

KIP-738: Removal of Connect's internal converter properties

- [Status](#)
- [Motivation](#)
- [Public Interfaces / Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
 - [Migration](#)
 - [User Insight](#)
- [Rejected Alternatives](#)
 - [Official migration tooling](#)

Status

Current state: *Accepted*

Discussion thread: [here](#)

JIRA: [here](#)

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

With [KAFKA-5540](#) / [KIP-174](#), the `internal.key.converter` and `internal.value.converter` Connect worker properties, as well as all properties prefixed with these names, were deprecated.

Despite this, users still sometimes try to experiment with these properties—for debugging, after accidental conflation with the non-deprecated key and value converter-related properties, or just blind misconfiguration of their worker. The results of these experiments can be disastrous; workers configured with a new internal converter are unable read internal topic data produced by a worker with an older internal converter, which leads to loss of offsets and connector configurations at best, and worker failure at worst.

Since these properties have been fully deprecated since the 2.0.0 release nearly three years ago and the upcoming 3.0 release allows for breaking changes, it'd be a great time to remove these properties from Connect for good.

Public Interfaces / Proposed Changes

The following Connect properties will be removed:

- `internal.key.converter`
- `internal.value.converter`
- passthrough of properties prefixed with `internal.key.converter.` to the worker's internal key converter
- passthrough of properties prefixed with `internal.value.converter.` to the worker's internal value converter

Connect will behave as if none of the above are provided. Specifically, for both its key and value converter, it will use the [JsonConverter](#) that it comes with out of the box, configured with `schemas.enable` set to `false`.

Compatibility, Deprecation, and Migration Plan

Migration

Users running Connect clusters that do not use the `JsonConverter` with `schemas.enable` set to `false` can follow these steps to upgrade their Connect clusters to 3.0:

1. Stop all workers on the cluster
2. For each internal topic (config, offsets, and status):
 - a. Create a new topic to take the place of the existing one
 - b. For every message in the existing topic:
 - i. Deserialize the message's key and value using the Connect cluster's old internal key and value converters
 - ii. Serialize the message's key and value using the [JSON converter](#) with schemas disabled (by setting the `schemas.enable` property to `false`)
 - iii. Write a message with the new key and value to the new internal topic
3. Reconfigure each Connect worker to use the newly-created internal topics from step 2
4. Start all workers on the cluster

User Insight

Users may not know about these changes and be confused when their workers fail to start or behave incorrectly after upgrading to 3.0. In order to help them discover the cause of failure as quickly as possible, a warning message will be logged on startup if the worker detects that it has been configured with any of these to-be-removed properties. Users will be linked to the upgrade notes for 3.0 in this message and instructed to follow them.

Rejected Alternatives

Official migration tooling

Summary: provide an official tool to users to perform the above-described migration.

Rejected because: based on feedback on the existing tickets related to the internal converter deprecation and removal (

 Unable to render Jira issues macro, execution error.

 Unable to render Jira issues macro, execution error.

 Unable to render Jira issues macro, execution error.

), there don't appear to be many (if any) users currently running Connect clusters

that would be affected by this change. The cost to develop, test, document, and support this tool would more than triple the work required for the current proposal. Given these two facts, the tradeoff does not seem worth it.