Integrate Collaboration into your Workflow
Existing workflow

... → Checkout → Modify Content → Commit → ...

<oxygen/>
Integrating collaboration workflow

... → Checkout → Modify Content → Commit → ...

Sharing Content → Collaborators changes → Integrate changes
oXygen Content Fusion

Collaboration on XML content using a browser!
(DITA, DocBook, XHTML, your own XML)
Why oXygen Content Fusion?

You need to **collaborate**

You have **access to specific tools**

You work within a specific **workflow**

You have **access** to a repository

You want to **approve changes** before reaching the repository

You need a **less formal** way to receive feedback
It cannot be easier to collaborate!

**oXygen Content Fusion** allows you to:

1. **Create** a review/collaboration task from XML files
2. **Share** task URL with your contributors
   - Contributors need only a web browser!
     - a) Follow link
     - b) Make changes
     - c) Mark as done
3. **Get changes** back into the original files
Tools

oXygen XML Editor or oXygen XML Author
Content Fusion Connector plugin
oXygen Content Fusion server
Process

1. **Create** review/contribute task
2. **Share** it with reviewers/contributors
3. **Integrate** changes
Let’s see how it works

Examples:

1. Create a simple review task with a few topics
2. Provide a place for contributors to enter content
3. Restrict editing to specific content
Supported functionality

One or multiple XML files
DITA Map to provide the editing context for DITA
Multiple reviewers/contributors
Automatic locking
Change tracking
Visual XML-aware 3-way merge with automatic merge support for pseudo-conflicts
Task messages and email notifications
Enable/optimize workflows
Distributed services

- Storage
- Versioning
- Publishing
- Communication
- Continuous Integration
Distributed services

- Storage
- Versioning
- Publishing
- Communication
- Continuous Integration
- XML Authoring?
XML Authoring as a Service

Access XML Authoring by following a URL

http://server/service?url=file/to/edit
Technical support to documentation

Create a channel between technical support team and documentation team to enable continuous improvement of documentation
Technical support to documentation

Provide a URL the technical support engineer can follow to suggest changes to documentation!
Identify documentation topic

Working with Markdown Documents in DITA
Oxygen XML Editor includes some unique features that allow you to easily integrate Markdown documents in a DITA project. This is especially helpful for teams that have contributors who are familiar...
topics/markdown-dita-x-dita2.html

DITA Maps Manager
Oxygen XML Editor provides a view for managing and editing DITA maps. The DITA Maps Manager view presents a DITA map as a Table of Contents. It allows you to navigate the topics and maps, make...
topics/dita-maps-manager.html

Markdown Editor
Oxygen XML Editor provides an intuitive, dynamic, and easy-to-use Markdown editor for writing and converting Markdown documents. It is a split-screen editor with two panels that are synchronized in...
topics/markdown-editor.html
See if it responds to user issue

Working with Markdown Documents in DITA

Oxygen XML Editor includes some unique features that allow you to easily integrate Markdown documents in a DITA project. This is especially helpful for teams that have contributors who are familiar with the Markdown syntax, but they want their output to be generated from DITA projects. The integration between the Markdown editor and DITA includes actions to export or convert Markdown documents to DITA topics and the DITA tab in the Preview pane provides a visualization of how the topic will look after conversion.

- **Export Markdown as a DITA Topic**

  The Markdown editor includes an option to quickly convert the current Markdown document into a DITA topic. The Export as DITA Topic action is available in the contextual menu of the left-side text editor and the right-side Preview pane when the DITA tab is selected.

  The conversion creates a new XML file that is defined as a DITA topic and opens it in the Text editing mode. You can then work with the document as you would with any other DITA topic, although you may need to manually correct some issues where the parser could not properly map Markdown syntax to DITA markup.

- **Working with Markdown Documents in the DITA Maps Manager**

  Oxygen XML Editor has some specialized features that allow you to integrate Markdown documents directly into your DITA project using the DITA Maps Manager. The following features are available for Markdown documents in the DITA Maps Manager view:

  - **Insert Reference to Markdown Document** - You can use the New, Reference, and Reference to the currently edited file actions from the Append Child, Insert Before, or Insert After submenu when invoking the contextual menu in the DITA Maps Manager to insert a reference to a Markdown document at the selected location in the map.
Work with Markdown Documents in DITA

---

**Keywords:**

- (Markdown documents (DITA))
- (DITA and Markdown documents)

---

Oxygen XML Editor includes unique features that allow you to easily integrate Markdown documents into a DITA project. This is especially helpful for teams that have contributors who are familiar with the Markdown syntax, but they want their output to be generated from DITA projects. The integration between the Markdown editor and DITA includes actions to export or convert Markdown documents to DITA topics and the DITA tab in the Preview pane provides a visualization of how the topic will look after conversion.

**Export Markdown as a DITA Topic**

The Markdown editor includes an option to quickly convert the current Markdown document into a DITA topic. The Export as DITA Topic action is available in the contextual menu of the left-side text editor and the right-side Preview pane.
Developers to review documentation

Enable developers to review the incremental documentation related to the issue they resolve.
Developers to review documentation

Provide a URL the developer can follow to get access to the modified documentation!
Automatic message with links to modified topics
Follow link to review

Detecting Master Files

Metadata

Keywords:

Master Files (Detecting)
Projects (Master files (Detecting))
Detect Master Files
Detect Master Files from Project

Oxygen XML Editor allows you to detect the master files using the Detect Master Files option. This action applies to the folders you select in the project.

To detect master files over the entire project, do one of the following:

- Right-click the root of the project and select Detect Master Files (if product is 'author' or 'developer' or 'editor').
- Use the Detect Master Files from Project option, available in the contextual menu of the master files folder.

Both of these options display the Detect Master Files wizard. In the first panel you can select the type of master files you want. The detected master files are presented in a tree-like fashion. The resources are grouped into three categories:
Follow link to see changes

EXM-38653: Added a note for DITA projects
StevenHiggs committed on 24 Feb

Showing 1 changed file with 3 additions and 1 deletion.

```diff
- presented in a tree-like fashion. /p> The resources are grouped into three
categories: /ul>
- Possible /i>master files</i> - The files presented on the first
level in
this category are not imported or included from other files. These files are most
likely
- to be set as /i>master files</i>

<p>/i>Cycles</i> - The files that are presented on the first level have circular
dependencies between them. Any of the files presented on the first level of a cycle
is a
possible /i>master file</i>
```
Immediate Review

Review directly the source

Review a published format

- PDF
- HTML
- Etc.
Classic vs Direct XML review

**Classic XML Review Process**
- XML → PDF → Review (comments only)
- Align comments with the XML
- Manual process

**Direct XML Review Process**
- XML → Direct XML Review
- Direct XML Review comments + changes (tracked)
- Accept/reject changes
- Act on comments
Immediate Review

Provide a URL to edit the source directly from the published format!

- XML → PDF → Review (comments only)
  - align comments with the XML
  - manual process!
  - act on comments and make changes to the XML source

- Direct XML review
  - comments + changes (tracked)
  - accept/reject changes
  - act on comments
Working with Archives

Oxygen XML Editor includes a useful Archive Browser view that offers the means to work with files directly from various types of archives (for example, opening and saving files directly in archives, or browsing and modifying archive structures). The archive support is available for all ZIP-type archives, including:

- ZIP archives
- EPUB books
- JAR archives
- Office Open XML (OOXML) files
- Open Document Format (ODF) files
- IDML files

You can transform, validate, and perform many other operations on files directly from an archive. For instance, you can transform, or validate files directly from OOXML or ODF packages, and the structure and content of the ZIP archives can be opened, edited, and saved, similar to any other ZIP archive browsing tool. Also, when browsing for a URL in various dialog boxes, you can use the Browse for archived file action to browse and select files from a particular archive.
Immediate access to source

Working with Archives

Metadata
Keywords:
(Archive)

Oxygen XML Editor includes a useful Archive Browser view that offers the means to work with files directly from various types of archives (for example, opening and saving files directly in archives, or browsing and modifying archive structures). The archive support is available for all ZIP-type archives, including:

- ZIP archives
- EPUB books
- JAR archives
- Office Open XML (OOXML) files
- Open Document Format (ODF) files
- .idml/.dml files

You can transform, validate, and perform many other operations on files directly from an archive. For instance, you can transform, or validate files directly from OOXML or ODF packages, and the structure and content of the ZIP archives can be opened, edited, and saved, similar to any other ZIP archive browsing tool. Also, when browsing for a URL in various dialog boxes, you can use the Browse for archived file action to browse and select files from a particular archive.
PDF-based output

Saving Documents

You can save the document you are editing with one of the following actions:

- **File > Save.**
  - **Save toolbar button** - If the document was not yet saved, it displays the **Save As** dialog box.
  - **File > Save As** - Displays the **Save As** dialog box, used either to name and save an open document to a file or to save an existing file with a new name.
  - **File > Save To URL** - Displays a **Save to URL** dialog box that can be used to save a file identified by its URL (defined by a protocol, host, resource path, and an optional port). Use the drop-down action list to choose one of the available save actions:
    - **Browse for local file** - Opens a local file browser dialog box allowing you to save the document locally.
    - **Browse for remote file** - Displays a **Save to URL** dialog box that allows you to save the document to a remote location (accessible through FTP, SFTP or WebDAV).
    - **Browse for archived file** - Displays the **Archive Browser** that allows you to save the document inside an archive.
    - **Browse Data Source Explorer** - Opens a **Data Source Explorer** that allows you to browse the data sources defined in the **Data Sources preferences page**.
      - **Tip:** You can get to the **Data Sources** preferences page, using the **Configure Database Sources** shortcut from the **Save to URL** dialog box.
    - **Search for file** - Displays the **Open/Find Resource dialog box**.

- **File > Save All** - Saves all open documents. If any document does not have a file, displays the **Save As** dialog box.
Immediate access to source

Saving Documents

You can save the document you are editing with one of the following actions:

- **File→Save.**
- **Save toolbar button** - If the document was not yet saved, it displays the **Save As** dialog box. (If product is editor or 'author' or 'developer')
- **File→Save As** - Displays the **Save As** dialog box, used either to name and save an open document to a file or to save an existing file with a new name.
- **File→Save To URL** - Displays a **Save to URL** dialog box that can be used to save a file identified by its URL (defined by a protocol, host, resource path, and an optional port). Use the drop-down action list to choose one of the available save actions:
  - **Browse for local file** - Opens a local file browser dialog box allowing you to save the document locally.
  - **Browse for remote file** - Displays a **Save to URL** dialog box that allows you to save the document to a remote location (accessible through FTP, SFTP or WebDAV).
  - **Browse for archived file** - Displays the #Archive Browser that allows you to save the document inside an
Crowd-sourcing docs

Enable anyone to contribute to your project documentation!
DITA-OT documentation

Services used by the DITA-OT documentation

• **GitHub**
  • storage
  • versioning
  • access control

• **GitHub Pages**
  • web publishing

• **Travis**
  • automate publishing

• **Slack**
  • project communication
Click on website to edit its content
Save to send a request for change
XML-based editable websites

http://xproc.org
- Based on DocBook as source format
- DocBook source available on GitHub

http://georgebina.github.io/ghd-wiki
- Based on DITA, Lightweight DITA and Markdown
- Source available on GitHub
Welcome to XProc.org

XProc.org is a website about XProc: An XML Pipeline Language, a specification under development at the W3C, and its use, implementation, and extensions.

What is it?

XProc is a language designed for describing operations to be performed on XML documents. The official specification for XProc is XProc: An XML Pipeline Language. It is a Recommendation.

The XProc specification was produced by the XML Processing Model Working Group. The XProc WG operated in the public, anyone is free to read the archives of its mailing list.

While the WG was in operation, public comments on the specification could be sent to the processing model comments mailing list. The archives of the comments list are also open to the public. Now that the WG has been disbanded, readers are encouraged to submit comments to the community through the GitHub issue tracker.

Why did you do it?

XProc is designed to address the common problem of how to compose XML processes. Many document processing scenarios involve some combination of XML technologies; canonical examples include XInclude, schema validation, and transformation.

Although it is possible to combine these technologies using general purpose tools such as make and ant (to name only two), these tools are not designed to deal specifically with the semantics of XML processing. As such they are often both more complicated and less useful than would be ideal.

XProc has been designed specifically to allow authors to compose XML processes and share these compositions in a standard way.
Access to the DocBook source

**Article: **XProc.org

XProc.org is a website about XProc: An XML Pipeline Language, a specification under development at the W3C, and its use, implementation, and extensions.

**Section 1: What is it?**

XProc is a language designed for describing operations to be performed on XML documents. The official specification for XProc is XProc: An XML Pipeline Language. It is a Recommendation.

The XProc specification was produced by the XML Processing Model Working Group. The XProc WG operated in the public, anyone is free to read the archives of its mailing list.

While the WG was in operation, public comments on the specification could be sent to the processing model comments mailing list. The archives of the comments list are also open to the public. Now that the WG has been disbanded, readers are encouraged to submit comments to the community through the GitHub issue tracker.

**Section 2: Why did you do it?**

XProc is designed to address the common problem of how to compose XML processes. Many document processing scenarios involve some combination of XML technologies; canonical examples include XInclude, schema validation, and transformation.
DITA+Markdown based Wiki

Website (GitHub Pages)
link to edit page

oXygen XML
Web Author

GitHub
• Documentation repository

Travis
Website content

https://georgebina.github.io/ghd-wiki/

Sample DITA Wiki

DITA magic with GitHub and oXygen

GitHub DITA Wiki

Lightweight DITA

Markdown

WebHelp output generated by <oXygen/> XML Author

https://georgebina.github.io/ghd-wiki/index.html
Lightweight DITA topic

Lightweight DITA

Lightweight DITA can be used for ghd-wiki taking advantage of controlled authoring experience by providing placeholders, hints, inline actions and inline form controls.

Lightweight DITA is a great alternative to Markdown, having the same or less complexity degree while being very clearly specified. The user interface takes advantage of controlling the authoring experience through placeholders, hints, inline actions and form controls to edit attribute values in place, all these combined to provide a simple user interface suitable for any contributors.

It is very easy to create content in LW-DITA, especially if you can benefit of some help from the authoring tool. The oXygen XML Web Author is an example of such a tool, that provides special support for working with LW-DITA, support that can be extended further to match your actual needs.
Revisions history

History for ghd-wiki / wiki / topic.dita

Commits on Oct 16, 2016

- Committing with oXygen Web Author
  georgebina committed 15 hours ago

- Make short description shorter.
  georgebina committed 15 hours ago

- Updated content.
  georgebina committed 20 hours ago
Lightweight DITA

Short Description

Use the short description to expand the title, providing additional information about the content of the topic. It should always be composed of complete sentences and form a comprehensive thought.

Lightweight DITA can be used for ghd-wiki taking advantage of controlled authoring experience by providing placeholders, hints, inline actions and inline form controls.

Lightweight DITA is a great alternative to Markdown, having the same or less complexity degree while being very clearly specified. The user interface takes advantage of controlling the authoring experience through placeholders, hints, inline actions and form controls to edit attribute values in place, all these combined to provide a simple user interface suitable for any contributors.
Travis project status

https://travis-ci.org/georgebina/ghd-wiki/builds

<table>
<thead>
<tr>
<th>Branch</th>
<th>Action</th>
<th>Message</th>
<th>Status</th>
<th>Duration</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>master</td>
<td>✓</td>
<td>Update README.md</td>
<td>#73 passed</td>
<td>46 sec</td>
<td>about 14 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>da8cf48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>master</td>
<td>✓</td>
<td>Add link to project home on GitHub.</td>
<td>#72 passed</td>
<td>52 sec</td>
<td>about 14 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8cd3646</td>
<td></td>
<td></td>
</tr>
<tr>
<td>master</td>
<td>✓</td>
<td>Committing with oXygen Web Author</td>
<td>#71 passed</td>
<td>53 sec</td>
<td>about 15 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>34569e6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>master</td>
<td>✓</td>
<td>Make short description shorter.</td>
<td>#70 passed</td>
<td>1 min 8 sec</td>
<td>about 15 hours ago</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2e7121b</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Take-aways

oXygen Content Fusion allows you to share your XML document with people you want to collaborate with and integrate their changes easily.

oXygen Web Author can be used as a service to simplify existing workflows or to enable new ones, such as:

- Link support to documentation
- Get developers to review relevant documentation
- Simplify the review process

Any XML-based system can be a wiki-like system where anyone can immediately interact with the content.
Thank you

Questions?

george@oxygenxml.com
@georgebina

http://www.oxygenxml.com
What oXygen products solve

Development  Authoring  Publishing  Integration

Web Authoring

Collaboration and Review
What oXygen products solve

Developer
Development

Author
Authoring
Publishing
Integration

Web Authoring

Collaboration and Review
What oXygen products solve

Editor
Developer
Author

Development
Authoring
Publishing
Integration

Web Authoring

Collaboration and Review
What oXygen products solve

- Editor
- Chemistry
- Developer
- Author
- Webhelp
- Development
- Authoring
- Publishing
- Integration
- Web Authoring

Collaboration and Review
What oXygen products solve

Editor
Chemistry
Developer
Author
Webhelp
Authoring
Web Authoring
Integration
Publishing
Scripting
Development
Authoring
Collaboration and Review
What oXygen products solve

Editor
Chemistry
Developer
Author
Webhelp
Scripting
Development
Authoring
Publishing
Integration
Web Author
Web Authoring

Collaboration and Review
What oXygen products solve

- **Editor**
- **Chemistry**
- **Developer**
- **Author**
- **Webhelp**
- **Scripting**
- **Development**
- **Authoring**
- **Publishing**
- **Integration**
- **Web Author**
- **Web Authoring**
- **Collaboration and Review**
- **oXygen Content Fusion**