reason behind the dismounted database.

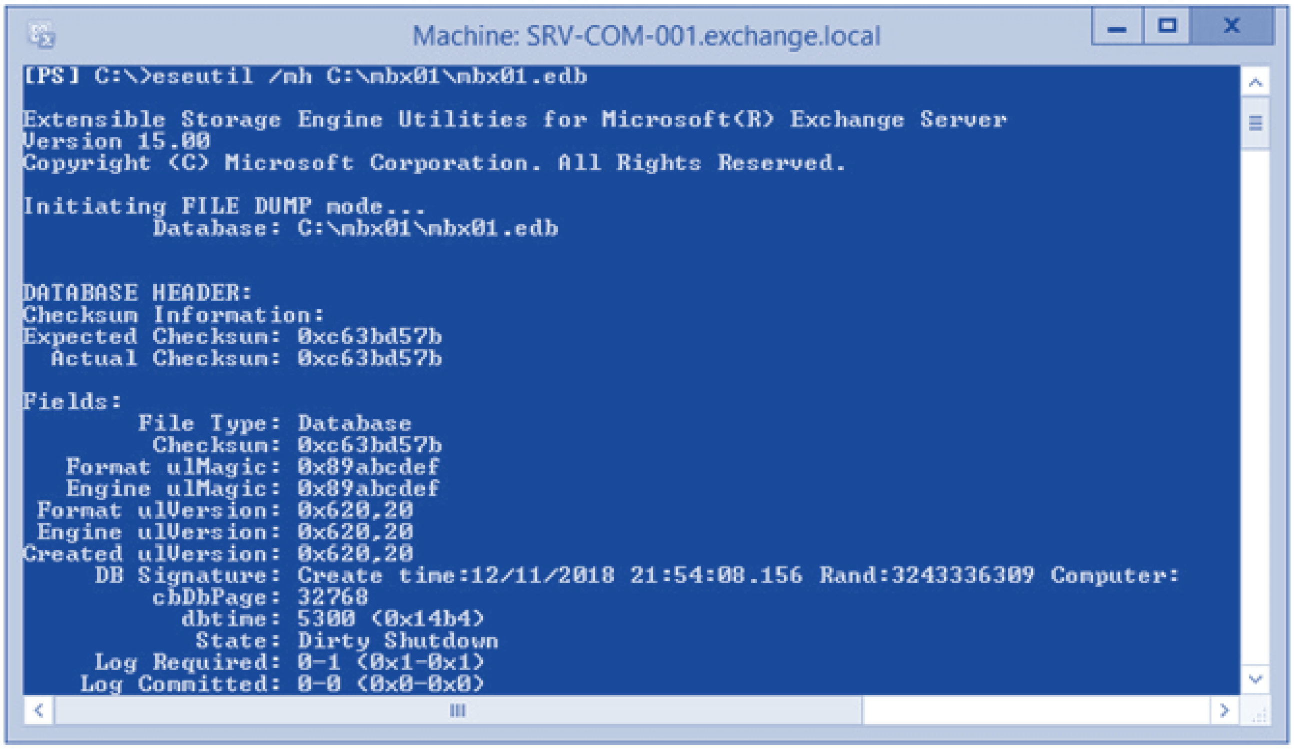


```
[PS] C:\db1>eseutil /mh db1.edb

Extensible Storage Engine Utilities for Microsoft(R) Exchange Server
Version 15.00
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating FILE DUMP mode...
         Database: db1.edb


DATABASE HEADER:
Checksum Information:
Expected Checksum: 0xff5ae2ab
  Actual Checksum: 0xff5ae2ab

Fields:
        File Type: Database
         Checksum: 0xff5ae2ab
   Format ulMagic: 0x89abcdef
   Engine ulMagic: 0x89abcdef
 Format ulVersion: 0x620,20
 Engine ulVersion: 0x620,20
Created ulVersion: 0x620,20
     DB Signature: Create time:12/03/2015 14:54:57.970 Rand:625297155 Computer:
         cbDbPage: 32768
           dbtime: 158401 (0x26ac1)
            State: Clean Shutdown
     Log Required: 0-0 (0x0-0x0)
    Log Committed: 0-0 (0x0-0x0)
   Log Recovering: 0 (0x0)
```

What happens when you run the Eseutil /mh command on a database that is in a dirty shutdown state? The following example illustrates —

```
C:\>eseutil /mh C:\mbx01.edb
```

The below screen now displays the State as “Dirty Shutdown” and log 1 as the required log file. These attributes convey that the transaction log is either partially committed or needed to restore the database to a consistent state. **However, the file is certainly required in a complete and uncorrupted state to mount the database.**



```
[PS] C:\>eseutil /mh C:\mbx01\mbx01.edb

Extensible Storage Engine Utilities for Microsoft(R) Exchange Server
Version 15.00
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating FILE DUMP mode...
         Database: C:\mbx01\mbx01.edb


DATABASE HEADER:
Checksum Information:
Expected Checksum: 0xc63bd57b
  Actual Checksum: 0xc63bd57b

Fields:
        File Type: Database
         Checksum: 0xc63bd57b
   Format ulMagic: 0x89abcdef
   Engine ulMagic: 0x89abcdef
 Format ulVersion: 0x620,20
 Engine ulVersion: 0x620,20
Created ulVersion: 0x620,20
     DB Signature: Create time:12/11/2018 21:54:08.156 Rand:3243336309 Computer:
         cbDbPage: 32768
           dbtime: 5300 (0x14b4)
            State: Dirty Shutdown
     Log Required: 0-1 (0x1-0x1)
    Log Committed: 0-0 (0x0-0x0)
```

Step 3: Perform Exchange database recovery using Eseutil

There are two native ways to perform database recovery using Eseutil, namely

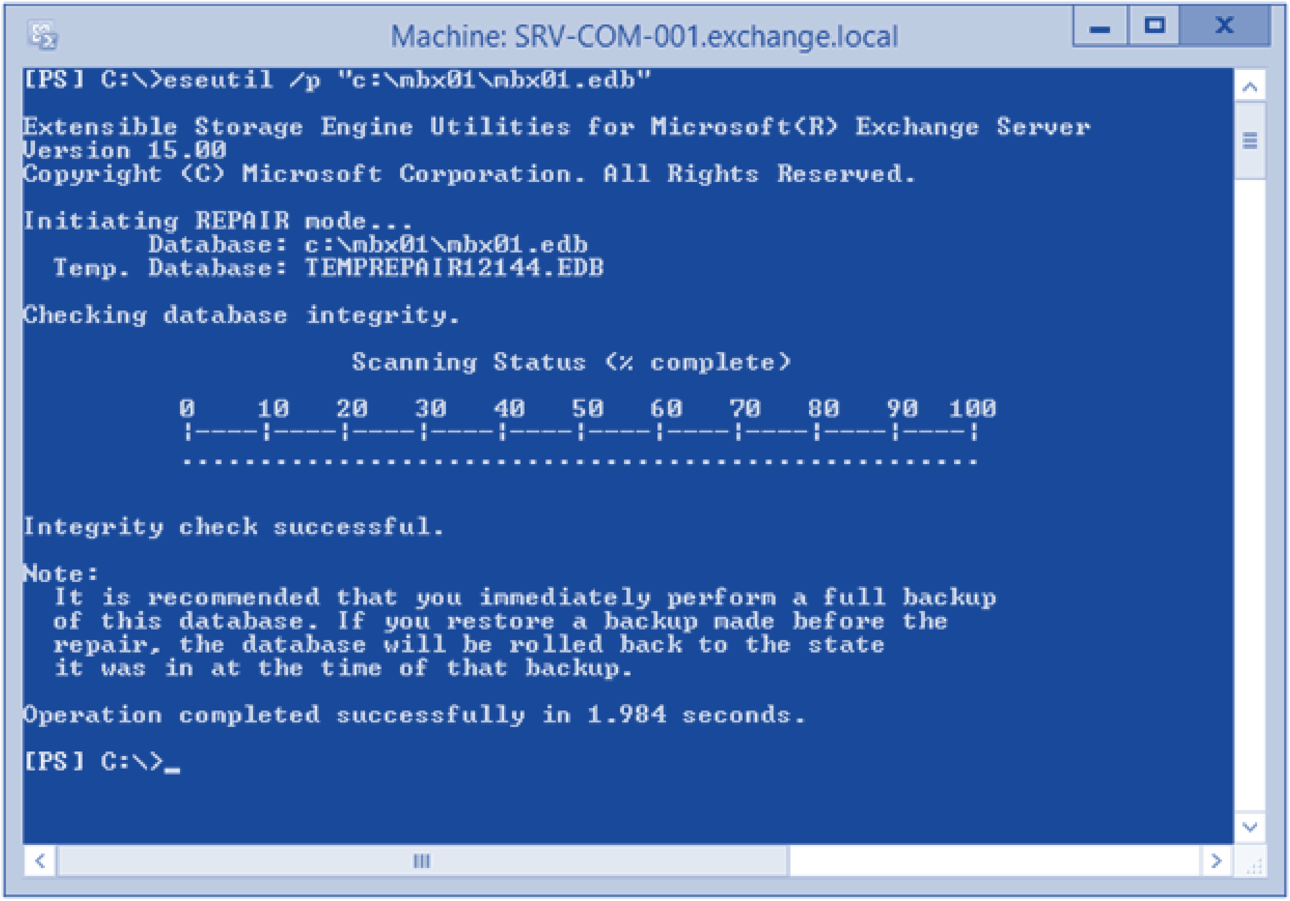
1. **Soft Recovery:** Performed using Eseutil /r, which replays the transaction logs to restore the database to a consistent state. In Exchange 2010 and later versions, Eseutil /r is run auto matically to replay the required log files, restoring the database consistency to allow mounting. You need to run the command manually when replaying the transaction logs from a backup or lagged copy.

The complete syntax of Eseutil /r command looks like this:

```
ESEUTIL /r eNN /L <log files path> /s <checkpoint file path> /d <database path> /i
```

- eNN – prefix of the required log file to be replayed
- /L – optional parameter used when running the command outside of the log file location
- /s – optional parameter to specify the checkpoint file path, used for establishing the "point in time" for replaying the logs. Not required if you need to replay all the logs.
- /i – ignores hanging attachments in the log files

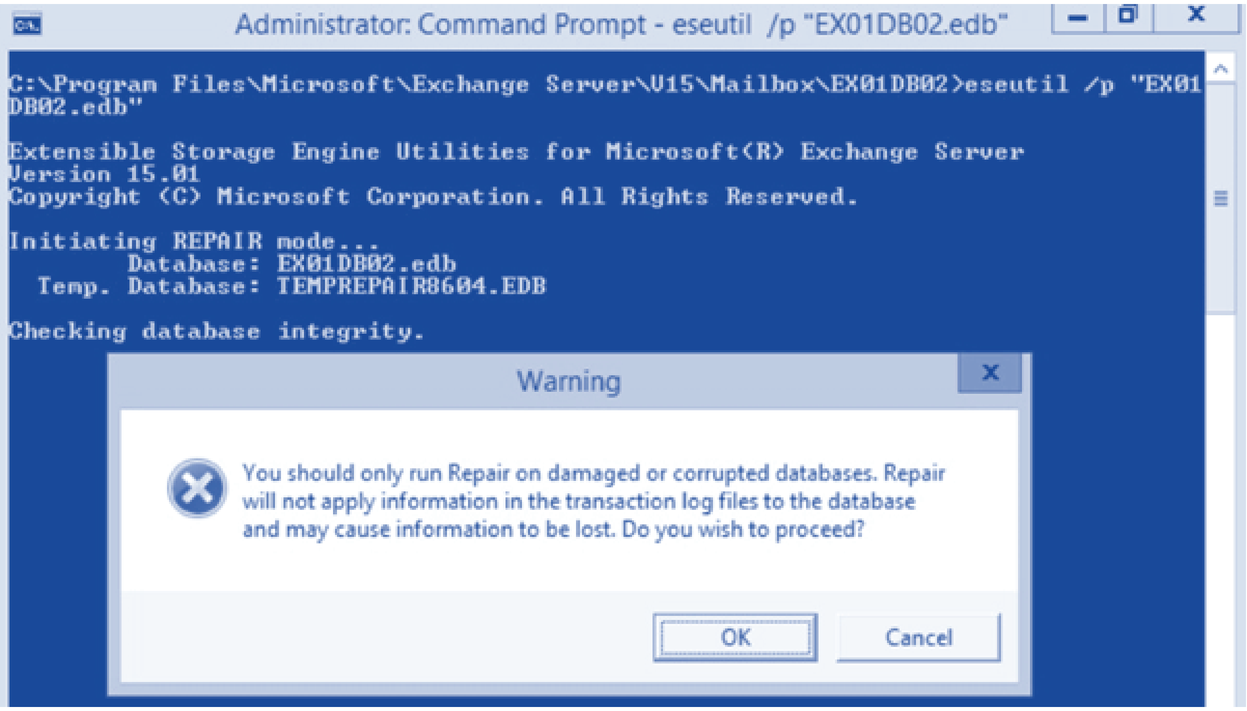
The screen illustrates the use of Eseutil /r on a database file named mbx01.edb.



```
[PS] C:\>eseutil /p "c:\mbx01\mbx01.edb"

Extensible Storage Engine Utilities for Microsoft(R) Exchange Server
Version 15.00
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating REPAIR mode...
        Database: c:\mbx01\mbx01.edb
  Temp. Database: TEMPREPAIR12144.EDB

Checking database integrity.

                     Scanning Status (% complete)

          0    10   20   30   40   50   60   70   80   90  100
          |----|----|----|----|----|----|----|----|----|----|
          ...................................................


Integrity check successful.

Note:
  It is recommended that you immediately perform a full backup
  of this database. If you restore a backup made before the
  repair, the database will be rolled back to the state
  it was in at the time of that backup.

Operation completed successfully in 1.984 seconds.

[PS] C:\>_
```

After replaying the logs, recheck the database state using the Eseutil /mh command described earlier. You should be able to mount the database if found in a clean state, i.e., restored to a consistent state after performing soft recovery. If the database is still in a dirty shutdown state, then probably it is corrupt beyond the scope of Eseutil /r, and you need to perform hard recovery.

2. **Hard Recovery:** Performed using the Eseutil /p command that purges any data from the database file, it deems corrupted and rolls back transactions to restore the database to a consistent state. However, hard recovery leads to data loss, so use it only as the last resort when options like backup restore are exhausted.



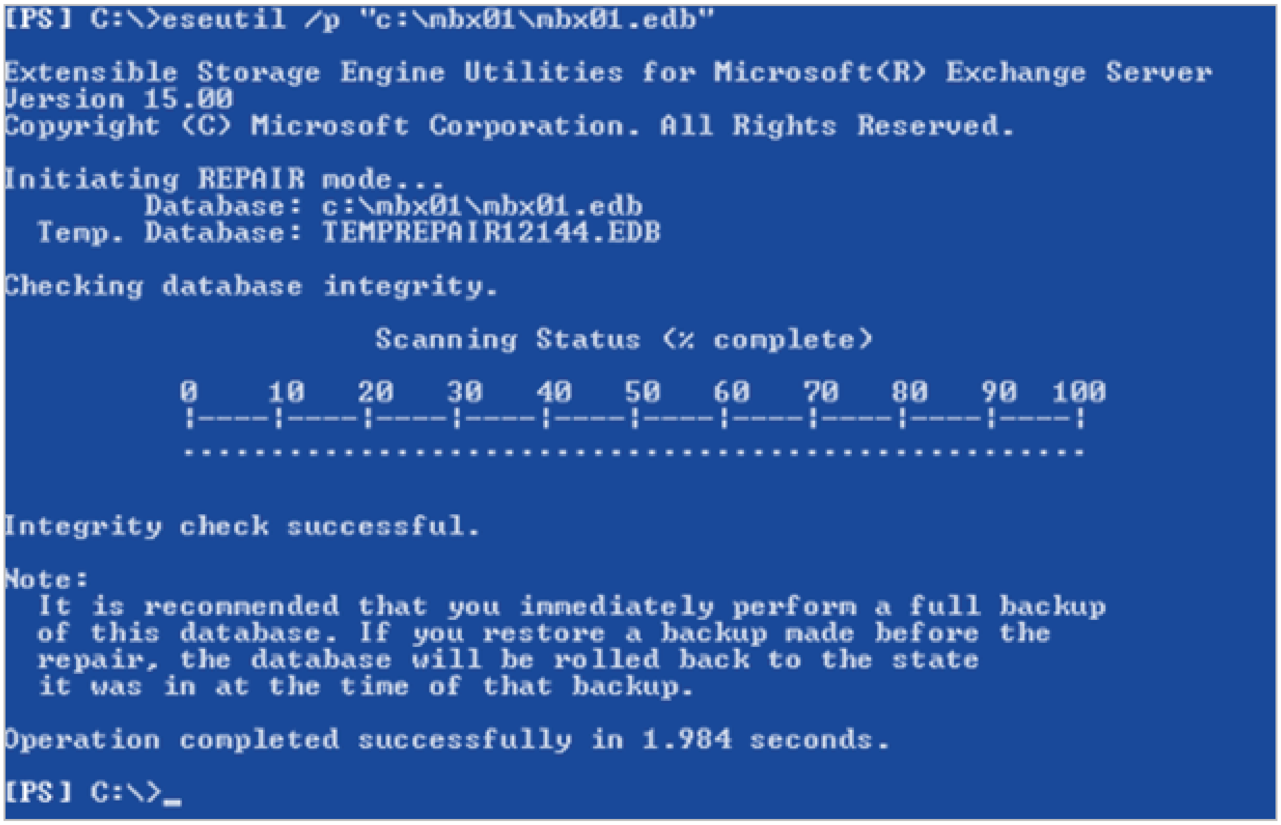
```
C:\Program Files\Microsoft\Exchange Server\V15\Mailbox\EX01DB02>eseutil /p "EX01
DB02.edb"

Extensible Storage Engine Utilities for Microsoft(R) Exchange Server
Version 15.01
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating REPAIR mode...
        Database: EX01DB02.edb
  Temp. Database: TEMPREPAIR8604.EDB

Checking database integrity.
```

You should only run Repair on damaged or corrupted databases. Repair will not apply information in the transaction log files to the database and may cause information to be lost. Do you wish to proceed?

The syntax of Eseutil /p command is as follows:

```
C:> ESEUTIL /p <DatabaseFileName>
```

The screen shows the use of Eseutil /p command on a database file named mbx01.edb.



```
[PS] C:\>eseutil /p "c:\mbx01\mbx01.edb"

Extensible Storage Engine Utilities for Microsoft(R) Exchange Server
Version 15.00
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating REPAIR mode...
        Database: c:\mbx01\mbx01.edb
  Temp. Database: TEMPREPAIR12144.EDB

Checking database integrity.

                     Scanning Status (% complete)

          0    10   20   30   40   50   60   70   80   90  100
          |----|----|----|----|----|----|----|----|----|----|
          ...................................................


Integrity check successful.

Note:
  It is recommended that you immediately perform a full backup
  of this database. If you restore a backup made before the
  repair, the database will be rolled back to the state
  it was in at the time of that backup.

Operation completed successfully in 1.984 seconds.

[PS] C:\>_
```

After completing the hard recovery process, purge all the database files and previous transaction logs, as you cannot replay them into the repaired database. Next, check the database state using the Eseutil /mh command and attempt to mount the database.

As said, use Eseutil/p as a last resort since you are accepting data loss with no guarantee that the database will be restored to a healthy state. Also, you need to consider that the database will be marked with a hardcode that it has been repaired. If you contact Microsoft, it will breach the support contract, and you will not get any support on the database if it is found to be repaired.
If able to mount, move all the mailboxes from the recovered database to a new database and run the New-MailboxRepairRequest cmdlet to detect and repair corruption in the mailboxes, as follows:

The command moves Angus' mailbox to a new database named DB05.

```
New-MoveRequest -Identity 'angus@alpineskihouse.com' -TargetDatabase "DB05"
```

The command detects and repairs all corruption types in Angus' mailbox.

```
New-MailboxRepairRequest -Mailbox angus -CorruptionType
ProvisionedFolder,SearchFolder,AggregateCounts,Folderview
```

EXCHANGE DATABASE REPAIR TOOL

So far, the Guide shared information on Exchange Server database corruption types and reasons, error codes related to database corruption, dirty shutdown scenarios, and troubleshooting methods. It reinforced a few takeaways regarding database corruption being is a serious business concern as it can lead to prolonged downtime, and troubleshooting can be daunting considering the vast scenarios and root causes. Add factors like high turnaround durations and risks associated with manual troubleshooting, and you can understand the utility and relevance of specialized tools designed to repair corrupt Exchange databases.

Third-party tools like **Stellar Repair for Exchange** are listed in the category of enterprise-grade software that focus on repairing and recovering from corrupted databases, irrespective of the type or root cause. Such tools equip you with a readymade capability to fix the database corruption with high assurance. Their core value lies in supporting fast and precise database recovery in urgent and severe downtime situations.

ENDING NOTES

This guide shared an in-depth understanding of Exchange Server database corruption and ways to troubleshoot using the prevalent methods, including the use of Eseutil, PowerShell cmdlets, and system-level measures. Our aim is to keep this resource up to date and provide well-informed guidance on Exchange Server database corruption-related matters.

REFERENCES

Extensible Storage Engine
Troubleshooting JET database errors
Exchange Dirty Shutdown Scenarios

stellar

Stellar Repair for Exchange

DOWNLOAD FROM:
https://www.stellarinfo.com/edb-exchange-server-recovery.htm

☎ +1-877-778-6087 (Tollfree)   ✉ support@stellarinfo.com   🌐 www.stellarinfo.com