



Help and Information Page



Access Levels

These are the access levels of access codes that are displayed on the editor page of each resolution.

Holder	Level	Description
Everyone (Delegates)	DE	This level means that there is no specific access restriction and the resolution can be viewed by anyone with the token. No access code is needed for this level.
The Approval Panel	AP	Editing is restricted to the Approval Panel. Resolutions are restricted to this access level when they are in stage #2 .
The Formal Clearing	FC	Editing is restricted to the Formal Clearing. Resolutions are restricted to this access level when they are in stage #3 .
The Secretariat	SC	The Secretariat enters its access code in stage #4 to signify that the resolution is being printed and is to be advanced to stage #5 . It doesn't have editing access though.
Chairs	CH	Chairs always have unrestricted access to resolutions. This overrides other restrictions, such as the levels FC and AP. This is the maximum level found on a resolution. Chairs can use LiveView when a resolution is being debated.
Administration	MA	This level represents unrestricted access to all available functions of the website including LiveView. This also includes changing access levels of specific codes or giving out new access codes.

A Formatting Advice and Special Characters

This section will inform you about the formatting rules and special character restrictions in resolutions. This is about the formal and syntactical aspects only; for content advice, see the [Content Guidelines](#). Ask your Chair for assistance with formatting and if you are unsure of how to achieve a certain formal element of your resolution. Look in `latexGenerator.js` and `editorInit.js` for the code.

A basic clause consists of a phrase and a content body, optionally followed by a single enumeration. The enumeration items behave similarly to a regular clause, except for the lack of a phrase. Additionally, a single block of extended content can follow the enumeration. If you're trying to create a clause with more than two content blocks, more than one enumeration or a different ordering of these elements, restructure your clause to fit the formatting restrictions. This will also increase the legibility and semantic separation of your resolution.

Special Syntax

The editor will attempt to make syntax become obvious to the user by providing interface elements that facilitate the different aspects of writing and formatting a resolution. However, in some special cases it is unavoidable to use special syntax to achieve some formatting types within clause content. Use this special formatting sparingly, only when necessary and not as a generic stylistic device. Special syntax only applies to the body of clauses and enumeration items and phrases will not be formatted.

Styling	Examples	Syntax	Explanation
Italics	<i>A/RES/69/154,</i> <i>UNESCO</i> <i>Declaration...</i>	<code>*A/RES/69/154*</code> , <code>*UNESCO Declaration...*</code>	Insert Asteriks <code>*</code> around the text you want to be rendered in <i>italics</i> . Apply italics to emphasise resolution identifiers like <i>A/RES/69/154</i> or names of formal documents, bodies or institutions.
Single character subscript	H ₂ O, CO ₂	<code>H_2O</code> , <code>CO_2</code>	Characters preceded by an underscore <code>_</code> will be rendered as subscript. Use this to correctly display molecular formulas.

Styling	Examples	Syntax	Explanation
Multiple character subscript	C ₆ H ₁₂ O ₆	C_6H_12 O_6	Use the underscore character <code>_</code> , the text you want to be rendered as a subscript followed by the pipe character <code> </code> to correctly display molecular formulas that involve subscripts consisting of multiple characters.
Single character superscript	Na ⁺	Na^+	Characters preceded by an circumflex <code>^</code> will be rendered as superscript.
Multiple character superscript	7 th of May, 42 nd edition	7^th of May, 42^nd edition	Use the circumflex character <code>^</code> , the text you want to be rendered as a superscript followed by the pipe character <code> </code> to correctly display ordinal numbers.

Whitespace

Please refrain from inserting any abnormal whitespace into clause text areas or any other input field. The editor will most likely remove all unnecessary whitespace before rendering the resolution. Line breaks, tabs and more obscure whitespace characters will be removed from clauses in order to preserve formatting and aesthetics in the rendered document. Do not use tabs for indentation of sub clauses and instead use the integrated button, that appears in the clause when you click the edit button or start typing, to create sub clauses.

Special Characters

Some characters have a special meaning and are removed or transformed by the editor and the renderer. Try to find another way of expressing the meaning of a special character if it doesn't have the expected purpose in the editor. Phrases will be stripped of all non-alphanumeric characters. If any character is observed to behave differently than described here or produces an error, please immediately contact your Chair and IT-Management!

Character(s)	Explanation
<code>tab</code> , <code>new line</code> , <code>carriage return</code>	See the above section on Whitespace . All abnormal whitespace will be removed.
\$	The dollar sign is reserved for internal use. Please spell out names of currencies instead of using shorthand symbols. This recommendation applies to all currencies, in particular those which don't use the dollar sign.
%	The percent sign is a reserved character. Spell out the percent sign as "per cent".
#	Instead of the pound sign, use ordinal numbers to express counted items. See Special Syntax
&	The ampersand is used internally as a reserved character but can still be used where required for the correct representation of names or similar descriptors.
\, {, }, ~	These miscellaneous reserved characters should not be used within resolutions in general and will be automatically removed. This includes the tilde ~ character.
` , ' , ’	Detached backticks ` , detached forwardticks ' and regular keyboard apostrophes ’ will be normalized to the typographically correct apostrophe '. <i>Letters</i> with diacritical marks are allowed and won't be modified. Do not use diacritical marks as apostrophes or quotation marks, as they will most likely be removed.
" , “ , ” , ‹ , › , « , »	All quote-like characters are automatically converted into correctly formatted and placed typographic quotation marks “ ” and “ ”. Inner quotes will not be modified or correctly formatted and therefore their use is not recommended, as they may result in unpredictable formatting.
, , : , ; , .	Do not end your clauses or enumeration items with resolution specific punctuation. The renderer will reliably and correctly insert formal punctuation where necessary.

Character(s)

Explanation

`□, ^, *, |`

See [Special Syntax](#). These characters are used to denote the application of special styling in the rendered document. Using them for a different purpose will produce unexpected results.

`ö, é, ì`

If you need to represent names in their romanized but still foreign form, use the letters with the different marks on them instead of adding combining diacritical marks, as they will be removed and are unsupported by the renderer.

`other`

Refrain from using characters that are not listed in this list of allowed characters. Additionally the most common letters with diacritical marks in the unicode range `U+00C0–U+024F` are supported as well. They can be viewed [here](#), [here](#) and [here](#).

```
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
* _ ^ | & ' " - . , ( ) / +
```

Handy Textbox

Use this [handy textbox](#) for whatever you want. It's meant to be used as a temporary text area in which you can type amendments before copying them into editor's amendment interface to debate them.



Attribute States

No special state The resolution is in its original state and doesn't have any additional saving or advancement restrictions. This is the state that all resolutions are created with.

Read-only This resolution is read-only and can't be edited without administrative privileges. The editor is displayed and can be used but automatic and user invoked saving is restricted until this flag is unset.

Non-advanceable This resolution cannot be advanced without administrative privileges and will stay at this stage until this flag is unset. The editor can still be used by authorized users though.

Static This resolution cannot be advanced or edited without administrative privileges and will stay in this exact state until this flag is unset. The editor can be used if authorized to do so but changes won't be saved. This attribute combines *Read-only* and *Non-advanceable*.



Resolution Workflow Stages

Initial Creation #0

This resolution has not been opened with the resolution token yet. The token will be released back into the system if the resolution remains unsaved at this stage for longer than 24 hours.

Advance to next stage by: Saving the resolution

Delegate Writing #1

This resolution is being worked on by the Delegate and has been opened with the resolution token. The resolution token is now bound to this saved resolution. The Delegate has to have their Chair enter their access code to approve the written resolution by the Chair.

Advance to next stage by: Entering a Chair access code for approval by the Chair

Approval Panel Correcting #2

The Approval Panel is correcting this resolution. Only Chairs, who always have access, and the Approval Panel can edit this resolution. After having entered the Approval Panel access code to open the editor, spelling and grammar mistakes in the resolution are corrected. Then a button is pressed to approve and advance this resolution to the next step.

Advance to next stage by: Entering an AP or Chair access code for approval by the Approval Panel

Formal Clearing Viewing #3

The Formal Clearing Office is looking at this resolution and determining its fulfillment of the formal constraints. The Formal Clearing approves the resolution following the same scheme as the Approval Panel; a button is pressed by the Formal Clearing to approve and advance the resolution.

Advance to next stage by: Entering an FC or Chair access code for approval by the Formal Clearing Office

Waiting to Print #4

This resolution has completed its validation process and will soon be printed by the Secretariat. From this point on only Chairs have editing access to this resolution.

Advance to next stage by: Printing the resolution and entering an SC (Secretariat) or Chair access code to signify this action

Waiting to Debate #5

The Secretariat has printed this resolution for the committee and it will be debated sometime after the printed copies have been distributed to the Delegates.

Advance to next stage by: Entering a Chair access code to begin the debate

In Debate #6

This resolution is being actively debated by its committee and will frequently change because of this. LiveView is enabled for this resolution during this stage so that this resolution can be accessed by another device for presentation.

Advance to next stage by: Entering a Chair access code to end the debate and finalize the resolution

Finished #7

The final stage of the resolution has been reached. Congratulations! The corresponding committee has debated and finalized the content of this resolution. It may still be edited for use in the resolution booklet though.

Unknown #?

Could not determine the status of this resolution. There seems to have been an error in the system. Please report this immediately as it may negatively affect the whole resolution workflow.

Please note that users with the access level MA have unrestricted access to all functions of the editor in all stages. See [Access Levels](#)



Resolution Structure Format Definition

Click [here](#) to view the JSON representation of the definition object of the resolution structure format.

Made for MUNOL

Created in need of a better way of formatting and distributing resolutions for MUNOL.

Contact

Repository
File a Bug
Contribute
MUNOL
MUNOL-App

Stack

Material Design (+Icons)
Materialize
jQuery
Pug
node.js
GitHub

