ACCORD PROJECT

Google Summer of Code 2021

Project: CiceroMark <->DOCX(OOXML)

Project Mentors: Aman Sharma, Dan Selman, Tom Brooke

About Me

Name: Kushal Kumar
Gender: Male
Github: k-kumar-01
Linkedin: kushal-kumar-979575170
Slack Handle: Kushal Kumar
Email: kushalkumargupta4@gmail.com
Alternative Mail: kushal.kumar.mat19@itbhu.ac.in
Contact Number: +91 - 6388828288
Alternative Contact Number: +91-9695585870
Country: India
Timezone: UTC + 5:30

Intro

I am Kushal Kumar, a sophomore currently pursuing Mathematics and Computing(5-year program) from the Indian Institute of Technology - BHU (Varanasi), India. I expect to graduate in the year 2024.
I got familiarized with computer programming in class 11th. However, my interest rose during the first year when I gave a hand in web development. I started exploring the field and made some projects with HTML, CSS, and Javascript. After gaining some experience, I started learning React in frontend and Node, Express, and MongoDB in the backend thus moving to a path of MERN stack developer.

Apart from programming in my leisure hours, I play computer games.

Education

Degree: Integrated Dual Degree(IDD - B.Tech + M.Tech): 5-year program
Branch: Mathematics and Computing
Current Year: 2nd

Expected Graduation Year: 2024
Institute: Indian Institute of Technology - BHU (Varanasi)

My Skills

Level	**Novice**	**Intermediate**	**Advanced**
Skill	Tailwind, GatsbyJs, Graphql, Django, C, Redux, Webpack	SCSS, Typescript, Express, Node, C++, Python, Git, REST API's	HTML, CSS, Javascript, Bootstrap, React, NextJs, jQuery

CONTRIBUTIONS

Accord Project

I have contributed to Accord Project in the following ways:

In accordproject/web-components

Issue	**Pull Request**	**Status**
#219	#222	Merged
#218	#225	Merged
#224	#227	Merged
#228	#229	Merged
#185	#236	Merged
#242	#251	Merged
#158	#252	Merged
#137	#277	Merged
#318	#319	Merged
#301	#303	Merged

#40	#312	Open
#30	#289	Open
#320	#330	Open

In accordproject/cicero-word-add-in

Issue	Pull Request	Status
#37	#38	Merged
#25	#44	Open

Besides making the above PRs, I have opened some issues also. They are:
In accordproject/web-components
In accordproject/cicero-word-add-in

Other Orgs

Apart from contributions to the Accord Project, I have also contributed to the following:

elastic/eui

Issue	Pull Request	Status
#4239	#4383	Merged
#4330	#4452	Merged
#4183	#4465	Merged

navidrome/navidrome

Issue	**Pull Request**	**Status**
#816	#826	Merged
#827	#831	Merged

PROJECT DETAILS

Currently for non-technical users, using the accord project is a bit difficult. It involves parsing and drafting using command line tools which may cause discomfort to the users. The word add-in which was developed last year has removed this inconvenience a lot. However, the add-in still has much more development to go through. This project aims at that.

The project can be classified into two mini-projects of their own.

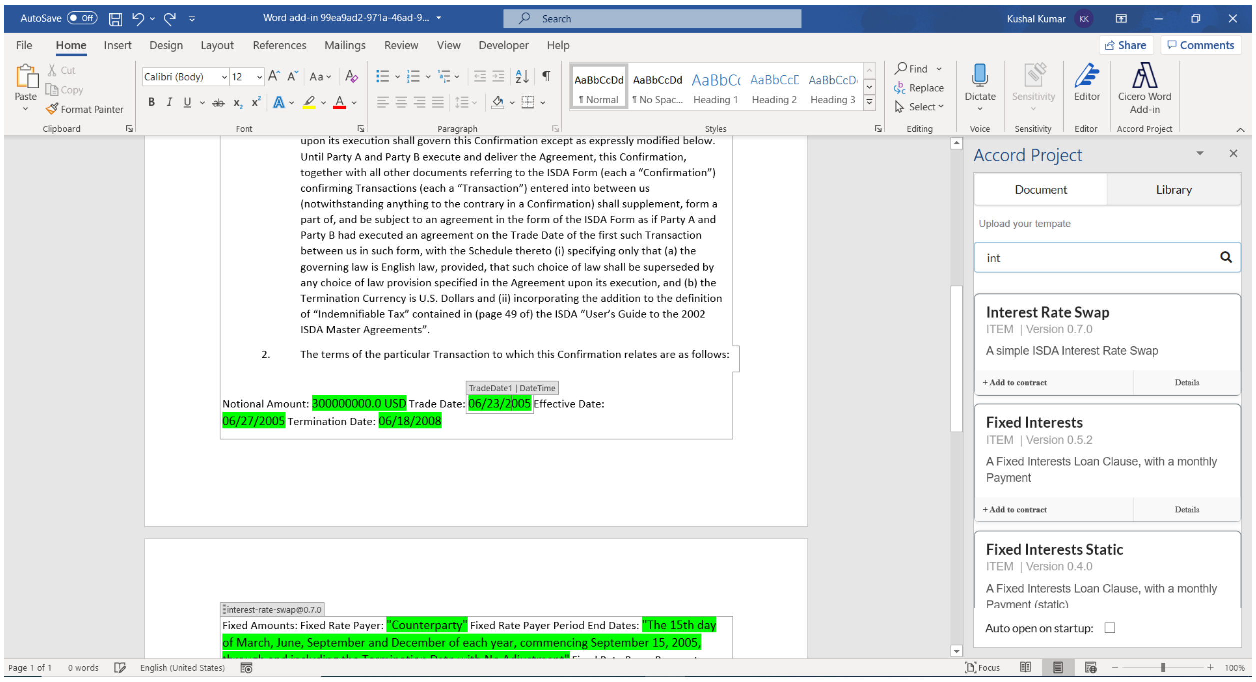
CiceroMark to DOCX (OOXML)

It involves converting the CiceroMark which is a JSON object into OOXML.
Conversion of the above itself can be categorized into the following:

➢ Improving the existing features

Currently when a variable is inserted into the add-in, then the editing of a variable can be done freely i.e. there is no type checking to prevent the insertion of wrong data. An example image to depict the following.

Initial Rendering



upon its execution shall govern this Confirmation except as expressly modified below. Until Party A and Party B execute and deliver the Agreement, this Confirmation, together with all other documents referring to the ISDA Form (each a "Confirmation") confirming Transactions (each a "Transaction") entered into between us (notwithstanding anything to the contrary in a Confirmation) shall supplement, form a part of, and be subject to an agreement in the form of the ISDA Form as if Party A and Party B had executed an agreement on the Trade Date of the first such Transaction between us in such form, with the Schedule thereto (i) specifying only that (a) the governing law is English law, provided, that such choice of law shall be superseded by any choice of law provision specified in the Agreement upon its execution, and (b) the Termination Currency is U.S. Dollars and (ii) incorporating the addition to the definition of "Indemnifiable Tax" contained in (page 49 of) the ISDA "User's Guide to the 2002 ISDA Master Agreements".

2. The terms of the particular Transaction to which this Confirmation relates are as follows:

TradeDate1 | DateTime

Notional Amount: 300000000.0 USD Trade Date: 06/23/2005 Effective Date: 06/27/2005 Termination Date: 06/18/2008

interest-rate-swap@0.7.0

Fixed Amounts: Fixed Rate Payer: "Counterparty" Fixed Rate Payer Period End Dates: "The 15th day of March, June, September and December of each year, commencing September 15, 2005,

Accord Project

Document | Library

Upload your tempate

int

Interest Rate Swap
ITEM | Version 0.7.0
A simple ISDA Interest Rate Swap
+ Add to contract | Details

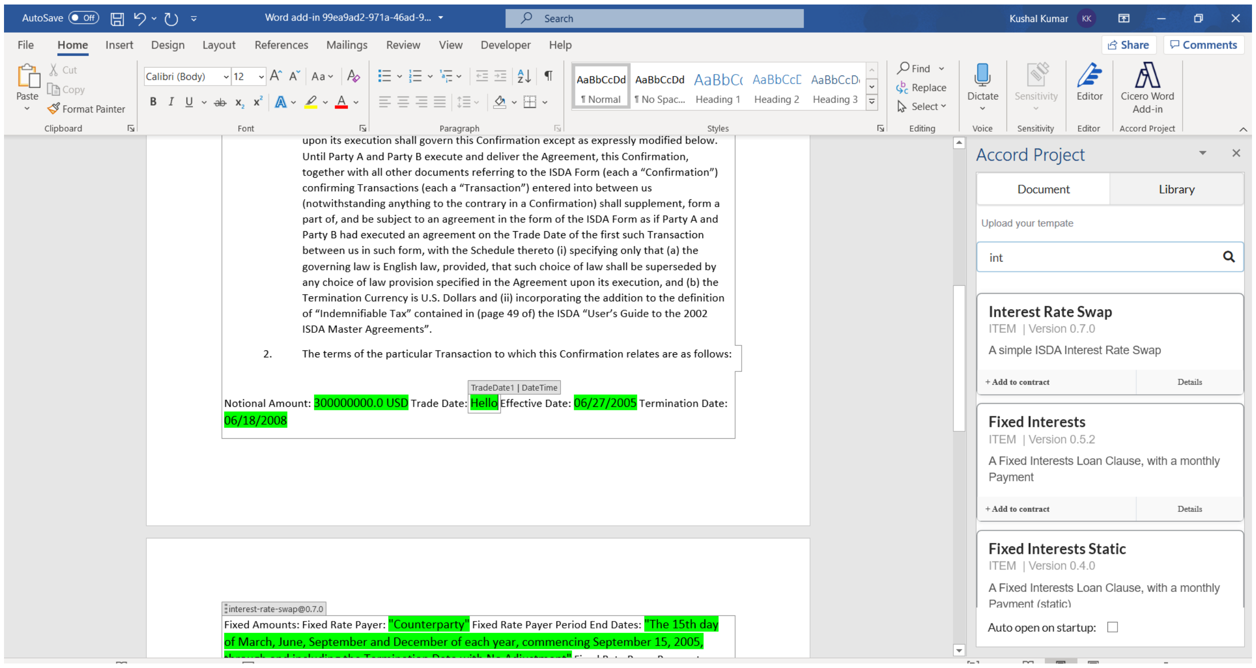
Fixed Interests
ITEM | Version 0.5.2
A Fixed Interests Loan Clause, with a monthly Payment
+ Add to contract | Details

Fixed Interests Static
ITEM | Version 0.4.0
A Fixed Interests Loan Clause, with a monthly Payment (static)

Auto open on startup: ☐

AutoSave Off
Word add-in 99ea9ad2-971a-46ad-9...
Search
Kushal Kumar
File Home Insert Design Layout References Mailings Review View Developer Help
Share Comments
Clipboard Font Paragraph Styles Editing Voice Sensitivity Editor Accord Project
Cicero Word Add-in
Page 1 of 1 0 words English (United States) Focus 100%

Changed Rendering



upon its execution shall govern this Confirmation except as expressly modified below. Until Party A and Party B execute and deliver the Agreement, this Confirmation, together with all other documents referring to the ISDA Form (each a "Confirmation") confirming Transactions (each a "Transaction") entered into between us (notwithstanding anything to the contrary in a Confirmation) shall supplement, form a part of, and be subject to an agreement in the form of the ISDA Form as if Party A and Party B had executed an agreement on the Trade Date of the first such Transaction between us in such form, with the Schedule thereto (i) specifying only that (a) the governing law is English law, provided, that such choice of law shall be superseded by any choice of law provision specified in the Agreement upon its execution, and (b) the Termination Currency is U.S. Dollars and (ii) incorporating the addition to the definition of "Indemnifiable Tax" contained in (page 49 of) the ISDA "User's Guide to the 2002 ISDA Master Agreements".

2. The terms of the particular Transaction to which this Confirmation relates are as follows:

TradeDate1 | DateTime

Notional Amount: 300000000.0 USD Trade Date: Hello Effective Date: 06/27/2005 Termination Date: 06/18/2008

interest-rate-swap@0.7.0

Fixed Amounts: Fixed Rate Payer: "Counterparty" Fixed Rate Payer Period End Dates: "The 15th day of March, June, September and December of each year, commencing September 15, 2005,

Accord Project

Document | Library

Upload your tempate

int

Interest Rate Swap
ITEM | Version 0.7.0
A simple ISDA Interest Rate Swap
+ Add to contract | Details

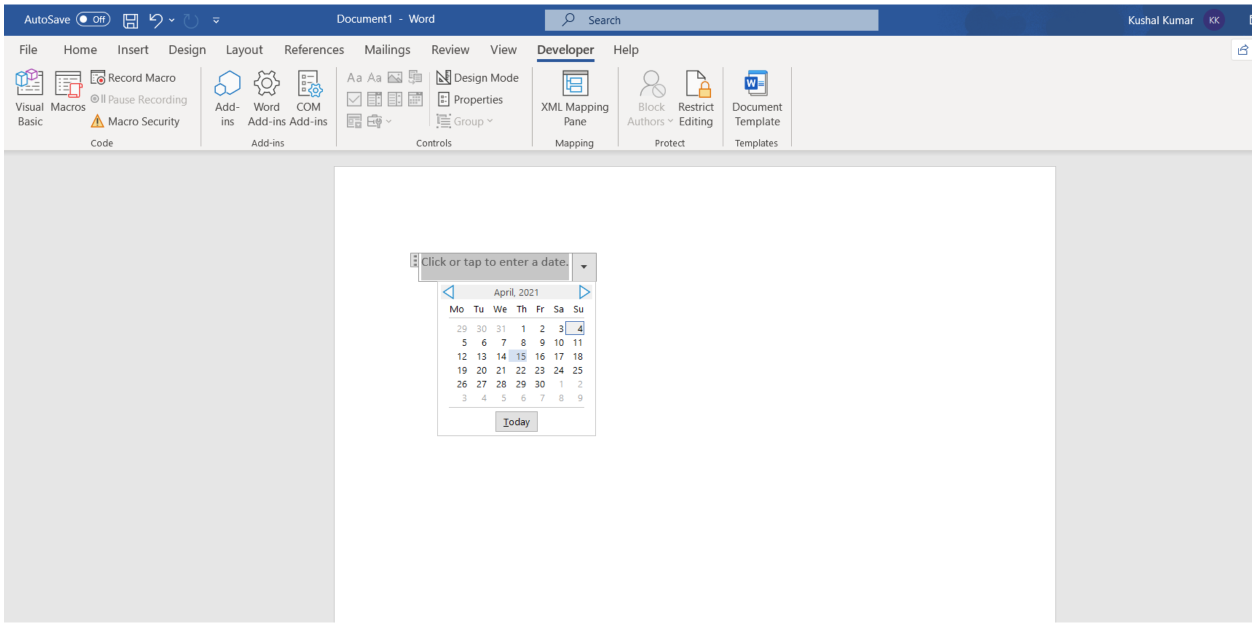
Fixed Interests
ITEM | Version 0.5.2
A Fixed Interests Loan Clause, with a monthly Payment
+ Add to contract | Details

Fixed Interests Static
ITEM | Version 0.4.0
A Fixed Interests Loan Clause, with a monthly Payment (static)

Auto open on startup: ☐

AutoSave Off
Word add-in 99ea9ad2-971a-46ad-9...
Search
Kushal Kumar
File Home Insert Design Layout References Mailings Review View Developer Help
Share Comments
Clipboard Font Paragraph Styles Editing Voice Sensitivity Editor Accord Project
Cicero Word Add-in

As visible, we can change the DateTime to a simple hello string.
It would be beneficial to render them as DateTimePickers like:



The OOXML for this is also available as stated here.
Sample OOXML when the date is set to 05-04-2021 (dd-mm-yy)

```
<w:sdt>
  <w:sdtPr>
    <w:id w:val="963392120"/>
    <w:placeholder>
      <w:docPart w:val="DefaultPlaceholder_-1854013437"/>
    </w:placeholder>
    <w:date w:fullDate="2021-04-05T00:00:00Z">
      <w:dateFormat w:val="dd-MM-yyyy"/>
      <w:lid w:val="en-IN"/>
      <w:storeMappedDataAs w:val="dateTime"/>
      <w:calendar w:val="gregorian"/>
    </w:date>
  </w:sdtPr>
  <w:sdtContent>
    <w:p w14:paraId="680B112F" w14:textId="28945DE5" w:rsidR="00A700DD"
w:rsidRPr="00353D8C" w:rsidRDefault="00353D8C" w:rsidP="00353D8C">
      <w:r>
        <w:t>05-04-2021</w:t>
      </w:r>
    </w:p>
  </w:sdtContent>
</w:sdt>
```

Currently, `org.accordproject.ciceromark.Formattedvariable` are not rendered.
It is similar to `org.accordproject.ciceromark.Variable` with the difference that Formatted Variable has an extra format field.

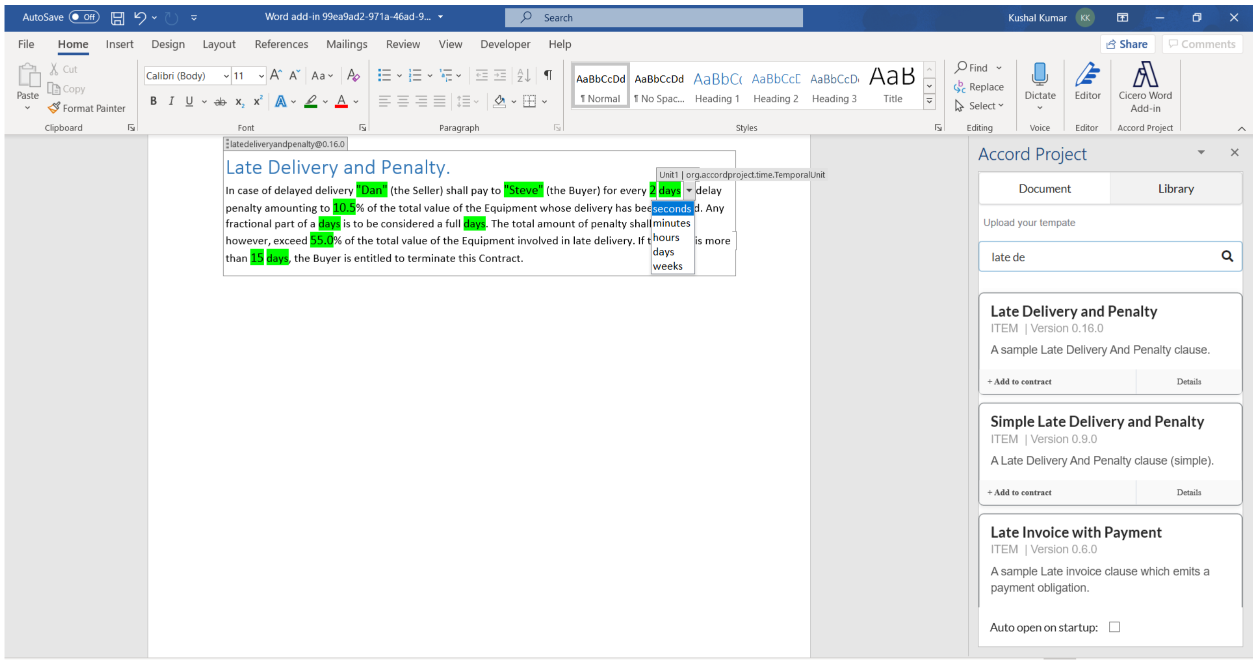
```
    $class: "org.accordproject.ciceromark.FormattedVariable"
    elementType: "org.accordproject.money.MonetaryAmount"
    format: "0,0.00 CCC"
    name: "loanAmount"
    value: "100,000.00 USD"
```

So we can have a function say `renderFormattedVariable()` which will format the variable according to the field given.

The add-in is also unable to render `org.accordproject.ciceromark.Enumvariable`.
It is also similar to `org.accordproject.ciceromark.Variable` with the difference that we have an enumValues field. This field can be created using dropdown content controls using the developer tab in word.

```
    $class: "org.accordproject.ciceromark.EnumVariable"
    elementType: "org.accordproject.time.TemporalUnit"
enumValues: Array
  0: "seconds"
  1: "minutes"
  2: "hours"
  3: "days"
  4: "weeks"
    name: "fractionalPart"
    value: "days"
```

A sample implementation of the above enum approach.



Late Delivery and Penalty.

In case of delayed delivery "Dan" (the Seller) shall pay to "Steve" (the Buyer) for every 2 days delay penalty amounting to 10.5% of the total value of the Equipment whose delivery has bee d. Any fractional part of a days is to be considered a full days. The total amount of penalty shall however, exceed 55.0% of the total value of the Equipment involved in late delivery. If t is more than 15 days, the Buyer is entitled to terminate this Contract.

Late Delivery and Penalty
ITEM | Version 0.16.0
A sample Late Delivery And Penalty clause.

Simple Late Delivery and Penalty
ITEM | Version 0.9.0
A Late Delivery And Penalty clause (simple).

Late Invoice with Payment
ITEM | Version 0.6.0
A sample Late invoice clause which emits a payment obligation.

For ensuring that the variables match a specific data type like *Integer* or *Double*, we can create a function like `checkConsistency(value, type)`. If the match occurs then we will update all the variable instances. Else, we will update the value of the variable from which the variableChangeListener was called to the previously accepted value.
Note: I was able to solve the problem in my local machine.
Pseudo code snippet:
components/TemplateLibrary/index.js

```
// using a state to store the values of different variables and their
type
  const [variableValues, setVariableValues] = useState({});

const setup = async (ciceroMark, template) => {
    .
    .
    .
    let countVal = {...variableValues};
    for (const variableText in counter) {
        for (let index=1; index<=counter[variableText].count; ++index) {
          attachVariableChangeListener(

titleGenerator(`${variableText.toUpperCase()[0]}${variableText.substring
(1)}${index}`, counter[variableText].type),
            countVal
          );
        }
      }
```

utils/AttachVariableChangeVisitor.js

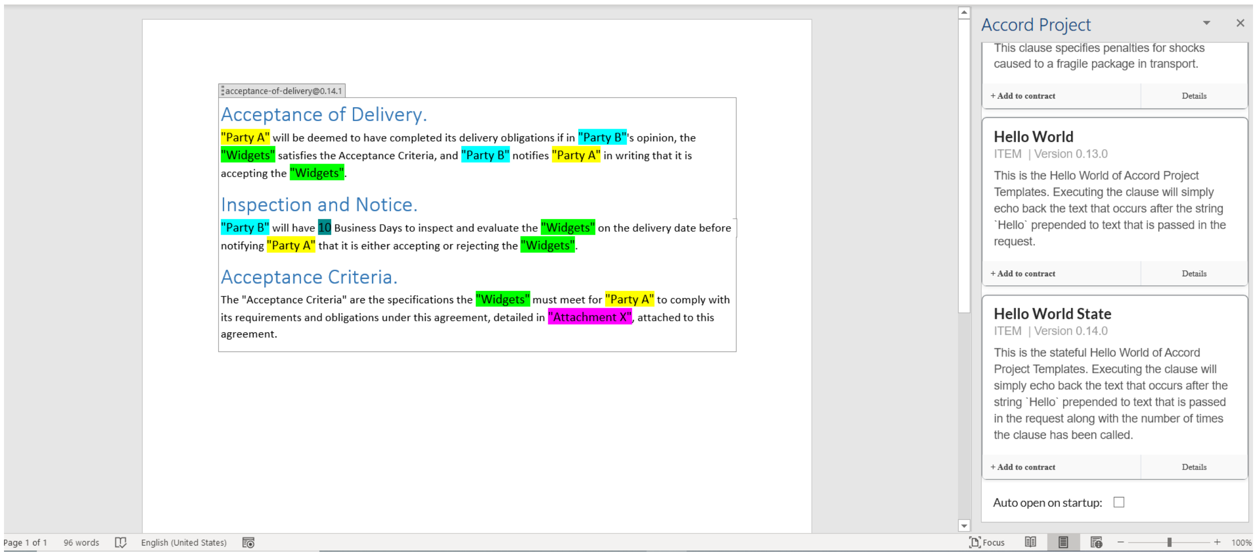
```
const attachVariableChangeListener = (title, values) => {
  Office.context.document.bindings.addFromNamedItemAsync(title,
Office.CoercionType.Text, { id: title }, res => {
    if (res.status === Office.AsyncResultStatus.Succeeded) {
      res.value.addHandlerAsync(Office.EventType.BindingDataChanged,
e=>variableChangeListener(e, values), res => {

const variableChangeListener = (event, values) => {
    ...
    binding.getDataAsync(result => {
        // The text typed by user to change it
        const data = result.value;
        // if data is invalid
        if(data==='invalid'){
          // update the current value to the value stored in the values
state
          // do not update the other values
        }else{
          // run the code as written and update the state values also
        }
```

To check if variable value is consistent:

```
const checkConsistency = (value, type) => {
  if(type==='Long'){
    if(Number.isInteger(value) && value<=Number.MAX_SAFE_INTEGER &&
value>= Number.MIN_SAFE_INTEGER){
      return true;
    }else{
      return false;
    }
  }
  if(type==='Double'){
    if(Number.isInteger(x)){
      return false;
    }else{
      return false;
    }
  }
  if(type==='Integer'){
    return Number.isInteger(value);
  }
  return value ? true : false;
};
```

The above code snippet is a very rough estimation of how a function can be written to check if a variable value is consistent with the type.
We can also add a third argument to the function which will also check for the formatting of the variable using the regex.
Also, documentation regarding the same including some gifs can be done for a proper explanation.

Currently, all the variables in the add-in are displayed in the same color. It would be better if the variables which are linked are displayed in different colors.


acceptance-of-delivery@0.14.1

Acceptance of Delivery.

"Party A" will be deemed to have completed its delivery obligations if in "Party B"'s opinion, the "Widgets" satisfies the Acceptance Criteria, and "Party B" notifies "Party A" in writing that it is accepting the "Widgets".

Inspection and Notice.

"Party B" will have 10 Business Days to inspect and evaluate the "Widgets" on the delivery date before notifying "Party A" that it is either accepting or rejecting the "Widgets".

Acceptance Criteria.

The "Acceptance Criteria" are the specifications the "Widgets" must meet for "Party A" to comply with its requirements and obligations under this agreement, detailed in "Attachment X", attached to this agreement.

Accord Project

This clause specifies penalties for shocks caused to a fragile package in transport.

+ Add to contract | Details

Hello World
ITEM | Version 0.13.0
This is the Hello World of Accord Project Templates. Executing the clause will simply echo back the text that occurs after the string `Hello` prepended to text that is passed in the request.

+ Add to contract | Details

Hello World State
ITEM | Version 0.14.0
This is the stateful Hello World of Accord Project Templates. Executing the clause will simply echo back the text that occurs after the string `Hello` prepended to text that is passed in the request along with the number of times the clause has been called.

+ Add to contract | Details

Auto open on startup: ☐

Page 1 of 1 96 words English (United States)

This can be done via the following:

```
// using a state to store the colors for different tags of variables
  // where colors[tag]='color code'
  const [colors, setColors] = useState({});
```

Pseudo Code snippet:

```
let diffTags; // the different tags we encounter while we traverse the CiceroMark JSON
// can be acquired from the keys of counter variable which stores tag names
for (tag in diffTags){
  colors[tag] = `unique color code`
}
```

➢ Adding the new features

While doing the conversion from CiceroMark to JSON, there are many JSON fields that we have left untouched. These can be mainly classified into two categories

1. Elements having class **`org.accordproject.ciceromark.*`**:
   The following elements have no conversion criteria defined for them currently.
   `org.accordproject.ciceromark.Formula,`
   `org.accordproject.ciceromark.Clause,`
   `org.accordproject.ciceromark.Conditional,`
   `org.accordproject.ciceromark.Optional`

   For the Formula checking, this can be used as an inspiration. It has currently implemented the logic to update the formula whenever the variable name changes. Something similar to the above can be implemented.

2. Elements having class **`org.accordproject.commonmark.*`** :
   The following elements have no conversion criteria defined for them currently.
   `org.accordproject.commonmark.Strong,`
   `org.accordproject.commonmark.CodeBlock,`
   `org.accordproject.commonmark.Code,`
   `org.accordproject.commonmark.BlockQuote,`
   `org.accordproject.commonmark.ThematicBreak,`
   `org.accordproject.commonmark.Link,`
   `org.accordproject.commonmark.Image`

   Though most of the elements above would not be present in a clause generally, however one would still benefit if they are parsed. I keep these as low priority*(except the Strong and BlockQuote)*.

3. Implement any of the expressions left here.

I would also like to add an **index for add-in** which will depict the different types of data present in the add-in. Currently, we are dealing only with variables so it is easy to distinguish. However, as we move towards more features for add-in like optional, block, conditional it would become difficult to understand the features. An index would surely help the users to get familiar.

➢ UI Features

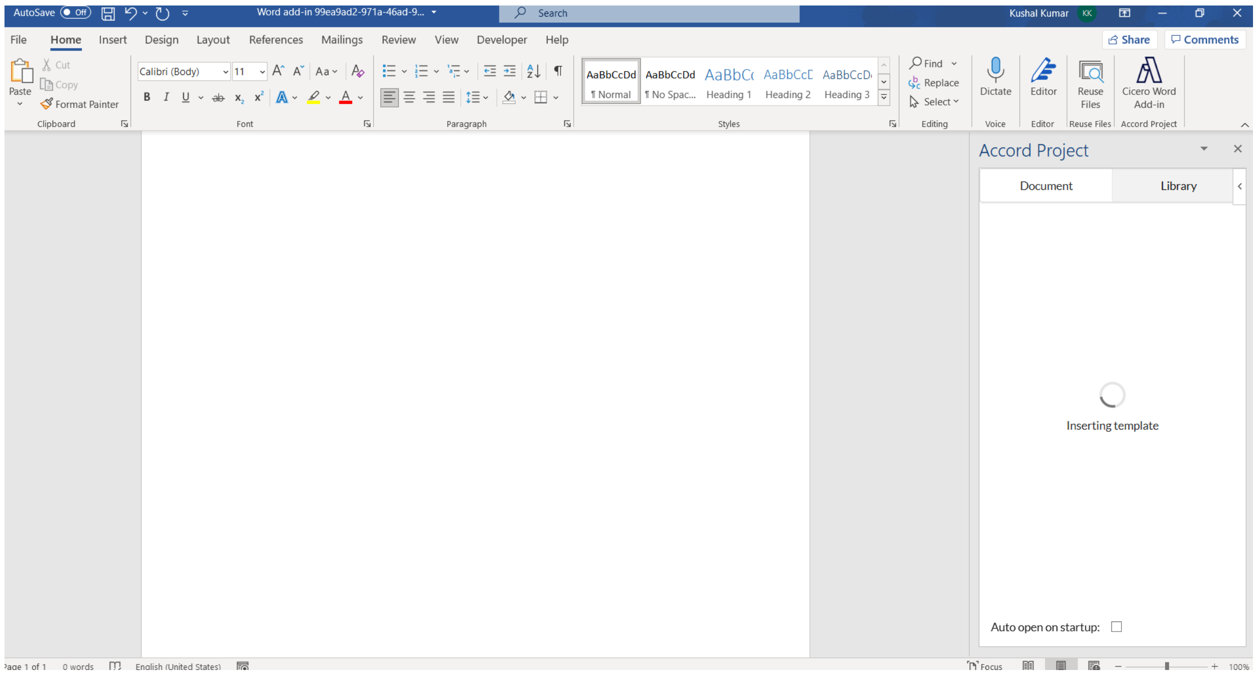
All the features implemented till now were related to the parser i.e. CiceroMark JSON -> OOXML. Below are the features which will be implemented to improve the UI of the add-in.

Template Insertion Loader

Whenever we insert a template there is no indication of whether the template is being inserted. A loader whenever we insert a template would be beneficial in the following ways:

- ➔ A user will be able to see that the template is rendered.
- ➔ Currently, a user can click multiple times continuously on the + Add to contract button. This will ensure that the user clicks only once at a time.

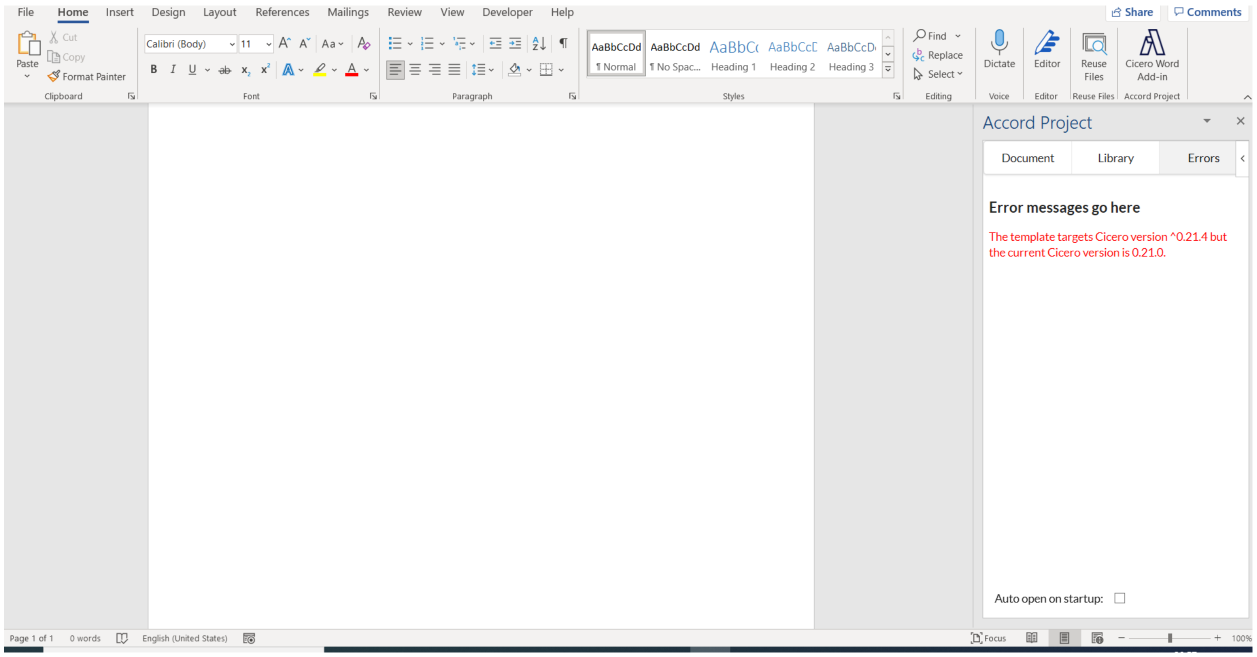
Sample Implementation



Improved Error Handling

Improving the error handling is another feature that will be seen. Currently, if there is some error adding the template, no display or error message is displayed. Showing error messages would certainly improve the user experience.

Sample Implementation

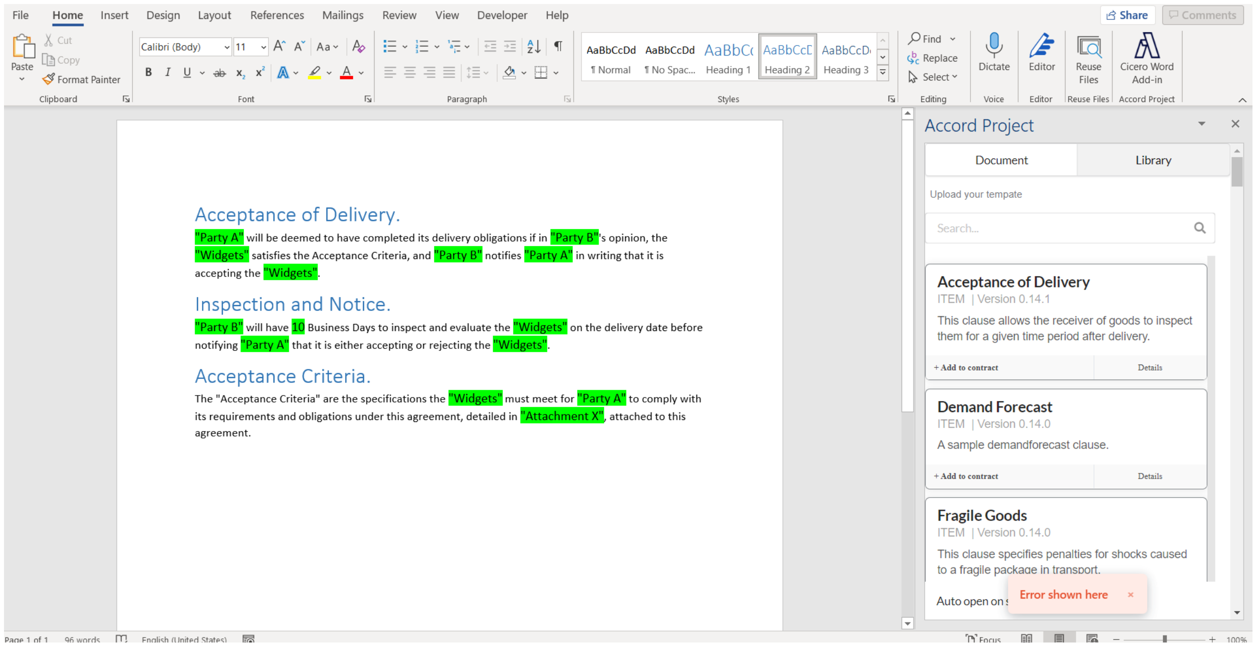


Error messages go here

The template targets Cicero version ^0.21.4 but the current Cicero version is 0.21.0.

The above features can also be implemented by showing the toasts inside add-in using [this]() package.

Sample Implementation



Acceptance of Delivery.

"Party A" will be deemed to have completed its delivery obligations if in "Party B"'s opinion, the "Widgets" satisfies the Acceptance Criteria, and "Party B" notifies "Party A" in writing that it is accepting the "Widgets".

Inspection and Notice.

"Party B" will have 10 Business Days to inspect and evaluate the "Widgets" on the delivery date before notifying "Party A" that it is either accepting or rejecting the "Widgets".

Acceptance Criteria.

The "Acceptance Criteria" are the specifications the "Widgets" must meet for "Party A" to comply with its requirements and obligations under this agreement, detailed in "Attachment X", attached to this agreement.

Acceptance of Delivery
ITEM | Version 0.14.1
This clause allows the receiver of goods to inspect them for a given time period after delivery.

Demand Forecast
ITEM | Version 0.14.0
A sample demandforecast clause.

Fragile Goods
ITEM | Version 0.14.0
This clause specifies penalties for shocks caused to a fragile package in transport.

Error shown here

This feature can also be extended to show notifications when listeners are attached to the document or any other.

OOXML to CiceroMark

This involves the roundtrip conversion i.e. converting the OOXML back to CiceroMark. We have generated the OOXML from CiceroMark. Now, we would also need some means to convert this OOXML back to the original CiceroMark. This mini-project aims at dealing with that.
The work is already started here. However, it still lacks many things and I would like to improve this.
Currently, we don't have any conversion for lists that will be implemented. Apart from that, the OOXML which we generate during this project will also need to be converted to CiceroMark. That work will also be included in this part.

Implementation list

- isEmphasized()
- isList() and lisItem()
- isStrong()
- isBlockQuote()
- isConditional()
- isOptional()
- isClause()
- isFormula()

We already use jest to match whether the parsed OOXML is correct or not. I also intend to continue with jest for testing purposes.

Project Deliverables

After the end of the coding period, the add-in will be supposed to have the following features:

- Support for more CommonMark classes.
- Support for more CiceroMark classes particularly optional, conditional, clause, and ergo expressions(formula).
- Testing of the variable values to ensure correct data.
- Improved UI experience.

Apart from the add-in, the project will also add more features to the OOXML->CiceroMark transformer along with proper tests.

The project outcome will also include proper documentation for the changes.

Flowchart containing the structure and relevant details.

TIMELINE

The following is a rough timeline sketch that I will follow:

Period	**Task**
Pre GSOC	➢ Contribute to the Accord Project ➢ Solve existing issues ➢ Understand the codebase more thoroughly
COMMUNITY BONDING	
May 17-June 07	➢ Discuss with mentors about the best possible implementations of the above approach. ➢ Install any new or necessary dependency for the project. ➢ Learn any new technology which might be required. ➢ Resolve existing issues(majorly minor) of the project to understand the codebase better.
PHASE 1	
Week 01(June 07 - June 14)	➢ Improve the existing features ➢ Adding the DateTime picker and dropdown in the word.
Week 02(June 15 - June 22)	➢ Improving the existing features ➢ Ensure that the text adheres to some specifications in the variables i.e. testing the properties of variable
Week 03(June 23 - June 29)	➢ Addition of new features ➢ Adding remaining converters for

	commonmark.
Week 04-05(June 30 - July 11)	➢ Addition of new features ➢ Converters for the optional, clause, and block CiceroMark features. ➢ Implementation of UI features
Evaluation(July 12 - July 16)	➢ Submit the evaluation ➢ Refactoring of the code. ➢ Verify that the current implementation works correctly.
PHASE 2	
Week 06-07(July 17- July 31)	➢ Addition of features ➢ Implement the ergo expressions i.e. formula in the DOCX and remaining ones. ➢ Wrapping up the CiceroMark->OOXML. ➢ Implementing the index for add-in.
Week 08(August 01 - August 07)	➢ OOXML to CiceroMark ➢ Start the conversion back to CiceroMark. ➢ Do the conversion of basic things which are left(from the previous conversion).
Week 09(August 08 - August 16)	➢ Improvisation of the algorithm ➢ Conversion for complex data types. ➢ Conversion of the implemented OOXML in the project.
Final Evaluation(August 17- August 23)	➢ Submit the final evaluation ➢ Wrapping up the conversion and writing the necessary tests. ➢ Refactoring of the code.

I also intend to add the documentation each week depicting the progress that has been made with the add-in and capturing the necessary things which a user might want to know.

ADDITIONAL WORK

Post completing all my work within the GSoC Timeline, I will give a try to integrate the OOXML <> CiceroMark transform into the transformation graph so that it works with CLI.

Apart from the above-mentioned, I would also love to work on a custom template feature.

MY COMMITMENTS

I have made no prior commitments to anyone so I will be available till the end of July completely. During this time, I will be available for around 40 hours per week. When my college reopens in August, I will be able to denote around 30 hrs per week. I will keep the community updated about my progress.

EXPECTATIONS FROM MENTOR

- ➢ Help me choose the best possible way when I have more than one way of implementation.
- ➢ Understanding some part of the codebase if I am unable to understand on my own.
- ➢ Feedback for the work implemented.

REASON FOR APPLYING

Being working on web development for a year, I have generated keen interest. I have contributed to some open source organizations before. I got to know about the Accord project in December 2020. The codebase of the Accord Project was in React which is my plus point so I decided to give it a try, and have been contributing since.

This project involves working on add-in which is an exclusive field. While exploring the project and its prerequisites, I understood a lot about the different things that are possible in the MS-WORD and it grabbed my attention. The project also interests me in the sense that it would be a great addition to those people who want to create smart legal contracts but do not possess much technical knowledge. Improving their experience with the add-in and making their lives easier while working with the add-in would be very nice and it would be great if I can become a part of this.

COMMUNICATION

I am reachable on the following platforms: Slack, Github, Linkedin, Discord, email, or any other meeting app with scheduled meetings.
Generally the time slots which suits me are:

- 15:30 - 18:30 IST (UTC 10:00 - 13:00)
- 22:30 - 02:30 IST (UTC 17:00 - 21:00)
- 10:00 - 14:00 IST (UTC 04:30 - 08:30)

POST GSoC

I intend to stay with the community after GSoC and improve the services which are provided by the Accord Project. I will contribute to the community by improving the editor features or any other thing which is possible. It would give me immense pleasure to be a part of the Accord Project community and improve the lives of those who are using the Accord Project.