

Conditionality and Versioning in Author-it



What We'll Cover

- ▶ Definitions
- ▶ Methods for creating conditional content
- ▶ Methods for managing conditional content and versions
- ▶ Publishing controls
- ▶ Archiving strategies
- ▶ Considerations for localization and translation
- ▶ Architectural considerations
- ▶ Choosing the right techniques for your requirements



What Is Conditionality?

- ▶ Conditional content behaves differently depending on:
 - Where it is used
 - How it is published
- ▶ For Author-it, conditional structures enable a single content object to be used in different contexts, with variations that are appropriate to each context.
- ▶ There are different techniques that you can use to create conditional structures.

What Is Versioning?

- ▶ Versioning can mean different things to different people.
- ▶ Some different types of versioning:
 - Ensuring that the most current information is included in latest publications
 - Archiving historical content that is not part of current publications
 - Producing content collections that vary by product release, edition, audience, or other factors
 - Maintaining related content for multiple active versions
 - Archiving published content that is part of each software release
- ▶ Defining what you mean by versioning is crucial.
- ▶ You can build conditional structures to create different versions of content.

Creating Conditional Content



Methods for Creating Conditional Content

- ▶ Create different books with combinations of shared and unique content.
- ▶ Duplicate topics or other objects to modify independently of the original objects (use naming conventions to distinguish similar objects).
- ▶ Use variables for short pieces of content that can have different values depending on context (library, book, or topic).
- ▶ Create **composite topics** that contain both unique text and **embedded topics** that provide shared content.
- ▶ Create and apply media-specific styles for content that should only be published to print, help, or web output.
- ▶ Create media-specific topics or other objects that are only published to print, help, or web output.

Versioning Features

- ▶ Use the Author-it **version control feature** for a linear versioning process (only one version can be active at a time).
- ▶ Use **object variants feature** or other conditional content techniques when you need to maintain multiple versions simultaneously.
- ▶ Object variants contain content that is specific to predefined criteria, such as release or operating system.



Books as Conditional Constructs



- ▶ An Author-it book is a container for content to be published for a specific audience or purpose (e.g., User Guide or Training Guide).
- ▶ Two books may contain many of the same topics, as well as topics that are unique to that book.
- ▶ Updating shared topics affects all of the books where those topics are used.
- ▶ Updating unique topics only affects the book where those topics are used.
- ▶ You can duplicate a book as a quick way to create a starting point for customization.

Duplicating Topics

- ▶ Duplicating topics is a quick and easy way to customize a topic for a new book - but it means you are *not reusing the content* (may have to maintain it in two places).
- ▶ To duplicate a topic in a book:
 - Select the topic in the book contents.
 - Right-click and choose Locate to find the topic in a folder.
 - Go to the Manage tab in the main window command ribbon and choose Duplicate the selected object from the Versioning command group.
 - After confirming the operation, the object is duplicated.
 - Open the object in Properties view and modify the Description value on the General tab to indicate where this object is applicable.
 - Add the duplicated topic to the book, and (if necessary) remove the original topic.



Using Variables

- ▶ Create variables in Author-it Administrator.
- ▶ Text, List of Values, or File Object variables.
- ▶ At library level, you can define a default value.
- ▶ At the book level, you can assign a value to be used throughout the book (overrides library default).
- ▶ At the topic level, you can assign a value that will *always* be used for that topic (overrides book value).
- ▶ Variables can also be used to apply **metadata**:
 - To view library variables, select Manage Variables from the Author-it icon menu on the main window. Right-click on a variable to see usage or assignment information throughout the library.
 - You can perform searches in the form <VarName> or <VarName>=value.
- ▶ Variable assignments in subbooks are ignored.

Embedded Topics

- ▶ Embedded topics are used within other topics.
- ▶ Useful when you have short pieces of text that are used in multiple topics (e.g., menu option or field labels).
- ▶ When dragging a topic into another topic's contents, drop the topic and choose Embed Topic from the pop-up menu.
- ▶ To copy content and edit independently of the original, choose Paste As Text when dropping a topic.
- ▶ An embedded topic cannot be edited in the topic where it is included; double-click on the embedded topic to edit it.
- ▶ Changes are reflected everywhere the topic is embedded.
- ▶ Select text and press Alt + Enter to create an embedded topic.
- ▶ Store embedded topics in a separate folder and apply folder permissions to restrict the ability to modify the information.

Composite Topics

- ▶ Composite topics combine customized text with standard content
- ▶ Customized text is included directly within a composite topic
- ▶ Standard content is included as one or more embedded topics within a composite topic
- ▶ Used within a book structure where the customized text is applicable



Media-Specific Output



- ▶ On the General tab of an object, the Include object in settings enable you to control whether the object is published to Print, Help, and Web outputs:
 - Define paragraph and character styles so that content marked with those styles is published only to selected output types (or no outputs).
 - Define topics or other objects so that they are published only to selected output types (or no outputs).
- ▶ Examples:
 - Comment style - does not publish to any output (indicates planned revisions or revision history for a topic).
 - Question for Reviewer style - only published to review draft formats. (Use style substitution in a "final" publishing profile to ensure that questions are not published in final publications.)
 - Non-publishing Revision History topic - use at the beginning of a book, to document the revisions made to the book over time.

Version Control Feature

- ▶ Version control is typically used for a linear versioning process - only one version of the content can be active.
- ▶ Useful when content is transitioned to a new version and older versions are archived.
- ▶ Versions can be Active, Inactive, and Redundant.
- ▶ New versions are inactive until they are made active, and can be edited independently of the original.
- ▶ When a new version is made active, the previous version is made redundant (read-only), and the new version is substituted for the old in any active books.
- ▶ Only one version can be active at a time.

Object Variants

- ▶ Create and maintain different variants of objects.
- ▶ Variants are based on specific **criteria variables** that you define (e.g., release, operating system, domain, ...).
- ▶ Variants remain linked to their primary (parent) object.
- ▶ Multiple variants can be active at one time.
- ▶ Filter against criteria to display variants within a book.
- ▶ Choose appropriate criteria when publishing.

Managing Conditional Content and Versions



Methods for Managing Versions

- ▶ Use folders and folder permissions to organize and control access to content for a particular version or condition, or content that is heavily reused.
- ▶ Use release states to mark content for a particular version (may also include permissions).
- ▶ Use **object history feature** to enable compare and revert (rollback) to older revisions of objects.
- ▶ Use non-publishing styles to track revision history for a topic, or a non-publishing topic to track revision history for a book.
- ▶ Use Show Relationships to see where content objects are used (or dependencies on other objects).
- ▶ Use the Author-it Localization Manager to create a **target library** that can be updated in a controlled way from the original **source library**.

Using Folders to Manage Versions



- ▶ Use folders to organize content that is:
 - Specific to a particular version
 - Shared across all versions
 - Embedded within other topics
- ▶ Use permissions to limit permissions to modify content:
 - Content that is specific to a version - e.g., if specific writers are responsible for content of different versions
 - Content that is shared, either directly or through embedded topics - e.g., if you want revisions to shared objects made by senior writers who understand the implications for how those changes would affect all of the books using the content



Using Release States to Manage Versions

- ▶ If your release state structure is fairly simple and you do not have a large number of versions to maintain, you may be able to use release states to mark version-specific content objects.
- ▶ Release states can be defined to use an object color.
- ▶ Object color provides a visual reminder of content status when objects are viewed in folders, books, or search results.
- ▶ Drawbacks:
 - This is not an intended use of release states, and does not work well if you have a large number of workflow stages or versions (not enough distinct colors).
 - This approach may not be helpful if some writers are color blind.
- ▶ Variants provide another way of showing visual markers of version-specific content.



Object History

- ▶ Set up object history settings in Author-it Administrator:
 - You can choose save points (for every change, release state changes, author changes, template changes, or publishing).
 - Object history can be stored in the database or in external files (no difference in usage).
- ▶ If object history is not enabled, you still see an audit trail for changes to objects when you right-click and choose Show History.
- ▶ If object history is enabled, you also have the option to Compare to a previous revision, or to Revert to a previous revision.
- ▶ When you use Revert, the rejected revision is still part of the object history.

Show Relationships



AuthorIt®

- ▶ Right-click on an object and choose Show Relationships to see relationships for the object:
 - Objects used by the selected object (dependencies on other objects)
 - Other objects that use the selected objects (e.g., books, topics, or other objects where the object is used)
 - If applicable, localizations or translation jobs that include the object
- ▶ On the Show Relationships window, you can Open, Locate, or Show Relationships for any related object.
- ▶ When working in the Book Editor, you can use the Books using this Topic tab to see what other books use the topic you have open. (This tab is dockable.)

Using the Localization Manager

- ▶ The Localization Manager manages translation content and provides a method for managing ongoing updates to content.
- ▶ A source library contains original content.
- ▶ A target library is a duplicate of the source library at a specific point in time.
- ▶ All objects in a target library retain the object codes used in the source library.
- ▶ New content can also be created in the target library.
- ▶ Updates to a target library identify any changes in the source library since the last update. You can review and accept or reject updates.
- ▶ This is not an intended use of the Localization Manager - talk to your Author-it account manager before implementing this approach!

Publishing Controls



Customizing Publishing



Author-it®

- ▶ Use ***publishing profiles*** to customize the publishing process:
 - Publishing profiles are defined in Author-it Editor, and stored in a subfolder under the Standards folder (Restricted in V4).
 - By default, Author-it 5 provides publishing profiles for each type of media output (Word, PDF, HTML Pages, etc.)
- ▶ If you have defined object variants, choose the variant criteria when you publish.



Setting Up Publishing Profiles

- ▶ Define additional publishing profiles to describe special requirements for a particular type of output.
- ▶ Be careful! Each publishable book or book template must be identified in a publishing profile:
 - To disable a publishing option for a particular output type, remove all book templates from the Applies To tab.
 - If you add a new book template, also add that book template to any applicable publishing profiles, or ensure that the publishing profile applies for *all types* of books.
- ▶ If variables are a) defined as Prompt Before Publish and b) assigned to a publishing profile, you are prompted to supply values for the variables when publishing.

Using Publishing Profiles



- ▶ Publishing profiles specify conditional publishing controls:
 - Outputs to which a type of book can be published
 - Which users can publish using a publishing profile (based on folder permissions)
 - Release states to be included in a publication
 - Substitution of Word templates, style objects, media objects (page layouts), book content objects, custom stylesheets, and external files
 - Exclusion of specific topics or types of topics based on release states, folder locations, or variables
 - Pre- or post-processing for published output.
- ▶ If you do not see any available publishing options for a book, it probably means that the book template (or object) is not included on the Applies To tab of any publishing profiles.

Archiving and Retrieval



Archiving Strategies



- ▶ Archive published documents for specific drafts or releases. (Word, PDF, HTML, help, ...)
- ▶ Archive online help outputs in a source code control system as part of a specific software build.
- ▶ Publish books to XML for archiving of all content source objects.
- ▶ Publishing to XML can also be used to duplicate a book in the same library.

Publishing to XML To Archive a Book



- ▶ When you publish a book to XML, the resulting XML file contains all objects needed to recreate the Author-it book (book, topics, templates, media, ...).
- ▶ XML can be imported into a new library to recreate the book and all supporting objects (be sure to import all objects).
- ▶ To reproduce the original output, you may also need to archive external publishing template files as they existed at that point in time.
- ▶ Important! The default behavior when importing an object in an Author-it XML file is for the imported object to overwrite the object if it already exists in the library. So be very careful if you are importing an XML file back into the same library.

Using XML To Duplicate a Book



- ▶ Publish to XML, or select the objects you want to duplicate and save as XML.
- ▶ Only import unique content objects that need to be duplicated - not templates, media objects, or objects that should be shared.
- ▶ Do not overwrite originals of unique book content (default behavior) - create new objects instead.

Localization & Translation



Managing Translation Content



- ▶ If topics are duplicated, both copies will require translation.
- ▶ Be sure that all variables are assigned to books if you want the values to be translated - even if you are using the library default values.
- ▶ Use embedded topics for labels that are used repeatedly but do not need to have different values (e.g., menu labels in a GUI).
- ▶ To minimize duplication of content for translation, do not create variants of complete topics. Instead, use embedded topics for paragraphs that represent content that needs to differ, and create variants using embedded topics.
- ▶ Target libraries can have object history enabled to keep track of changes to objects.

Managing Updates and Translation Jobs



AuthorIt®

- ▶ Exclude non-publishing objects or styles from translation jobs.
- ▶ Content in specific folders can be excluded from target libraries and translation jobs.
- ▶ Release states can be used to control updates from a source library to target libraries, and to determine which content is included in translation jobs.
- ▶ Show Relationships display provides information about localizations and translation jobs for objects.

Next Steps



Conditional Architecture

- ▶ Predefine specific scenarios for the types of conditional structures you need.
- ▶ Identify the types and methods of conditional structures needed for each scenario.
- ▶ Create model structures and test for all required outputs.
- ▶ Include localization in your testing if you need to manage content for translation.
- ▶ Document best practices and techniques used in your organization.



Choosing the Right Technique



AuthorIt®

Method	Usage
Unique Book Objects	Quickly create a new project based on an existing book. As needed, add topics or create customized topics (possibly using conditional constructs).
Duplicate Objects	Quickly create a copy of a topic that can be revised without affecting the original. Use Paste as Text to copy a topic's contents into another topic.
Variables	Use variables when content needs to vary from one project (book) to another. It is rarely useful (and sometimes harmful) to assign values to variables for specific topics.
Embedded Topics	Boilerplate text that needs to appear in multiple topics. Can also be used to include shared content in a topic that is unique to a particular version or audience.
Composite Topics	Include a composite topic in a book for a particular version or specialized audience. Any content that is unique to that version is included directly in the composite topic. Base content that is shared with other versions or audiences can be included using embedded topics.
Media-Specific Output	To include or exclude styles or objects from publishing to specific media types (print, help, or web).
Version Control feature	To develop version-specific information that is made active when it is released for publication. Only one version of an object can be active at a time.
Object Variants feature	To maintain multiple versions of content at the same time, based on any criteria that you define. To minimize duplication of content, use variants of embedded objects for content that differs among variants.
Folder Organization	Organize shared or version-specific content into folders. Use folder permissions to control modifications to content.
Naming Conventions	Use the Description of an object to make clear where it should be used. The Description can be changed independently of Heading values. For example, if you have 100 "Overview" topics, you can modify the Description values to indicate what each is an Overview of.
Object History feature	Provide the ability to compare and roll back to previous revisions of an object.
Publishing Profiles	Customize the publishing process. Substitute or exclude content, styles, page layouts, or perform operations and transforms on published output.

Additional Resources



AuthorIt®

- ▶ Other free webinars and training opportunities are described on our **Author-it web pages** (<http://www.sdicorp.com/authorit>).