

LASSELLERAMSAY
Business Content Development

Case Study

Reinventing Content Development using XML

Tom Voltz
Summer 2008

Summary: Using XML to boost productivity

Lasselle-Ramsay worked with a client to develop an automated process to generate technical reference guides straight from the program code. This process uses XML extraction from firmware and XLST transformations to multiple outputs. The process eliminates costly and time-consuming update and revision tracking and review meetings by multiple stakeholders, and enables quick re-purposing of guides for print and online formats.

Key Concepts and Keywords

XML transformation, documentation development productivity, re-purposing documentation sources, single-sourcing, automating publications.



Reinventing Content Development using XML

A new way of writing a technical reference guide

When one of our longstanding clients came to us to create a reference guide for a complex network switch, they expected the process would be business as usual: several draft review cycles, hands-on demonstration time between the writer, programmers, and other stakeholders, and team meetings to go over bugs, errors, and technical updates. Creating an excellent, usable technical reference guide can be a long, hands-on process requiring extensive involvement from engineering subject matter experts and others across the organizational matrix, all of whom who prefer to focus on their respective tasks. Once we identified the business need, talked about what the client really wanted, and learned a more about their environment, we knew it was worth exploring a new way of getting the job done. And it was.

Working closely with our client, Lasselle-Ramsay developed a new approach which automated the process of creating the initial reference guide and virtually eliminated the manual effort associated with it. Our unique approach allowed us to harvest and extract the rich information already available in the embedded online help system and derive the syntax for complex commands needed for the reference guide. The result was the ability to generate and edit a 500 page PDF in a few days with almost minimal manual intervention.

Cost-benefits of automated XML documentation development

The benefits of this new process are summarized in Table 1:

Table 1. Automated Documentation Using XML

Feature	Benefit
Better completeness and accuracy	The new process gave us confidence that the content was complete and accurate for each switch, because it was derived directly from the firmware.
Faster time to market	Once the new process was in place, it allowed us to generate a 500 page reference guide in a few days, and also report on what changes were made to the product that might affect the documentation.
Cheaper with each iteration	While the effort required to build the first one or two manuals was similar to the effort and cost of creating them the traditional way, subsequent manuals could be created and updated at a fraction of the cost.



Reinventing Content Development using XML

Feature	Benefit
Less disruptive to organizational process	Because we were able to get 80% to 90% of the content directly from the firmware extract, we did not need to take much time from the Subject Matter Experts (in this case, the lead engineers).
XML flexibility and single-sourcing to multiple output formats	The new process generated clean XML output for the reference, allowing us to generate output in both HTML, PDF and CHM formats. Because the content is now transformed into XML, it's also much easier to transform it into new outputs that are not currently required or perhaps not even anticipated.
Unanticipated bonus benefits	The new process also allowed the documentation writers to provide specific feedback to improve the engineer-written online help very little additional cost.

Collaborating to solve productivity challenges

Because our client trusted us based on our many years of collaboration, they were open to looking at a new way of solving their challenges. Also, together we were able to look beyond the current project's immediate needs to see how we could build a solution that was going to be more cost-effective across multiple product lines.

Discovery: XML Extract

During the discovery phase, we were able to create an XML extract of the firmware that included all the help text, command parameters, and context specific options for the entire switch. This was a large XML file, nearly 700,000 lines of text. In addition, the data in the file had a complex, highly recursive data structure. It was also not in any standard DTD or Schema, but just raw data with XML wrappers. However, after a few iterations with the engineering team, we were able to tweak the output so that we could parse it into a much more useful XML format.

Prototype: HTML iterations

During the initial prototype phase, we kept our output transformations to basic HTML views, so that we could confirm that we were correctly extracting and organizing the information from the switch. Using HTML output allowed us to do very quick iterations with the client and their SMEs.



Reinventing Content Development using XML

Once the client was comfortable with the rendering and organization of the content, we built more elaborate transformations to produce clean DocBook XML. At that point, we moved the content into a CMS to allow easier management of the files and publishing process.

Debugging: Finalizing Automated Publishing

During the first iteration, the writer still had several weeks of work to clean up the output before the content was ready to publish. When we started the process for the second switch, we were able to harvest all of the command examples, "related commands" relationships, and "command categories" metadata from the first project to use in the second switch. So we were able to automate the build process to go from the raw firmware extract to a PDF that merged the meta data (related commands and command categories) and supplemental data (examples) with the core reference data (syntax and help text), to create a complete and accurate reference in a few days.

We were also able to re-factor our original transformations and reduce the writer effort by about 75% compared to the manual effort required the first time through the process. We were also able to reuse the stylesheets and transformations that created the PDF, adding just a few enhancements or modifications.

After the second pass was complete, we were able to compare the unedited version of the content to the one modified by the documentation team to provide a report to engineering about where we had to enhance the original online help text. This allowed the engineers to improve the command line help text, and also reduces the effort required for future revisions of the automatically generated manual.

The future of XML on Slashdot:

"Word processors, spreadsheets, games, diagramming tools, and more are all migrating into the browser. This trend will only accelerate in the coming year as local storage in Web browsers makes it increasingly possible to work offline. But XML is still firmly grounded in Web 1.0 publishing, and that's still very important."

(Future of XML Blog, 2008)



Reinventing Content Development using XML

Lessons learned

This project was a success in part because rather than jumping right in, we were able to look at the bigger picture and find a better overall solution. We learned that

- Getting started right away is not always the shortest path.
- During the discovery phase of the project we were able to ask a key question that helped us find a better solution: "What if any aspects of this problem something that will need to be repeated for this manual, and similar manuals?"
- Automating content processes requires working beyond the traditional boundaries of most documentation groups. In this case, the documentation team was asking the engineering team to actually generate an extract. Once they saw the benefit to them, the engineers were very happy to collaborate on the effort.
- Choosing a standard XML format allowed us to use existing tools to transform and re-use the output, as well as pull content programmatically from one manual to another.
- Rapid HTML iterations helped us get good feedback and built enthusiasm and support for the project.
- Budgeting some time for re-factoring the work in the second iteration provided a huge savings in the effort required after the transformation.

Holistic approach gets business results

Lasselle-Ramsay takes a holistic approach to the issues it helps its clients solve. We have a team of experts who bring a blend of tools to the problem, including project management, information architecture, as well as XML and XSLT transformation expertise. We have a long tradition of providing best-of-class technical writing and both online and instructor-led training.

If you think you are facing some unique challenges with your documentation processes, or just think there may be an opportunity to do things better, please feel free to contact us to discuss your needs, and how we may help your company meet their business goals

For more on information development best practices, visit LR.com or [get in touch!](#)



Works Cited

Future of XML Blog. (2008, February 7). *The Future of XML*. Retrieved July 2, 2008, from Slashdot: News for Nerds. Stuff that matters.:
<http://developers.slashdot.org/article.pl?sid=08/02/07/2141221&from=rss>

About Lasselle-Ramsay

Lasselle-Ramsay is a leading content development services organization that develops and delivers critical business information and learning solutions for new products, business initiatives, and regulatory requirements. We integrate content, content development processes, and technology to provide solutions that are up-to-date, on-demand, and targeted to increase productivity and drive know-how. We focus on technical documentation, training development, and web-enabled content solutions

Since 1982, Lasselle-Ramsay has provided information and learning solutions to leading companies in a variety of industries, such as Fireman's Fund (insurance), Genentech (biotechnology), Tyco (finance), Cisco Systems and Hewlett-Packard (technology). Contact us at www.lr.com for effective solutions to your next content project.

Legal

COPYRIGHT © 2008 by Lasselle-Ramsay.

Lasselle-Ramsay and LR are trademarks of Lasselle-Ramsay Inc. All other trademarks and logos are the property of their respective owners. All rights reserved.

Reuse and reprinting of this material by permission only. To obtain permission contact us at:

Lasselle-Ramsay, Inc
The Landmark Bldg P,
2075 Landings Drive,
Mountain View, CA 94043

Tel: 650.968.1220

FAX: 650.968.0949

www.lr.com

