AI-POWERED XML EDITING

George Bina, Oxygen XML Editor

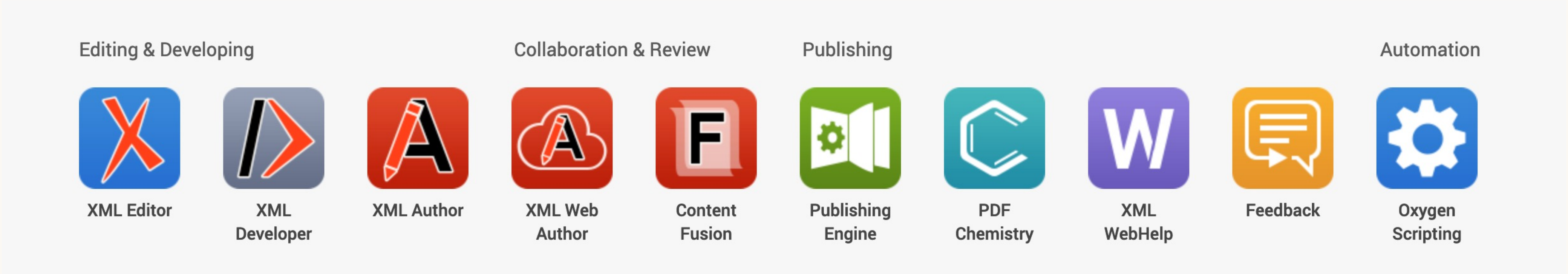
george@oxygenxml.com

© 2024 Syncro Soft SRL. All rights reserved.

About me

#syncrosoft #xml #oxygenxml #opensource #technology #tango

Oxygen Tools



Editing & Developing

Collaboration & Review

Publishing

Automation

XML Editor

XML Developer

XML Author

XML Web Author

Content Fusion

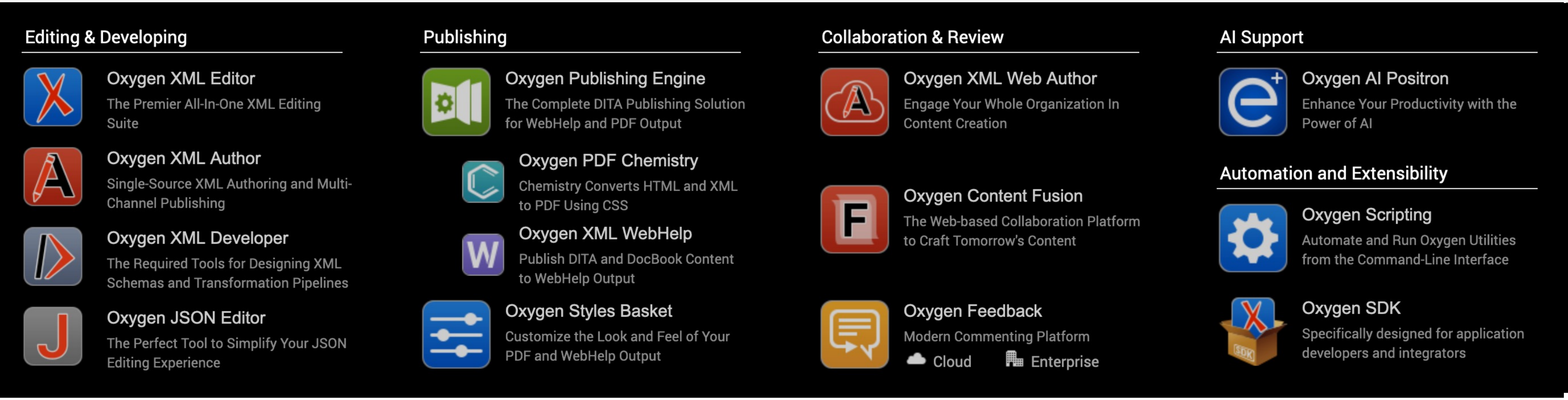
Publishing Engine

PDF Chemistry

XML WebHelp

Feedback

Oxygen Scripting



Editing & Developing

Oxygen XML Editor
The Premier All-In-One XML Editing Suite

Oxygen XML Author
Single-Source XML Authoring and Multi-Channel Publishing

Oxygen XML Developer
The Required Tools for Designing XML Schemas and Transformation Pipelines

Oxygen JSON Editor
The Perfect Tool to Simplify Your JSON Editing Experience

Publishing

Oxygen Publishing Engine
The Complete DITA Publishing Solution for WebHelp and PDF Output

Oxygen PDF Chemistry
Chemistry Converts HTML and XML to PDF Using CSS

Oxygen XML WebHelp
Publish DITA and DocBook Content to WebHelp Output

Oxygen Styles Basket
Customize the Look and Feel of Your PDF and WebHelp Output

Collaboration & Review

Oxygen XML Web Author
Engage Your Whole Organization In Content Creation

Oxygen Content Fusion
The Web-based Collaboration Platform to Craft Tomorrow's Content

Oxygen Feedback
Modern Commenting Platform
☁ Cloud　Enterprise

AI Support

Oxygen AI Positron
Enhance Your Productivity with the Power of AI

Automation and Extensibility

Oxygen Scripting
Automate and Run Oxygen Utilities from the Command-Line Interface

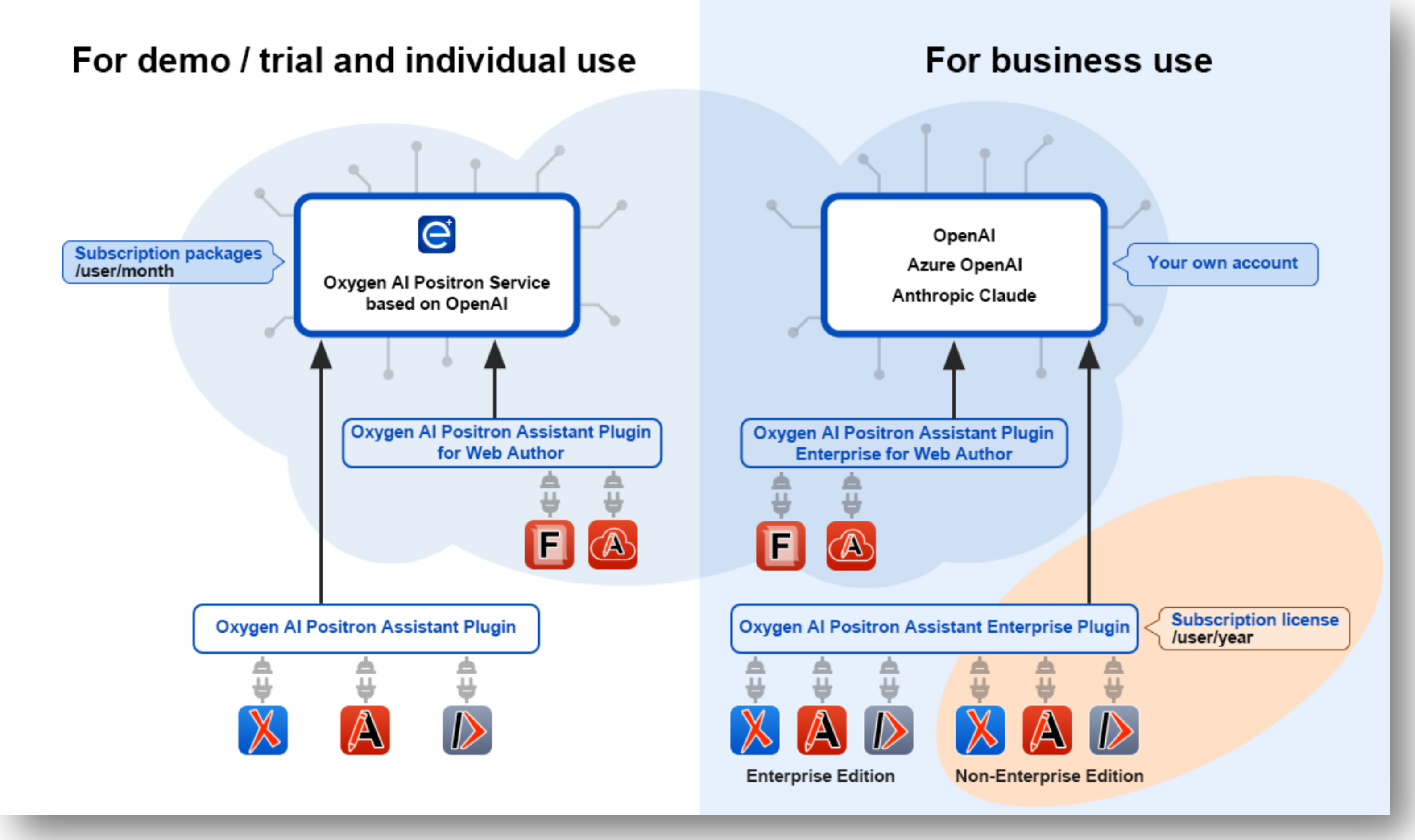
Oxygen SDK
Specifically designed for application developers and integrators

AI-Powered XML Editing

Is AI ready to be used?

Yes, but with care :)

Oxygen AI Positron Assistant add-ons



For demo / trial and individual use

Subscription packages /user/month

Oxygen AI Positron Service based on OpenAI

Oxygen AI Positron Assistant Plugin for Web Author

Oxygen AI Positron Assistant Plugin

For business use

OpenAI
Azure OpenAI
Anthropic Claude

Your own account

Oxygen AI Positron Assistant Plugin Enterprise for Web Author

Oxygen AI Positron Assistant Enterprise Plugin

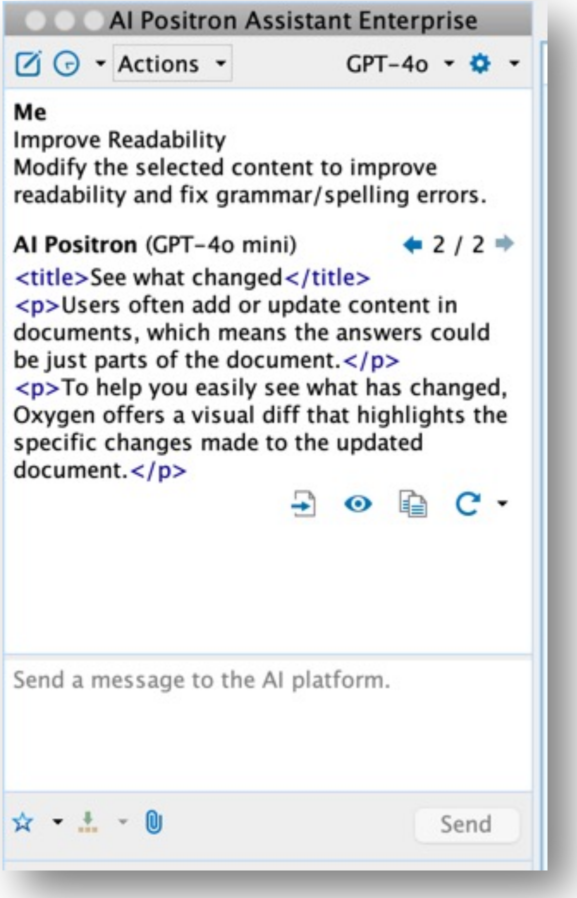
Subscription license /user/year

Enterprise Edition

Non-Enterprise Edition

Basic interactions with Chat

Multiple variants with editable chat history



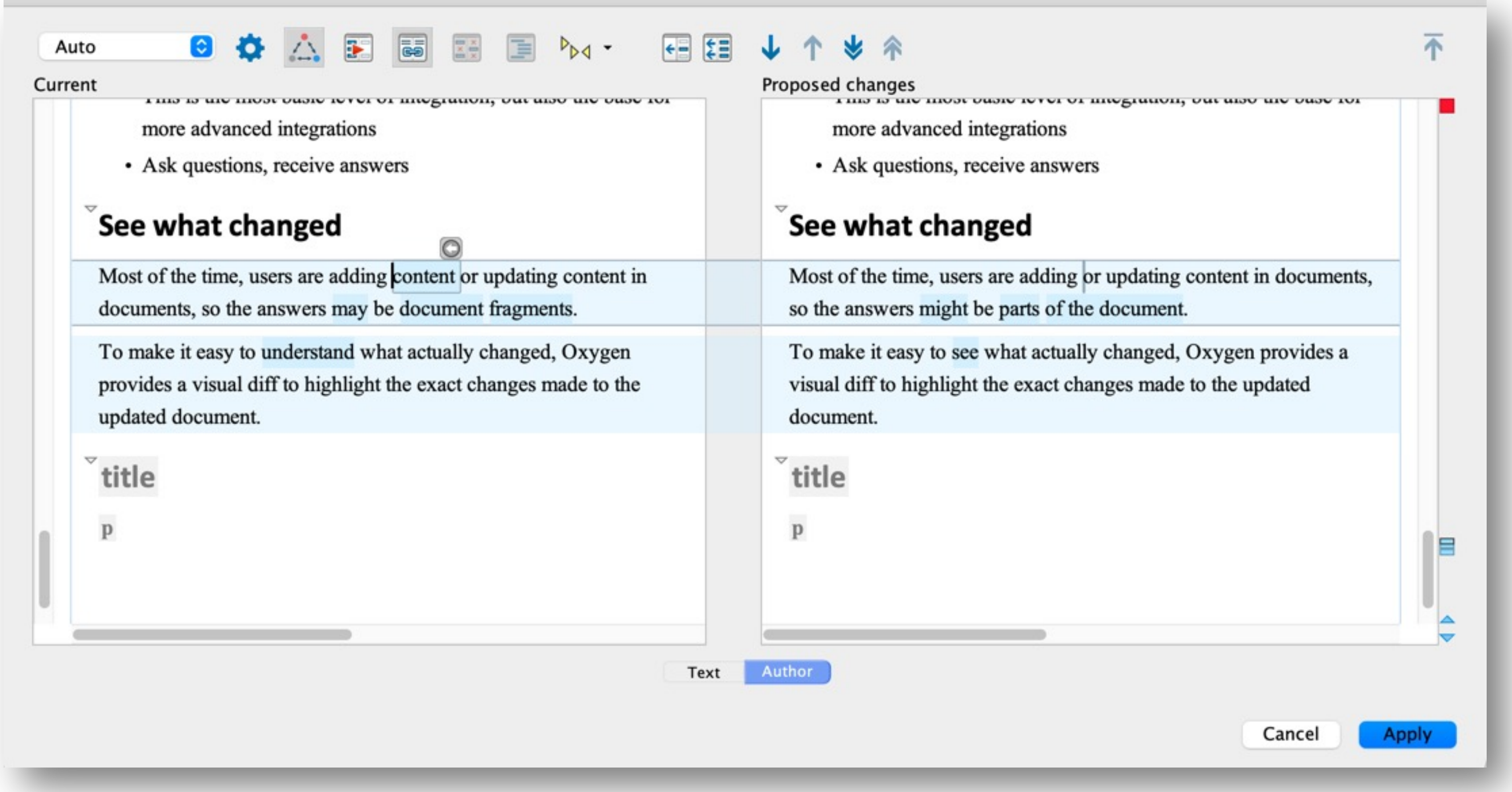
Me
Improve Readability
Modify the selected content to improve readability and fix grammar/spelling errors.

AI Positron (GPT-4o mini) 2 / 2

```
<title>See what changed</title>
<p>Users often add or update content in documents, which means the answers could be just parts of the document.</p>
<p>To help you easily see what has changed, Oxygen offers a visual diff that highlights the specific changes made to the updated document.</p>
```

Send a message to the AI platform.

Review what was changed and apply the changes



Current:

more advanced integrations

- Ask questions, receive answers

See what changed

Most of the time, users are adding content or updating content in documents, so the answers may be document fragments.

To make it easy to understand what actually changed, Oxygen provides a visual diff to highlight the exact changes made to the updated document.

title

p

Proposed changes:

more advanced integrations

- Ask questions, receive answers

See what changed

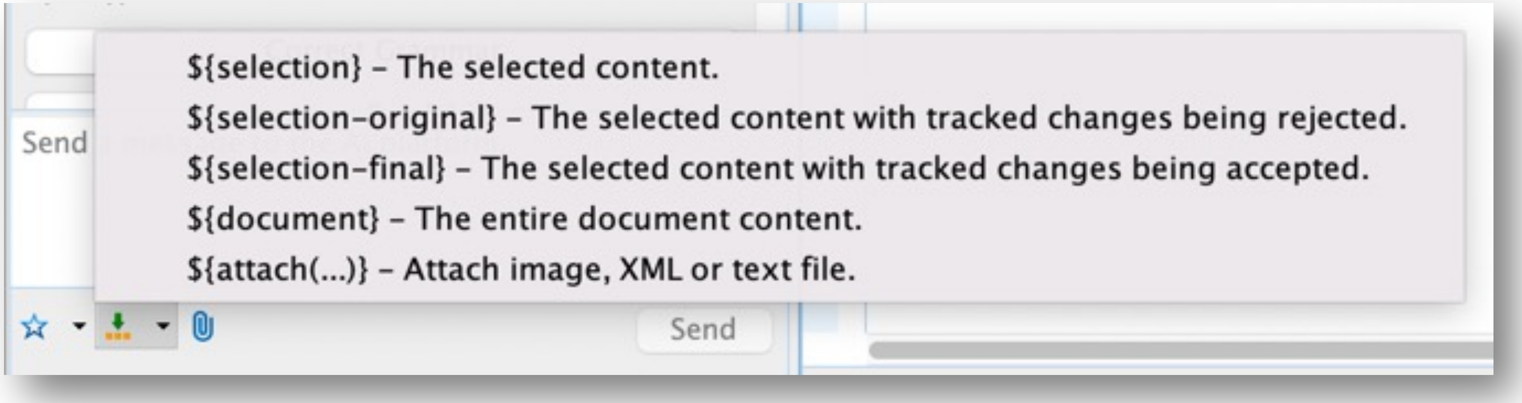
Most of the time, users are adding or updating content in documents, so the answers might be parts of the document.

To make it easy to see what actually changed, Oxygen provides a visual diff to highlight the exact changes made to the updated document.

title

p

Prompt variables and functions



${selection} – The selected content.
${selection-original} – The selected content with tracked changes being rejected.
${selection-final} – The selected content with tracked changes being accepted.
${document} – The entire document content.
${attach(...)} – Attach image, XML or text file.

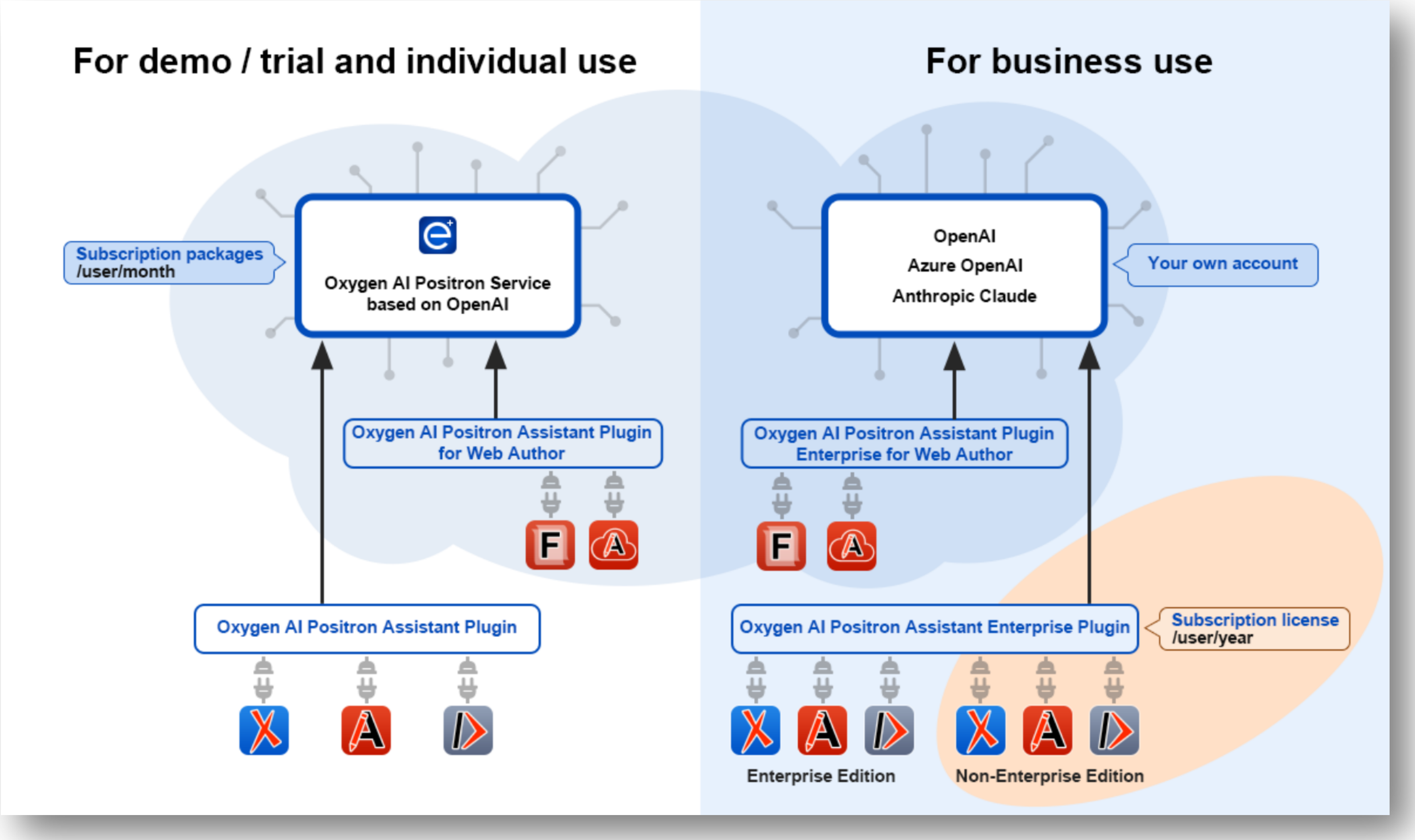
Enable the AI to find additional information

- RAG - Retrieval-Augmented Generation
- AI callback functions, such as:
  - get_related_content_from_project
  - get_related_resources_overview_from_project
  - get_current_document_plain_text_content
  - get_current_document_marked_up_content
  - get_text_around_caret

Vision support

- Combining text and vision is essential for XML documents and in particular for structured technical documentation
- Enables better results for some actions and new actions that would be impossible otherwise
- Integration of vision using a prompt function that can be placed within a prompt, so you can just write naturally the prompt and refer to an image.

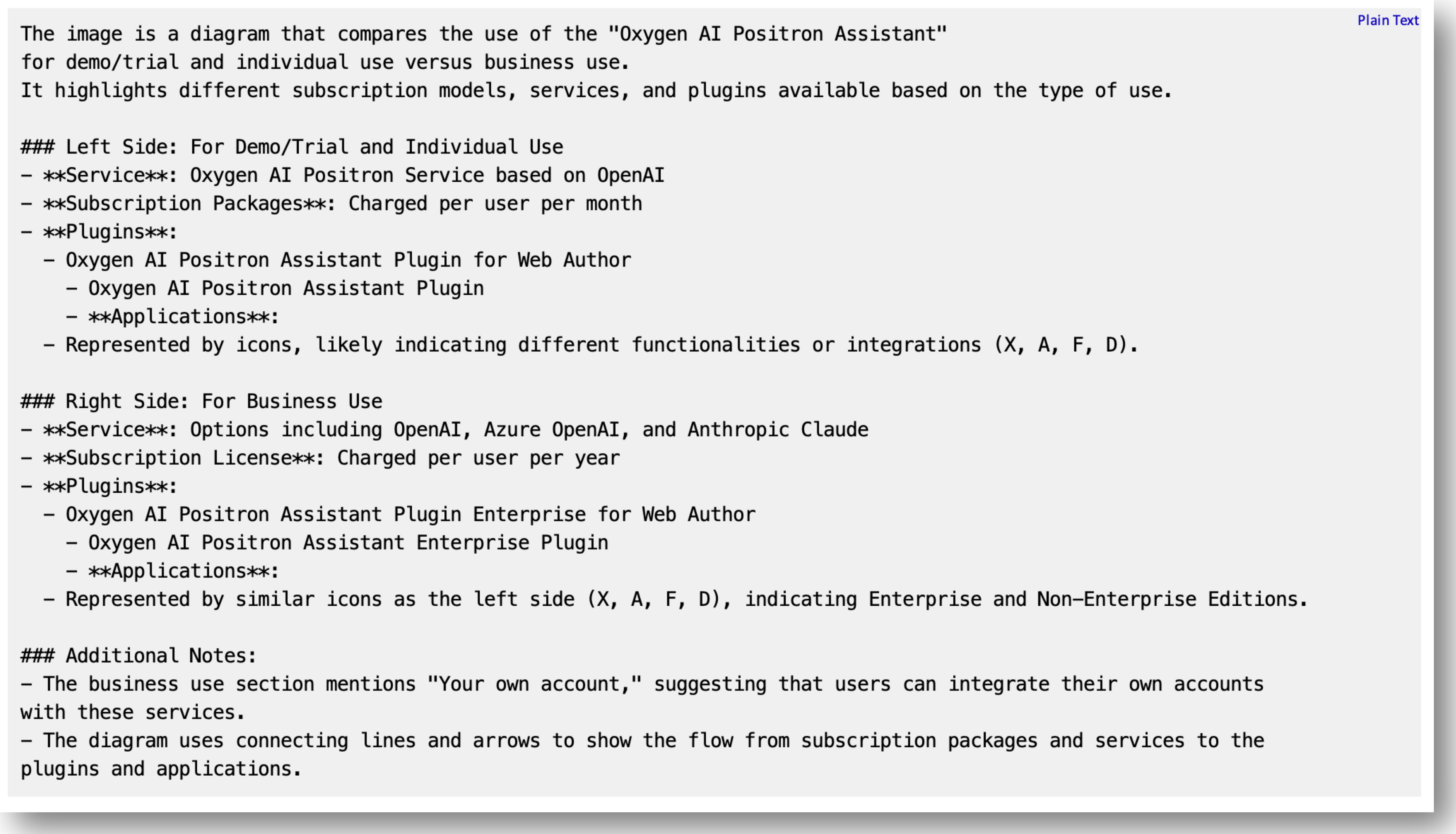
Describe this image: ${attach(.../addons.png)}



For demo / trial and individual use

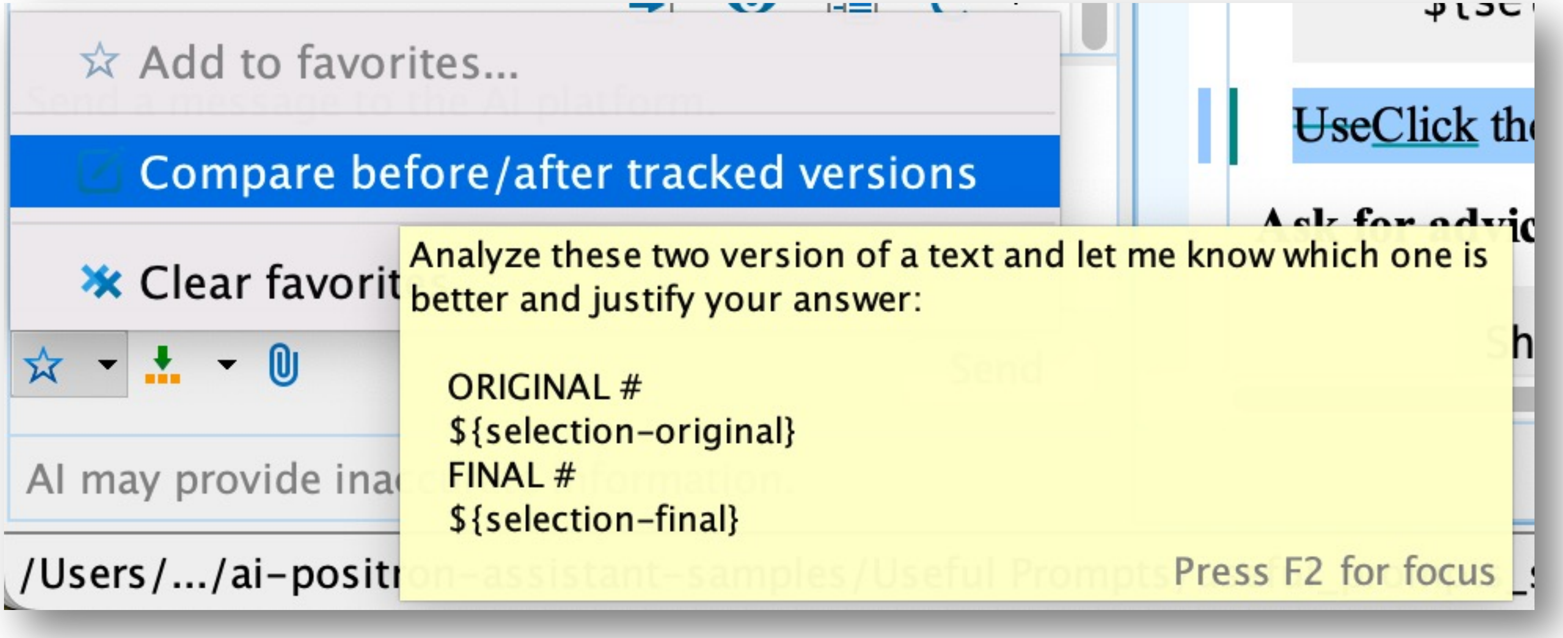
For business use

Subscription packages /user/month
Oxygen AI Positron Service based on OpenAI
Oxygen AI Positron Assistant Plugin for Web Author
Oxygen AI Positron Assistant Plugin
OpenAI
Azure OpenAI
Anthropic Claude
Your own account
Oxygen AI Positron Assistant Plugin Enterprise for Web Author
Oxygen AI Positron Assistant Enterprise Plugin
Subscription license /user/year
Enterprise Edition
Non-Enterprise Edition



```
The image is a diagram that compares the use of the "Oxygen AI Positron Assistant"
for demo/trial and individual use versus business use.
It highlights different subscription models, services, and plugins available based on the type of use.

### Left Side: For Demo/Trial and Individual Use
- **Service**: Oxygen AI Positron Service based on OpenAI
- **Subscription Packages**: Charged per user per month
- **Plugins**:
  - Oxygen AI Positron Assistant Plugin for Web Author
    - Oxygen AI Positron Assistant Plugin
    - **Applications**:
  - Represented by icons, likely indicating different functionalities or integrations (X, A, F, D).

### Right Side: For Business Use
- **Service**: Options including OpenAI, Azure OpenAI, and Anthropic Claude
- **Subscription License**: Charged per user per year
- **Plugins**:
  - Oxygen AI Positron Assistant Plugin Enterprise for Web Author
    - Oxygen AI Positron Assistant Enterprise Plugin
    - **Applications**:
  - Represented by similar icons as the left side (X, A, F, D), indicating Enterprise and Non-Enterprise Editions.

### Additional Notes:
- The business use section mentions "Your own account," suggesting that users can integrate their own accounts
with these services.
- The diagram uses connecting lines and arrows to show the flow from subscription packages and services to the
plugins and applications.
```

Quick AI prompts reuse with favorite prompts



Analyze these two version of a text and let me know which one is better and justify your answer:

ORIGINAL #
${selection-original}
FINAL #
${selection-final}

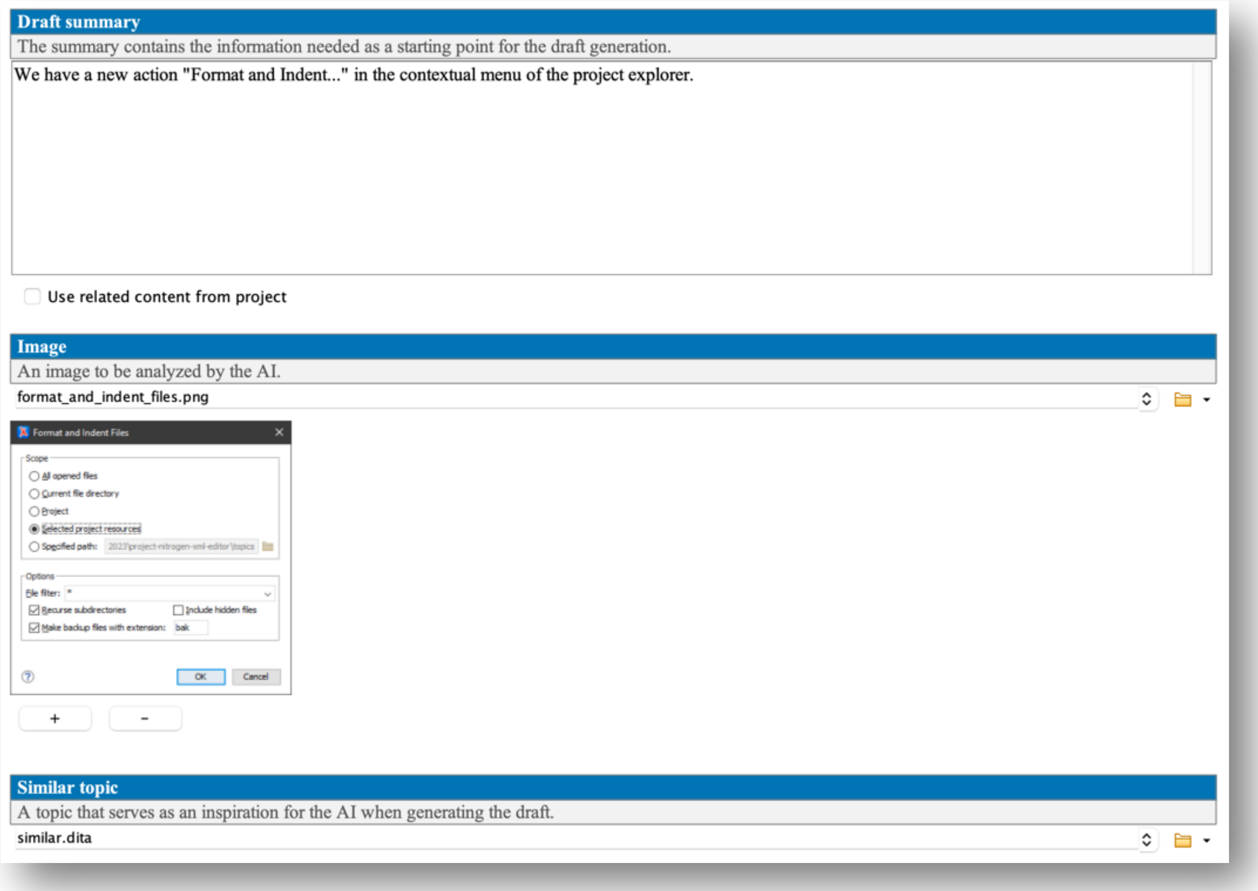
AI Actions

- Initiate a chat but offers much more than a favorite prompt.
- Specify initial prompt, AI model, context information, result type, settings
- Refine results by continuing the chat
- Defined by a configuration file (specific editing support provided to offer available keys and possible values)
- Many predefined AI Actions covering the common use cases and the actions we found valuable.

Predefined AI Actions

- Accessibility
- Content generation
- Rewrite
- Review - AI integration via comments/annotations
- Overview
- Translation
- Marketing
- Development

Accessibility - generate alternate text for image



Draft summary

The summary contains the information needed as a starting point for the draft generation.

We have a new action "Format and Indent..." in the contextual menu of the project explorer.

☐ Use related content from project

Image

An image to be analyzed by the AI.

format_and_indent_files.png

Format and Indent Files

Scope

○ All opened files
○ Current file directory
○ Project
● Selected project resources
○ Specified path: 2023\project-nitrogen-xml-editor\topics

Options

File filter: *
☑ Recurse subdirectories ☐ Include hidden files
☑ Make backup files with extension: bak

Similar topic

A topic that serves as an inspiration for the AI when generating the draft.

similar.dita

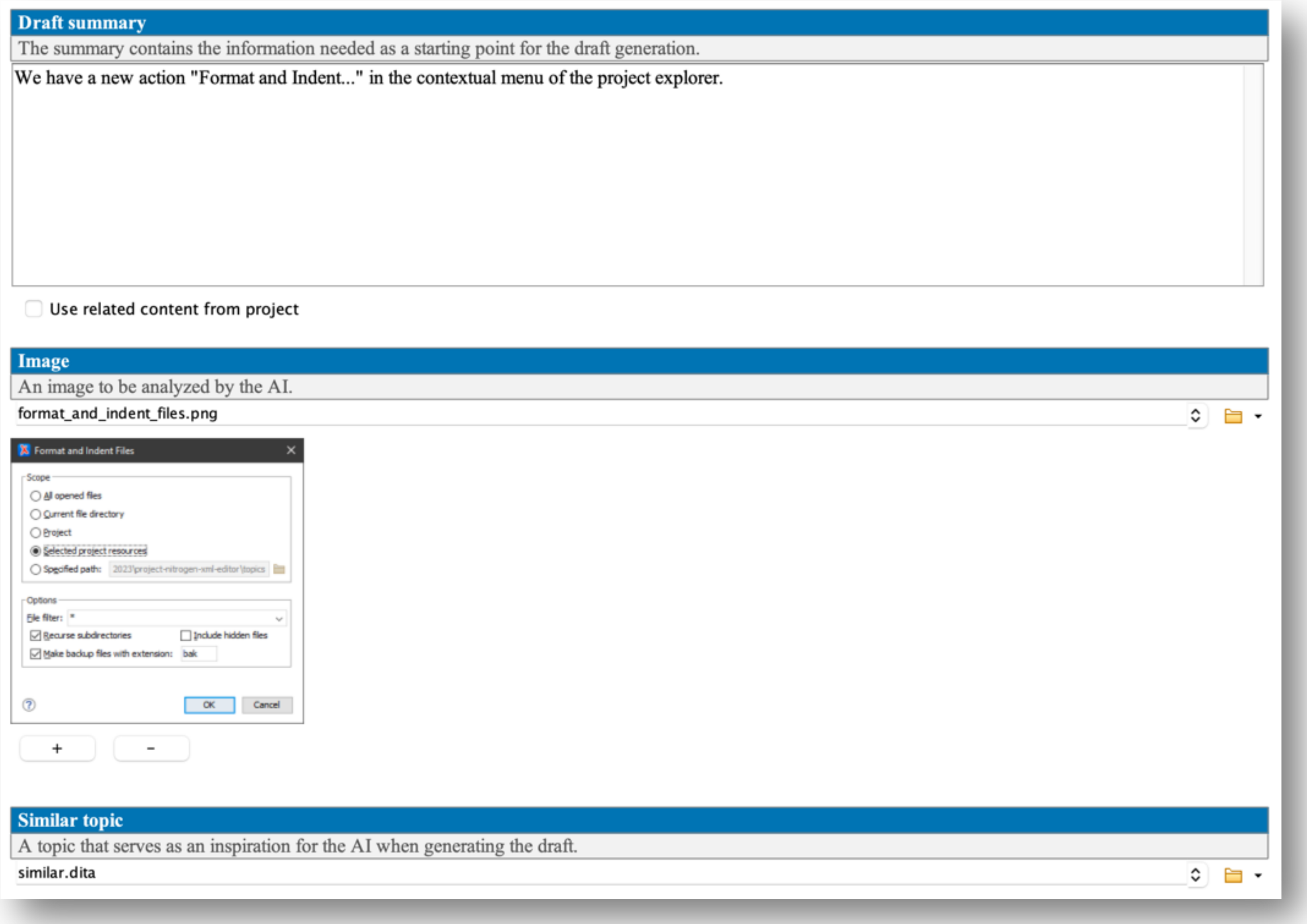
Plain Text

```
Draft configuration interface with options for summary, image analysis, and similar topic selection.
```

Content generation actions

- Generate documentation draft
- New topic
- Update content based on updated images
- Continue writing
- Generate short description
- Suggest index terms
- Add structured content

Generate documentation draft



Draft summary

The summary contains the information needed as a starting point for the draft generation.

We have a new action "Format and Indent..." in the contextual menu of the project explorer.

☐ Use related content from project

Image

An image to be analyzed by the AI.

format_and_indent_files.png

Format and Indent Files

Scope

○ All opened files
○ Current file directory
○ Project
● Selected project resources
○ Specified path: 2023\project-nitrogen-xml-editor\topics

Options

File filter: *

☑ Recurse subdirectories ☐ Include hidden files

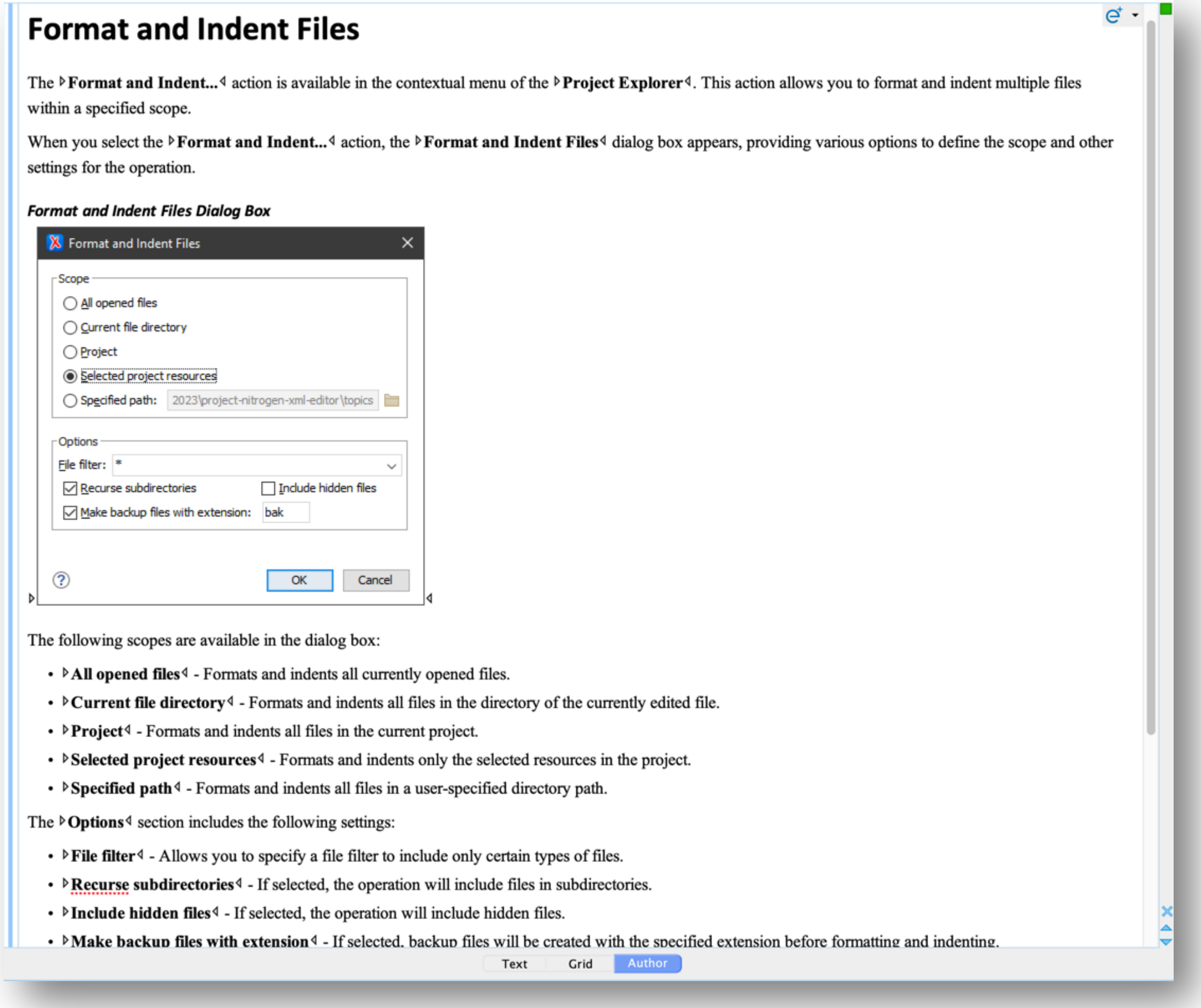
☑ Make backup files with extension: bak

Similar topic

A topic that serves as an inspiration for the AI when generating the draft.

similar.dita

Documentation draft result



Format and Indent Files

The **Format and Indent...** action is available in the contextual menu of the **Project Explorer**. This action allows you to format and indent multiple files within a specified scope.

When you select the **Format and Indent...** action, the **Format and Indent Files** dialog box appears, providing various options to define the scope and other settings for the operation.

Format and Indent Files Dialog Box

Format and Indent Files

Scope

○ All opened files
○ Current file directory
○ Project
● Selected project resources
○ Specified path: 2023\project-nitrogen-xml-editor\topics

Options

File filter: *
☑ Recurse subdirectories ☐ Include hidden files
☑ Make backup files with extension: bak

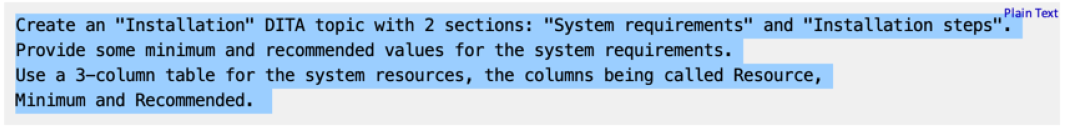
The following scopes are available in the dialog box:

- **All opened files** - Formats and indents all currently opened files.
- **Current file directory** - Formats and indents all files in the directory of the currently edited file.
- **Project** - Formats and indents all files in the current project.
- **Selected project resources** - Formats and indents only the selected resources in the project.
- **Specified path** - Formats and indents all files in a user-specified directory path.

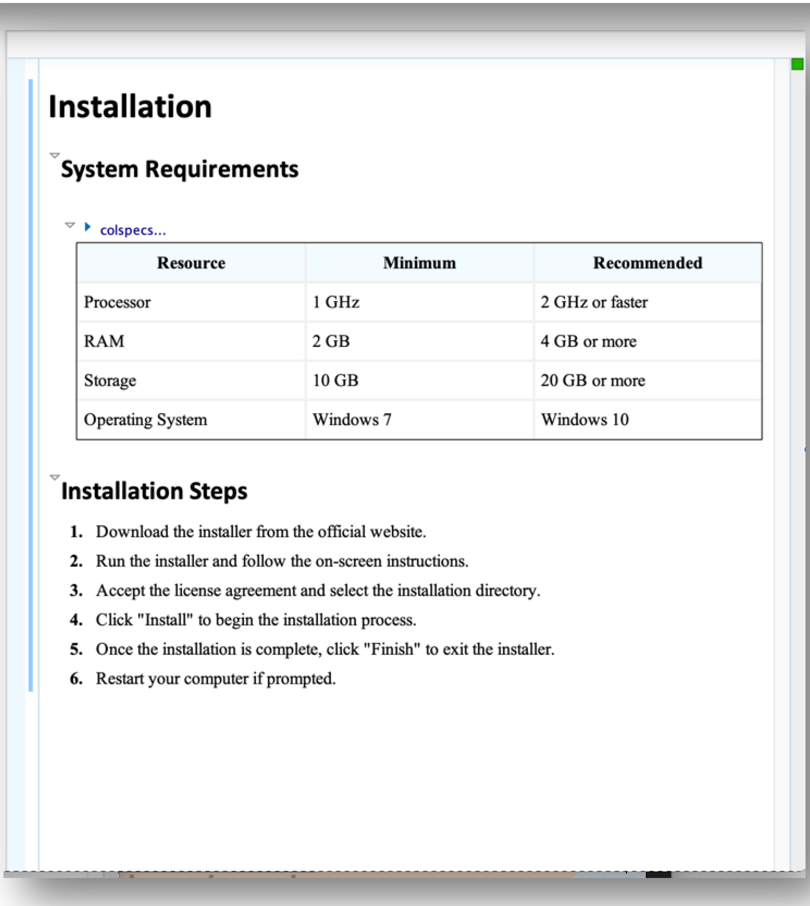
The **Options** section includes the following settings:

- **File filter** - Allows you to specify a file filter to include only certain types of files.
- **Recurse subdirectories** - If selected, the operation will include files in subdirectories.
- **Include hidden files** - If selected, the operation will include hidden files.
- **Make backup files with extension** - If selected, backup files will be created with the specified extension before formatting and indenting.

New topic



```
Create an "Installation" DITA topic with 2 sections: "System requirements" and "Installation steps".
Provide some minimum and recommended values for the system requirements.
Use a 3-column table for the system resources, the columns being called Resource,
Minimum and Recommended.
```



Installation

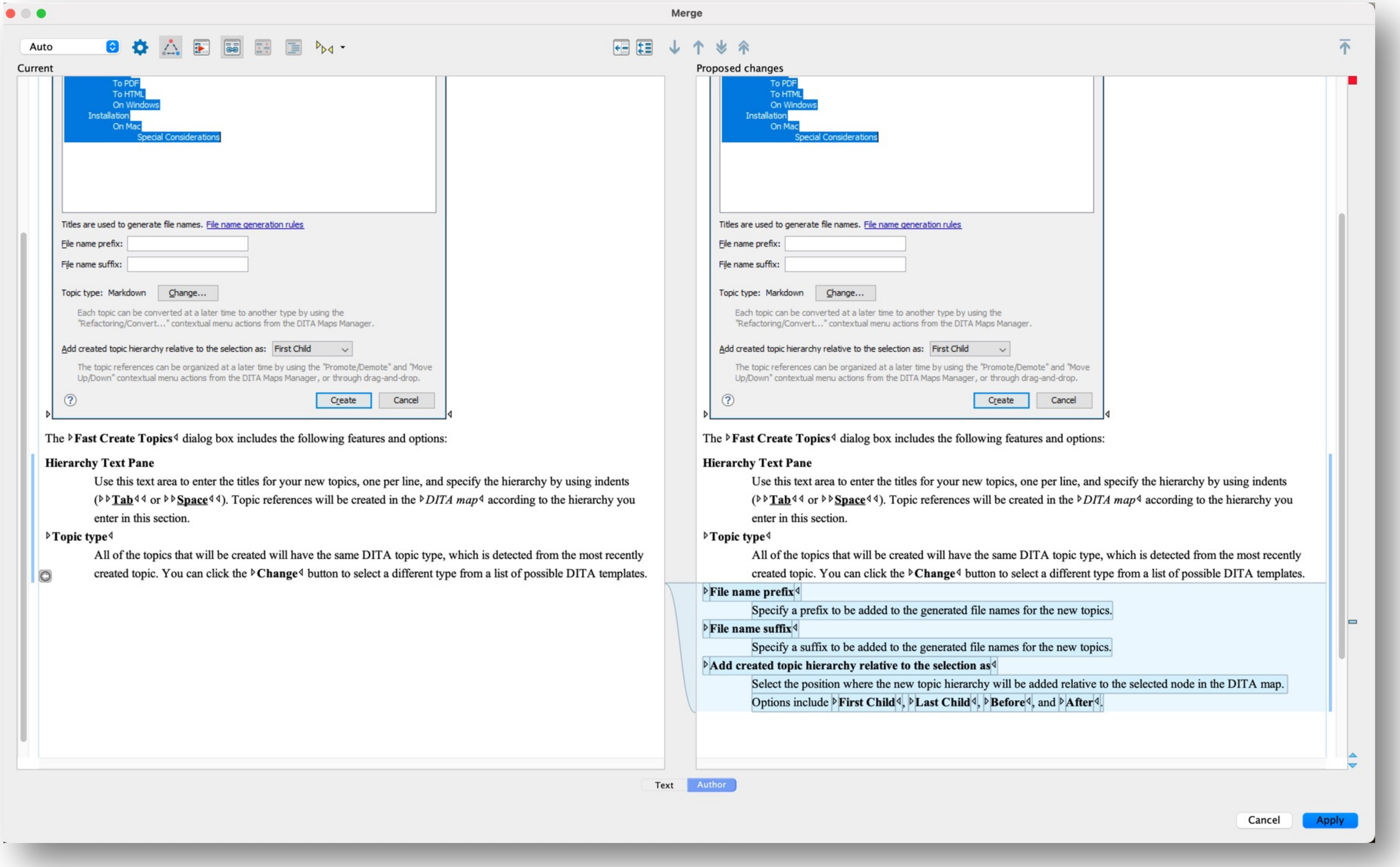
System Requirements

Resource	Minimum	Recommended
Processor	1 GHz	2 GHz or faster
RAM	2 GB	4 GB or more
Storage	10 GB	20 GB or more
Operating System	Windows 7	Windows 10

Installation Steps

1. Download the installer from the official website.
2. Run the installer and follow the on-screen instructions.
3. Accept the license agreement and select the installation directory.
4. Click "Install" to begin the installation process.
5. Once the installation is complete, click "Finish" to exit the installer.
6. Restart your computer if prompted.

Update content based on updated images



Current

To PDF
To HTML
On Windows
Installation
On Mac
Special Considerations

Titles are used to generate file names. File name generation rules

File name prefix:

File name suffix:

Topic type: Markdown Change...

Each topic can be converted at a later time to another type by using the "Refactoring/Convert..." contextual menu actions from the DITA Maps Manager.

Add created topic hierarchy relative to the selection as: First Child

The topic references can be organized at a later time by using the "Promote/Demote" and "Move Up/Down" contextual menu actions from the DITA Maps Manager, or through drag-and-drop.

Create Cancel

The **Fast Create Topics** dialog box includes the following features and options:

Hierarchy Text Pane

Use this text area to enter the titles for your new topics, one per line, and specify the hierarchy by using indents (**Tab** or **Space**). Topic references will be created in the *DITA map* according to the hierarchy you enter in this section.

Topic type

All of the topics that will be created will have the same DITA topic type, which is detected from the most recently created topic. You can click the **Change** button to select a different type from a list of possible DITA templates.

Proposed changes

To PDF
To HTML
On Windows
Installation
On Mac
Special Considerations

Titles are used to generate file names. File name generation rules

File name prefix:

File name suffix:

Topic type: Markdown Change...

Each topic can be converted at a later time to another type by using the "Refactoring/Convert..." contextual menu actions from the DITA Maps Manager.

Add created topic hierarchy relative to the selection as: First Child

The topic references can be organized at a later time by using the "Promote/Demote" and "Move Up/Down" contextual menu actions from the DITA Maps Manager, or through drag-and-drop.

Create Cancel

The **Fast Create Topics** dialog box includes the following features and options:

Hierarchy Text Pane

Use this text area to enter the titles for your new topics, one per line, and specify the hierarchy by using indents (**Tab** or **Space**). Topic references will be created in the *DITA map* according to the hierarchy you enter in this section.

Topic type

All of the topics that will be created will have the same DITA topic type, which is detected from the most recently created topic. You can click the **Change** button to select a different type from a list of possible DITA templates.

File name prefix

Specify a prefix to be added to the generated file names for the new topics.

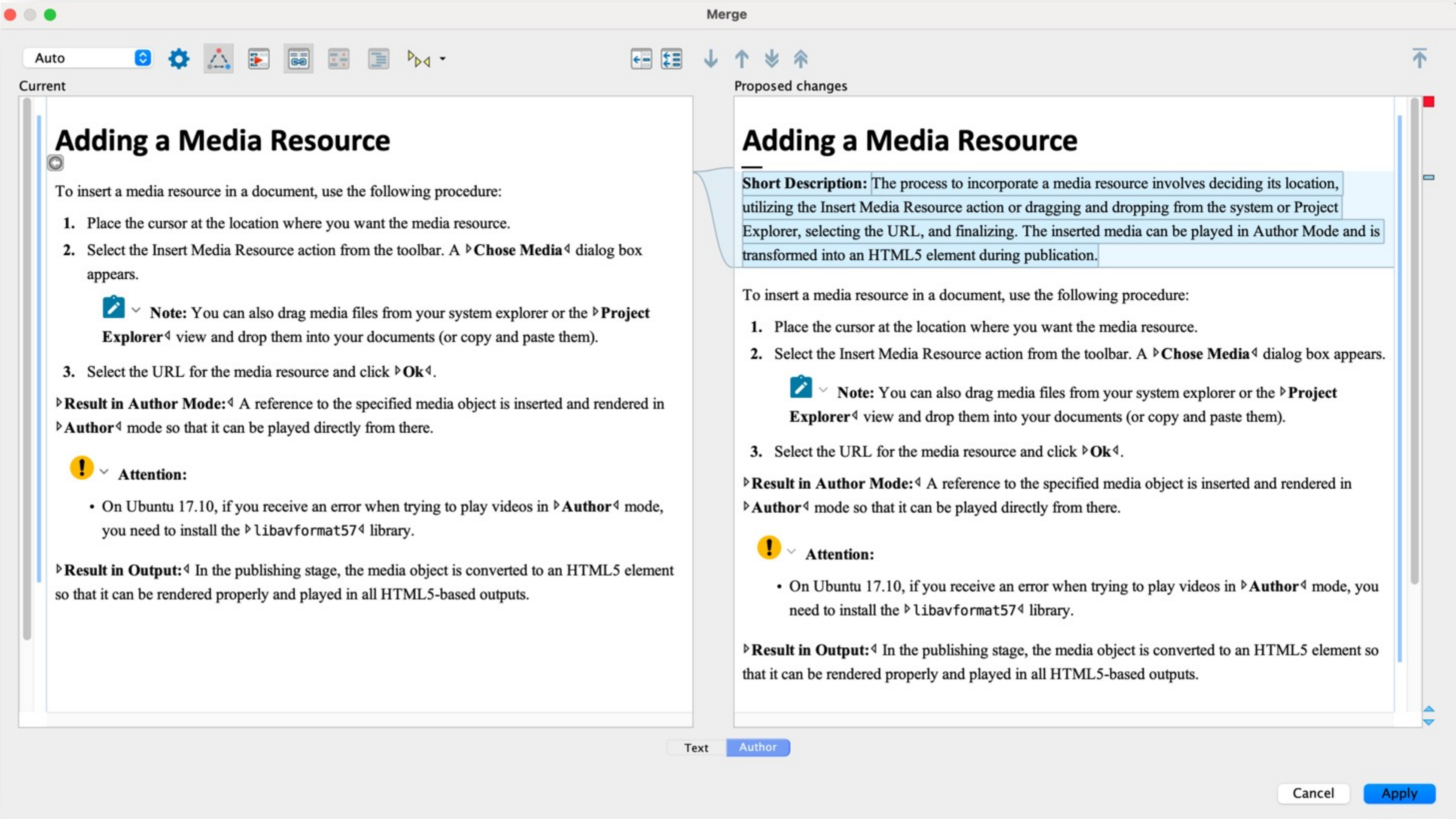
File name suffix

Specify a suffix to be added to the generated file names for the new topics.

Add created topic hierarchy relative to the selection as

Select the position where the new topic hierarchy will be added relative to the selected node in the DITA map. Options include **First Child**, **Last Child**, **Before**, and **After**.

Generate short descriptions



Current

Adding a Media Resource

To insert a media resource in a document, use the following procedure:

1. Place the cursor at the location where you want the media resource.
2. Select the Insert Media Resource action from the toolbar. A **Chose Media** dialog box appears.

   Note: You can also drag media files from your system explorer or the **Project Explorer** view and drop them into your documents (or copy and paste them).

3. Select the URL for the media resource and click **Ok**.

Result in Author Mode: A reference to the specified media object is inserted and rendered in **Author** mode so that it can be played directly from there.

Attention:

- On Ubuntu 17.10, if you receive an error when trying to play videos in **Author** mode, you need to install the `libavformat57` library.

Result in Output: In the publishing stage, the media object is converted to an HTML5 element so that it can be rendered properly and played in all HTML5-based outputs.

Proposed changes

Adding a Media Resource

Short Description: The process to incorporate a media resource involves deciding its location, utilizing the Insert Media Resource action or dragging and dropping from the system or Project Explorer, selecting the URL, and finalizing. The inserted media can be played in Author Mode and is transformed into an HTML5 element during publication.

To insert a media resource in a document, use the following procedure:

1. Place the cursor at the location where you want the media resource.
2. Select the Insert Media Resource action from the toolbar. A **Chose Media** dialog box appears.

   Note: You can also drag media files from your system explorer or the **Project Explorer** view and drop them into your documents (or copy and paste them).

3. Select the URL for the media resource and click **Ok**.

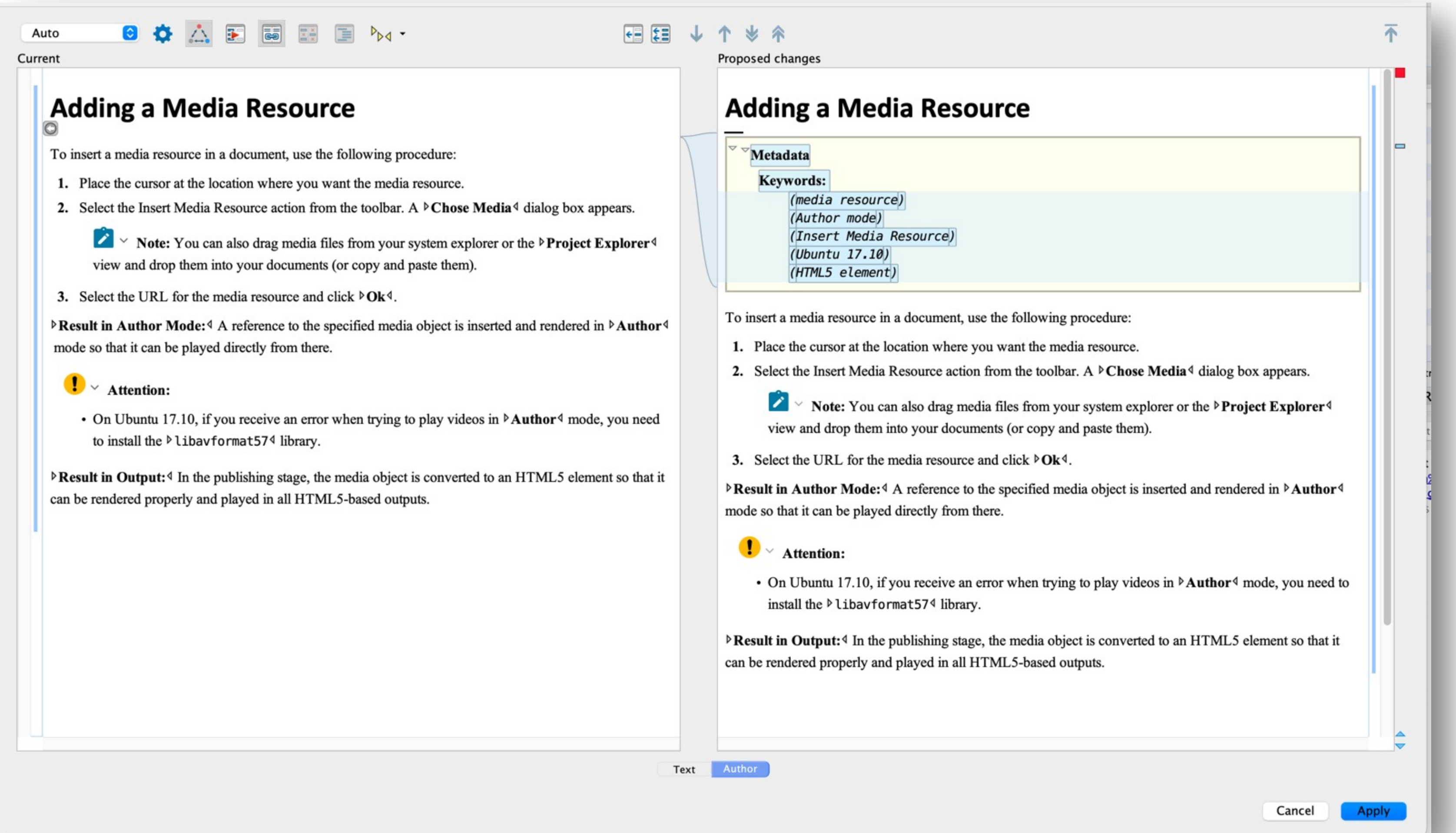
Result in Author Mode: A reference to the specified media object is inserted and rendered in **Author** mode so that it can be played directly from there.

Attention:

- On Ubuntu 17.10, if you receive an error when trying to play videos in **Author** mode, you need to install the `libavformat57` library.

Result in Output: In the publishing stage, the media object is converted to an HTML5 element so that it can be rendered properly and played in all HTML5-based outputs.

Generate index terms



Adding a Media Resource

To insert a media resource in a document, use the following procedure:

1. Place the cursor at the location where you want the media resource.
2. Select the Insert Media Resource action from the toolbar. A **Chose Media** dialog box appears.

   Note: You can also drag media files from your system explorer or the **Project Explorer** view and drop them into your documents (or copy and paste them).

3. Select the URL for the media resource and click **Ok**.

Result in Author Mode: A reference to the specified media object is inserted and rendered in **Author** mode so that it can be played directly from there.

Attention:

- On Ubuntu 17.10, if you receive an error when trying to play videos in **Author** mode, you need to install the `libavformat57` library.

Result in Output: In the publishing stage, the media object is converted to an HTML5 element so that it can be rendered properly and played in all HTML5-based outputs.

Adding a Media Resource

Metadata

Keywords:

(media resource)
(Author mode)
(Insert Media Resource)
(Ubuntu 17.10)
(HTML5 element)

To insert a media resource in a document, use the following procedure:

1. Place the cursor at the location where you want the media resource.
2. Select the Insert Media Resource action from the toolbar. A **Chose Media** dialog box appears.

   Note: You can also drag media files from your system explorer or the **Project Explorer** view and drop them into your documents (or copy and paste them).

3. Select the URL for the media resource and click **Ok**.

Result in Author Mode: A reference to the specified media object is inserted and rendered in **Author** mode so that it can be played directly from there.

Attention:

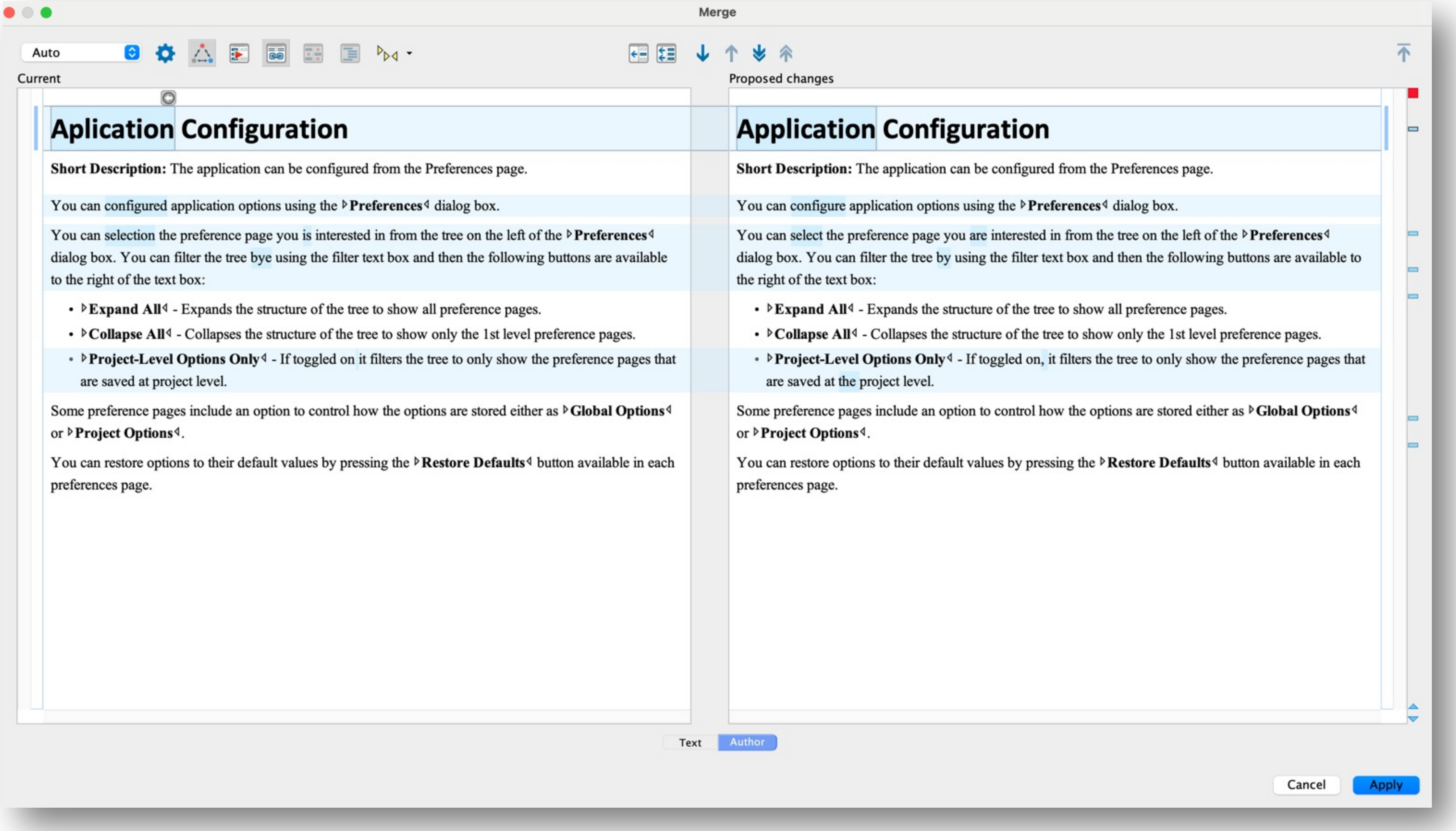
- On Ubuntu 17.10, if you receive an error when trying to play videos in **Author** mode, you need to install the `libavformat57` library.

Result in Output: In the publishing stage, the media object is converted to an HTML5 element so that it can be rendered properly and played in all HTML5-based outputs.

Rewrite actions

- Correct grammar
- Improve readability
- Use active voice
- Improve structure
- Itemize
- Join items

Correct grammar



Current

Aplication Configuration

Short Description: The application can be configured from the Preferences page.

You can configured application options using the **Preferences** dialog box.

You can selection the preference page you is interested in from the tree on the left of the **Preferences** dialog box. You can filter the tree bye using the filter text box and then the following buttons are available to the right of the text box:

- **Expand All** - Expands the structure of the tree to show all preference pages.
- **Collapse All** - Collapses the structure of the tree to show only the 1st level preference pages.
- **Project-Level Options Only** - If toggled on it filters the tree to only show the preference pages that are saved at project level.

Some preference pages include an option to control how the options are stored either as **Global Options** or **Project Options**.

You can restore options to their default values by pressing the **Restore Defaults** button available in each preferences page.

Proposed changes

Application Configuration

Short Description: The application can be configured from the Preferences page.

You can configure application options using the **Preferences** dialog box.

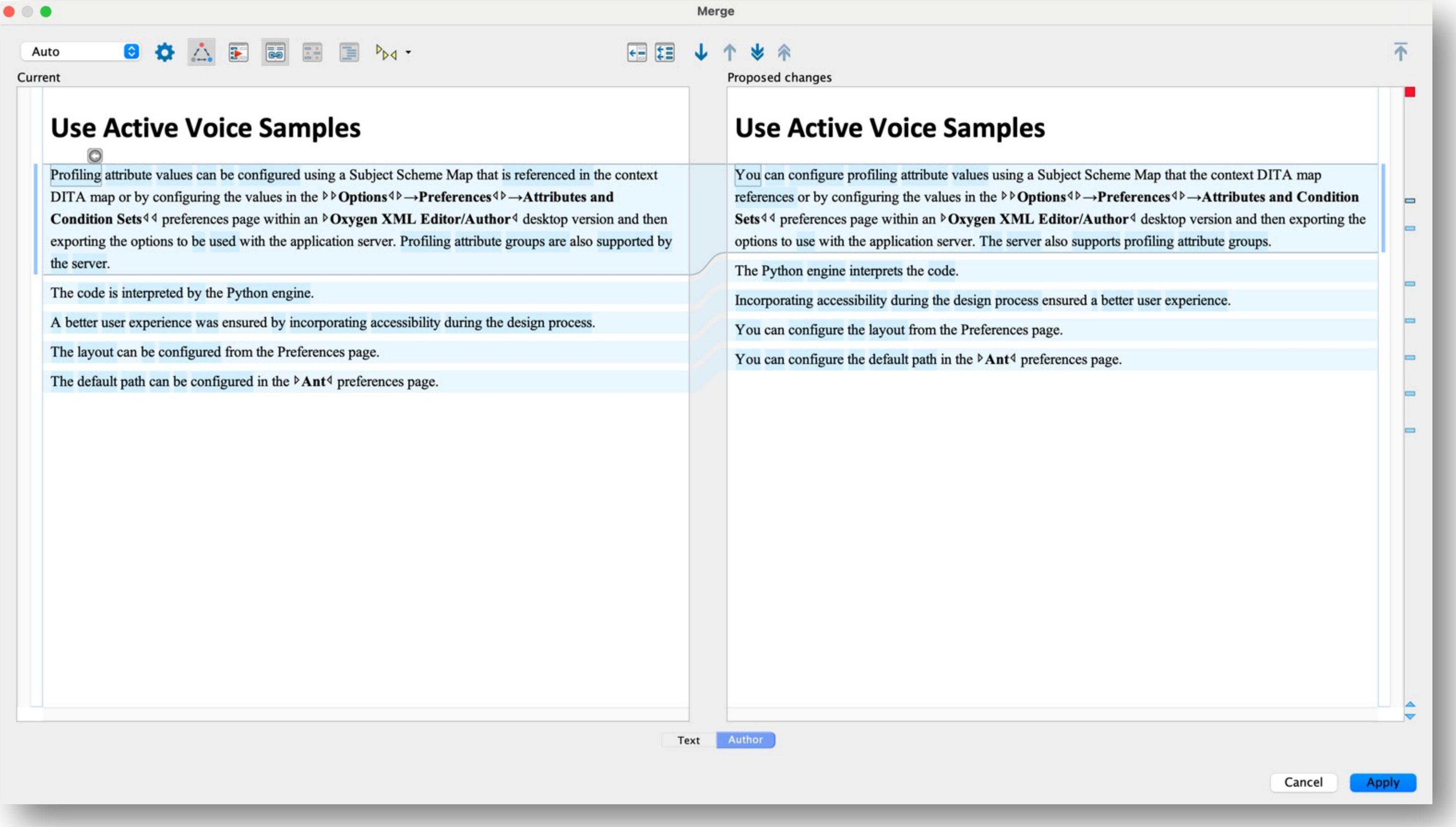
You can select the preference page you are interested in from the tree on the left of the **Preferences** dialog box. You can filter the tree by using the filter text box and then the following buttons are available to the right of the text box:

- **Expand All** - Expands the structure of the tree to show all preference pages.
- **Collapse All** - Collapses the structure of the tree to show only the 1st level preference pages.
- **Project-Level Options Only** - If toggled on, it filters the tree to only show the preference pages that are saved at the project level.

Some preference pages include an option to control how the options are stored either as **Global Options** or **Project Options**.

You can restore options to their default values by pressing the **Restore Defaults** button available in each preferences page.

Use active voice



Current

Use Active Voice Samples

Profiling attribute values can be configured using a Subject Scheme Map that is referenced in the context DITA map or by configuring the values in the **Options**→**Preferences**→**Attributes and Condition Sets** preferences page within an **Oxygen XML Editor/Author** desktop version and then exporting the options to be used with the application server. Profiling attribute groups are also supported by the server.

The code is interpreted by the Python engine.

A better user experience was ensured by incorporating accessibility during the design process.

The layout can be configured from the Preferences page.

The default path can be configured in the **Ant** preferences page.

Proposed changes

Use Active Voice Samples

You can configure profiling attribute values using a Subject Scheme Map that the context DITA map references or by configuring the values in the **Options**→**Preferences**→**Attributes and Condition Sets** preferences page within an **Oxygen XML Editor/Author** desktop version and then exporting the options to use with the application server. The server also supports profiling attribute groups.

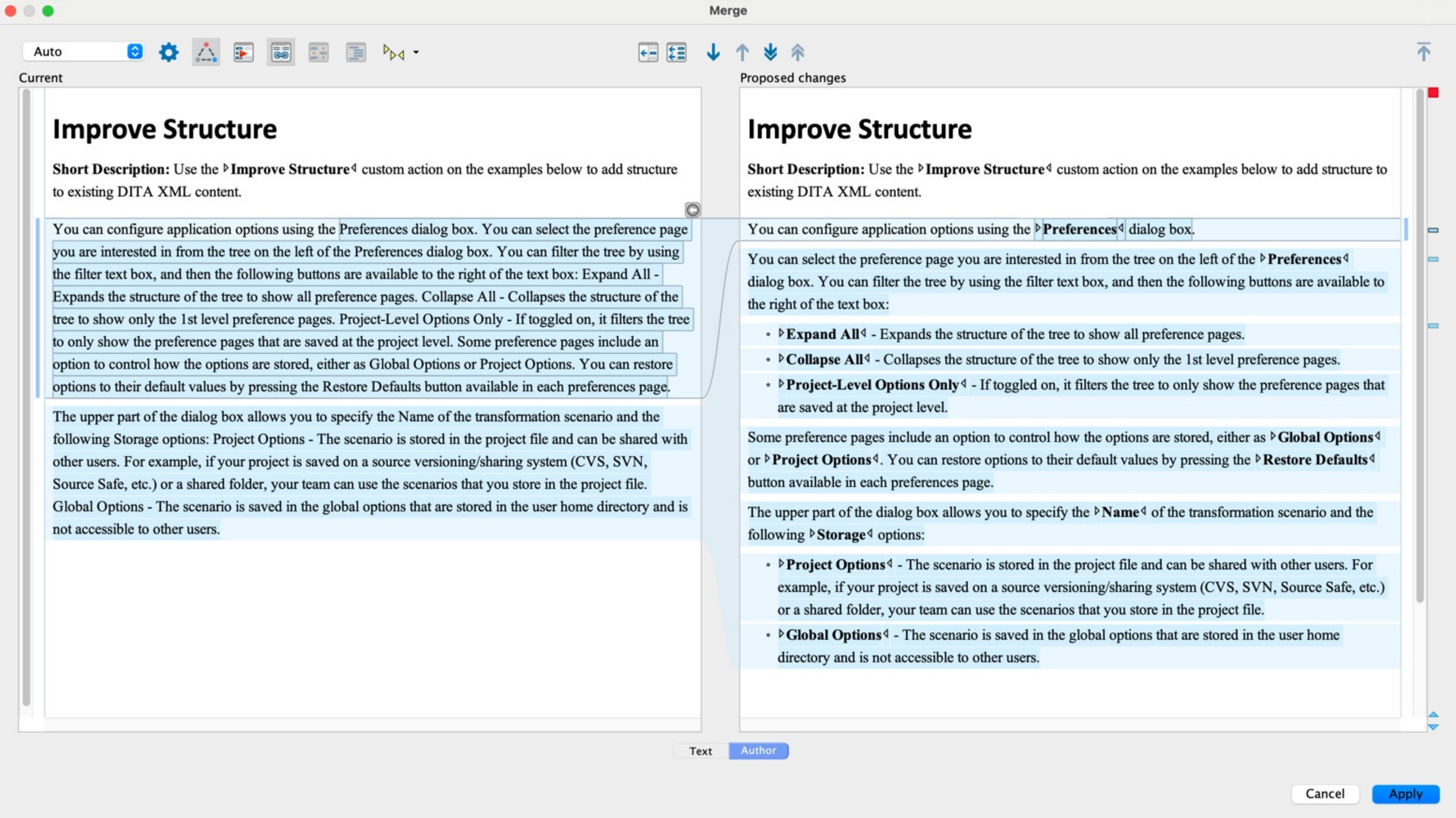
The Python engine interprets the code.

Incorporating accessibility during the design process ensured a better user experience.

You can configure the layout from the Preferences page.

You can configure the default path in the **Ant** preferences page.

Improve structure



Current

Improve Structure

Short Description: Use the **Improve Structure** custom action on the examples below to add structure to existing DITA XML content.

You can configure application options using the Preferences dialog box. You can select the preference page you are interested in from the tree on the left of the Preferences dialog box. You can filter the tree by using the filter text box, and then the following buttons are available to the right of the text box: Expand All - Expands the structure of the tree to show all preference pages. Collapse All - Collapses the structure of the tree to show only the 1st level preference pages. Project-Level Options Only - If toggled on, it filters the tree to only show the preference pages that are saved at the project level. Some preference pages include an option to control how the options are stored, either as Global Options or Project Options. You can restore options to their default values by pressing the Restore Defaults button available in each preferences page.

The upper part of the dialog box allows you to specify the Name of the transformation scenario and the following Storage options: Project Options - The scenario is stored in the project file and can be shared with other users. For example, if your project is saved on a source versioning/sharing system (CVS, SVN, Source Safe, etc.) or a shared folder, your team can use the scenarios that you store in the project file. Global Options - The scenario is saved in the global options that are stored in the user home directory and is not accessible to other users.

Proposed changes

Improve Structure

Short Description: Use the **Improve Structure** custom action on the examples below to add structure to existing DITA XML content.

You can configure application options using the **Preferences** dialog box.

You can select the preference page you are interested in from the tree on the left of the **Preferences** dialog box. You can filter the tree by using the filter text box, and then the following buttons are available to the right of the text box:

- **Expand All** - Expands the structure of the tree to show all preference pages.
- **Collapse All** - Collapses the structure of the tree to show only the 1st level preference pages.
- **Project-Level Options Only** - If toggled on, it filters the tree to only show the preference pages that are saved at the project level.

Some preference pages include an option to control how the options are stored, either as **Global Options** or **Project Options**. You can restore options to their default values by pressing the **Restore Defaults** button available in each preferences page.

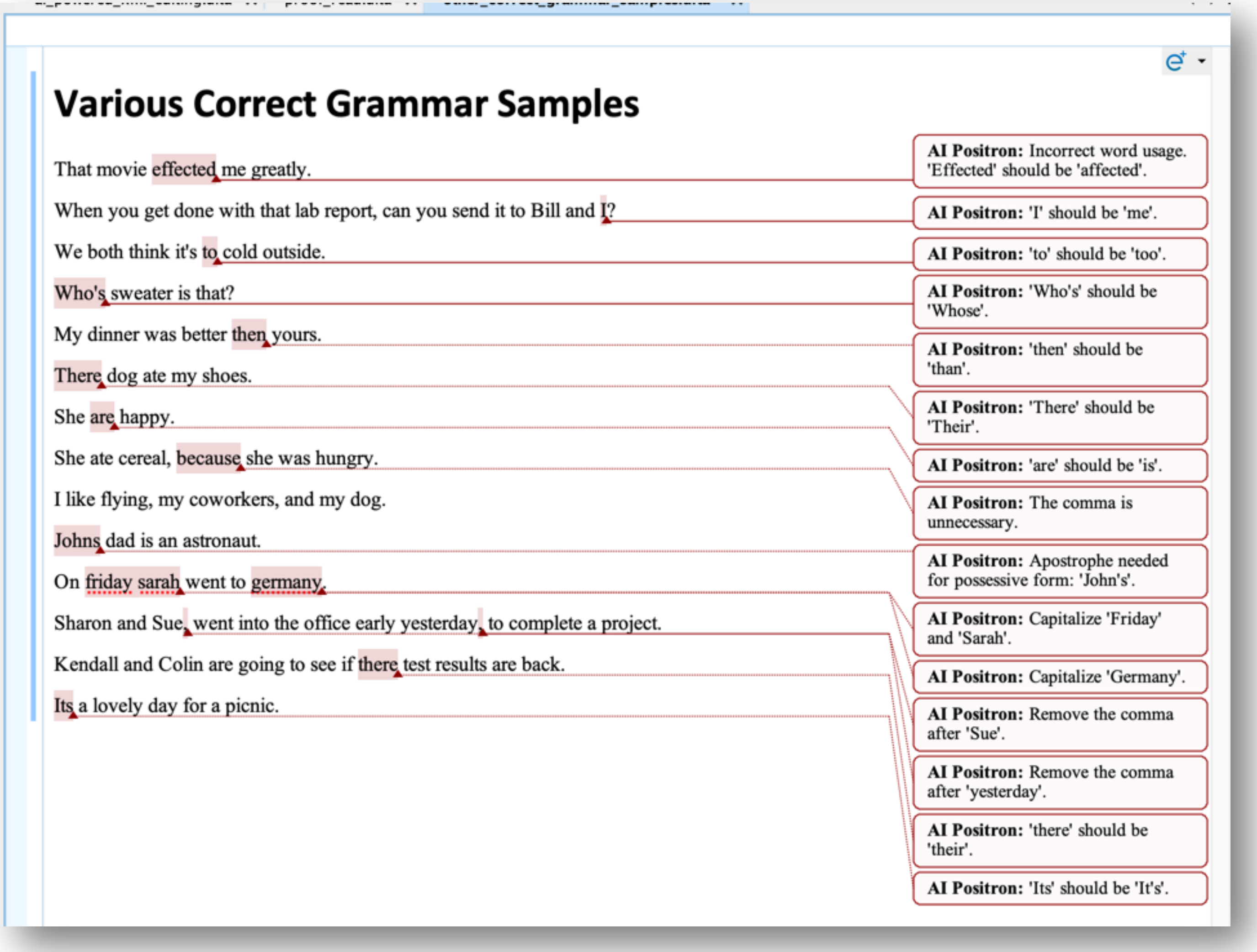
The upper part of the dialog box allows you to specify the **Name** of the transformation scenario and the following **Storage** options:

- **Project Options** - The scenario is stored in the project file and can be shared with other users. For example, if your project is saved on a source versioning/sharing system (CVS, SVN, Source Safe, etc.) or a shared folder, your team can use the scenarios that you store in the project file.
- **Global Options** - The scenario is saved in the global options that are stored in the user home directory and is not accessible to other users.

Review - AI integration via Oxygen comments

- Proofread
- Resolve comments

Proofread grammar issues



Various Correct Grammar Samples

That movie effected me greatly.

When you get done with that lab report, can you send it to Bill and I?

We both think it's to cold outside.

Who's sweater is that?

My dinner was better then yours.

There dog ate my shoes.

She are happy.

She ate cereal, because she was hungry.

I like flying, my coworkers, and my dog.

Johns dad is an astronaut.

On friday sarah went to germany.

Sharon and Sue, went into the office early yesterday, to complete a project.

Kendall and Colin are going to see if there test results are back.

Its a lovely day for a picnic.

AI Positron: Incorrect word usage. 'Effected' should be 'affected'.

AI Positron: 'I' should be 'me'.

AI Positron: 'to' should be 'too'.

AI Positron: 'Who's' should be 'Whose'.

AI Positron: 'then' should be 'than'.

AI Positron: 'There' should be 'Their'.

AI Positron: 'are' should be 'is'.

AI Positron: The comma is unnecessary.

AI Positron: Apostrophe needed for possessive form: 'John's'.

AI Positron: Capitalize 'Friday' and 'Sarah'.

AI Positron: Capitalize 'Germany'.

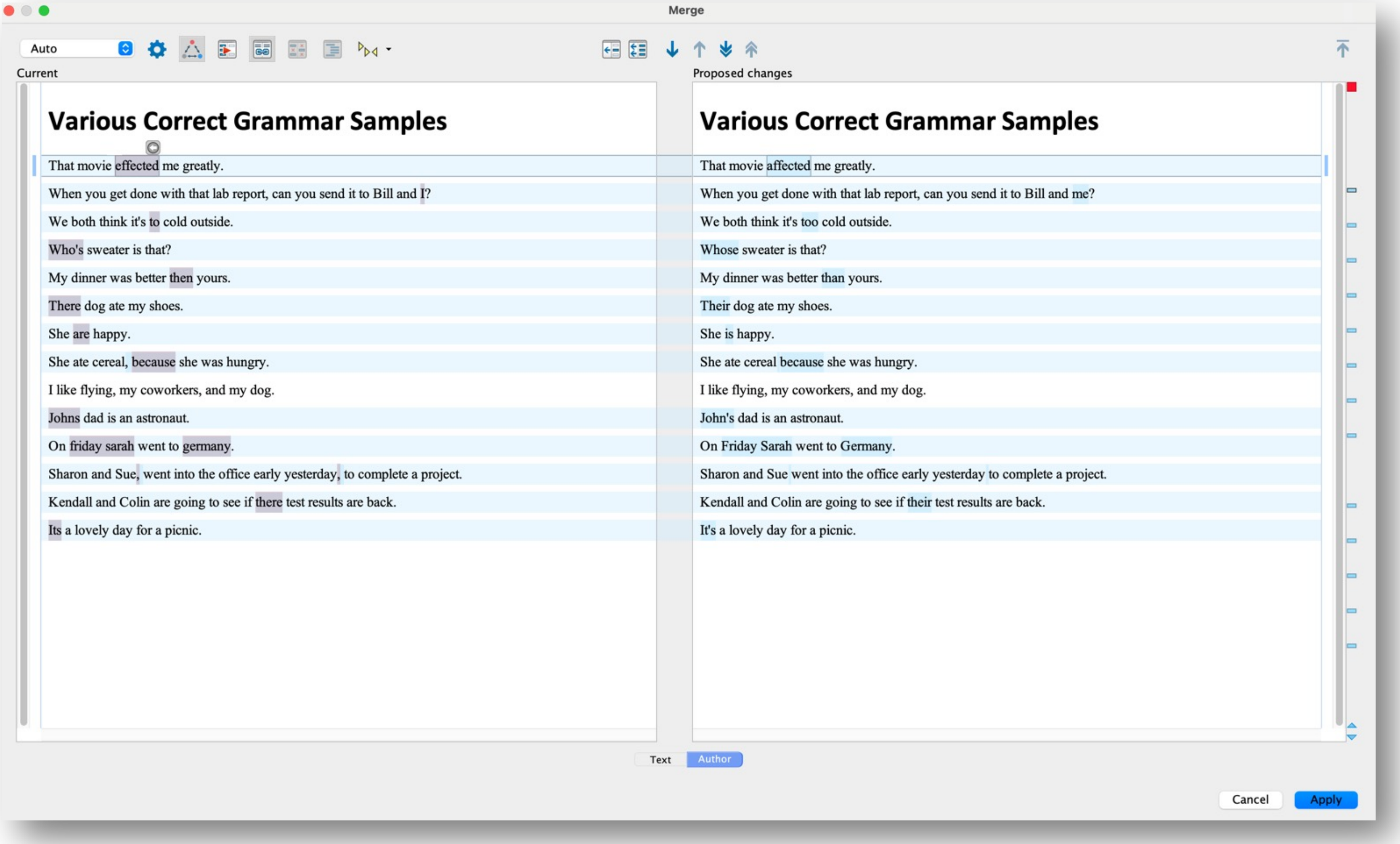
AI Positron: Remove the comma after 'Sue'.

AI Positron: Remove the comma after 'yesterday'.

AI Positron: 'there' should be 'their'.

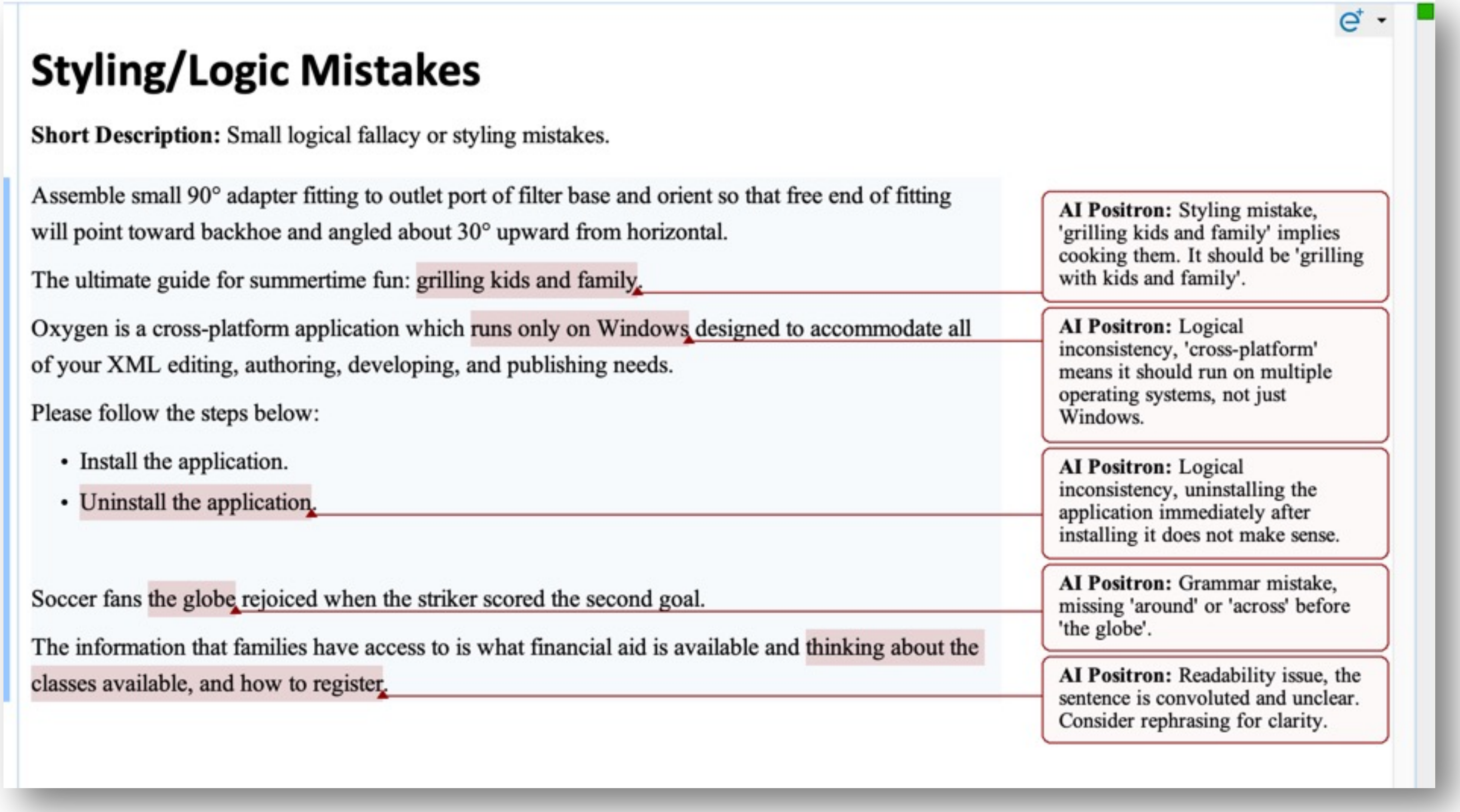
AI Positron: 'Its' should be 'It's'.

Resolve comments for grammar issues



Current	Proposed changes
Various Correct Grammar Samples	**Various Correct Grammar Samples**
That movie effected me greatly.	That movie affected me greatly.
When you get done with that lab report, can you send it to Bill and I?	When you get done with that lab report, can you send it to Bill and me?
We both think it's to cold outside.	We both think it's too cold outside.
Who's sweater is that?	Whose sweater is that?
My dinner was better then yours.	My dinner was better than yours.
There dog ate my shoes.	Their dog ate my shoes.
She are happy.	She is happy.
She ate cereal, because she was hungry.	She ate cereal because she was hungry.
I like flying, my coworkers, and my dog.	I like flying, my coworkers, and my dog.
Johns dad is an astronaut.	John's dad is an astronaut.
On friday sarah went to germany.	On Friday Sarah went to Germany.
Sharon and Sue, went into the office early yesterday, to complete a project.	Sharon and Sue went into the office early yesterday to complete a project.
Kendall and Colin are going to see if there test results are back.	Kendall and Colin are going to see if their test results are back.
Its a lovely day for a picnic.	It's a lovely day for a picnic.

Proofread styling/logical mistakes



Styling/Logic Mistakes

Short Description: Small logical fallacy or styling mistakes.

Assemble small 90° adapter fitting to outlet port of filter base and orient so that free end of fitting will point toward backhoe and angled about 30° upward from horizontal.

The ultimate guide for summertime fun: grilling kids and family.

Oxygen is a cross-platform application which runs only on Windows designed to accommodate all of your XML editing, authoring, developing, and publishing needs.

Please follow the steps below:

- Install the application.
- Uninstall the application.

Soccer fans the globe rejoiced when the striker scored the second goal.

The information that families have access to is what financial aid is available and thinking about the classes available, and how to register.

AI Positron: Styling mistake, 'grilling kids and family' implies cooking them. It should be 'grilling with kids and family'.

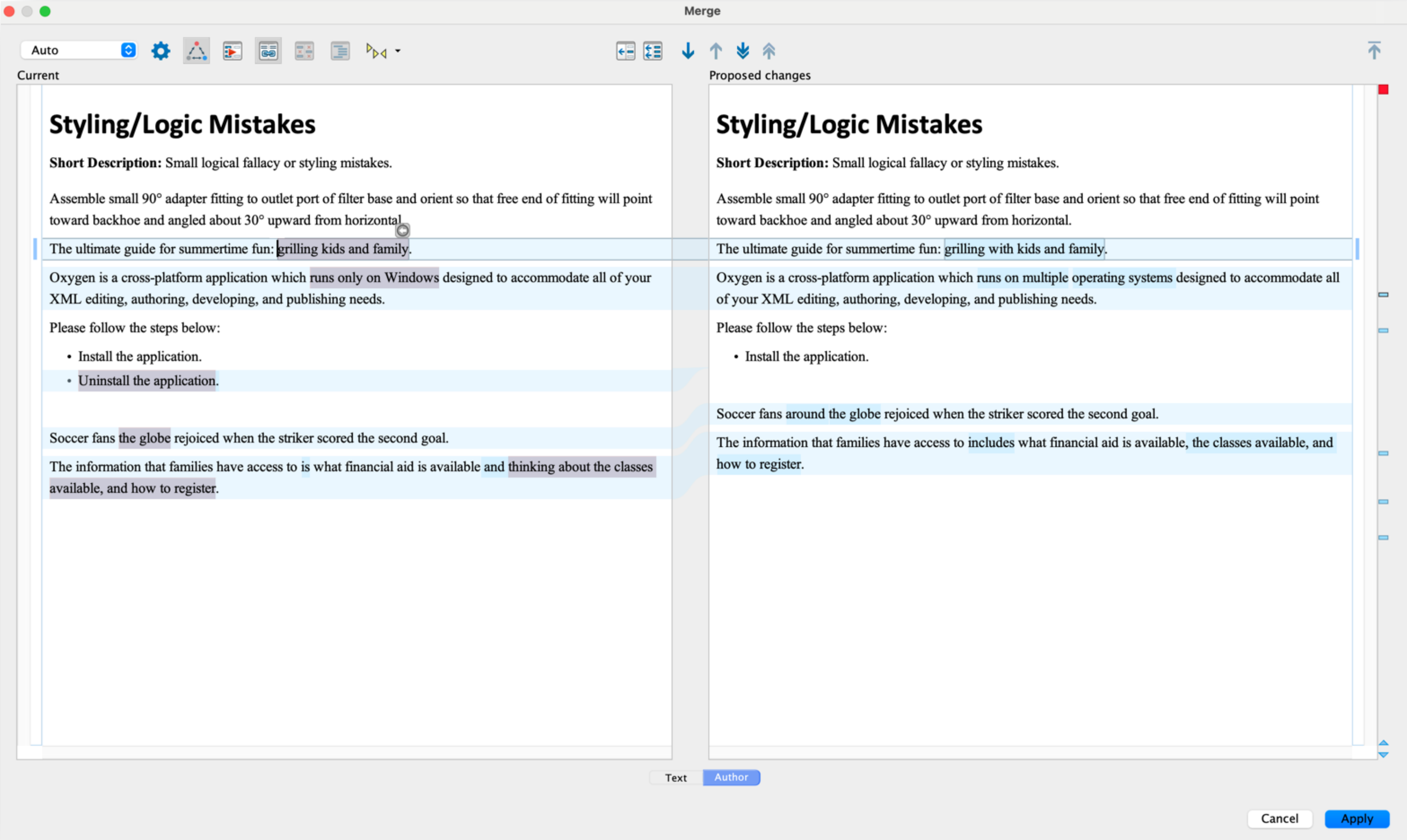
AI Positron: Logical inconsistency, 'cross-platform' means it should run on multiple operating systems, not just Windows.

AI Positron: Logical inconsistency, uninstalling the application immediately after installing it does not make sense.

AI Positron: Grammar mistake, missing 'around' or 'across' before 'the globe'.

AI Positron: Readability issue, the sentence is convoluted and unclear. Consider rephrasing for clarity.

Resolve comments for styling/logical mistakes



Current

Styling/Logic Mistakes

Short Description: Small logical fallacy or styling mistakes.

Assemble small 90° adapter fitting to outlet port of filter base and orient so that free end of fitting will point toward backhoe and angled about 30° upward from horizontal.

The ultimate guide for summertime fun: grilling kids and family.

Oxygen is a cross-platform application which runs only on Windows designed to accommodate all of your XML editing, authoring, developing, and publishing needs.

Please follow the steps below:

- Install the application.
- Uninstall the application.

Soccer fans the globe rejoiced when the striker scored the second goal.

The information that families have access to is what financial aid is available and thinking about the classes available, and how to register.

Proposed changes

Styling/Logic Mistakes

Short Description: Small logical fallacy or styling mistakes.

Assemble small 90° adapter fitting to outlet port of filter base and orient so that free end of fitting will point toward backhoe and angled about 30° upward from horizontal.

The ultimate guide for summertime fun: grilling with kids and family.

Oxygen is a cross-platform application which runs on multiple operating systems designed to accommodate all of your XML editing, authoring, developing, and publishing needs.

Please follow the steps below:

- Install the application.

Soccer fans around the globe rejoiced when the striker scored the second goal.

The information that families have access to includes what financial aid is available, the classes available, and how to register.

Overview actions

- Answer questions
- Generate questions
- Summarize
- Readability report

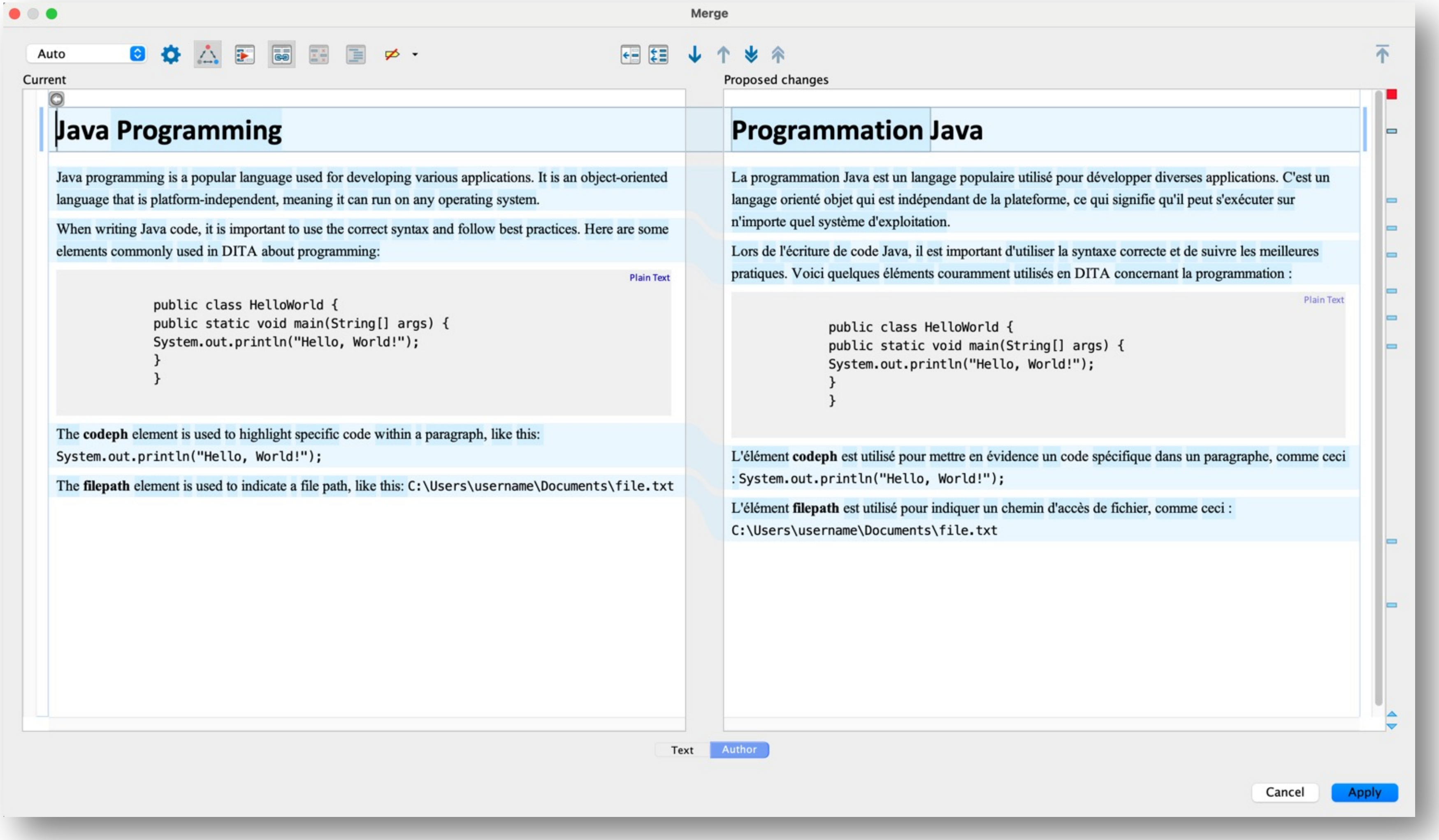
Generate and answer questions

- What are the main questions your topic responds to
- What answers an AI-based bot will provide to questions if this topic will be added to its context (in a Retrieval-Augmented Generation scenario)

Translation actions

- English
- French
- German
- Japanese
- Other languages...

Translation from English to French



Current

Java Programming

Java programming is a popular language used for developing various applications. It is an object-oriented language that is platform-independent, meaning it can run on any operating system.

When writing Java code, it is important to use the correct syntax and follow best practices. Here are some elements commonly used in DITA about programming:

```
public class HelloWorld {
public static void main(String[] args) {
System.out.println("Hello, World!");
}
}
```

The **codeph** element is used to highlight specific code within a paragraph, like this: `System.out.println("Hello, World!");`

The **filepath** element is used to indicate a file path, like this: `C:\Users\username\Documents\file.txt`

Proposed changes

Programmation Java

La programmation Java est un langage populaire utilisé pour développer diverses applications. C'est un langage orienté objet qui est indépendant de la plateforme, ce qui signifie qu'il peut s'exécuter sur n'importe quel système d'exploitation.

Lors de l'écriture de code Java, il est important d'utiliser la syntaxe correcte et de suivre les meilleures pratiques. Voici quelques éléments couramment utilisés en DITA concernant la programmation :

```
public class HelloWorld {
public static void main(String[] args) {
System.out.println("Hello, World!");
}
}
```

L'élément **codeph** est utilisé pour mettre en évidence un code spécifique dans un paragraphe, comme ceci : `System.out.println("Hello, World!");`

L'élément **filepath** est utilisé pour indiquer un chemin d'accès de fichier, comme ceci : `C:\Users\username\Documents\file.txt`

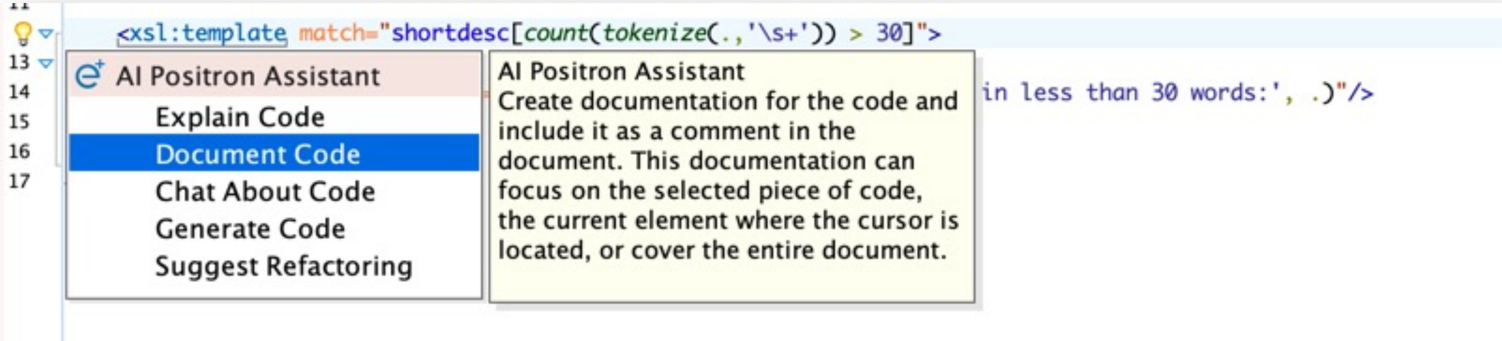
Marketing actions

- Release notes
- Marketing post
- Improve SEO
- Pain-Agitate-Solution
- Features-Advantages-Benefits

Development actions

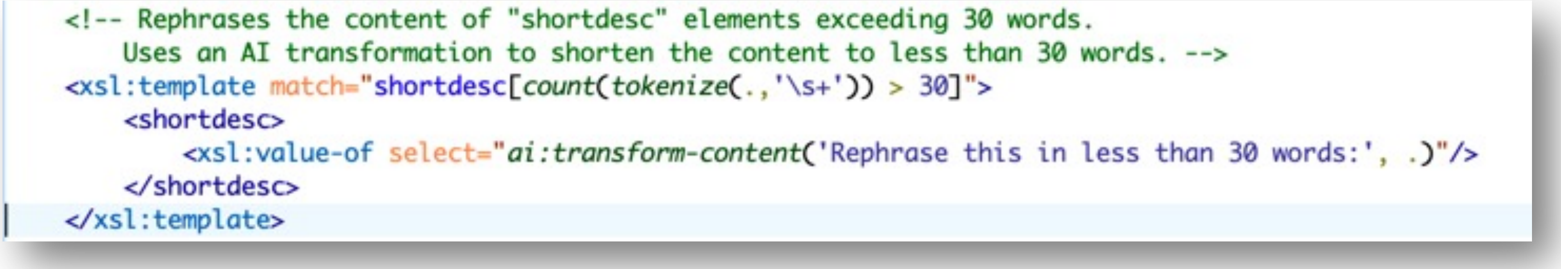
- Explain code
- Document code
- Suggest Refactoring
- Chat about code
- Generate code

Document code



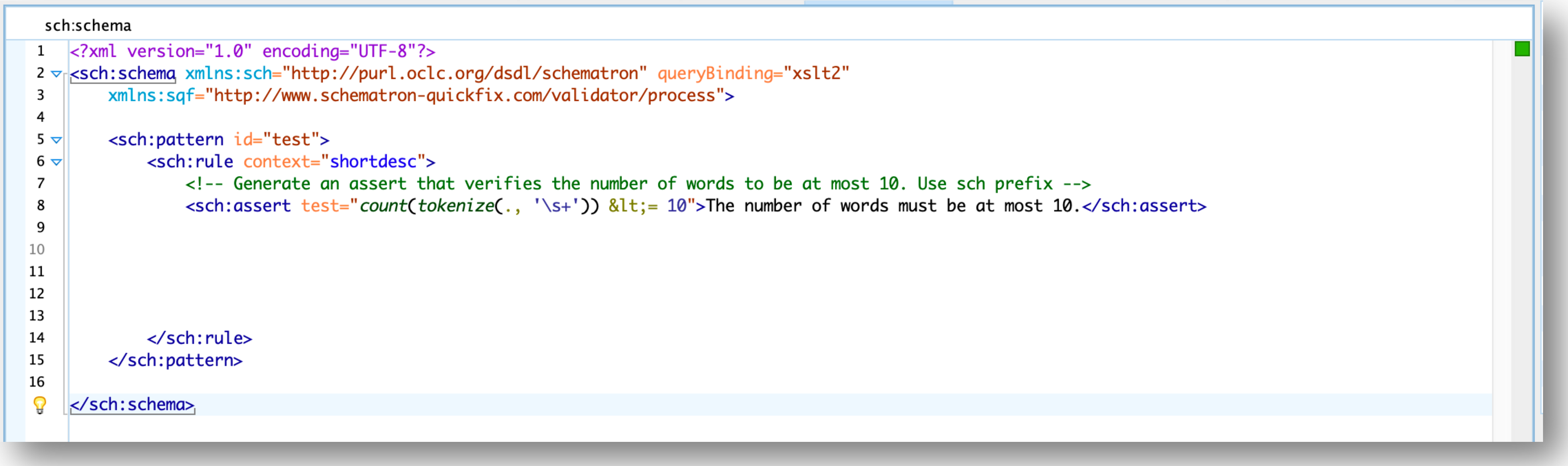
```
<xsl:template match="shortdesc[count(tokenize(.,'\s+')) > 30]">
                                                      in less than 30 words:', .)"/>
```

AI Positron Assistant
Create documentation for the code and include it as a comment in the document. This documentation can focus on the selected piece of code, the current element where the cursor is located, or cover the entire document.



```
<!-- Rephrases the content of "shortdesc" elements exceeding 30 words.
    Uses an AI transformation to shorten the content to less than 30 words. -->
<xsl:template match="shortdesc[count(tokenize(.,'\s+')) > 30]">
    <shortdesc>
        <xsl:value-of select="ai:transform-content('Rephrase this in less than 30 words:', .)"/>
    </shortdesc>
</xsl:template>
```

Generate code



```
<?xml version="1.0" encoding="UTF-8"?>
<sch:schema xmlns:sch="http://purl.oclc.org/dsdl/schematron" queryBinding="xslt2"
    xmlns:sqf="http://www.schematron-quickfix.com/validator/process">

    <sch:pattern id="test">
        <sch:rule context="shortdesc">
            <!-- Generate an assert that verifies the number of words to be at most 10. Use sch prefix -->
            <sch:assert test="count(tokenize(., '\s+')) &lt;= 10">The number of words must be at most 10.</sch:assert>





        </sch:rule>
    </sch:pattern>

</sch:schema>
```

Define AI actions (or prompts) by examples

Automatically record the changes to a document and use them as few shots samples to define AI actions

- Describe the changes you want to make
- Make the changes to a document
- Save as custom action or favorite prompt

Other integration points

Ai Fixes

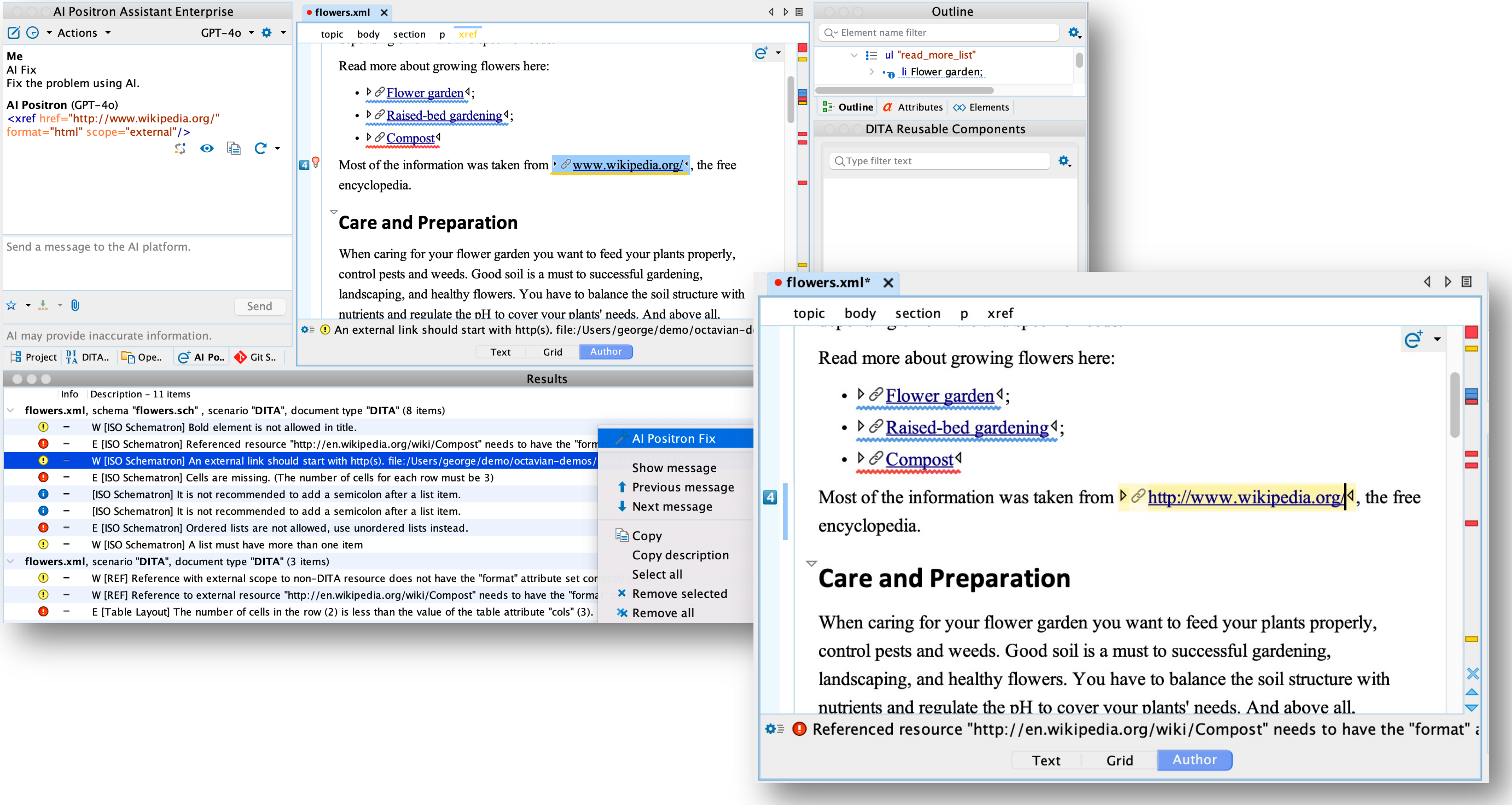
AI Refactoring

XPath AI-based functions

AI Fixes

- Automatically propose solutions to error messages by the AI model
- On any error we can ask the AI model to provide a fix
- Schematron error messages are best candidates, because the error message is written by the schema author for the user

AI Positron Fix



Me
AI Fix
Fix the problem using AI.

AI Positron (GPT-4o)

```
<xref href="http://www.wikipedia.org/"
format="html" scope="external"/>
```

Send a message to the AI platform.

AI may provide inaccurate information.

Read more about growing flowers here:

- Flower garden;
- Raised-bed gardening;
- Compost

Most of the information was taken from www.wikipedia.org/, the free encyclopedia.

Care and Preparation

When caring for your flower garden you want to feed your plants properly, control pests and weeds. Good soil is a must to successful gardening, landscaping, and healthy flowers. You have to balance the soil structure with nutrients and regulate the pH to cover your plants' needs. And above all.

An external link should start with http(s). file:/Users/george/demo/octavian-d

flowers.xml, schema "flowers.sch" , scenario "DITA", document type "DITA" (8 items)

- W [ISO Schematron] Bold element is not allowed in title.
- E [ISO Schematron] Referenced resource "http://en.wikipedia.org/wiki/Compost" needs to have the "form
- W [ISO Schematron] An external link should start with http(s). file:/Users/george/demo/octavian-demos/
- E [ISO Schematron] Cells are missing. (The number of cells for each row must be 3)
- [ISO Schematron] It is not recommended to add a semicolon after a list item.
- [ISO Schematron] It is not recommended to add a semicolon after a list item.
- E [ISO Schematron] Ordered lists are not allowed, use unordered lists instead.
- W [ISO Schematron] A list must have more than one item

flowers.xml, scenario "DITA", document type "DITA" (3 items)

- W [REF] Reference with external scope to non-DITA resource does not have the "format" attribute set cor
- W [REF] Reference to external resource "http://en.wikipedia.org/wiki/Compost" needs to have the "forma
- E [Table Layout] The number of cells in the row (2) is less than the value of the table attribute "cols" (3).

Read more about growing flowers here:

- Flower garden;
- Raised-bed gardening;
- Compost

Most of the information was taken from http://www.wikipedia.org/, the free encyclopedia.

Care and Preparation

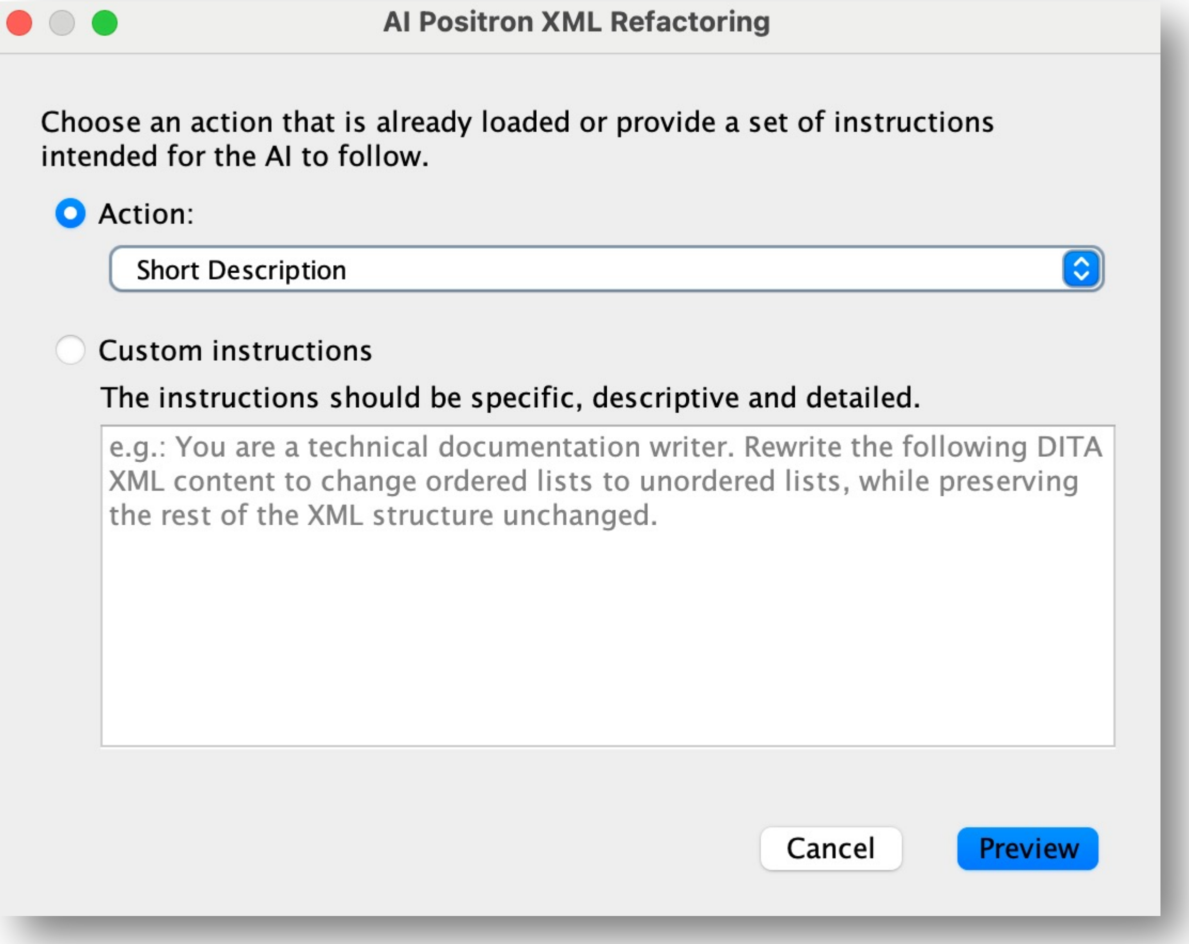
When caring for your flower garden you want to feed your plants properly, control pests and weeds. Good soil is a must to successful gardening, landscaping, and healthy flowers. You have to balance the soil structure with nutrients and regulate the pH to cover your plants' needs. And above all.

Referenced resource "http://en.wikipedia.org/wiki/Compost" needs to have the "format" a

AI Refactoring

- Apply any AI action on a set of documents
- Preview proposed changes and be able to exclude the ones you do not want

Trigger AI Refactoring



AI Positron XML Refactoring

Choose an action that is already loaded or provide a set of instructions intended for the AI to follow.

● Action:

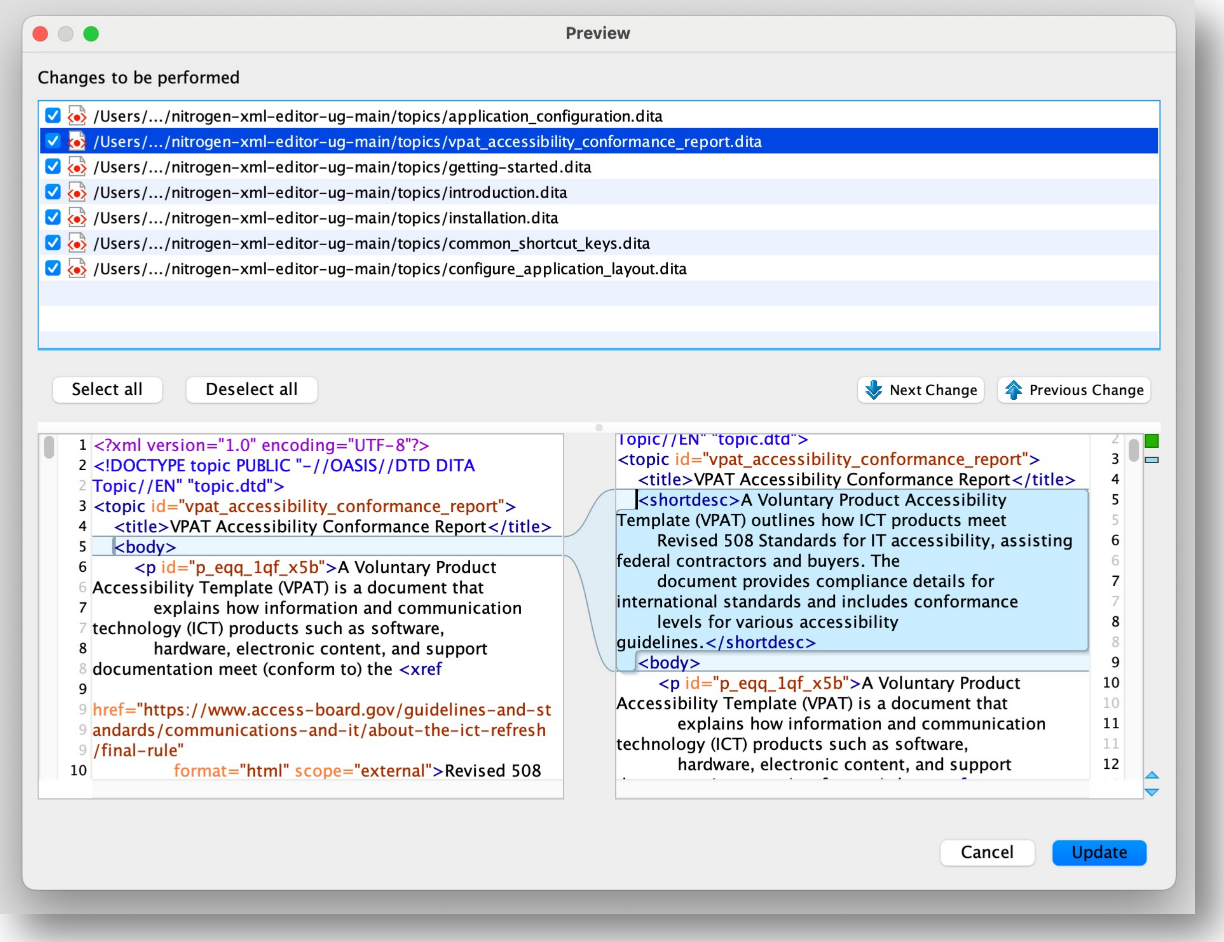
Short Description

○ Custom instructions

The instructions should be specific, descriptive and detailed.

e.g.: You are a technical documentation writer. Rewrite the following DITA XML content to change ordered lists to unordered lists, while preserving the rest of the XML structure unchanged.

Preview AI Refactoring



- ☑ /Users/…/nitrogen-xml-editor-ug-main/topics/application_configuration.dita
- ☑ /Users/…/nitrogen-xml-editor-ug-main/topics/vpat_accessibility_conformance_report.dita
- ☑ /Users/…/nitrogen-xml-editor-ug-main/topics/getting-started.dita
- ☑ /Users/…/nitrogen-xml-editor-ug-main/topics/introduction.dita
- ☑ /Users/…/nitrogen-xml-editor-ug-main/topics/installation.dita
- ☑ /Users/…/nitrogen-xml-editor-ug-main/topics/common_shortcut_keys.dita
- ☑ /Users/…/nitrogen-xml-editor-ug-main/topics/configure_application_layout.dita

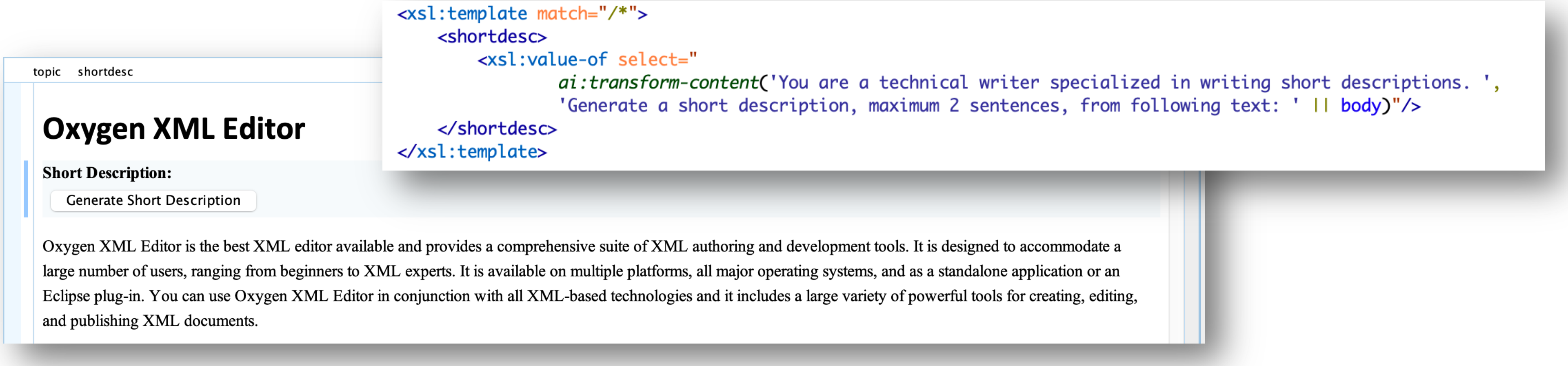
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD DITA
Topic//EN" "topic.dtd">
<topic id="vpat_accessibility_conformance_report">
   <title>VPAT Accessibility Conformance Report</title>
   <body>
      <p id="p_eqq_1qf_x5b">A Voluntary Product
Accessibility Template (VPAT) is a document that
         explains how information and communication
technology (ICT) products such as software,
         hardware, electronic content, and support
documentation meet (conform to) the <xref

href="https://www.access-board.gov/guidelines-and-st
andards/communications-and-it/about-the-ict-refresh
/final-rule"
            format="html" scope="external">Revised 508
```

```
Topic//EN" "topic.dtd">
<topic id="vpat_accessibility_conformance_report">
   <title>VPAT Accessibility Conformance Report</title>
   <shortdesc>A Voluntary Product Accessibility
Template (VPAT) outlines how ICT products meet
      Revised 508 Standards for IT accessibility, assisting
federal contractors and buyers. The
      document provides compliance details for
international standards and includes conformance
      levels for various accessibility
guidelines.</shortdesc>
   <body>
      <p id="p_eqq_1qf_x5b">A Voluntary Product
Accessibility Template (VPAT) is a document that
         explains how information and communication
technology (ICT) products such as software,
         hardware, electronic content, and support
```

XPath AI-based functions

- Interact with AI models from any language that uses XPath, such as
  - XSLT
  - XQuery
  - XQuery Update
  - Schematron
  - Schematron Quick Fix
- Enables interesting use cases, such as:
  - Use AI from custom actions that can be triggered by users in visual editing mode - these custom actions can be defined in many ways, including by triggering XSLT or XQuery scripts
  - Use AI from custom XML refactoring actions, the refactoring actions are based on XSLT or XQuery Update scripts
  - Use AI in Schematron to check various conditions - Schematron uses XPath for tests
  - Use AI in quick fixes defined by Schematron Quick Fix, this also uses XPath as its expression language

AI-based custom Author actions

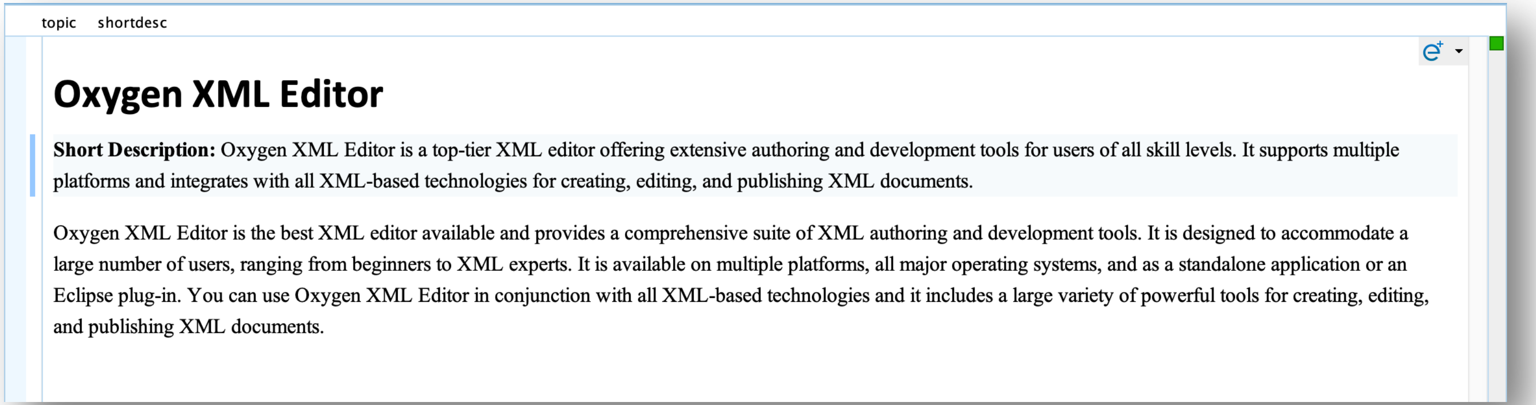


```
<xsl:template match="/*">
    <shortdesc>
        <xsl:value-of select="
                ai:transform-content('You are a technical writer specialized in writing short descriptions. ',
                'Generate a short description, maximum 2 sentences, from following text: ' || body)"/>
    </shortdesc>
</xsl:template>
```

Oxygen XML Editor

Short Description:

Oxygen XML Editor is the best XML editor available and provides a comprehensive suite of XML authoring and development tools. It is designed to accommodate a large number of users, ranging from beginners to XML experts. It is available on multiple platforms, all major operating systems, and as a standalone application or an Eclipse plug-in. You can use Oxygen XML Editor in conjunction with all XML-based technologies and it includes a large variety of powerful tools for creating, editing, and publishing XML documents.

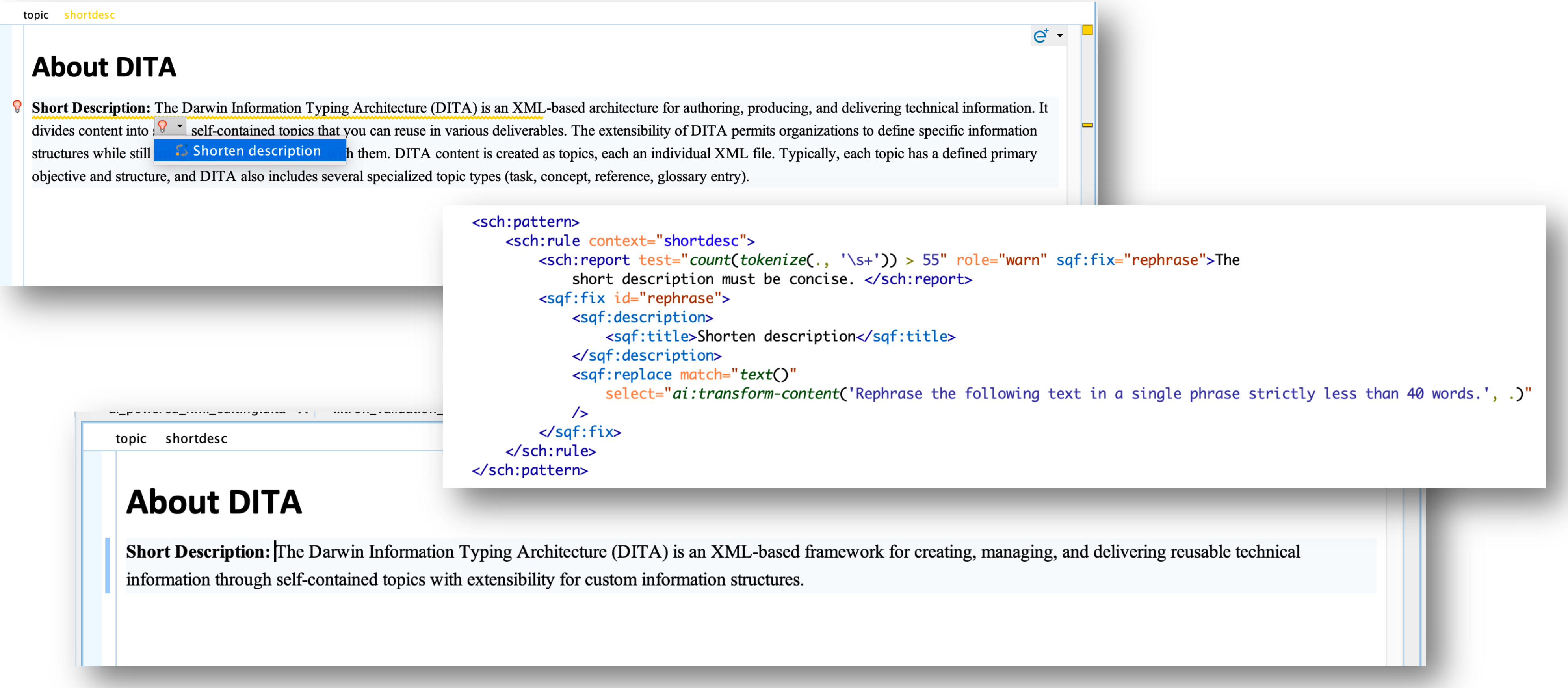


Oxygen XML Editor

Short Description: Oxygen XML Editor is a top-tier XML editor offering extensive authoring and development tools for users of all skill levels. It supports multiple platforms and integrates with all XML-based technologies for creating, editing, and publishing XML documents.

Oxygen XML Editor is the best XML editor available and provides a comprehensive suite of XML authoring and development tools. It is designed to accommodate a large number of users, ranging from beginners to XML experts. It is available on multiple platforms, all major operating systems, and as a standalone application or an Eclipse plug-in. You can use Oxygen XML Editor in conjunction with all XML-based technologies and it includes a large variety of powerful tools for creating, editing, and publishing XML documents.

Schematron Quick Fix with AI



About DITA

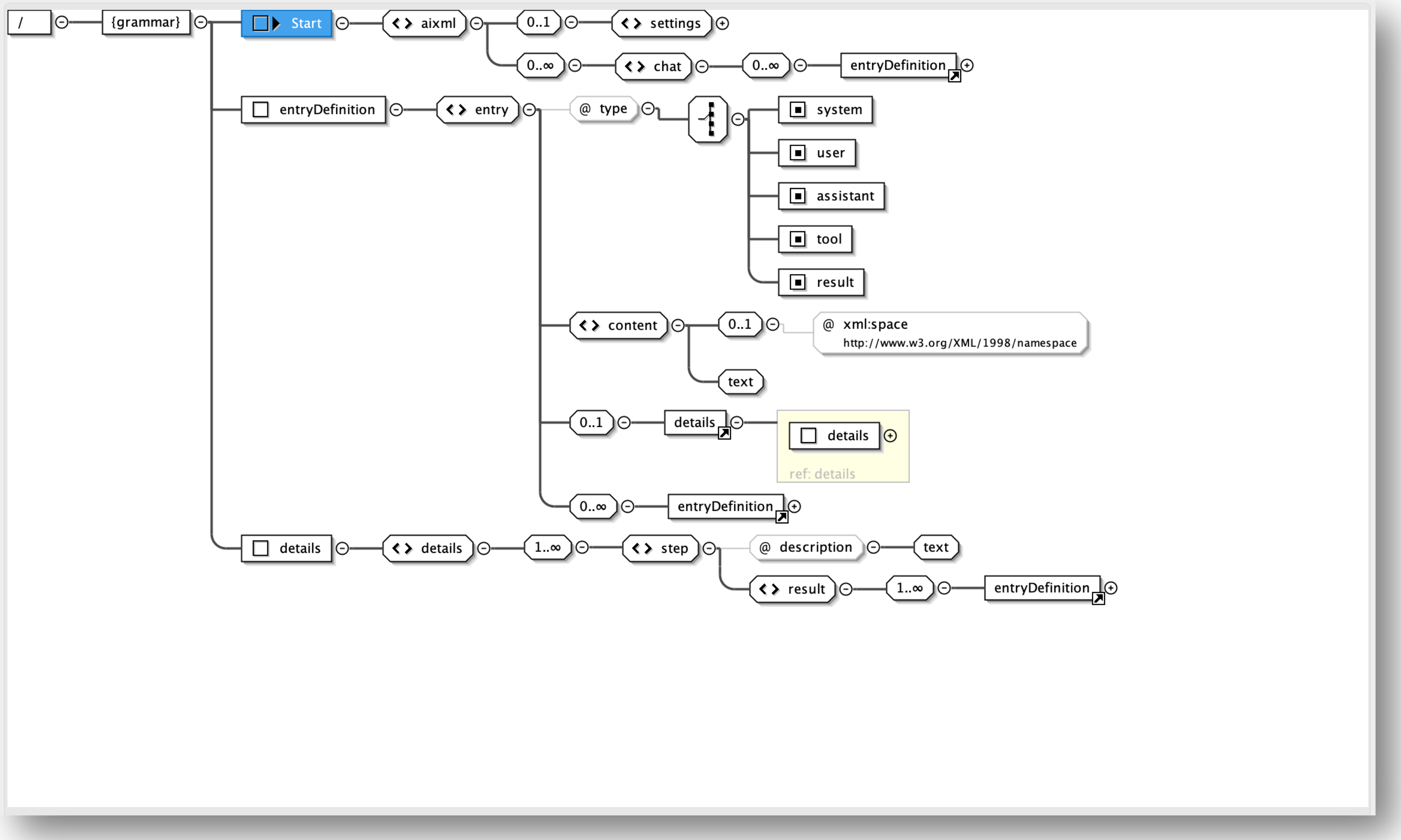
Short Description: The Darwin Information Typing Architecture (DITA) is an XML-based architecture for authoring, producing, and delivering technical information. It divides content into s self-contained topics that you can reuse in various deliverables. The extensibility of DITA permits organizations to define specific information structures while still h them. DITA content is created as topics, each an individual XML file. Typically, each topic has a defined primary objective and structure, and DITA also includes several specialized topic types (task, concept, reference, glossary entry).

```
<sch:pattern>
    <sch:rule context="shortdesc">
        <sch:report test="count(tokenize(., '\s+')) > 55" role="warn" sqf:fix="rephrase">The
            short description must be concise. </sch:report>
        <sqf:fix id="rephrase">
            <sqf:description>
                <sqf:title>Shorten description</sqf:title>
            </sqf:description>
            <sqf:replace match="text()"
                select="ai:transform-content('Rephrase the following text in a single phrase strictly less than 40 words.', .)"
            />
        </sqf:fix>
    </sch:rule>
</sch:pattern>
```

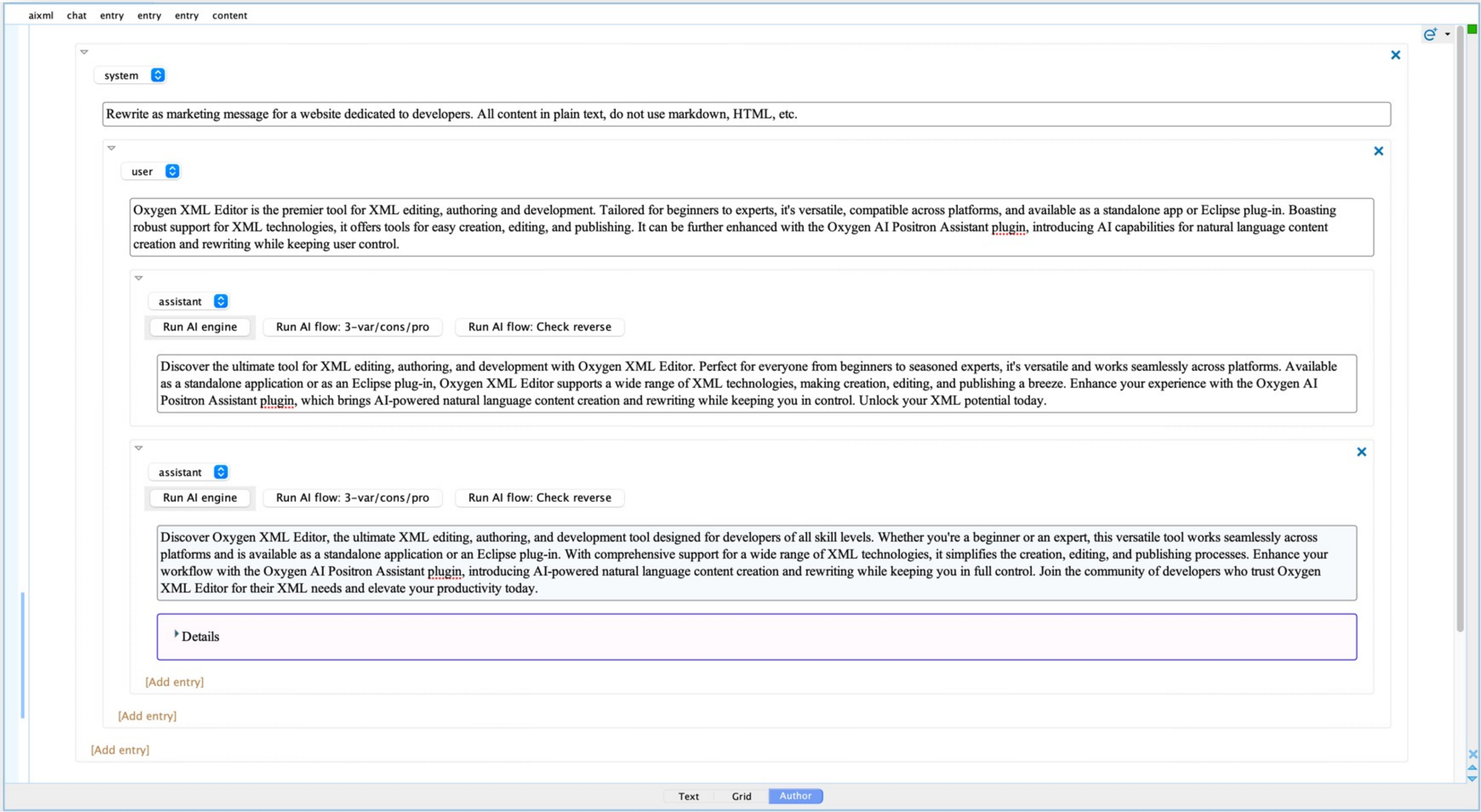
About DITA

Short Description: The Darwin Information Typing Architecture (DITA) is an XML-based framework for creating, managing, and delivering reusable technical information through self-contained topics with extensibility for custom information structures.

Implement a chat framework – AIXML language



Implement a chat framework



Rewrite as marketing message for a website dedicated to developers. All content in plain text, do not use markdown, HTML, etc.

Oxygen XML Editor is the premier tool for XML editing, authoring and development. Tailored for beginners to experts, it's versatile, compatible across platforms, and available as a standalone app or Eclipse plug-in. Boasting robust support for XML technologies, it offers tools for easy creation, editing, and publishing. It can be further enhanced with the Oxygen AI Positron Assistant plugin, introducing AI capabilities for natural language content creation and rewriting while keeping user control.

Discover the ultimate tool for XML editing, authoring, and development with Oxygen XML Editor. Perfect for everyone from beginners to seasoned experts, it's versatile and works seamlessly across platforms. Available as a standalone application or as an Eclipse plug-in, Oxygen XML Editor supports a wide range of XML technologies, making creation, editing, and publishing a breeze. Enhance your experience with the Oxygen AI Positron Assistant plugin, which brings AI-powered natural language content creation and rewriting while keeping you in control. Unlock your XML potential today.

Discover Oxygen XML Editor, the ultimate XML editing, authoring, and development tool designed for developers of all skill levels. Whether you're a beginner or an expert, this versatile tool works seamlessly across platforms and is available as a standalone application or an Eclipse plug-in. With comprehensive support for a wide range of XML technologies, it simplifies the creation, editing, and publishing processes. Enhance your workflow with the Oxygen AI Positron Assistant plugin, introducing AI-powered natural language content creation and rewriting while keeping you in full control. Join the community of developers who trust Oxygen XML Editor for their XML needs and elevate your productivity today.

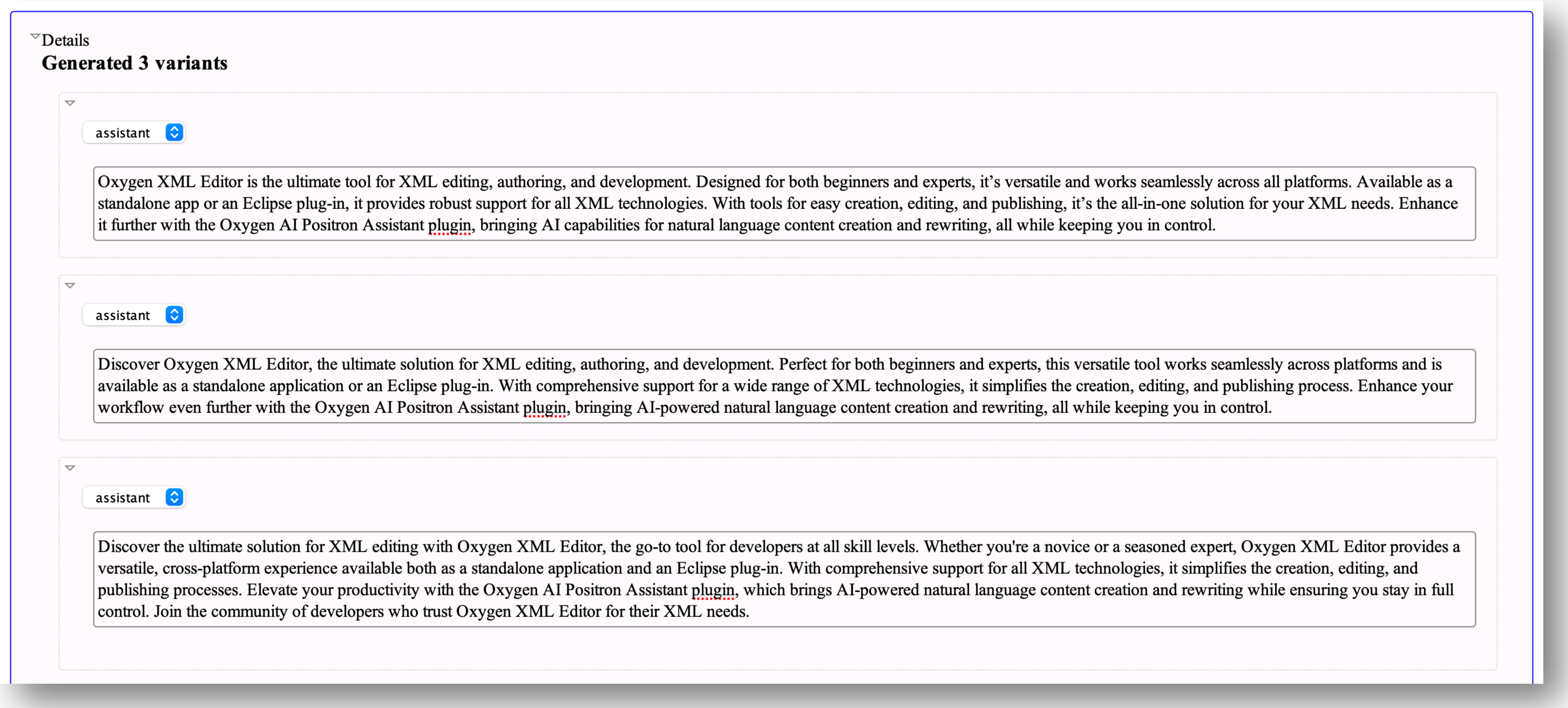
Conversation workflow

- Generate alternative responses
- Determine cons/pros for each
- Select the best response from the 3 variants based on pros and cons
- Explain how the best response was obtained

Implementation

- with the chat framework using multiple interactions with the AI model
- using structured output support

Workflow using multiple interactions



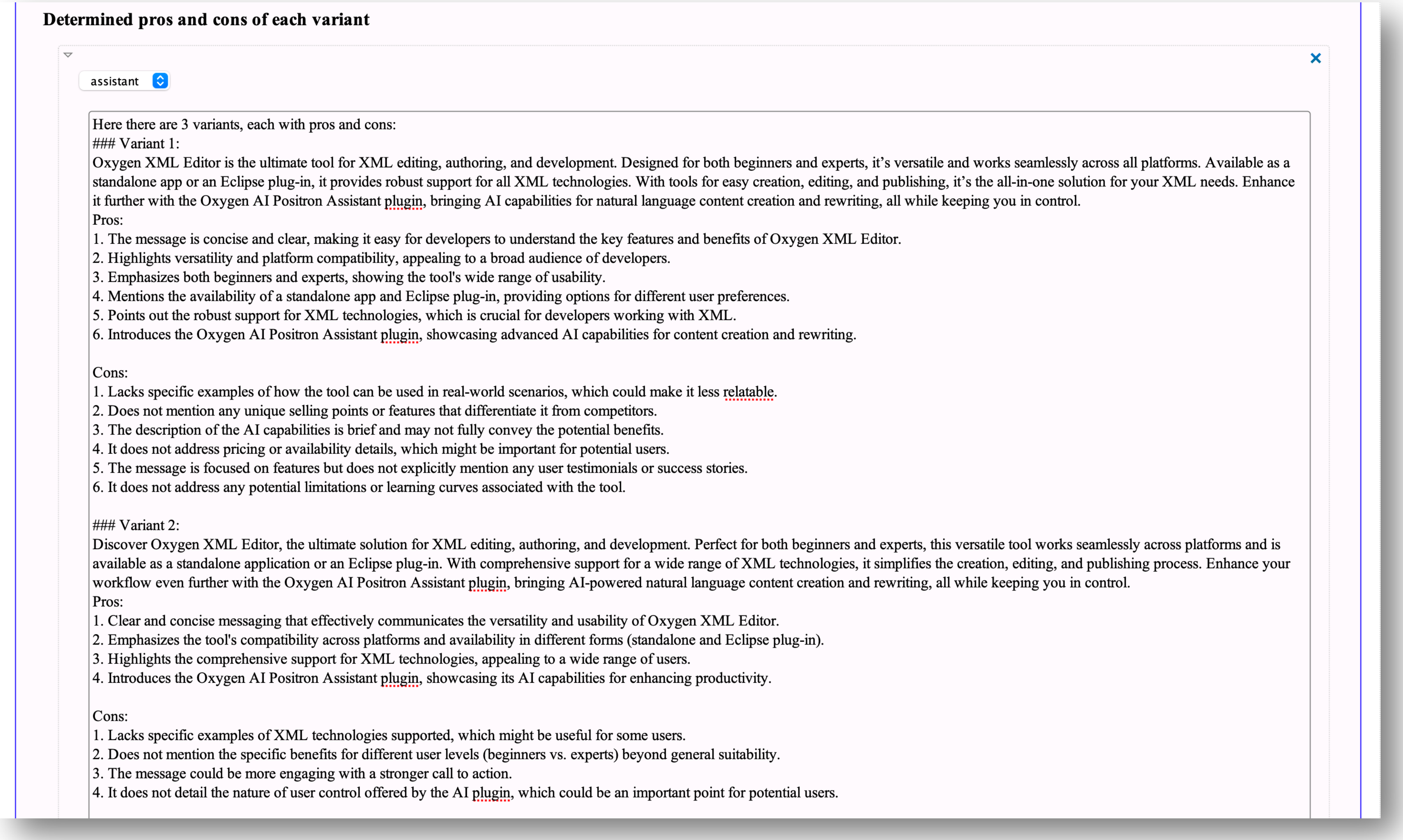
Details

Generated 3 variants

Oxygen XML Editor is the ultimate tool for XML editing, authoring, and development. Designed for both beginners and experts, it’s versatile and works seamlessly across all platforms. Available as a standalone app or an Eclipse plug-in, it provides robust support for all XML technologies. With tools for easy creation, editing, and publishing, it’s the all-in-one solution for your XML needs. Enhance it further with the Oxygen AI Positron Assistant plugin, bringing AI capabilities for natural language content creation and rewriting, all while keeping you in control.

Discover Oxygen XML Editor, the ultimate solution for XML editing, authoring, and development. Perfect for both beginners and experts, this versatile tool works seamlessly across platforms and is available as a standalone application or an Eclipse plug-in. With comprehensive support for a wide range of XML technologies, it simplifies the creation, editing, and publishing process. Enhance your workflow even further with the Oxygen AI Positron Assistant plugin, bringing AI-powered natural language content creation and rewriting, all while keeping you in control.

Discover the ultimate solution for XML editing with Oxygen XML Editor, the go-to tool for developers at all skill levels. Whether you're a novice or a seasoned expert, Oxygen XML Editor provides a versatile, cross-platform experience available both as a standalone application and an Eclipse plug-in. With comprehensive support for all XML technologies, it simplifies the creation, editing, and publishing processes. Elevate your productivity with the Oxygen AI Positron Assistant plugin, which brings AI-powered natural language content creation and rewriting while ensuring you stay in full control. Join the community of developers who trust Oxygen XML Editor for their XML needs.



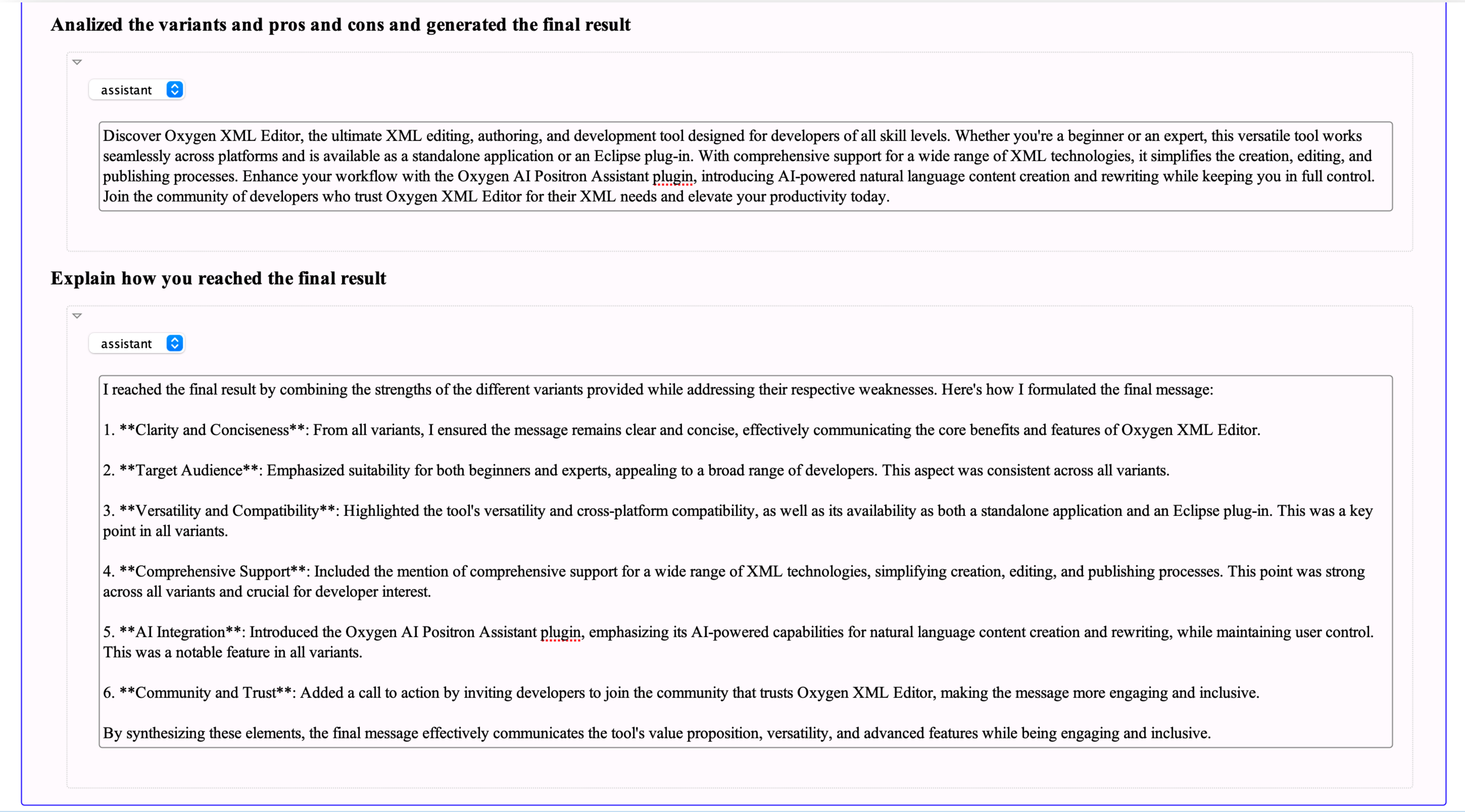
Determined pros and cons of each variant

Here there are 3 variants, each with pros and cons:
Variant 1:
Oxygen XML Editor is the ultimate tool for XML editing, authoring, and development. Designed for both beginners and experts, it's versatile and works seamlessly across all platforms. Available as a standalone app or an Eclipse plug-in, it provides robust support for all XML technologies. With tools for easy creation, editing, and publishing, it's the all-in-one solution for your XML needs. Enhance it further with the Oxygen AI Positron Assistant plugin, bringing AI capabilities for natural language content creation and rewriting, all while keeping you in control.
Pros:
1. The message is concise and clear, making it easy for developers to understand the key features and benefits of Oxygen XML Editor.
2. Highlights versatility and platform compatibility, appealing to a broad audience of developers.
3. Emphasizes both beginners and experts, showing the tool's wide range of usability.
4. Mentions the availability of a standalone app and Eclipse plug-in, providing options for different user preferences.
5. Points out the robust support for XML technologies, which is crucial for developers working with XML.
6. Introduces the Oxygen AI Positron Assistant plugin, showcasing advanced AI capabilities for content creation and rewriting.

Cons:
1. Lacks specific examples of how the tool can be used in real-world scenarios, which could make it less relatable.
2. Does not mention any unique selling points or features that differentiate it from competitors.
3. The description of the AI capabilities is brief and may not fully convey the potential benefits.
4. It does not address pricing or availability details, which might be important for potential users.
5. The message is focused on features but does not explicitly mention any user testimonials or success stories.
6. It does not address any potential limitations or learning curves associated with the tool.

Variant 2:
Discover Oxygen XML Editor, the ultimate solution for XML editing, authoring, and development. Perfect for both beginners and experts, this versatile tool works seamlessly across platforms and is available as a standalone application or an Eclipse plug-in. With comprehensive support for a wide range of XML technologies, it simplifies the creation, editing, and publishing process. Enhance your workflow even further with the Oxygen AI Positron Assistant plugin, bringing AI-powered natural language content creation and rewriting, all while keeping you in control.
Pros:
1. Clear and concise messaging that effectively communicates the versatility and usability of Oxygen XML Editor.
2. Emphasizes the tool's compatibility across platforms and availability in different forms (standalone and Eclipse plug-in).
3. Highlights the comprehensive support for XML technologies, appealing to a wide range of users.
4. Introduces the Oxygen AI Positron Assistant plugin, showcasing its AI capabilities for enhancing productivity.

Cons:
1. Lacks specific examples of XML technologies supported, which might be useful for some users.
2. Does not mention the specific benefits for different user levels (beginners vs. experts) beyond general suitability.
3. The message could be more engaging with a stronger call to action.
4. It does not detail the nature of user control offered by the AI plugin, which could be an important point for potential users.



Analized the variants and pros and cons and generated the final result

assistant

Discover Oxygen XML Editor, the ultimate XML editing, authoring, and development tool designed for developers of all skill levels. Whether you're a beginner or an expert, this versatile tool works seamlessly across platforms and is available as a standalone application or an Eclipse plug-in. With comprehensive support for a wide range of XML technologies, it simplifies the creation, editing, and publishing processes. Enhance your workflow with the Oxygen AI Positron Assistant plugin, introducing AI-powered natural language content creation and rewriting while keeping you in full control. Join the community of developers who trust Oxygen XML Editor for their XML needs and elevate your productivity today.

Explain how you reached the final result

assistant

I reached the final result by combining the strengths of the different variants provided while addressing their respective weaknesses. Here's how I formulated the final message:

1. **Clarity and Conciseness**: From all variants, I ensured the message remains clear and concise, effectively communicating the core benefits and features of Oxygen XML Editor.

2. **Target Audience**: Emphasized suitability for both beginners and experts, appealing to a broad range of developers. This aspect was consistent across all variants.

3. **Versatility and Compatibility**: Highlighted the tool's versatility and cross-platform compatibility, as well as its availability as both a standalone application and an Eclipse plug-in. This was a key point in all variants.

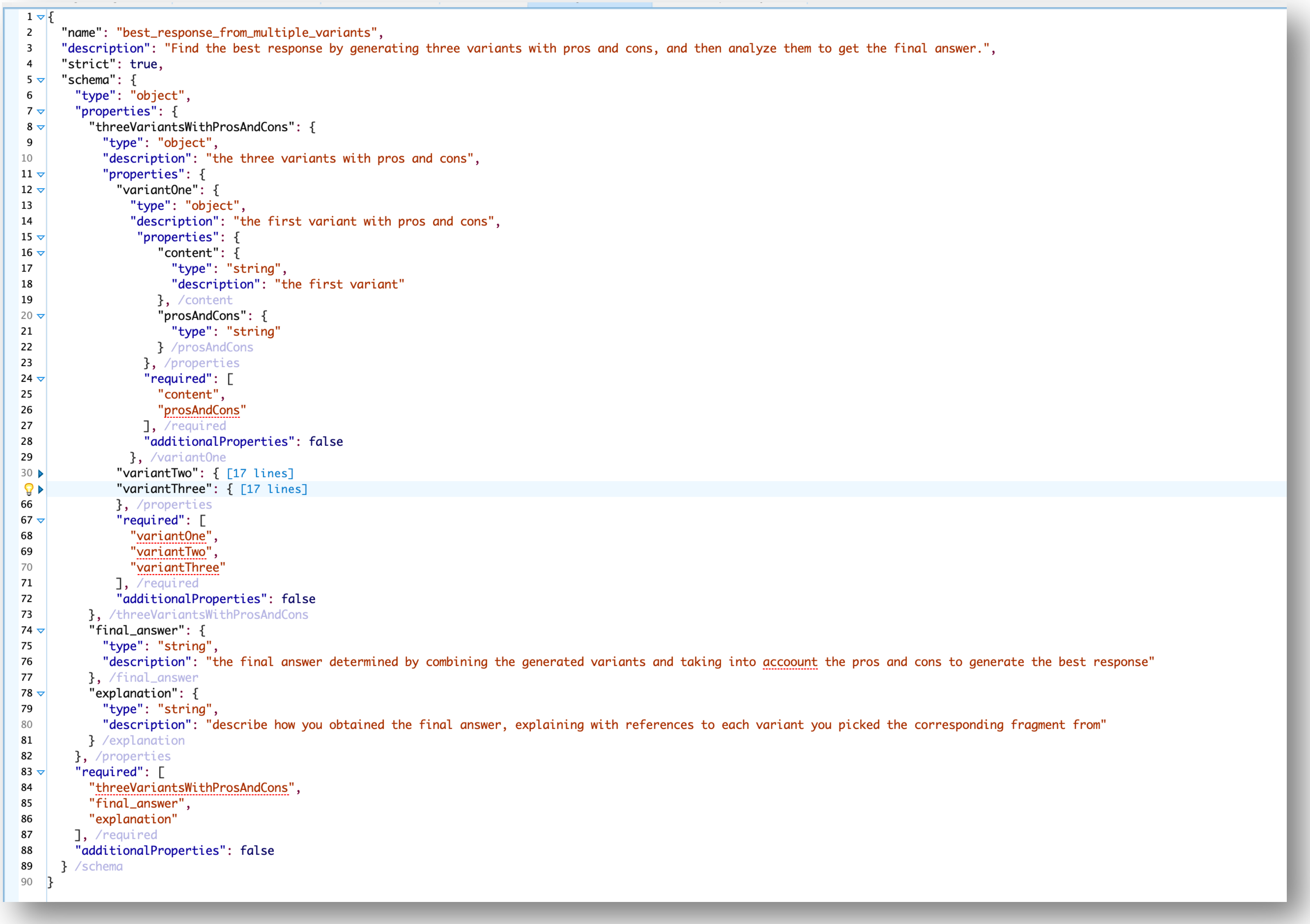
4. **Comprehensive Support**: Included the mention of comprehensive support for a wide range of XML technologies, simplifying creation, editing, and publishing processes. This point was strong across all variants and crucial for developer interest.

5. **AI Integration**: Introduced the Oxygen AI Positron Assistant plugin, emphasizing its AI-powered capabilities for natural language content creation and rewriting, while maintaining user control. This was a notable feature in all variants.

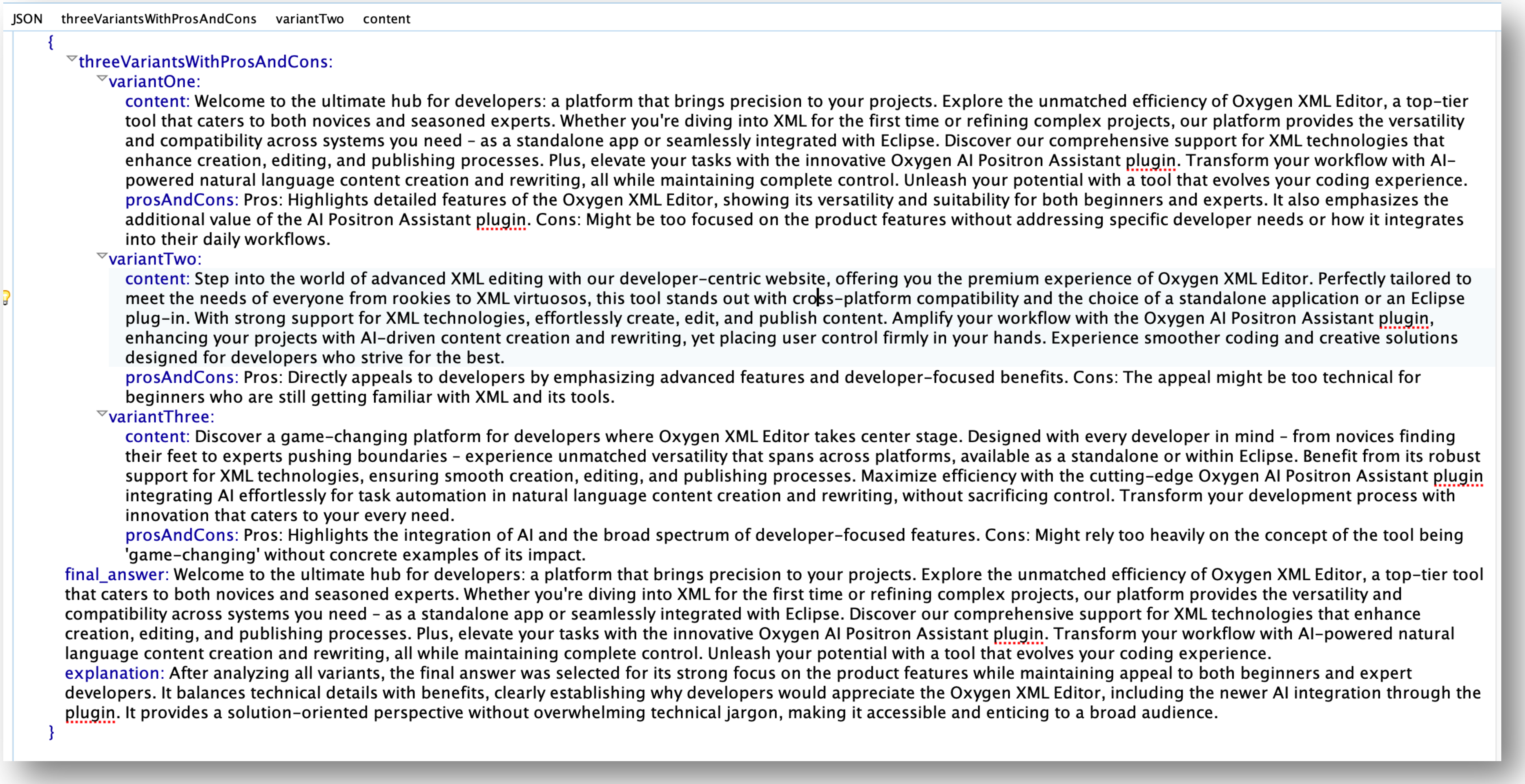
6. **Community and Trust**: Added a call to action by inviting developers to join the community that trusts Oxygen XML Editor, making the message more engaging and inclusive.

By synthesizing these elements, the final message effectively communicates the tool's value proposition, versatility, and advanced features while being engaging and inclusive.

Workflow using structured output



```
{
  "name": "best_response_from_multiple_variants",
  "description": "Find the best response by generating three variants with pros and cons, and then analyze them to get the final answer.",
  "strict": true,
  "schema": {
    "type": "object",
    "properties": {
      "threeVariantsWithProsAndCons": {
        "type": "object",
        "description": "the three variants with pros and cons",
        "properties": {
          "variantOne": {
            "type": "object",
            "description": "the first variant with pros and cons",
             "properties": {
                "content": {
                  "type": "string",
                  "description": "the first variant"
                }, /content
                "prosAndCons": {
                  "type": "string"
                } /prosAndCons
              }, /properties
              "required": [
                "content",
                "prosAndCons"
              ], /required
              "additionalProperties": false
            }, /variantOne
          "variantTwo": { [17 lines]
          "variantThree": { [17 lines]
          }, /properties
          "required": [
            "variantOne",
            "variantTwo",
            "variantThree"
          ], /required
          "additionalProperties": false
      }, /threeVariantsWithProsAndCons
      "final_answer": {
        "type": "string",
        "description": "the final answer determined by combining the generated variants and taking into accoount the pros and cons to generate the best response"
      }, /final_answer
      "explanation": {
        "type": "string",
        "description": "describe how you obtained the final answer, explaining with references to each variant you picked the corresponding fragment from"
      } /explanation
    }, /properties
    "required": [
      "threeVariantsWithProsAndCons",
      "final_answer",
      "explanation"
    ], /required
    "additionalProperties": false
  } /schema
}
```



{

threeVariantsWithProsAndCons:

variantOne:

content: Welcome to the ultimate hub for developers: a platform that brings precision to your projects. Explore the unmatched efficiency of Oxygen XML Editor, a top-tier tool that caters to both novices and seasoned experts. Whether you're diving into XML for the first time or refining complex projects, our platform provides the versatility and compatibility across systems you need - as a standalone app or seamlessly integrated with Eclipse. Discover our comprehensive support for XML technologies that enhance creation, editing, and publishing processes. Plus, elevate your tasks with the innovative Oxygen AI Positron Assistant plugin. Transform your workflow with AI-powered natural language content creation and rewriting, all while maintaining complete control. Unleash your potential with a tool that evolves your coding experience.

prosAndCons: Pros: Highlights detailed features of the Oxygen XML Editor, showing its versatility and suitability for both beginners and experts. It also emphasizes the additional value of the AI Positron Assistant plugin. Cons: Might be too focused on the product features without addressing specific developer needs or how it integrates into their daily workflows.

variantTwo:

content: Step into the world of advanced XML editing with our developer-centric website, offering you the premium experience of Oxygen XML Editor. Perfectly tailored to meet the needs of everyone from rookies to XML virtuosos, this tool stands out with cross-platform compatibility and the choice of a standalone application or an Eclipse plug-in. With strong support for XML technologies, effortlessly create, edit, and publish content. Amplify your workflow with the Oxygen AI Positron Assistant plugin, enhancing your projects with AI-driven content creation and rewriting, yet placing user control firmly in your hands. Experience smoother coding and creative solutions designed for developers who strive for the best.

prosAndCons: Pros: Directly appeals to developers by emphasizing advanced features and developer-focused benefits. Cons: The appeal might be too technical for beginners who are still getting familiar with XML and its tools.

variantThree:

content: Discover a game-changing platform for developers where Oxygen XML Editor takes center stage. Designed with every developer in mind - from novices finding their feet to experts pushing boundaries - experience unmatched versatility that spans across platforms, available as a standalone or within Eclipse. Benefit from its robust support for XML technologies, ensuring smooth creation, editing, and publishing processes. Maximize efficiency with the cutting-edge Oxygen AI Positron Assistant plugin integrating AI effortlessly for task automation in natural language content creation and rewriting, without sacrificing control. Transform your development process with innovation that caters to your every need.

prosAndCons: Pros: Highlights the integration of AI and the broad spectrum of developer-focused features. Cons: Might rely too heavily on the concept of the tool being 'game-changing' without concrete examples of its impact.

final_answer: Welcome to the ultimate hub for developers: a platform that brings precision to your projects. Explore the unmatched efficiency of Oxygen XML Editor, a top-tier tool that caters to both novices and seasoned experts. Whether you're diving into XML for the first time or refining complex projects, our platform provides the versatility and compatibility across systems you need - as a standalone app or seamlessly integrated with Eclipse. Discover our comprehensive support for XML technologies that enhance creation, editing, and publishing processes. Plus, elevate your tasks with the innovative Oxygen AI Positron Assistant plugin. Transform your workflow with AI-powered natural language content creation and rewriting, all while maintaining complete control. Unleash your potential with a tool that evolves your coding experience.

explanation: After analyzing all variants, the final answer was selected for its strong focus on the product features while maintaining appeal to both beginners and expert developers. It balances technical details with benefits, clearly establishing why developers would appreciate the Oxygen XML Editor, including the newer AI integration through the plugin. It provides a solution-oriented perspective without overwhelming technical jargon, making it accessible and enticing to a broad audience.

}

Oxygen Feedback
AI Assistant for published content

- AI response together with the search results
- RAG Chat on published content
- It is also a feedback tool for publishers
  - successful sessions reduce support costs
  - unsuccessful sessions are in fact an opportunity for improvement
    - the information is there but was not found - increase it relevance for that question
    - the information is not there, if it is relevant then it needs to be added - improve the documentation coverage

Future plans

- Enable AI to perform actions on the editor itself (via functions or by interpreting the result).
- Support complex workflows
  - via multiple interactions
  - via structured output support
- Suggest continuations for actions/chat sessions
- Chat memory - make chat history available via AI callback functions (via the RAG support)
- Support multiple engines to be used in the same time so we can implement workflows when we automatically compare results from various models.

Takeaways

- AI changed the tools in content creation and development
- All related conferences have an AI-dominant schedule
- Integrating AI into a product is a non-trivial
- Rapid and continuous development

Thank you!

Happy to answer your questions!

George Bina
george@oxygenxml.com