

Content Reduction and Reuse in a Multichannel Publishing Environment

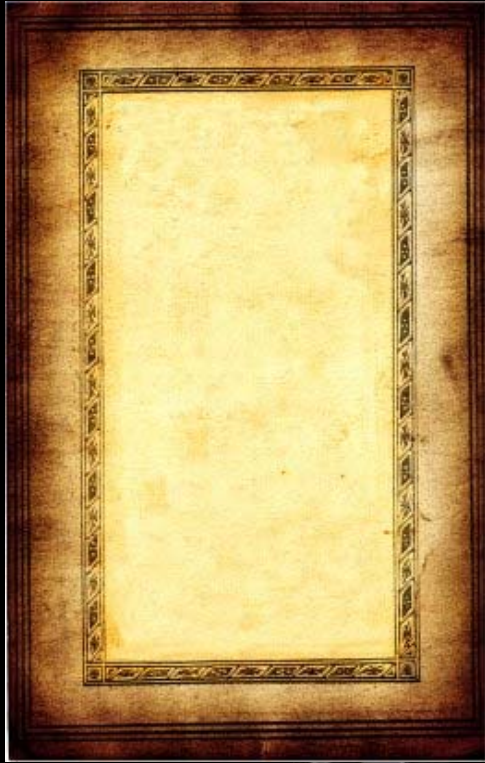
*Elizabeth Fraley and
Maria Ortmann
Single-Sourcing Solutions, Inc.*



Reuse Is Not New

We have always tried to improve efficiency and reduce effort

- Woodblock printing
- Illuminated pages
- Engraving plates
- Stamps
- Movable type
- Standard components
- Mass production
- Recycling



USA



Work Is Never “Done”

- Content always changes.
 - Errors
 - Changes
 - New feature
 - New product
 - New company
- Changes to any one of those can trigger other changes.
 - Data Model
 - Metadata
 - Stylesheets
 - Processes

Increased Efficiency Reduces Work

- Reusing and repurposing content is all about improving efficiency
 - Publications staff, support staff, and customers!
- Opportunities for reuse occur everywhere information intersects customers
 - At all points in the authoring and publication process (creation, construction, aggregation, and production)
 - All across the enterprise (engineering, marketing, customer support, product/project management, training, legal)

“Our goal is to provide the right content to the right person, at the right time, in the right format, and in the right media.”

*– Lori Fisher, IBM
Content Management
Strategies Conference 2007*

Example: Fictional Product Document

Based on the Neuros OSD Documentation

- About the Neuros OSD

- An open-source media center hardware appliance that encourages customer software development, modification, and enhancement (<http://www.neurosaudio.com/>)

- Customer-viewable information set:

Community Wiki

Bug Repository

Online Store

Google Group

Quick-start Guide

FAQ

Forums

Corporate Website

Corporate Wiki

Subversion Repository

Support Website

Product Specs

IRC Channel

Software Download

Legal bits

neuros

the power of freedom

neuros support

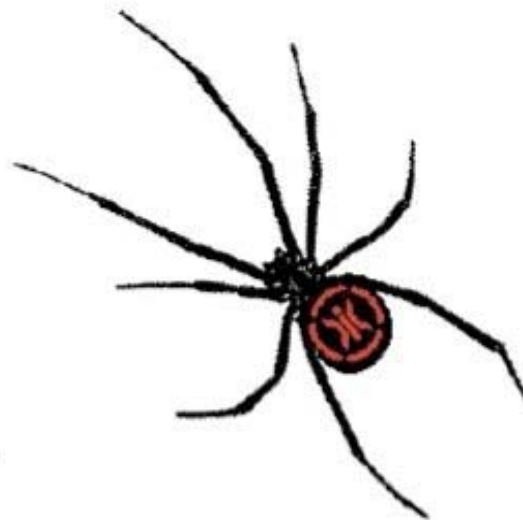
Untangle the neuros osd Web of Information

Welcome to the Neuros OSD information page. We will strive to keep this updated with the latest news and sources of information concerning the Neuros OSD.

As you may have figured out by now, the Neuros OSD is a rapidly evolving product that will continue to expand in functionality long after your initial out-of-the box experience.

This will come from a variety of sources:

- The first one being the official Neuros OSD firmware downloads. In it's early stages, you will see many of these to expand the functionality of the Neuros OSD and bring it up to the high standards that we expect out of our products.
- The second source of added features to your Neuros OSD will be from the open source community. We designed the Neuros as an Open Source Device (hence the name OSD). This means that anyone with some hacking ability can create applications and custom versions of OSD firmware. This also means that anyone can benefit from these plug-ins and 3rd party applications once they are finished. Some of these features are already well on their way to making it to your Neuros OSD.



Below are some handy links to keep you updated on all Neuros OSD information and news:

FIRMWARE

Need the latest official Neuros OSD firmware? Check [here](#).

BUGZILLA

Need to report a bug or see if it's already on our to-do list? Check out [Bugzilla](#). We have also created a nifty little pre-bugzilla landing page to soften the blow of Bugzilla format and guide you through the process. Check it out [here](#).

NEWS and ODNT

Want to dive into Neuros OSD hacking or just want to peak into the geeky behind the scenes action? [ODNT](#) (Open.NeurosTechnology.com) is the place for you. Links to source code, binaries, plugins, hacks, IRC channel and latest development news can all be found on ODNT.

LET'S TALK

Are you feeling the need to share your joy or ask for a little community help? The Neuros [forums](#) may be the right place for you. Another good source is the [Google group dedicated to the OSD](#). Feeling a little more hardcore and aren't afraid of supreme geek talk? The [Neuros IRC channel](#) may be better suited for you.

3RD PARTY APPLICATIONS AND PLUGINS

This is still in it's ubergeek stage. Eventually we will have a place for those that need the squeaky clean, Neuros stamp of approval type things, but for now [NEUROSVM](#) is probably your best bet in tandem with whatever info you can get from [ODNT](#).

INFO

The [Neuros Wiki](#) is the unofficial Neuros place for additional info. Anyone can add to it and thus can be great for getting the very latest or the place that will tell you how to toast bread in the CF slot of your OSD. We try to keep it up to date and as accurate as possible. Feel free to add to it for others to benefit from your gained OSD knowledge.

An [FAQ](#) is available on the wiki as well.

Additional documentation will also be found on the official Neuros Technology site in the [support](#) section. As the Neuros OSD is rapidly evolving, a nailed down version of the user's guide is yet to be available, but should appear there shortly and will be updated as new features are added in the Neuros OSD firmware. For now, a copy of the [quickstart guide](#) is currently available for your viewing pleasure.

free your media

Add An Imaginary Document To The Set

Containing information applicable to a new release

- Hardware product specs & feature list
 - Sources: HW-ENG, PLM/PDM, Marketing documentation
- Hardware appliance details
 - Sources: HW-ENG CAD files
- API details
 - Sources: SW-ENG specifications, implementation and build systems
- Outstanding bugs
 - Sources: Internal bug database, community bug database
- Troubleshooting information
 - Sources: Customer Support pages, IRC transcripts, FAQ, mailing list

Define Desired Output Targets

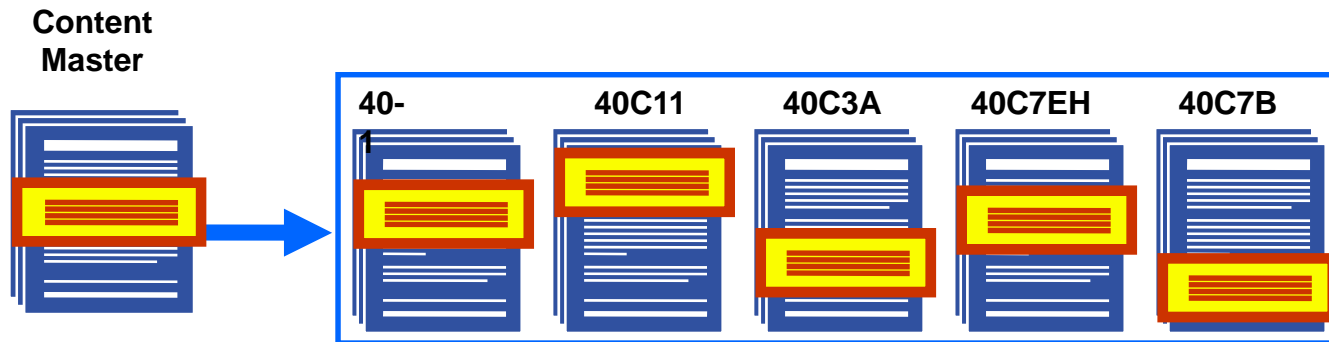
At release time

- Online Help (back into software build)
- HTML for corporate web
- Text/HTML mail for mailing lists
- Visio Stencil
- Diagnostic application
- PDF
- Other compiled documents (eReader, Palm, CHM, etc.)
- Wiki (both)

The Dimensionality Problem

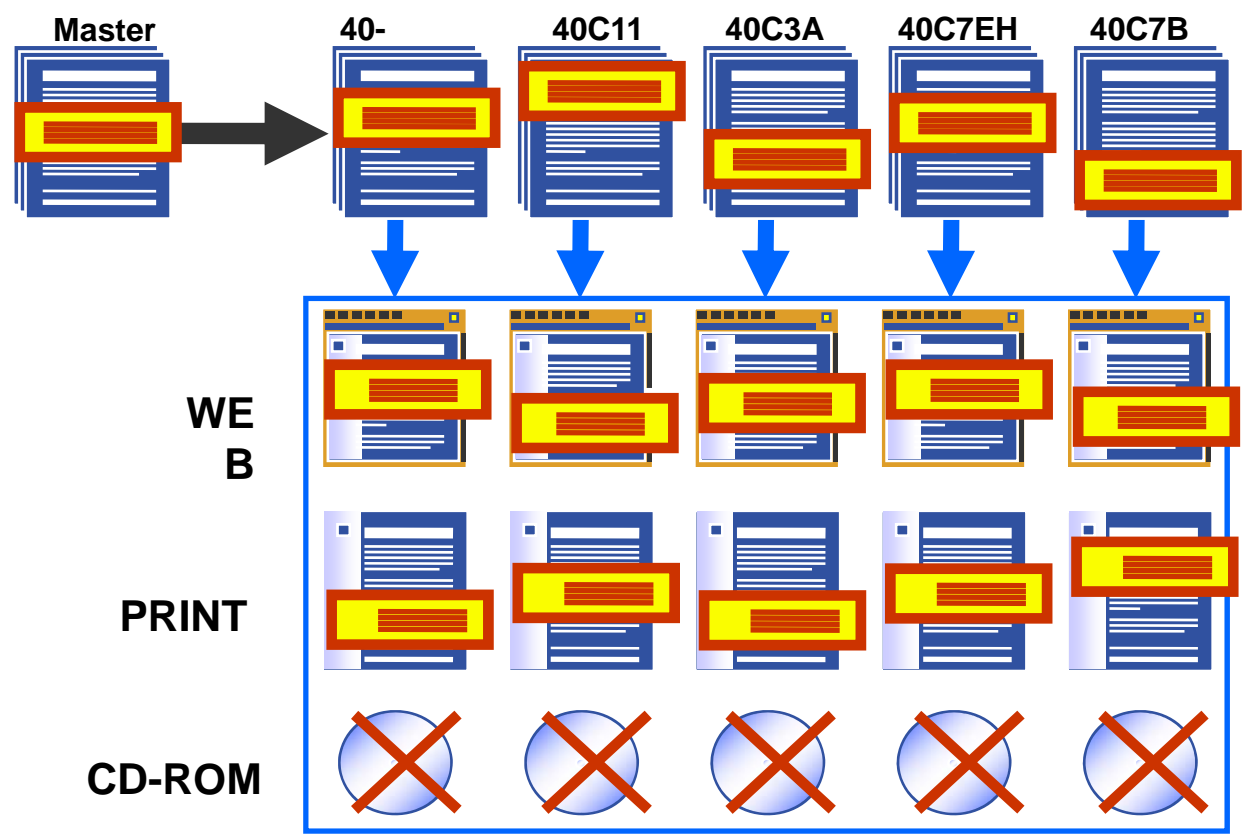
- Next 3 slides
 - Originally from “Authoring & Document Management at Hamilton Sundstrand” AUGI 2002 by Ray Hicks, Manager Tech Pubs, Hamilton Sundstrand, and Eric Helgerman, Architect, Creative Concepts
 - Exemplary depiction of the “Dimensionality Problem”

Many Changes Impact Other Documentation



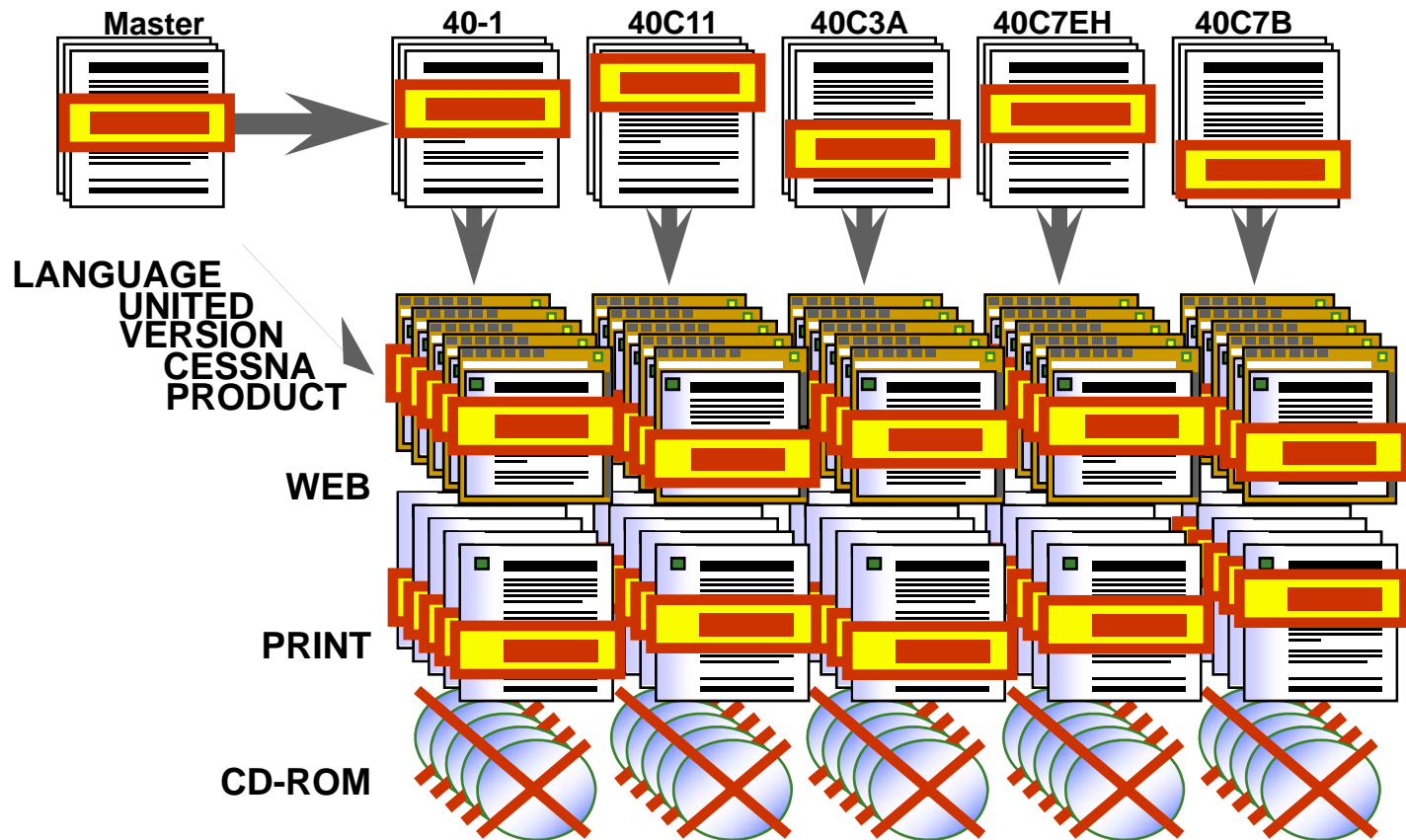
From: "Authoring & Document Management at Hamilton Sundstrand" AUGI 2002 by Ray Hicks, Manager Tech Pubs, Hamilton Sundstrand, and Eric Helgerman, Architect, Creative Concepts

One Change Affects Every Medium in Which the Documentation Is Delivered



From: "Authoring & Document Management at Hamilton Sundstrand" AUGI 2002 by Ray Hicks, Manager Tech Pubs, Hamilton Sundstrand, and Eric Helgerman, Architect, Creative Concepts

The Number of Documents Impacted by Just One Change Is Immense



From: "Authoring & Document Management at Hamilton Sundstrand" AUGI 2002 by Ray Hicks, Manager Tech Pubs, Hamilton Sundstrand, and Eric Helgerman, Architect, Creative Concepts

Now What Happens When...

- The next Neuros product arrives?
 - Overlapping source code, features, hardware components
- The company acquires or is acquired by someone else?
 - Look and feel changes, at the minimum
- A third party decides to be an OEM vendor and wants to rebrand and add hardware or software?
 - Look and feel changes, as well as information aggregation
- The company decides to add a new output target, format, or information repository?
 - New output format, new technology, or some other unknown factor

Things Suddenly Get More Complicated

- Content shared can be n-dimensional
 - Within a book
 - Across versions of a book
 - Between multiple books
 - Between multiple books and across versions
 - Between books, across versions, and across multiple product lines
 - To multiple formats
 - To multiple languages
- Complexity can increase exponentially

N-Dimensional Thinking is Hard



Image from <http://www.hawking.org.uk/about/mgallery1.html>

Mechanisms to Help?

- XML Mechanisms:
 - DITA's specialization, conref, copy-to attribute, variables
 - Xinclude (for content and formatting)
- Content Management:
 - Fileserver
 - XML database
 - SQL-based database

Content Reuse and Effort Reduction Guidelines

- Develop truly architected information
 - Modular writing is “the culture, not the tagging”
 - Integration and reuse requires thinking at a higher level
- Understand the whole
 - How any particular information module fits into the overall content picture
 - The relationships between information
- Find out why some things are reused and others are not
- Module size is important
 - Large modules are less likely to be reused
 - Small modules magnify the effects of overhead

Best Practices Recommendations

- You should chunk -
 - Core components that will be shared between documents
 - Major components that are redundant between documents
 - Components that form a logical unit of work for authors
 - When you want to allow authors to edit smaller pieces of the overall document without locking other authors out
 - When you want separate approval workflows for pieces of the document, or to be able to control access to chunks

- You should not chunk –
 - XML components at too low a level
 - When you don't have a clear reason to do so
 - Just because some of the information in a component is redundant

From: "Integrating XML Publishing with CMS: Best Practices"
AUGI 2003 by Eric Severson, CTO, Flatirons Solutions

Best Practices Recommendations

- Minimize chunks overall, and at each level of the document
- Adjust the DTD if necessary to fit the chunking approach
 - E.g. to put attributes that will be linked to metadata inside the chunk
- Optimize organization of chunks in the CMS
 - Create folders for logical groupings
 - Optimize navigation structure for authors
 - Provide metadata on chunks for optimum search / query
 - Group chunks with similar access security

From: "Integrating XML Publishing with CMS: Best Practices"
AUGI 2003 by Eric Severson, CTO, Flatirons Solutions

Best Practices Recommendations

- Suppose that specific content within the chunk is only applicable in certain contexts – does this mean that each of these must be treated as a separate chunk?
- No! Rather than chunking at a very low level, use “applicability” metadata to condition the use of content within the chunk
- An XSLT transformation can then be used to “filter” out content that isn’t applicable for a particular output

From: “Integrating XML Publishing with CMS: Best Practices”
AUGI 2003 by Eric Severson, CTO, Flatirons Solutions

Medtronic at PTC/User (AUGI) 2002

Reuse results for authors and translators

Authoring reuse

- Modules 82.3%
- Words 71.1%

Translating reuse

- Segments, 100% matches 70% avg
- Segments, fuzzy matches 91% avg

From: "The Road to Reusability: From Promise to Problems to Profits – Part Deux"
AUGI 2002 by Greg Johnson, Medtronic

Potential Guidelines for Content Reduction

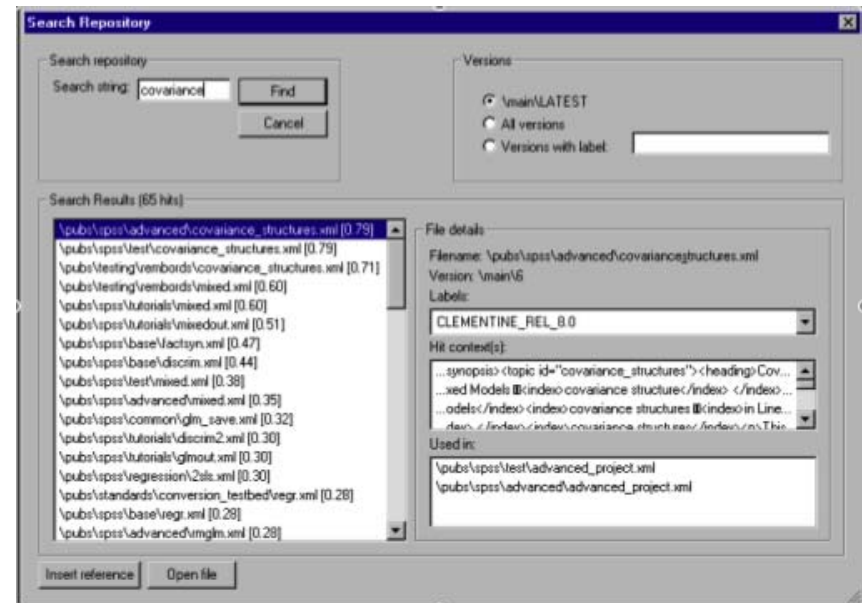
- Don't worry about whether you're starting too big or too small or whether everything is covered
 - It's not; It never will be.
- Look for ways to make small improvements
 - Where can your stylesheet do the work?
 - Where can tooling improve experience or efficiency (anyone)
- Refocus staff to “fill” desired roles
 - People who can think at the vision level start to take on Information Architect role
 - Change classifications as necessary to adjust requisition pool

More Guidelines for Content Reduction

- Docplans should include all information that relates to or affects the information contained in the document that is being generated across the enterprise
 - What is Marketing writing?
 - What about Customer Support?
 - What don't you have to write?
- Staff members can use the time they get not writing duplicate information to keep up or take on the responsibilities of the new roles

SPSS at PTC/User (AUGI) 2004

- CMS-based Search Dialog
 - Text search
 - Limit by version
 - List of hits
 - File details for selected hit
 - Link backtracking (“Used where”)
 - Autogenerate reference to selected file



From: “Content Management From the Ground Up”
AUGI 2002 by Clay Helberg, SPSS

Potential Guidelines for Managing Reuse

- Corporate-level tools vs. local tools
 - Tool team should be corporate
 - Tools to improve efficiency should be standardized and subject to a managed release process
 - This does not mean that tools must all be from one vendor or of one language.
- Tight vs. loose integration
 - Opt for loose integration!
 - Upgrading customized code is worse than upgrading custom code

More Guidelines for Managing Reuse

- Use standards whenever possible
- Know the standards you choose
 - Mailing lists get the following question all the time: "how do I do do X?"
 - The answer is always "use this, the spec has already done the work for you"
- Keep an eye on developing standards, languages, and technology
 - Ever changing pool of technical availability
 - New output targets/channels appear every day

Questions

Talk to People!

Single-Sourcing Solutions, Inc.

Liz Fraley

liz.fraley@single-sourcing.com

Maria Ortmann

maria.ortmann@single-sourcing.com

