

Who is who in JATS?

Servei de Publicacions, UAB

Marc Bria Ramírez

Departament de Psicologia Social, UAB

Juan Muñoz Justicia

v.3.0

(CC-by 2019)

INTRODUCTION

During the last years we have been talking a lot about JATS (Journal Article Tag Suite), a format to structure, share and archive academic articles that it's close to become an international standard.

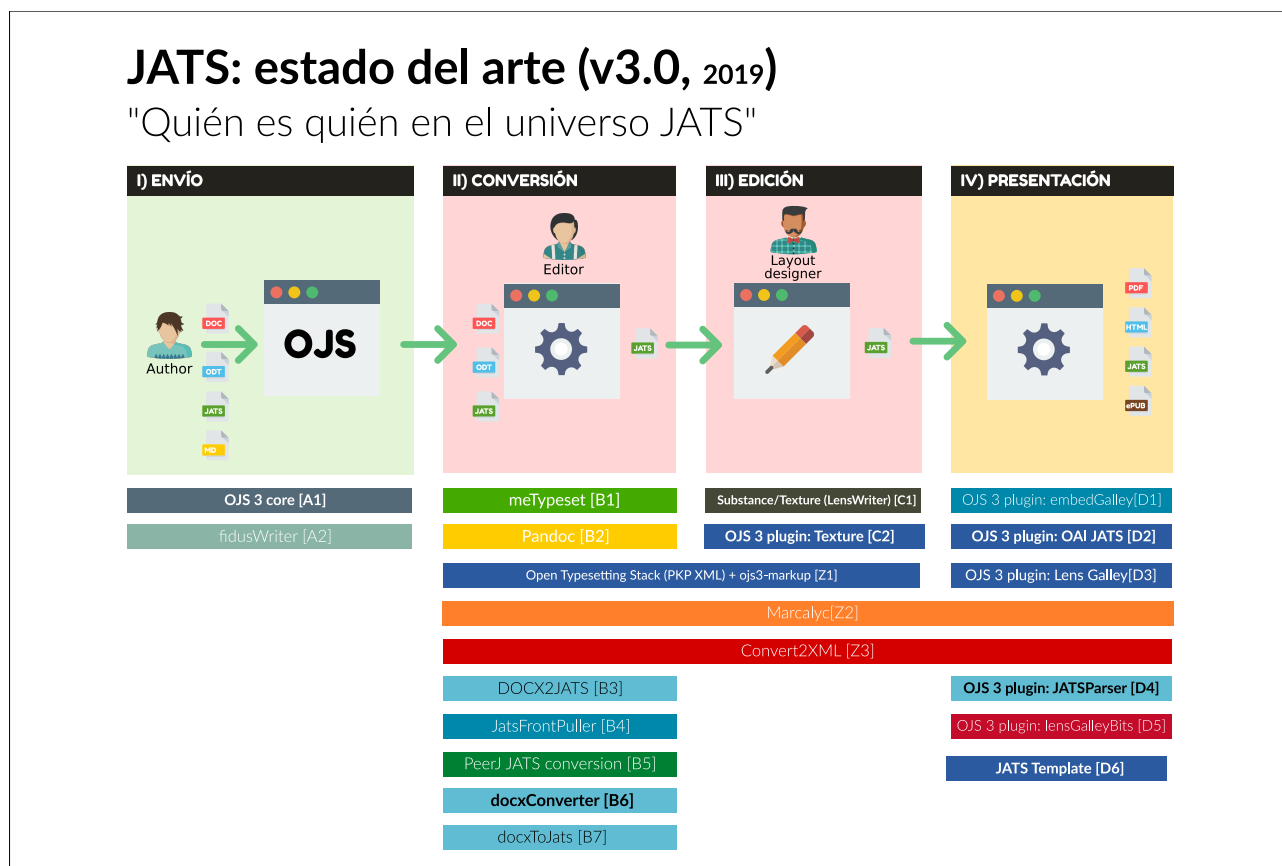
PKP have been a main actor in this movie and it's forum show an increasing interest: <http://forum.pkp.sfu.ca/search?q=jats>

May be because of the "hype" but the fact is, when asked, most of our journals said they like to start "JATSing" (we need a new verb :-)) so, before start developing a new solution, we made some research trying to discover "who is who" in the JATS universe.

This report is the result of a research around existing OJS free software tools. This year, proprietary solutions where intentionally excluded because they are far of our ideological umbrella: we have a commitment with open knowledge and freedom and proprietary applications usually limit both.

We titled this summary "state of the art of JATS". The graphic shows a picture of the whole puzzle and show all the involving pieces. It will let you find if some parts are still uncovered.

We like to update this graphic and report yearly till we have a solid and definitive solution.



Let us explain this research was not an easy task because some tools are incomplete, abandoned or changed its name 2-3 times... so please, comment in forum if you think we missed any relevant actors in this map or a description is not accurate.

We divided the JATS workflow in 4 phases:

1. **SUBMISSION:** authors send their work (usually in DOCx or ODT) to the platform.
2. **CONVERSION:** the originals are transformed to JATS.
3. **EDITING:** conversions are not perfect or authors asked us to make modifications to our final galley, so we need something to make changes directly to the JATS files.
4. **PRESENTATION:** JATS is not for humans. It need to be converted to something that humans can read (HTML, PDF, EPUB...) or they can be shown with JS helpers.

Based in conversations, testing and our experience, this year we decide we feel brave enough to recommend for each stage, the tool that is doing a better job, seam more perdurable, standard compliant, robust, but...

DISCLAIMER

The authors of this report are humans and don't have a crystal ball, so please don't claim then if they fail with their predictions. Take their recommendations as suggestions that need to be evaluated in each context.

So, without further delay, here you have the inventory¹ of the tools in the OJS ecosystem that, alone or in combination with others, let you work with JATS:

1 Report updated September of 2019

1) SUBMISSION

[A1] **OJS 3 core:** by default, OJS is able to work with originals in any format. DOC, DOCX, ODT, PDF even JATS are accepted by the platform. Markdown originals are supposed to be accepted, but not found much info about it. At this moment this is the best **solution for Submission** stage.

- Authors: PKP
- Status: Active (stable)
- Source: <https://github.com/pkp/ojs>

[A2] **OJS3-plugin ~ fidusWriter:** A development over fidusWriter to allow authors submit papers directly in JATS seems feasible. there are a couple more of plugins to integrate fidus in OJS, but we didn't test them yet.

- Authors: Alec (PKP) + fidusWriter
- Announcement: <https://www.fiduswriter.org/2016/05/29/fidus-writer-3-0-0-beta-4-released>
- Status: Active (beta)
- Source: <https://github.com/fiduswriter/fiduswriter-ojs>

2) CONVERSION

[B1] **myTypeset**: standalone command-line tool wrote in python to convert from DOCX format to NLM/JATS-XML. Is a fork of the TEI Consortium's OxGarage stylesheets and was integrated as part of the PKP XML parsing stack (now called OTS). It could be useful if you have plenty of DOCX and you want to batch convert them to JATS.

- Authors: M.P. Eve, Rodrigo, Alex (PKP), Dulip (PKP-UH)...
- Status: Active (stable)
- Source: <https://github.com/MartinPaulEve/meTypeset>

[B2] **Pandoc**: is the universal file converter. A standalone command-line tool wrote in Haskell, to convert from a lot of source formats to whatever destination. The list of source/destination formats [is impressive \(and beautiful\)](#). Again, it could be useful for batch conversion from your existing format to JATS, but it's not auto-magical (you will need work to make things fit). OTS also integrates this tool for it's conversions.

- Authors: MacFarlane (et. al.)
- Status: Active (stable)
- Documentation: <http://pandoc.org/MANUAL.html>

[B3] **DOCX2JATS**: standalone Java project developed to that facilitate the DOCX to JATS XML transformation for scientific articles.

- Author: Vitaliy (PKP community)
- Status: Deprecated
- Source: <https://github.com/Vitaliy-1/DOCX2JATS>
- Forum: <https://forum.pkp.sfu.ca/t/herramienta-docx-to-jats/31678/4>

[B4] **OJS3 plugin ~ JatsFrontPuller**: OJS3 plugin for generating the JATS XML Front section based on OJS metadata. Author says "is not need any more", because is included in the ojs3-markup plugin.

- Author: ajnyga (PKP community)
- Status: Deprecated
- Source: <https://github.com/ajnyga/JatsFrontPuller>
- Forum: <https://forum.pkp.sfu.ca/t/workflow-with-galleys-from-jats-xml-to-html-and-pdf/30361/4>
- Deprecation: <https://github.com/ajnyga/OJS3XMLWorkflow>

[B5] **PeerJ JATS conversion**: is not a tool ready to use "out of the box", but you can find there some helper files from PeerJ to convert, fetch and validate JATS.

- Author: PeerJ
- Status: Active
- Source: <https://github.com/PeerJ/jats-conversion>

[B6] **OJS3 plugin ~ docxConverter**: a plugin for OJS that converts from docx (created or exported from M\$Word, libre/openOffice or googleDocs) to JATS. The plugin is sothly integrated in OJS workflow: shows you a new button to make the conversion and creates a xml document attached to your submission. Figures and references are not yet implemented but the developer is working on it. At this moment, with an uncertain situation around OTS, this is the **best solution for Conversion** stage.

- Author: Vitaliy (PKP)
- Status: Active (beta)
- Source: <https://github.com/Vitaliy-1/docxConverter>
- Forum: <https://forum.pkp.sfu.ca/t/docx-to-jats-xml-converter/30224/136>

[B7] **docxToJats**: refactoring in PHP of former DOCX2JATS [B3]. The new tool still works as a standalone application and let you make batch transformations of scientific articles from DOCX to JATS-XML.

- Author: Vitaliy (PKP)
- Status: Active (beta)
- Source: <https://github.com/Vitaliy-1/docxToJats>

3) EDITION

[C1] **Substance/Texture (aka. LensWriter)**: it's kind of tricky to understand the naming of this tool. Making long story short: Substance is an generic web editor, that was customized to be JATS compliant. This "jats-specific-web-editor" was initially called "LensWriter" and today is "Texture". It's implementation is not still complete (don't cover the full JATS 1.2 specification) but last version it's solid as a rock and works with most common tags. Texture is released as a standalone-desktop application (that unfortunately don't read JATS, but reads a great format called DAR²) and also delivered as a JS application (only reads JATS) that could be integrated anywhere as a web editor.

- Authors: Substance Consortium (formed PKP, CoKo, SciELO and Érudit).
- Status: Active (stable)
- Site: <http://substance.io/texture/>
- Documentation: <http://substance.io/docs/beta5/about.html>
- Forum: <https://forum.pkp.sfu.ca/t/jats-xml-wysiwyg-editor/38329/3>
- Source: <https://github.com/substance/texture>
<https://github.com/substance/texture-jats>

[C2] **OJS3 plugin ~ texture**: a plugin for OJS3 that integrates the texture web editor in OJS. The plugin is included in the official release (need to be enabled in the plugin Gallery). Once installed and activated, you will see a new button under every submitted XML file. This button will drive you to a new page to let you edit the XML-JATS file and save it. Each edition will create a new version to your JATS and will attach images as article's dependent files. Although (as was explained) texture is still not full JATS 1.2 compliant, it's "good enough" for common usage. Both, the plugin and the texture-JS, are really solid and offer a very easy and clean user experience. It's, by far, the **best solution for Edition** stage that we have ever seen.

- Author: Alec (PKP) and Dulip (PKP-UH)
- Status: Active (stable)
- Source: <https://github.com/pkp/texture>
- Forum: <https://forum.pkp.sfu.ca/t/herramienta-docx-to-jats/31678/4>

2 A Document ARchive (.dar) is essentially a zip archive, containing the manuscript as a standard JATS XML file, as well as additional assets such as images and data files.

4) PRESENTATION

[D1] **OJS3 plugin ~ embedGalley:** OJS3 plugin to automatically converts a JATS XML galley to embedded HTML, which is shown on the abstract page. The plugin is using some code and XSL files from [B5] (PeerJ JATS Conversion).

- Author: Ajnyga (PKP community)
- Status: Active (stable)
- Source: <https://github.com/ajnyga/embedGalley>

[D2] **OJS3 plugin ~ OAI JATS:** OJS3 plugin to expose JATS XML via the OAI-PMH interface.

- Author: Alec (PKP)
- Status: Active (stable)
- Source: <https://github.com/pkp/oaiJats>

[D3] **OJS3 plugin ~ Lens Galley:** Galley viewer plugin integrating eLife Lens for OJS 3.0. Warning: This plugin collides with texture plugin.

- Author: Alec (PKP)
- Status: Active (stable)
- Source: <https://github.com/asmecher/lensGalley>

[D4] **OJS3 plugin ~ JATSParser:** OJS3 plugin to parse JATS XML and displaying it on article's abstract page as HTML (including a link to an auto-generated PDF). Parses JATS content (<body> and <back> sections), but metadata is extracted directly from OJS. Some tags still need to be implemented (figures, formulas, footnotes...), references are shown in AMA citation format and PDFs are not very customizable. For correct displaying should be used in conjunction with the theme but, as a reference, there are a couple of themes adapted to be JATSParser compliant (recommended: [OldGregg](#)). Check this post to find a [list of themes](#) and discover if they are "JATSParser friendly". Soon all themes will be compliant. Right now this the **best solution for Presentation** stage.

- Author: Vitaliy
- Status: Active (beta)
- Update: <https://forum.pkp.sfu.ca/t/jats-xml-wysiwyg-editor/38329>
- Source: <https://github.com/Vitaliy-1/JATSParserPlugin>

[D5] **OJS3 plugin ~ lensGalleyBits:** (aka. OJS3 Lens-viewer plugin for JATS and BITS) Galley viewer plugin integrating a version of eLife Lens for OJS 3.0 able to read JATS and BITS. This plugin can work with texture plugin.

- Author: Dulip (PKP-UH)
- Status: Active (stable)
- Source: <https://github.com/asmecher/lensGalley>

[D6] **OJS3 plugin ~ jatsTemplate**: This plugin permits OJS to use a basic JATS XML document generated from the OJS metadata and full-text extraction in cases where a better JATS XML document is not available. It is intended to be used in concert with the OAI JATS plugin to deliver JATS via OAI for journals that do not have better JATS XML available..

- Author: Alec (PKP)
- Status: Active (stable)
- Source: <https://github.com/pkp/jatsTemplate>

***) JAT SUITES**

There are some tools that work in conjunction to cover all or part of the JATS workflow. Those tools are:

[Z1] **OTS + ojs3-markup**: PKP released that combination of tools for the conversion stage. Full integrated in OJS, but with a server-client approach, this tandem of tools will let you convert from usual formats (rtf, odt, docx...) to JATS. This development was initially based on meTypeset and pandoc, but with the release of [GROBID](#) developers decide to stop and refactor to include this new tool.

[Z1a] **Open Typesetting Stack (OTS)**: this is the server component of the tandem. To facilitate the adoption, PKP offers this tool as a SaaS platform (but far as is released under GPL3, and also distributed in vagrant flavor, you can raise your own OTS server if you like). Build upon a stack of transformation tools (meTypeset, pandoc, grobid) is a “universal format converter” able to transform from whatever format to JATS and, during the process, it will also deliver additional versions of the file in HTML, PDF, EPUB, etc. Although it was started so long ago (initial project was called “Lemon8 XML”, and then moved to “PKP XML”), right now it is under a process of redefinition.

- Author: Alex Garnett (PKP)
- Status: Active (refactoring)
- Site: <https://pkp.sfu.ca/open-typesetting-stack>
- Source: <https://github.com/pkp/ots>
- Article: (2015) <https://www.ncbi.nlm.nih.gov/books/NBK279666>
- Video: (2015) <https://videocast.nih.gov/summary.asp?Live=16114&start=5130&bhcp=1>

[Z1b] **ojs3-markup**: is the client component of this tandem. Integrates OTS into OJS workflow for the Conversion phase (also editing with texture, but now this part it's has been decoupled and not need any more). As far as OTS is under development, the plugin alone is not useful right now.

- Author: Kassim (PKP)
- Status: Active (refactoring)
- Forum: <https://forum.pkp.sfu.ca/t/ojs3-markup-thanks-and-questions/29538>
- Source/Documentation: <https://github.com/kaschioudi/ojs3-markup>

[Z2] **Marcalyc**: it's not OJS but this SaaS application let you convert, mark and publish in JATS4R. Strong discourse about open knowledge and aims be free software, but I asked for the code plenty of times and the organization refuses to share. If you use their service you need to sign a contract with Redalyc (the organization behind Marcalyc) and you accept to add a redalyc cover to the exported files, publish in JATS the last 2 years and deliver to Redalyc all your articles in Microsoft Word format.

- Author: Redalyc
- Status: Unknown
- Site: <http://marcalyc.redalyc.org>
- Asking for code: <https://twitter.com/eaguadoredalyc/status/741000216912924672>

[Z3] **Convert2XML**: self described as “Converter based on JATS standard and fully integrated at Open Journal Systems (OJS) for the automatic conversion of articles to XML format. This converter follows the most important parameters required for SciELO Organization for markup language”. It's designed to fit well with OJS, but incomprehensibly it was developed over Microsoft server stack (.Net, ASP).

- Authors: SciELO
- Status: Abandoned (2017)
- Demo: <http://ojs.escire.net/demo248/index.php/escire> (user: demo / pwd: convert2xml) ← Error 404
- Source: <https://github.com/escire/Convert2XML>

Additional notes

At the moment JATS and OJS are able to keep article's metadata, the information can be inconsistent. PKP is discussing now about the right model to keep metadata synchronized and avoid propagation errors (JATS over OJS? OJS over JATS? Warn if data is not synchronized and ask me what to do ?...).

Conclusions

We believe in adopting a technology when formats are really consolidated and tools are solid enough to offer a fluid and pleasant experience.

In this sense, although we are pretty close (between 6 month and a year), our conclusion is that today, we can not recommend OJS editors to start migrating their content to JATS. As we have seen, we have a very rich toolset, but some issues still need to be addressed to make the workflow pleasant and fluid.

Even in 2018 we already known the **SUBMISSION** stage was going to be covered by OJS3 itself, without any need of extra plugins. With OJS3, you can submit DOCX documents for transformation, but not only this. The managements system is also ready to upload any other format (ODT, DAR, TEI, LaTeX, Markdown...), attach associated/dependent images, media or submit remote content (URL).

So here, we are convinced that **OJS3 “out of the box”** is the winner in usability, stability, but also in keeping you in a green-safe zone, knowing you are adopting and learning a tool won't be replaced in a long time.

At the end of January 2020, PKP will release **OJS 3.2**, that is deeply tested and include new great features, so you don't have more excuses to delay your OJS migration... and start asking your authors for SUBMISSIONS in DOCX format.

For the **CONVERSION** stage this year we have seen some plugins consolidated, while others where abandoned or moved their development in a different direction. If in 2018 we were talking enthusiastically about OTS, but the irruption of GROBID changed the scenario and today, without an estimation about when OTS development will be finish, we can't recommend it any more.

What to do if somebody is in a hurry and likes or need to start converting to JATS? Keep calm and go with **docxConverter**. The tool works really well (except for the images-upload and the references-parsing that are features underdevelopment, and are not a real drama because that could be addressed later with texture).

So, you should be thinking “why do you mark it in orange?”. Well, the fact is that it's quite probable this tool will be superseded someday by OTS (that will include more features as well as transformations to the most common formats), so yes.. you will be adopting a technology that could change, but the fact is the plugin's installation is a piece of cake³ and with docxConverter you will get a rock-solid and very usable tool. At the end, once converted to JATS, who minds who made it?

We have been asked about Marcalyc and SciELO toolsets. The idea is good and we are sure they are plenty of good intentions, but...

³ Vitaliy (the developer of docxConverter) intentionally reduced dependencies of the plugin to minimum to let you install it just with 2 clicks without any technical knowledge.

As far as we don't have access to the code and we disagree with the aggressive conditions of Redalyc's SaaS service, we can not recommend it. In the other hand, SciELO development is abandoned and also was based in a technology that collides with OJS stack (LAMP) so in our opinion, it won't be a good decision. Anyway, we had the opportunity of testing both tools, and JATS conversion with docxConverter (over OJS) is, as good or even better than the conversion services offered by both platforms.

So our recommendation for CONVERSION is keep waiting a little bit more, till the image-upload and the refence-parsing issues are fixed, install **docxConverter** and then start with the JATS migration.

For EDITION the texture plugins is the undisputed winner. The work done in the Substance Consortium is impressive and the integration is very intuitive and robust. With texture in the sand, we can't imagine why somebody will like to keep tagging with Marcalyc or even any commercial/proprietary software. Yes... some minor issues still need to be fixed (importing references without DOI from zotero, multi-language issues with figure, table names...) and would be nice to work with texture-desktop (¿what about remote publish to the journal?), and yes, PKP still need to define where the metadata will be placed, but the tool is absolutely "green zone" in terms of durability.

The **texture** plugin is the tool you want for EDITION (period).

Finally, PRESENTATION is the complicate part. A few years ago everybody wanted lens in their journals. It was an error. Yes, Lens interface is clean, we all like it a lot, but javascript is the wrong solution because it's obtrusive in terms of SEO and you still need to publish your article in HTML and/or PDF. So, what to do? JATSParser it's the perfect piece to fill the gap and convert your JATS to HTML and PDF. The PDF generation is done in the fly and development is based in tools that you can find in quite every server (so again looks like it will be a "double-click" installation) but unfortunately, right now, the PDF could not be easily customized to the journal needs (headers, single/double columns, typography, styles, citation format...). Fortunately, the developer (Vitaliy) is working to find a solution to this.

For PRESENTATION, if you don't need an automatic PDF generation, **OldGregg** (over JATSParser) is your base theme. Wait till it's migrated to OJS3.2 and create your child theme based on it... or develop your own theme compliant with JATSParser. If you want to expose JATS metadata in your OAI, then enable OAI-JATS plugin.

Summarizing, the final workflow will be:

SUBMISSION >	CONVERSION	EDITION	PRESENTATION
OJS3	docxConverter	texture-plugin	OldGregg (JATSParser)

Any comments and corrections are wellcomed.

Happy Jasting!