

# Daisy Books Overview

---

## Introduction

The purpose of Daisy Books is to publish a set of aggregated Daisy documents as one whole, thus like a book or manual. Books can be published in various forms, including PDF and HTML. The HTML can be published as as one big page or chunked into multiple pages, the chunks don't need to correspond to the original document boundaries.

Technically, Daisy Books is part of (= integrated in) the Daisy Wiki, though the process of publishing a book is completely unrelated to that of publishing a Wiki page (except that both reuse the services offered by the [Publisher](#)<sup>1</sup> component).

Daisy Books offers a lot of common features required for books, including:

- Table Of Content generation
- Lists of figures and tables
- cross-references (to chapters, sections, figures or tables, in different styles)
- numbering of sections, figures and tables
- index
- footnotes

Books are published 'off-line' (in batch). This means a user will trigger the process to aggregate and format the Daisy documents into an actual book. The resulting book can then be consulted by other users (the book readers).

## Terminology

Let us first agree on some terminology. Here we only give some very general definitions, which are described in more detail later on.

**Book definition.** A book definition is a Daisy document that defines the structure and content of the book (which documents to include), it defines general book metadata, and it defines default parameters for the publication of the book.

**Publication process.** A publication process generates an actual book (a book instance) based on a book definition, according to one ore more publication types.

**Book instance.** A book instance, or simply book, is the result of running a book publication process. It is a collection of files (organised in directories) among which the published PDF file and/or HTML files, but also a snapshot of the data which is included in the book, intermediary book production files, a book publication log and a link errors log. Book instances are created and managed by the book store.

**Book store.** The book store manages book instances. In other words, the book store is where published books are stored.

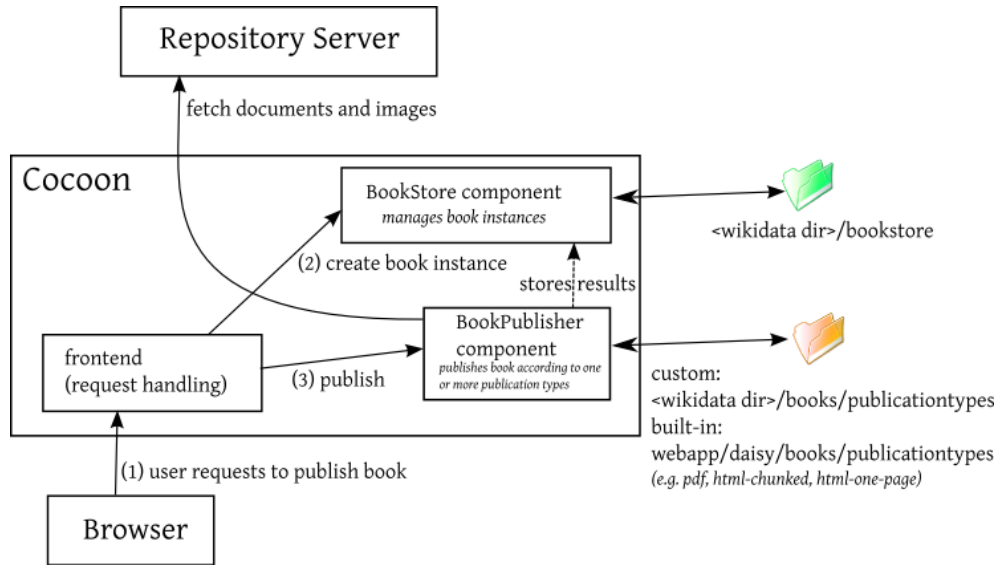
**Publication type.** A publication type defines one way of how a book can be published (for example "PDF" or "Chunked HTML"). Publication types can be customised to a certain extent through properties (for example, to define the depth of the generated TOC), though for most layout changes you will usually create your own publication types (starting from the default ones).

---

1. daisy:194-cd (Publisher)

**Publication output.** A publication output, or simply publication, is the result of publishing a book according to a publication type (thus a set of files).

## The book publication process



The process of publishing a book consists of two main parts:

1. *Book data retrieval:* retrieves, from the repository server, all the data (documents and images) needed for inclusion in the book. This data is stored as part of the book instance.
2. *Publication type specific publishing:* performs the publishing according to a publication type (this step is performed multiple times if multiple publication types were selected)

These two parts are now described in some more detail.

### Book data retrieval

The needed book data is retrieved and stored in the book instance. There are multiple reasons for doing this:

- all the publications (PDF, HTML, ...) will be performed on exactly the same snapshot of the book data. Otherwise, it might be that a document gets updated between the publishing of e.g. the PDF and the HTML.
- it allows to re-publish the book in the future based on the same snapshot
- it is more efficient to retrieve the data only once

The documents are retrieved using the [Publisher<sup>2</sup>](#) component in the form of 'prepared documents'.

While retrieving the data, a list is compiled of all retrieved documents with their exact version retrieved. This information can be used later on to check what documents were updated since the publishing of a book.

### Publication type specific publishing

This process is defined by the publication type, but usually follows these steps:

- apply document type specific styling on each of the documents

---

2. [daisy:194-cd \(Publisher\)](#)

- shift the headers in the documents based on their hierarchical position in the book definition
- assembling the individual documents into one big XML document
- add numbers to sections, figures and tables
- add the index (only for PDFs)
- checking for double IDs, broken links, and adding links to all elements that require one (i.e. all headers, and images/tables that have a caption)
- add the table of contents and lists of figures and tables
- publication type specific steps:
  - applying a Cocoon pipeline to generate customised HTML or XSL-FO
  - chunking the HTML into multiple parts
  - ....