


ADTU Table of Content Plan

Table of Content Feature for the Policy Page and the Worksite Representatives Page

About > Governance > Constitution and Bylaws

Constitution and Bylaws

View the complete ADTU Constitution and Bylaws PDF 

Jump to section

- Article I - Name
- Article II - Objectives
- Article III - Membership
- Article IV - Officers
- Article V - Election of Officers
- Article VI - Duties of the Executive
- Article VII - Representative Assembly (RA)
- Article VIII - Money, Fees and Levies
- Article IX - Meetings
- Article X - Sub-Locals
- Article XI - Committees
- Article XII - Ratification of Local Collective Agreements
- Article XIII - Ratification of Provincial Collective Agreement
- Article XIV - Audit
- Article XV - Amendments
- Article XVI - Conflict of Interest Policy

Revised May 21, 2024

Article I: Name

The name of the organization shall be the Alberni District Teachers' Union, hereinafter called the Union. The Union shall be a Local Union of the British Columbia Teachers' Federation.

About > Committees and Representatives > Worksite Representatives

Worksite Representatives

Lorem ipsum dolor sit amet consectetur. Mi id blandit bibendum euismod est nibh. In id nulla semper feugiat suspendisse nisi. Nulla lobortis mi neque lorem turpis in. Arcu id lectus consectetur ac amet habitasse convallis pellentesque. Leo vel elementum elit dignissim gravida semper scelerisque sed.

Jump to worksite location

- 8th Ave Learning Centre
- Alberni District Secondary
- Alberni Elementary
- Bamfield Community
- EJ Dunn Elementary
- John Howitt Elementary
- Maquinna Elementary
- School Board Office
- Student Support Services
- Tsumo-as Elementary
- Ucluelet Elementary
- Ucluelet Secondary
- Wickaninnish Community
- Wood Elementary

Worksite Contacts

8th Ave Learning Centre	
Jane Sawyer	member rep role #1
	member rep role #2

Method 1

Create a new Paragraph for the Body Content: **Long Body Content**

Long Body Content: Paragraph

- Chapter Content: Long formatted Text, unlimited entry

In the Twig template of this Paragraph, assign an id attribute to each Chapter Content

In Javascript, grab the unique ID in custom.js and render those ID into a table of content. Link will look like /about/policy#{id}

Advantage:

- The process of creating new content block is straightforward since the entire article can be created in one page

Disadvantage:

- Using Javascript to construct the Table of Contents may cause unexpected issues here and there
- Not very intuitive for future editing - client would have to navigate to the page that contains the list through multiple layers of Drupal UI

Method 2

In custom.js, grab the original PDF (currently hosted here:

<https://adtu.ca/sites/default/files/news/ADTU%20Constitution%20Revised%20May%202021%202024.pdf>),

figure out a way to parse the PDF file and insert it in the HTML structure when the user enters the policy page.

Potential ways of parsing the PDF file:

- use PDF.js, <https://mozilla.github.io/pdf.js/>
- use generative AI's API

<https://chatgpt.com/share/684343c4-81f0-800c-98b4-a733f3146664>

Advantage:

Most convenient for the client - assuming the ADTU has to have a PDF copy of the policy, they don't have to re-create the policy again in Drupal with this method. This saves time and avoids errors

Disadvantage

This is the most technically difficult method - converting PDF with a complex hierarchy could be very tricky and may have inconsistent results

Method 3

Create a new Content Block Type for each chapter of the article and use a View to display the article and the Table of Contents.

Structure:

Long Article: View

- Fields from **Article Chapter: Content Block**
 - For Policy, just a Long formatted text may be enough
 - Or Policy Title (unformatted short text) and Policy Body (Long formatted text)
 - For Worksite Representatives, use a different block structure
 - Use Custom Text field, and assign a unique ID to each chapter. The ID should be from one of the fields

Long Article Table of Contents: View

- Fields from **Article Chapter: Content Block**
 - Fetch the specific field used for chapter ID in **Long Article: View** and use Custom Text field to show it as a link in the Table of Content

```
<a href="#{{ field_title }}">{{ field_title }}</a>
```

Advantage

- Easy for the client to edit since View can provide an edit link right beside the content on the website
- Easy and stable to implement since it mainly uses native Drupal technology

Disadvantage

- The process of creating or adding new content is not very intuitive compared to Paragraph approach as the client needs to create each chapter separately
- Content Block page may become messy since lots of new content blocks will be created

Conclusion

If we can figure out how to stably parse and output PDF contents, **Method 2** would be the best solution as it only requires the client to update the Policy once. Otherwise, **Method 3** would be the best alternative in my opinion because minor updates from the client are likely to be the norm for the policy document, and Method 3's easier editing flow will better suit the situation.

One other note is that we might need to consider using **Method 3** for pages other than the policy page that requires a table of content (such as the Worksite Representatives) in order to keep things consistent.