

Customizing AI Positron for Your DITA XML Project

The [Oxygen AI Positron Assistant Add-on](#) includes a variety of helpful AI actions designed to boost productivity. However, **DITA XML** projects have specific requirements. They often need particular **terminology** and adherence to **style guide rules**, or they may rely on company-specific **DITA** specialization DTDs/schemas that differ from the standard **DITA DTDs** that the AI models are trained on. Therefore, to generate high-quality content using the AI actions, the add-on must be configured to understand the unique aspects of the **DITA XML** project it will be used with.

AI Tools Pros/Cons

With their ability to **rewrite**, **research**, and **generate documentation drafts** AI models can be useful in a variety of cases as long as aspects specific to the current DITA XML project can be controlled and enforced.

Good	Bad/Ugly
Re-write content (correct grammar, active voice, style, readability improvements).	The company specific style guide might impose certain rules that, by default, are not taken into account by the AI model.
Generating draft content by combining multiple sources (including images, attached PDF , Word or Markdown documents).	Hallucinations when generating content if not enough quality context information is provided. The overall terminology or style guide is not taken into account. The DITA XML specializations are also not taken into account by default.
Research / Explain concepts (chat).	By default an AI model responds to questions based on its training and not on the content from the current user's manual.

Access Content From Your Project

By default, when **researching** or **asking** the AI questions, [Retrieval Augmented Generation \(RAG\)](#) tools give the AI access to **search for specific keywords** in your current DITA XML documentation and thus provide the AI access to the content of topics which are related and relevant to your discussion.

Bundled AI actions like **Generate Documentation Draft**, **Expand Draft**, **Create Topics** give the AI model access to the current indexed DITA XML project to retrieve the topics with content similar to the new content that needs to be created.

You can create your own [custom AI actions](#) that allow RAG access to the current project contents for the AI engine. An example can be found [here](#).

Overall Project Context and Audience

In the Oxygen **Preferences AI Positron Assistant** preferences page you can set in the [Context](#) text area general information about the current project and the intended audience. This context information will be used subsequently for all AI actions and chat sessions.

Example of a context which establishes an overall project scope and audience:

```
# Project context

The current project contains technical documentation for a fun and easy to drive
dune vehicle named the Sharkfin Rover.

# Audience

Technical documentation must be precise and easy to understand.

Our clients are not technical, they are usually beginners when it comes to driving
and their understanding of English may vary.
```

The same context information should be expanded to include terminology and style guide information for the current project. An example can be found [here](#).

Terminology Rules

A DITA XML project may define (in various formats) allowed terms and certain synonyms that are not recommended to be used instead of the allowed terminology.

The [Context](#) field in the **AI Positron Assistant** preferences page can also contain rules for the most important terms and disallowed synonyms.

Example of context that establishes some of the most important terms:

```
# Terminology

Specific terms in the DITA XML content must be wrapped in the <term> element. Preserve all such terms and do not
attempt to replace them with possible synonyms.

Always prefer to use the terms below instead of possible synonyms:
...

```

```

Sharkfin Rover (instead of 'Sharkfin Buggy')

vehicle (instead of 'car')

lorry (instead of 'truck')

rear position lamp (instead of 'tailgate')

powertrain (instead of 'drivetrain')

petrol engine (instead of 'Otto engine')

motorcycle (instead of 'motor-bike')

engine (instead of 'motor')

electronic stability control (instead of 'ESP')

motor vehicle chassis (instead of 'vehicle frame')

...

```

Once the AI model receives the imposed terminology rules as part of the prompt context, it will take them into account for all interactions (e.g. for creating documentation drafts from content that was written without taking the terminology into account).

The Oxygen [Terminology Checker add-on](#) can be configured with a terminology rules file that allows it to highlight terminology problems in real time and propose replacement for them. You can find a sample terminology rules file [here](#). There is also a [sample AI action](#) that can generate an XML rules file from the add-on based on a style guide.

Once the **Terminology Checker** add-on is installed and flags the terminology problems in the **Author** visual editor mode, you can use the **Fix Terminology Problems** AI action to fix terminology problems in the currently opened topic's contents.

As a conclusion, you can add your most important terminology rules in the **Context** setting to create and process DITA XML content while directly obeying the rules. However, installing the terminology checker add-on allows you to see terminology problems highlighted in the visual editor and to fix them using the AI engine with the bundled **Fix Terminology Problems** AI action.

Style Guide

A style guide is a set of project-specific standards and guidelines used to ensure consistency and clarity in technical documentation. A company's style guide may come in various formats (DITA XML, Markdown, Word, etc.) but some of the most important rules in the style guide can be distilled and set in the [Context](#) field within the **AI Positron Assistant** preferences page.

Example of context that establishes some of the most important style guide rules:

```
# Style guide
```

```

There is a set of company specific Style Guide rules which are important to follow when rewriting or creating new
content, the most important rules are mentioned below:

- Never use a metaphor, simile, or other figure of speech.
- Never use a long word where a short one will do.
- Keep sentences short. If it is possible to cut a word out, always cut it out.
- Never use the passive where you can use the active.
- Never use a foreign phrase, a scientific word, or a jargon word if you can think of an everyday English equivalent.
- Don't use text that points to other topics such as "In the document below" or "in the following section."
- Some words or phrases are disallowed to be used. Mapping of disallowed phrases to their valid equivalent constructs:
...

as well as, in addition to => and
at this time, at this/that point in time => currently, now/then
advance planning => planning
by means of => by
by reason of => because of
due to the fact => because
during the course of => during
end result => result
...

```

The Oxygen [Terminology Checker add-on](#) can also match disallowed words and phrases in the terminology rules file, allowing you to see in the visual editor the style guide problems and to correct them using the bundled **Fix Terminology Problems** AI action.

Style Guide rules may also impose structural aspects for the DITA XML content and such rules can be automated by converting them to [Schematron rules](#). Once such Schematron rules are added to the DITA XML project and start showing errors in the edited content, the bundled **Fix Validation Problems** AI action can be used to ask the AI to fix the reported problems in the current document. If the AI engine attempts to save content to a topic and the saved content has validation problems, the AI receives the validation problems and attempts to auto fix the problems until the topic becomes valid.

An alternative to defining a style guide in the AI context prompt or to impose them after the content is generated would be to create a fine-tuned AI model trained with small examples based on the internal style guide and created also using an AI model.

DITA XML DTD/Schemas Specialization

Quite often, a company that uses the DITA XML standard may decide to create a DTD or RelaxNG based specialization that defines new XML elements that are not known by the AI engine.

Most bundled AI actions attempt to preserve the existing XML structure, as much as possible. AI actions like **Expand Draft** start from a newly created DITA XML topic and thus preserves the existing DTD/schema reference and XML elements.

Actions like **Generate Documentation Draft** can specify (in their definition) that a certain XML structure must be generated. An example, for this can be found [here](#).

The most important XML vocabulary changes (DTD/schema references, newly added XML elements) can be distilled and set in the **Context** field within the **AI Positron Assistant** preferences page.

Example of vocabulary changes described in the prompt context:

```
# Vocabulary Changes
```

```
DITA XML topics generated for our project's user's manual have the following structure and DOCTYPE declaration:
```

```
```
```

```
<!DOCTYPE oxyTopic PUBLIC "OXY-PUB-TOPIC" "oxy-system">
```

```
<oxyTopic>
```

```
 <title></title>
```

```
 <body>
```

```
 </body>
```

```
</oxyTopic>
```

```
```
```

```
DITA XML concepts generated for our project's user's manual have the following structure and DOCTYPE declaration:
```

```
```
```

```
<!DOCTYPE oxyConcept PUBLIC "OXY-PUB-CONCEPT" "oxy-system">
```

```
<oxyConcept>
```

```
 <title></title>
```

```
 <body>
```

```
 </body>
```

```
</oxyConcept>
```

```
'''
```

New DITA XML elements are defined in our specific DITA XML vocabulary as follows:

There is a new <part-term> XML element which must be used to tag various car parts.

There is a new <product> XML element which must be used to tag product names.

Example:

The <product>Hisense</product> <part-term>engine</part-term> is more efficient.

## Conclusion

You need to invest time to adapt the AI Positron add-on to generate better content for your particular project and organization particularities.

By specifying your specific project particularities in a context prompt, content created and re-written by an AI engine can become much more precise and fit better with your existing ways of working. Some of your project's rules can be automated allowing you to also detect and fix such problems after the content is generated using the AI engine.

An example of a detailed context prompt that contains an overview of the project and audience, terminology, style rules, described vocabulary changes can be found [here](#).