

# KIP-949: Add flag to enable the usage of topic separator in MM2 DefaultReplicationPolicy

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

## Status

**Current state:** *Adopted*

**Discussion thread:** [here](#)

**JIRA:** [here](#)

## Motivation

Since [KIP-690: Add additional configuration to control MirrorMaker 2 internal topics naming convention](#), the `replication.policy.separator` has been used to control the name of internal topics for `DefaultReplicationPolicy`, causing backward compatibility issues with offset syncs and checkpoint topics when users set a customized `replication.policy.separator`. This impacted 3.1, 3.2, 3.3, 3.4, 3.5 versions.

Currently, the only way to work around this is either by providing a new version of `ReplicationPolicy` (which can optionally subclass the `DefaultReplicationPolicy` class) that overrides the `ReplicationPolicy.offsetSyncsTopic` and `ReplicationPolicy.checkpointsTopic` methods to use old topics if users still want to use the old internal topics, or by deleting the old internal topics and letting MM2 set up the new ones.

In this KIP, we are proposing to add `replication.policy.internal.topic.separator.enabled` to allow customers to control this using configurations instead of providing a new implementation for both checkpoints and offset sync topics. The heartbeat topic isn't controlled by the separator at all.

## Public Interfaces

Property	Default value	Description
<code>replication.policy.internal.topic.separator.enabled</code>	true	whether or not to use <code>replication.policy.separator</code> to control internal topic names

## Proposed Changes

```

public class DefaultReplicationPolicy implements ReplicationPolicy, Configurable {

    private Boolean isInternalTopicSeparatorEnabled;

    @Override
    public void configure(Map<String, ?> props) {
        if (props.containsKey(INTERNAL_TOPIC_SEPARATOR_ENABLED_CONFIG)) {
            isInternalTopicSeparatorEnabled = (Boolean) props.get(INTERNAL_TOPIC_SEPARATOR_ENABLED_CONFIG);
        }
    }

    private String internalSeparator() {
        return isInternalTopicSeparatorEnabled ? separator : ".";
    }

    public String offsetSyncsTopic(String clusterAlias) // should use internalSeparator() to name the offset
sync topic
    public String checkpointsTopic(String clusterAlias) // should use internalSeparator() to name the offset
topic
    }
}

```

## Backporting plan

- The KIP need to be released as part of the bugfix release to the last 3 versions
- The KIP need to be backported into 3.1 and 3.2 (any other version reached EOL post KIP-690).

## Compatibility, Deprecation, and Migration Plan

Users who upgrade from Kafka < 3.1 and wish to disable controlling internal topics using the separator will need to set `replication.policy.internal.topic.separator.enabled` to `false`. Any users upgrading from > 3.1 don't need to do anything.

## Rejected Alternatives

- Introduce a `LegacyReplicationPolicy` that inherits `DefaultReplicationPolicy` that provide an override for `ReplicationPolicy.offsetSyncsTopic` and `ReplicationPolicy.checkpointsTopic` methods to use old topics and ignore the usage of the topic's separator. The advantage of this approach is avoiding introducing a new property to only fix a backward compatibility bug. However, this approach has a high risk of diversion between this new class and `DefaultReplicationPolicy` at some point in the future. It also adds another class to maintain for MM2 without adding a big value for the connector besides fixing the backward compatibility bug.