

Configuration Migration Best Practice Guide

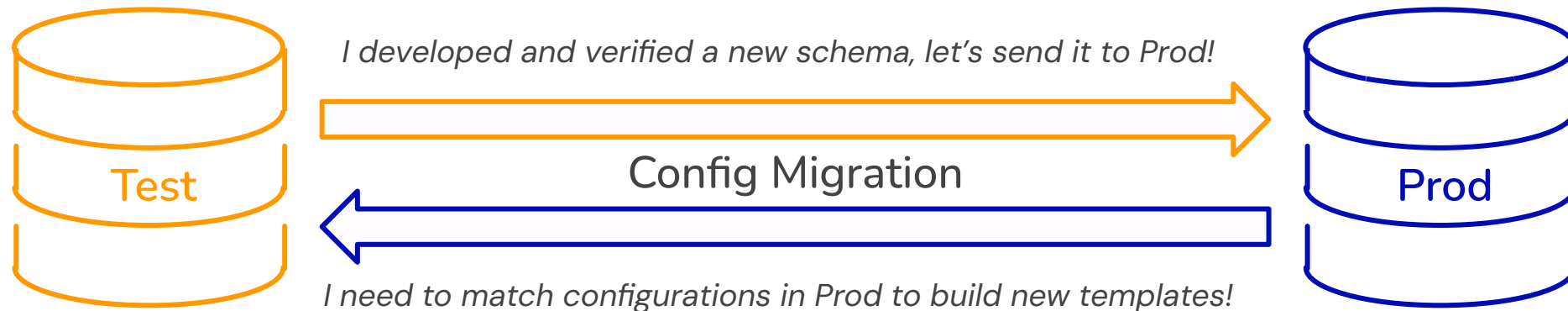


What is Config Migration?

We recommend having two separate environments when building Benchling objects:

- A test environment where you trial your schemas, templates, and other objects
- A production environment where you actually collect your data, with verified schemas

The Config Migration tools are there to help you easily move these configurations from one environment to another. If you need a test environment, reach out to your Benchling rep!



Use your test environment to develop and verify objects before moving them to production

Your production environment should have verified schemas and templates used to capture your data



What can be Migrated?

The list below is not exhaustive, but representative of the types of items that are supported.

May be migrated with the Tool

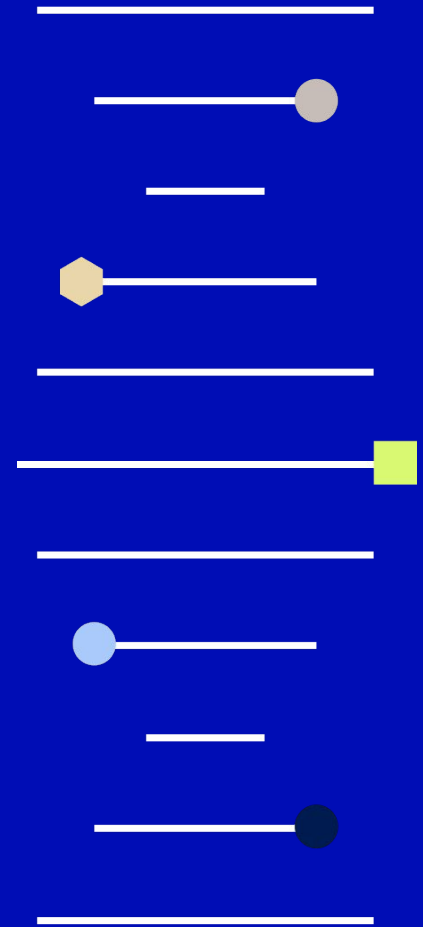
- Notebook Templates
- Notebook Sub-templates
- Template Collections
- All Schemas (Entity, Location, Container, Plate, Box, Workflow Task, Result, Run)
- Dropdowns

May **not** be migrated with the Tool

- Notebook Entries
- Specific Entity, Inventory, or Result Data
- Access Policies
- Template Collection Permissions
- Project/Folder Structure Hierarchy
- Teams, Users, Admins
- Application Provisioning
- Insights Dashboards
- Saved Searches

[Check our Help File for more information!](#)

Planning for Config Migration



Config Migration Best Practices: Before your Migration

Considerations	What to Prepare
Roles and Responsibilities	<ul style="list-style-type: none">● Identify Responsibilities<ul style="list-style-type: none">○ Who will validate configurations before and after○ Who will be performing the migration● Alert all End-Users of the Migration Switch-Over, as well as any downtime when data shouldn't be added
Documentation	<ul style="list-style-type: none">● Ensure any documentation tied to tenant configurations are updated
Configuration Preparation	<ul style="list-style-type: none">● Ensure you understand how all objects map to one another across tenants● Identify any changes to schema field types* so that you can change them later

(*) Schema field types cannot be changed via config migration and will need to be manually changed via the UI.

Schema Migration Considerations

- Schemas that were manually built in Prod and Test will require you to align those schemas as well as any fields associated with them
 - These alignments will be memorized after the initial sync
 - This process also means migrations between highly diverged tenants will take a greater amount of effort and time to align all diverged schemas
 - You can't undo this mapping! Do it correctly the first time

Entity schemas	Action ▾	1 to update ^
▼ Manually Built Schema	Update Manually Built Schema	▼
Schema fields		
Date	Update Date	▼
Month	Update Month	▼

Long Term Maintenance of Tenants

Discrepancies between Tenants can make migrations difficult, particularly with complex data models with lots of linked entities.



Schedule a cadence for performing checks on your tenants to ensure that all Benchling objects are closely aligned

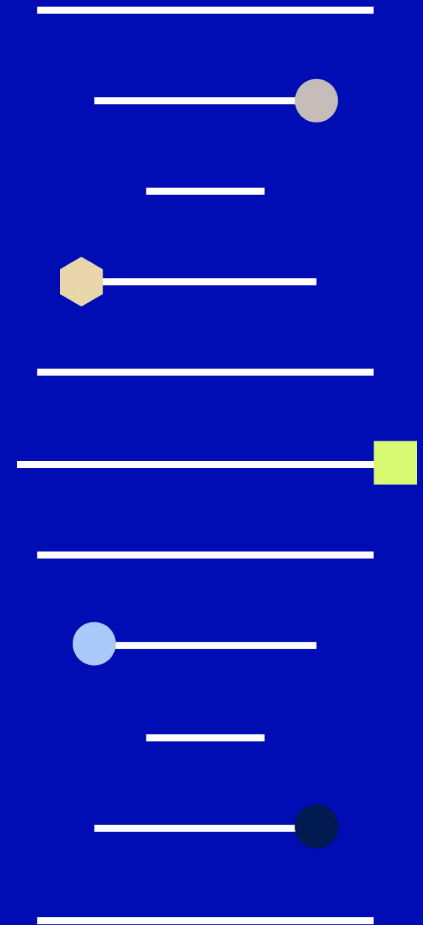


Perform limited migrations soon after changes to objects are made.



Create an SOP or Checklist for what routine checks must be performed for each migration

Config Migration Best Practices





Best Practices: Config Migration Management

Focus	Why?
Avoid Large Tenant Divergences	Regularly migrate all schema and template changes between tenants; Config migration is most difficult to use between tenants whose configurations have diverged significantly
Identify Export Warnings	Some export warnings do not block the export altogether, but still cause blocking issues on import. Take time to address these issues prior to migration. If you are unsure how to proceed, please contact support@benchling.com .
Avoid Repurposing Objects	<p>If you need an object for a new purpose, archive the old one and create a new one. Repurposing schema fields, dropdowns and dropdown options makes tenant alignment very hard, and in some cases impossible to do.</p> <p>E.g: You have a “Days of the week” dropdown that you no longer need and are looking to create a “Cell Type” dropdown. Instead of overwriting the dropdown, you should create a net new dropdown and archive the old one.</p>
Modifying Schema Field Types	<p>Schema fields types cannot be changed via config migration, so modifications to schema field type will require manual alignment across tenants.</p> <p>E.g. A Text field is being changed into a Number field, config migration will not overwrite this and will require modification after the migration</p> <p>Please refer to this article if you need more guidance on how to edit and manage schema fields.</p>



Best Practices: Config Migration Troubleshooting

Precaution	Why?
Download Migration Files only at the time of migration	Migration files expire seven days after export. A best practice is to export the file from the source tenant at the moment of the migration effort. Any Changes you make in the source tenant will also require you to re-download the migration files.
Take note of Schema dependencies for fields	<p>Warehouse Dependencies</p> <ul style="list-style-type: none">• The warehouse does not allow for duplicate values so ensure that warehouse names in your migration do not create duplicates. <p>Name Template Dependencies</p> <ul style="list-style-type: none">• Fields in naming templates can't be archived. Remove the field from the naming template/constraint in the destination or unarchive the field on the source. Then, re-export the file and continue the migration. <p>Result Dependencies</p> <ul style="list-style-type: none">• Result schema fields cannot be updated once data has been captured against them, so you cannot update dropdown fields to map to a new dropdown list.
Take note of Archived Entities	Archived entities can interfere in Migration of Templates - you may have to unarchive schemas or remove them from templates to be able to migrate again.
Keep an eye on Template Dependencies	Data cannot be migrated, so any @-mentions or prefilled structured tables in templates need to be re-done in your destination tenant.



Best Practices: Config Migration Error Messages

Error Type	Solution
General Import Error: "X cannot be imported"	It's possible that the migration file have expired, as they expire seven days after export.
Import error message on Schemas: "Another field of this schema has a similar name with the same warehouse column name. Please try another name."	The warehouse does not allow for duplicate values. Updating the warehouse name for the fields that are conflicting could solve the issue. Once these names are updated, re-export the file and continue the migration.
Import error message on Schemas: "Cannot archive field that is used in a name template or constraint."	A field cannot be archived while it's a part of a naming template or constraint. Remove the field from the naming template/constraint in the destination or unarchive the field on the source. Then, re-export the file and continue the migration.
Import error message on Schemas: "Schema field "X" cannot change dropdownld on a field that contains data."	Result schemas do not allow their fields to be updated once data has been captured against them. As a result, you cannot update a dropdown field to map to a new dropdown after data has been recorded. Options are to make a new result field or use the existing dropdown.



Best Practices: Config Migration Error Messages

Error Type	Solution
Import error message on Schemas: "Cannot add a required field once data has been received."	Result schemas do not allow their fields to be updated once data has been captured against them. As a result, you cannot update a field to be required after data has been recorded. Options are to make a new result field or use the existing dropdown as is.
Import error message on Templates or Sub-Templates: "Template X can be imported with changes. Replacing reference to Internal Link "XXX" with a placeholder."	@-mentions of users or objects within (sub)templates cannot be migrated. These should either be deleted prior to migration <i>or</i> replaced after migration. For the latter, there will be a highlighted "Placeholder" text in the imported (sub)template.
Import error message on Templates or Sub-Templates: "Template X cannot be imported. It depends on entity schema "Y", which is archived."	Try going to the source tenant and unarchiving entity schema "Y" or removing schema "Y" from the template. The migration file will need to be re-exported from the source to continue the migration.

For other error messages not shown here, please contact Benchling support via in-app chat or support@benchling.com to troubleshoot.

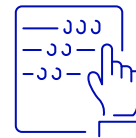
Resources for Configuration Migration



[Configuration Migration -
How To Guide](#)



[Benchling Support](#)



[Benchling Learning Labs -
How to Use the
Configuration Migration Tool](#)



[Benchling Community](#)