

EOFException

EOFException

You can get a EOFException `java.io.EOFException` in two main ways

EOFException during [FileSystem](#) operations

Unless this is caused by a network issue (see below), and EOFException means that the program working with a file in HDFS or another supported [FileS ystem](#) has tried to read or seek beyond the limits of the file.

There is an obvious solution here: don't do that.

EOFException during Network operations

You can see an EOFException during network operations, including RPC calls between applications talking to HDFS, the [JobTracker](#), YARN services or other Hadoop components.

It can mean

Unexpected Server shutdown

The far end of the network link shut down during the RPC operation.

1. Verify that the server at the end of the network operation is running -restart it if not.
2. If the service is an HA component, it may be that failover has occurred -but the client doesn't detect this and retry its operation. Try restarting the application.

Protocol Mismatch

There is some protocol mismatch between client and server which means that the server sent less data than the client expected. This is rare in the core Hadoop components, as the RPC mechanisms used versioned protocols precisely to prevent versioning problems. It is more likely in a third-party component, or a module talking to a remote filesystem.

1. Retry the operation, it may work this time.
2. Look at the stack trace and see if it occurs in a Hadoop class (`org.apache.hadoop` -especially an RPC one), or something else.
3. If it happens in one of the Hadoop remote filesystems (s3, s3n, ftp ...etc.), or in Apache HTTP libraries, it usually means the far end has finished early. Try again.

Attention: Developers of RPC clients

If your users see this a lot, it implies it is time to make your client applications use the `org.apache.hadoop.io.retry` package to make them more resilient to outages.