

KIP-400: Improve exit status in case of errors in ConsoleProducer

- [Status](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: *Under Discussion*

Discussion thread: [here](#) [Change the link from the KIP proposal email archive to your own email thread]

JIRA: [here](#) [ Unable to render Jira issues macro, execution error.]

Motivation

Async mode in the current implementation of ConsoleProducer always returns exit code 0, even if it fails to send messages. In sync mode, tool exits immediately with status 1 on first exception. But In async mode, exceptions are logged and the tool continues processing till the end and then exit with status 0. This behavior may be okay when the user doesn't care about the loss of data but may not work if the user wants to know about failures and dropped messages.

Public Interfaces

There will be new argument added to console producer

```
--error-handler-strategy [ignore|continue|abort]           Async error handler strategy: either  
  
'ignore' (default. Continue on errors and exit with status 0),  
  
'continue' (Continue on errors and exit with status 1),  
  
'abort' (Abort on error and exit with status 1)
```

Proposed Changes

We propose adding new argument for async mode, '--error-handler-strategy' with 3 possible values.

- *Ignore (Default)* - This will be the current and default behavior. Log the errors, continue processing all the messages and exit with code 0
- *Continue* - Log the errors, continue processing all the messages and exit with code 1
- *Abort* - Abort processing as soon as error is detected and exit with code 1

Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*
No impact. The default behavior for this new argument is same as the current behavior.

- *If we are changing behavior how will we phase out the older behavior?*
NA
- *If we need special migration tools, describe them here.*
NA
- *When will we remove the existing behavior?*
NA

Rejected Alternatives

NA