

# Writing for Maximum Reusability

A Case Study of DITA-based Content Production

Andrea Leszek  
Director, Documentation  
Salesforce.com

~~SOFTWARE~~



salesforce.com ~~SOFTWARE~~  
Success On Demand.™

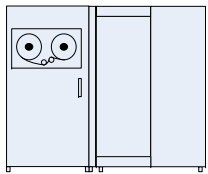
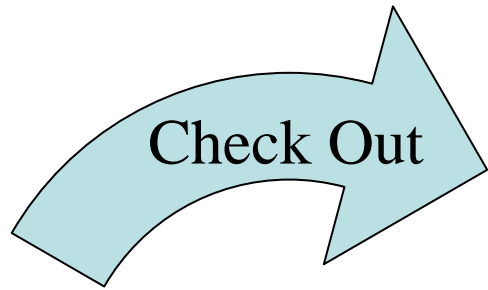
# Agenda

- **Salesforce.com documentation architecture**
- Types of content reuse in DITA
- Techniques for writing for content reuse
- Processes and tools for verifying reused content
- Producing different outputs
- Summary
- Lessons learned
- Q&A

# Salesforce.com Documentation Architecture

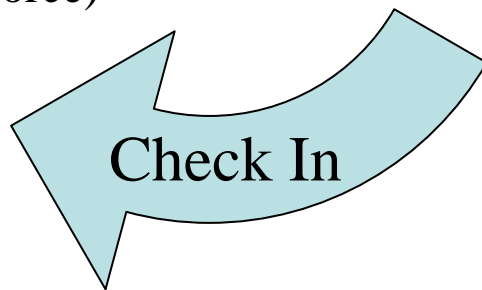
- Over 3000 XML topics, marked up using DITA
- Content management handled by Perforce source control system
- Content shared between 5 different types of guides
- 6 different output formats – PDF, vanilla HTML, Web help, Eclipse help, Wiki, CHM

# Authoring Workflow



SCMS (Perforce)

Author and Edit



# Agenda

- Salesforce.com documentation architecture
- **Types of content reuse and how we use them**
- Techniques for writing for content reuse
- Processes and tools for verifying reused content
- Producing different outputs
- Summary
- Lessons learned
- Q&A

# Types of Content Reuse in DITA

Reuse Opportunity	Solution
1. Multiple similar deliverables	Flag some content as conditional
2. Piece of content used in many different contexts	Include it in different topics using content references
3. Topic used in many different deliverables	Include it in different deliverables through DITA maps

# 1. Marking Content as Conditional

- Use *attributes* and *values* to flag content as conditional
- Primary use case: Similar, but not identical, products
- Example with **platform** attribute and **OT** value:

```
<p platform="OT">The Office Toolkit plug-in  
  makes it easy for developers to access the API  
  from Microsoft Office products.</p>
```

# More Granular Conditional Content

- Can be applied to sentences, words, phrases:

Use the `<ph platform="API">c</ph><ph platform="OT">C</ph>`reate call to create objects.

API doc output: Use the **create** call to create objects.

OT doc output: Use the **Create** call to create objects.

- Can be applied to other content, not just text
  - Images, index entries, tables, etc.

# Other Attributes for Conditional Content

- `platform` – for OS or hardware; we use for similar APIs
  - `audience` – for user of content; we use for “future” content
  - `product` – for different products
  - `rev` – for different revisions/releases
  - `otherprops` - for anything else
- 
- You choose what values of these attributes are valid.
  - You can create new attributes by *specializing* the `props` attribute.

# Scoping Conditional Content

- Do not conditionalize content that is “common to all.”
- To make an entire topic conditional, conditionalize the corresponding topic reference in the DITA map file.

# Producing Conditional Output – DITAVAL files

- DITAVAL files control which conditional content is part of the output
- We use a simple “exclude” model – if content has an attribute value specified in the DITAVAL file, it is excluded from the output
- Example DITAVAL for API doc output:

```
<prop att="platform" val="OT" action="exclude" />
```

Any content with `platform="OT"` is not in the API output.

## 2. Using Content References - conrefs

- Use `id` and `conref` attributes to reuse content
- Primary use case: Content that is the same in different topics
- Example:
  - Source file element with `id` attribute:

```
<p id="AppExchange">The AppExchange is an on-demand application-sharing service from salesforce.com that allows you to browse, install, and share apps and components built for the platform.</p>
```

- Target file element with `conref` attribute:

```
<p conref="appexchange_concepts.xml#appexchange_concepts/AppExchange"></p>
```

## Other Uses for Conrefs

- Apply at a phrase level for “variables”
- One common file for product names: reusable.xml

- Example:

- Reusable.xml Source file element with `id`

- ```
<ph id="ProdName">Salesforce</ph>
```

- Target file element with `conref`:

- ```
<ph conref="..\reusable.xml#reusable-  
topic/ProdName"></ph>
```

### 3. Reusing Entire Topics

- Primary use case: Topics common to multiple deliverables
- Examples: glossaries, “conventions” topics, different types of deliverables
- Simply add the topic to ditamap files as needed

# Agenda

- Salesforce.com documentation architecture
- Types of content reuse and how we use them
- **Techniques for writing for content reuse**
- Processes and tools for verifying reused content
- Producing different outputs
- Summary
- Lessons learned
- Q&A

# Techniques for Writing for Content Reuse

- Clear tagging guidelines
- Authoring tool locked down to allowed set of tags
- Consistent, shared style guide
- Consistent voice in addressing users
- Clear writing workflow using source control system
  - Why not a CMS?

# Agenda

- Salesforce.com documentation architecture
- Types of content reuse and how we use them
- Techniques for writing for content reuse
- **Processes and tools for verifying reused content**
- Producing different outputs
- Summary
- Lessons learned
- Q&A

# Tools for Verifying Content Reuse

- Custom tools within authoring tool to find reused content
  - Conreffed By – Finds all files that conref an element containing an id
  - Open Conref – Opens the file containing the source content for a content reference

Build	Ctrl+1
Conreffed By	Ctrl+2
Open Conref	Ctrl+3

# Processes for Verifying Content Reuse

- Writer-initiated test builds to verify content automatically
  - Verifies reused content exists
  - Verifies content reuse is structurally valid
  
- Quality testing to verify content is correct

# Agenda

- Salesforce.com documentation architecture
- Types of content reuse and how we use them
- Techniques for writing for content reuse
- Processes and tools for verifying reused content
- **Producing different outputs**
- Summary
- Lessons learned
- Q&A

# Producing Different Outputs

We do it using:

- Ant
- DITA Open Toolkit
- In-house customizations
- Automated nightly builds

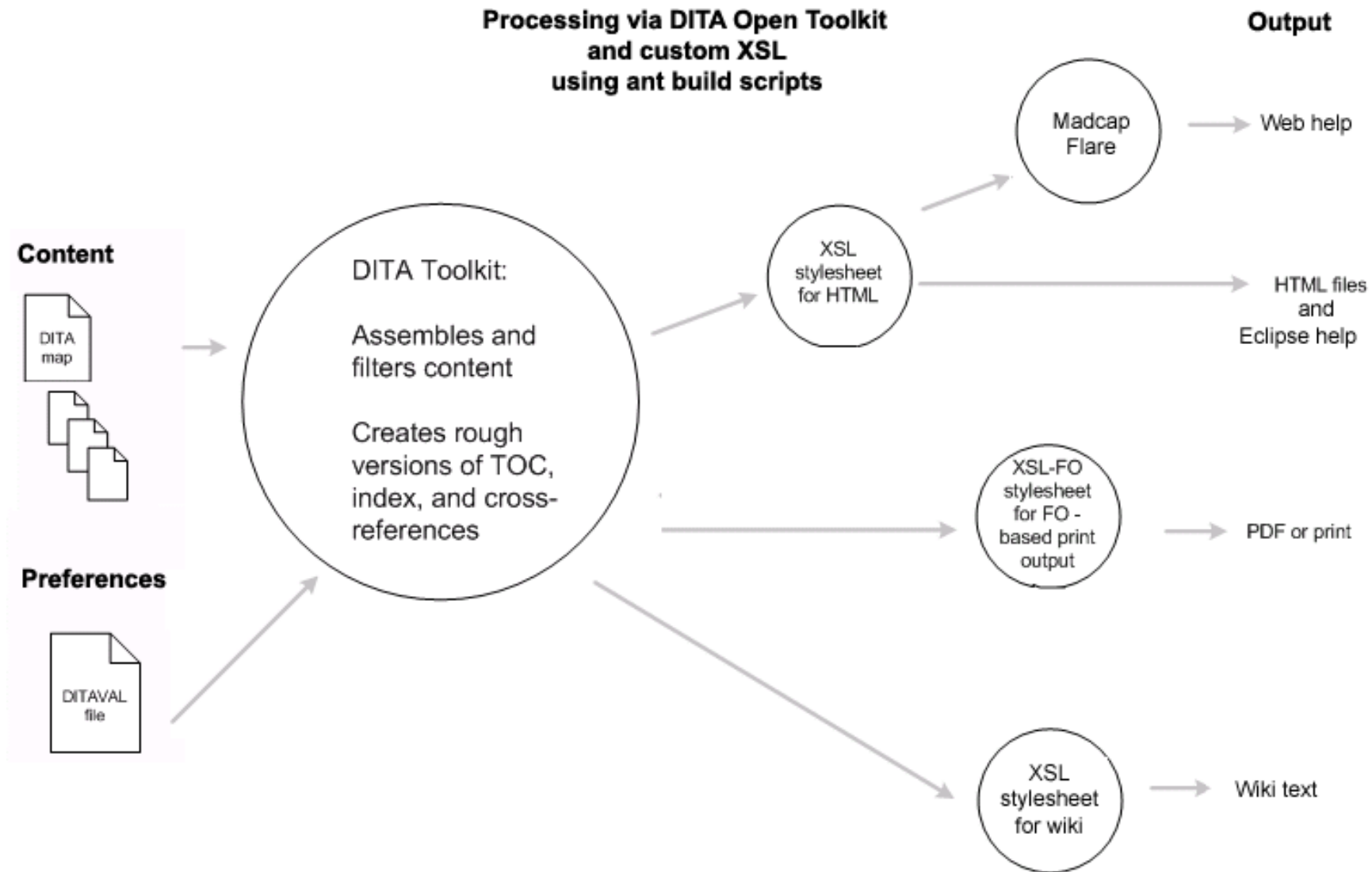
# Ant

- Apache ant, an automated build tool, is central to our process
- Using ant, the writer can build multiple types of outputs from many different sources
- Using ant ensures that every time output is built, it is done using a tested, reproducible process
- Build process becomes the bully, ensuring that nothing gets to customers without being verified for correctness

# DITA Open Toolkit (OT)

- The DITA OT is the framework for our builds
- It is an open-source collection of scripts and XSL (Extensible Stylesheet Language) files
- It does the following:
  - Removes conditional text according to settings recorded in the DITAVAL file
  - Glues together modular content
  - Generates index, table of contents, See Also links, and cross-references
  - Applies formatting
  - Produces deliverables in multiple formats; includes production-ready, but generic, output for HTML, PDF, RTF, and other formats

# How the DITA Open Toolkit Processes Content



# In-house customizations

- Using XSLT, we override the generic outputs to customize them for our needs, including:
  - Branding all content
  - Formatting
  - Inserting JavaScript
- Overrides are output specific, but sharing is possible
  - The override for inserting a copyright into HTML is different than the override for the PDF, but both share the same copyright text
  - Overrides that affect multiple output formats are maintained in one location and imported into the overrides for each appropriate output type
- Overrides allow automated modifications, with no content changes

# Agenda

- Salesforce.com documentation architecture
- Types of content reuse and how we use them
- Techniques for writing for content reuse
- Processes and tools for verifying reused content
- Producing different outputs
- **Summary**
- Lessons learned
- Q&A

# Summary

- DITA and content reuse has:
  - Increased our team’s productivity
  - Allowed us to produce more deliverables in more formats
  - Decreased localization costs – both from XML and reuse
  - Broadened the scope of integration possibilities
    - reusing content from code
    - sharing with curriculum developers
    - Single-sourcing knowledge base content
- Reuse has been successful with:
  - Clear, consistent guidelines for DITA tags and writing styles
  - Automated tools and processes

# Agenda

- Salesforce.com documentation architecture
- Types of content reuse and how we use them
- Techniques for writing for content reuse
- Processes and tools for verifying reused content
- Producing different outputs
- Summary
- **Lessons learned**
- Q&A

# Lessons We've Learned

- Don't reuse chunks smaller than a sentence  
(except for product names)
  - Impact on localized content is huge
- Inline links are bad
  - Leads to missed reuse opportunities
- Make sure writers understand model for conditional content
  - Excludes vs. includes

## Q&A

For more information on DITA, see the main DITA page at

<http://dita.xml.org>

For more information on the DITA OT, see

<http://dita-ot.sourceforge.net/>

For more information on ant, see the Apache ant page at

<http://ant.apache.org/>

Other questions? [aleszek@salesforce.com](mailto:aleszek@salesforce.com)