

# Guide to Removing Tabs from Property Flow Wizard

This document provides a step-by-step guide on how to remove a specific tab or step from a property flow in the Bhoomitali Real Estate Platform. This guide uses the removal of the "Commercial Details" tab from the "Commercial Rent" flow as a concrete example.

## Overview

Removing a tab from a property flow requires modifications to several key files in the wizard system. This is necessary to maintain proper navigation flow, ensure data integrity, and provide a consistent user experience.

## Files That Need Modification

1. **Flow Definition File:**  
`src/modules/owner/components/property/wizard/constants/flows.ts`
2. **Navigation Logic Hook:**  
`src/modules/owner/components/property/wizard/PropertyForm/hooks/useStepNavigation.ts`
3. **Form Content Component:**  
`src/modules/owner/components/property/wizard/PropertyForm/components/FormContent.tsx`
4. **Step Navigation Component:**  
`src/modules/owner/components/property/wizard/PropertyForm/components/StepNavigation.tsx`

## Step 1: Update Flow Definition

First, remove the step from the flow sequence in `flows.ts`:

```
// Before:
COMMERCIAL_RENT: [
  STEP_DEFINITIONS.details,
  STEP_DEFINITIONS.location,
  STEP_DEFINITIONS.rental,
  STEP_DEFINITIONS.commercial, // Remove this line
  STEP_DEFINITIONS.features,
```

```
    STEP_DEFINITIONS.review,  
    STEP_DEFINITIONS.photos  
]  
  
// After:  
COMMERCIAL_RENT: [  
  STEP_DEFINITIONS.details,  
  STEP_DEFINITIONS.location,  
  STEP_DEFINITIONS.rental,  
  STEP_DEFINITIONS.features,  
  STEP_DEFINITIONS.review,  
  STEP_DEFINITIONS.photos  
]
```

This change defines the new sequence of steps that will be used in the wizard.

## Step 2: Update Step Navigation Hook

Next, update the navigation logic in `useStepNavigation.ts`:

// Update the step sequences in the useMemo hook:

```
// Before:  
commercialRent: ['details', 'location', 'rental', 'commercial', 'features', 'review', 'photos'],  
  
// After:  
commercialRent: ['details', 'location', 'rental', 'features', 'review', 'photos'],
```

Also, in the same file, update the `getVisibleSteps` function to hide the removed tab for all property types:

```
// Before:  
(step.id === 'commercial' && !isCommercialRentMode)  
  
// After:  
(step.id === 'commercial') // Always hide the commercial step
```

## Step 3: Update Form Content Component

In `FormContent.tsx`, modify the handler for the removed tab to ensure it doesn't display content:

```
// Commercial Rent specific section - update the handler
if (currentStepId === 'commercial') {
  // This section is now not used in any flow
  return (
    <div className="py-6 text-center">
      <p className="text-muted-foreground">This section is not applicable for this property
type.</p>
    </div>
  );
}
```

This keeps the handler for backward compatibility but ensures it won't display actual content.

## Step 4: Update Step Navigation Component

Update the navigation logic in `StepNavigation.tsx` to reflect the new flow:

// Update the getFlowSteps function:

```
case 'commercial_rent':
  // Commercial Rent flow (updated to remove commercial details tab)
  return ['details', 'location', 'rental', 'features', 'review', 'photos'];
```

Enhance the `getPropertyType` function to better detect different property types:

```
const getPropertyType = () => {
  const urlPath = window.location.pathname.toLowerCase();

  if (urlPath.includes('flatmate')) {
    return 'flatmates';
  } else if (urlPath.includes('pghostel')) {
    return 'pghostel';
  } else if (urlPath.includes('commercial') && (urlPath.includes('sale') || urlPath.includes('sell'))) {
    return 'commercial_sale';
  } else if (urlPath.includes('commercial') && (urlPath.includes('rent') || urlPath.includes('lease'))) {
    return 'commercial_rent';
  } else if (urlPath.includes('coworking') || urlPath.includes('co-working')) {
    return 'coworking';
  }
```

```
} else if (urlPath.includes('land') || urlPath.includes('plot')) {  
  return 'land_sale';  
} else if (urlPath.includes('sale') || urlPath.includes('sell')) {  
  return 'sale';  
} else {  
  return 'rent';  
}  
};
```

## Step 5: Testing

After making these changes, thoroughly test the wizard to ensure:

1. The Commercial Rent flow no longer shows the Commercial Details tab
2. Navigation works correctly between the remaining steps
3. Data is saved properly without the removed tab
4. Existing properties created with the old flow can still be edited
5. All other property flows continue to work correctly

## Additional Considerations

### Version Management

Always update the version and modification timestamp in the file headers:

```
// src/modules/owner/components/property/wizard/constants/flows.ts  
// Version: 2.2.0  
// Last Modified: DD-MM-YYYY HH:MM IST  
// Purpose: Removed Commercial Details tab from Commercial Rent flow
```

### Backward Compatibility

When removing steps, consider these backward compatibility aspects:

1. **Existing Data:** Don't delete the component or validation logic for the removed step, as existing properties may still have that data
2. **API Endpoints:** Ensure backend APIs can still handle data structures that include fields from the removed step
3. **Form Submission:** Update any form submission logic to handle the absence of the removed step

## Documentation Updates

Update any related documentation to reflect the changes:

1. User manuals or help guides for property listing
2. Developer documentation about the wizard flow
3. Internal development team documentation about the flow structure

## Conclusion

Removing a tab from a property flow requires coordinated changes across multiple files to maintain the integrity of the wizard system. By following this guide, you can safely remove any tab from any property flow while ensuring a smooth user experience and data integrity.