



## Auto Save Entry & Workflow Migration

### Table of Contents

<b>Auto Save Entry</b>	<b>2</b>
How Auto Save Works	2
How Auto Save Helps Content Editors	3
Limitation of Auto Save	3
<b>Reference</b>	<b>3</b>
<b>Workflow Migration</b>	<b>4</b>
Step-by-Step Guide to Migrate Workflows	4
Exporting Workflow from the Source Stack	4
Importing Workflow into the Target Stack	5
Configure the Imported Workflow	5
Use Case for Workflow Migration	5
<b>Reference</b>	<b>6</b>



## Introduction

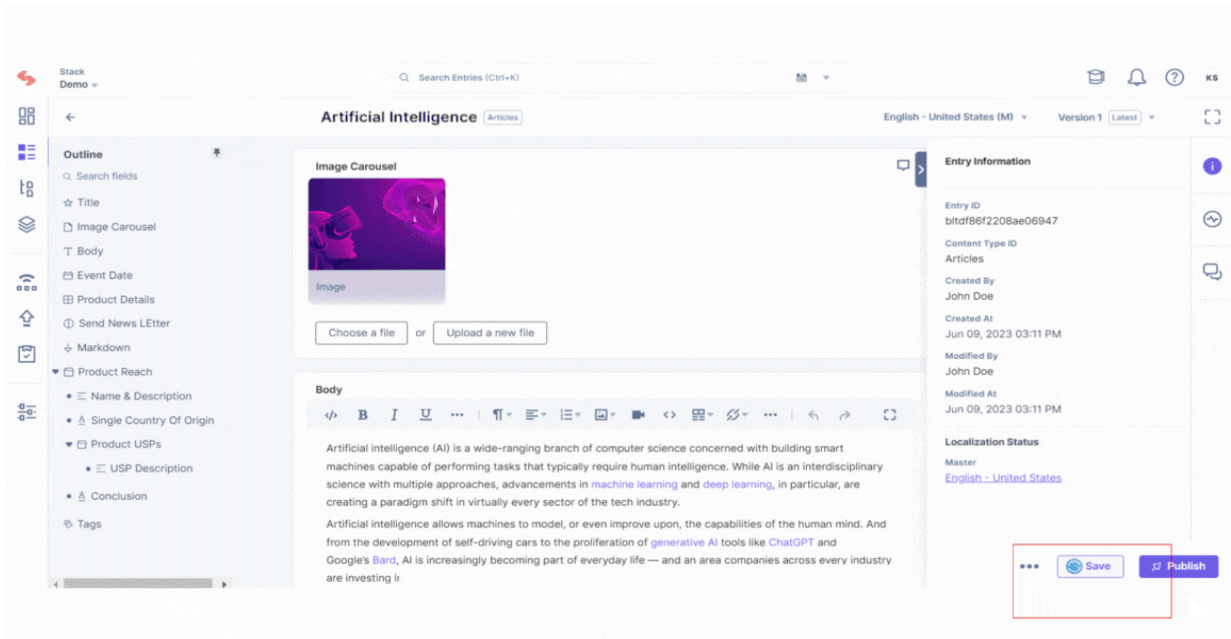
This document provides detailed information on two key features, that is **Auto Save Entry** and **Workflow Migration** in Contentstack's headless CMS. It explains how these features work, their benefits, limitations, and step-by-step implementation. The goal is to offer clear and structured guidance to help users understand and utilize these functionalities effectively.

## Auto Save Entry

Auto Save Entry feature in the Contentstack's entry editor automatically saves the entries at the regular intervals, eliminating the need for manual saving. This helps prevent the data loss in unforeseen instances such as internet disruption, accidental navigation, session timeouts or when users forget to save their work.

## How Auto Save Works

1. **Automation:** Auto Save activates during [creating](#) or [editing](#) the entries. System saves the data in the background without interrupting the user workflow.
2. **Save Frequency Indicator:** Entries are saved at the predefined intervals with timestamp showing when the entry was last saved, displayed below the **Auto Save** button at the bottom.
3. **Manual Publishing:** Auto Saved entries are saved as draft, and users can choose to manually publish the entries when it is ready.
4. **Version History:** This feature is integrated with the [versioning](#) system enabling users to review and restore the earlier version if necessary. Instead of creating a new version with each Auto Save, Contentstack updates a single Auto Save version, keeping it organized and tidy.



## How Auto Save Helps Content Editors

1. **Prevents Data Loss:** Auto save mitigates the risk of data loss during unforeseen situations. This ensures the reduced risk of losing data for content editors.
2. **Streamlined Workflow:** Content editors can create/edit the entries without the need of frequent saving, ensuring an uninterrupted and efficient workflow.
3. **Version Management:** Auto save, along with Contentstack's custom [version naming](#), helps editors to track changes over time, ensuring content accuracy and consistency.
4. **Improved Collaboration:** Auto save guarantees that all users working on the same entry in the stacks have access to the most recent version, preventing duplication of efforts.

## Limitation of Auto Save

1. **Auto Save Frequency Customization:** App users may not have the permission to configure the Auto Save intervals. The default interval will be fixed.
2. **Partial Saves:** Changes saved during Auto Save may not reflect the unfinished edits in the field if the user mistakenly overlooks it.



3. **Offline Mode:** Auto Save will not function in the offline mode. Changes made offline need manual save or synchronization upon reconnecting.

## Reference

- <https://payloadcms.com/docs/versions/autosave>
- [Contentstack Youtube](#)
- [Contentful Youtube Tutorials](#)
- [Contentstacks Docs](#)

## Workflow Migration

Workflow Migration functionality allows users to migrate the predefined [workflow](#) from one [stack](#) to another. It allows the corresponding stack's workflow to function without recreating it, thereby saving time and effort. This feature allows users to export workflows from an existing stack and import them to another stack and reuse the same workflow stages without having to start from scratch.



### Note:

- To export/import the workflow of a specific stack, ensure you have stack [ownership](#) or [Admin](#), or, [Developer](#) rights. Alternatively, you can proceed if the stack owner or admin has granted you edit access.
- [Content managers](#) and users with [custom roles](#) cannot access workflow migration settings but can use migrated workflows in entries once it is imported.
- Since workflows can be created for all or specific [content types](#), do ensure the compatibility between the source and target stack to avoid migration issues.

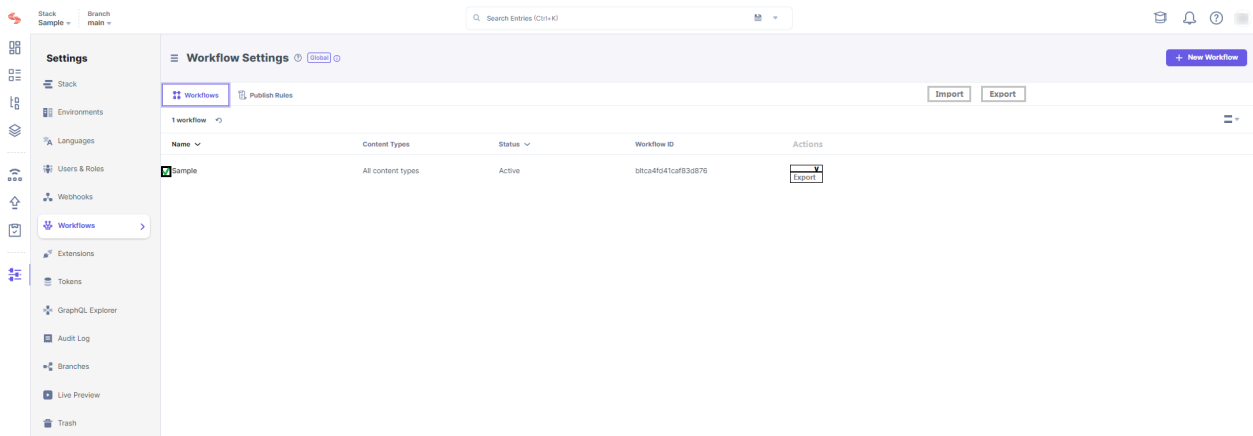
## Step-by-Step Guide to Migrate Workflows

To migrate a workflow in your stack, log in to your [Contentstack account](#) and perform the following steps:

### Exporting Workflow from the Source Stack

1. Go to the source stack, navigate to the "Settings" icon (press "S") on the left navigation panel.

2. Select Workflows (press "alt + F" for Windows OS, and "option + F" for Mac OS).
3. Under the Workflows tab, you will see a list of existing workflows. Locate the workflow you want to migrate and click on the respective vertical ellipsis under the Actions column.
4. From the dropdown menu that appears, select **Export**. Save the exported workflow file to your local system. Ensure the file format is compatible with the target stack.
5. Alternatively, you can select the checkbox for the specific workflow you want to export, which activates the 'Export' button. Click **Export** to initiate the workflow export. You can also export multiple workflows simultaneously.



## Importing Workflow into the Target Stack

1. Navigate to the Workflows section for the target stack via the Settings panel.
2. In the Workflows tab, click on the **+ New Workflow** button.
3. Click on the ellipses at the bottom of the page and select Import.
4. Alternatively click on the **Import** button at the top of the Workflow Settings page.
5. Upload the file exported from the source stack and verify it for any discrepancies.

## Configure the Imported Workflow

1. Update any stack-specific elements such as [adding stage transition and access rule](#) or [adding workflow superuser](#).
2. Activate the workflow by checking the **Enable Workflow** box.
3. Once the configuration is complete, click **Save** to finalize the migration.
4. Imported workflow will now be visible under the list of workflows.



## Use Case for Workflow Migration

1. Standardizing the workflow across environments (Staging, Development, and Production)

**Example:** QA can perform unit testing in a staging before moving the packaging to the production .

2. Workflow for onboarding new hires.

**Example:** Standard workflow can be migrated to the training stacks to helps new hires familiarize themselves with processes before working in the live environment.

## Reference

- [Import an Entry](#)
- [Export an Entry](#)
- [Setup Workflows and Publish Rules](#)