

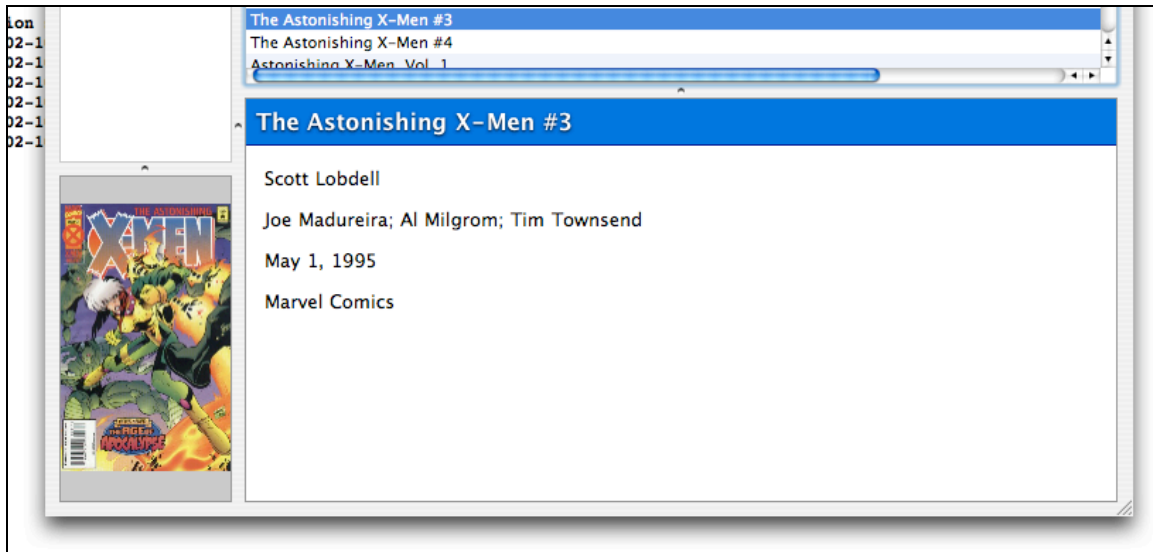
Writing a Books 3.0 Display Plug-In

Version 1.0
February 10, 2006

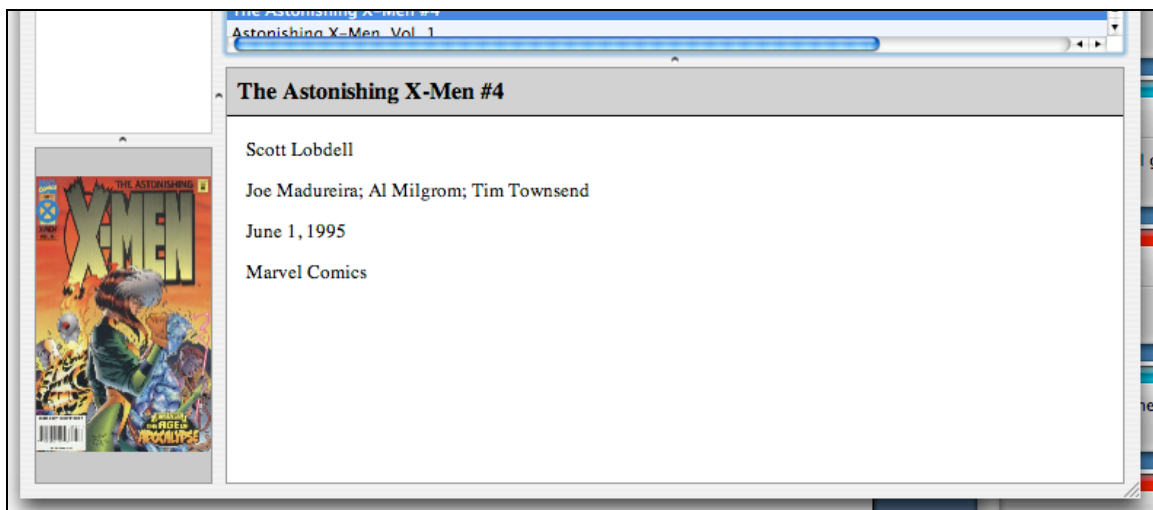
Introduction

In version 3.0, Books introduced a new feature called display plug-ins. Display plug-ins allow users to customize the way that the book record details are presented in the main window. (This is useful when the application developer is not all that aesthetically capable.)

This is an example of the Default Display plug-in that ships with Books:



This is an example of the Classic Display plug-in that ships with Books:



Both sets of plug-ins provide formatted display of data that can be represented in a textual format.

(The original inspiration for this idea came from the use of display themes in NetNewsWire.)

Anatomy of a display plug-in

Unlike other Books plug-ins, display plug-ins do not require any programming experience. Instead they rely upon the designer's skill with cascading stylesheets in a WebKit environment.

When Books is configured to use a plug-in's default set of fields when generating a page (as opposed to the user specifying their own field choices), Books parses a text file within the plug-in to determine which fields to use. These values are stored in a simple comma-delimited format. Such a file might contain contents like this:

```
title,authors,illustrators,genre,format,publisher,publishDate,listName
```

This file would generate a details window that displayed the fields above when the fields were available for a given record available.

Likewise, Books uses CSS in a relatively straightforward manner. When a user selects a book (or set of books) to be displayed in the main window, Books creates an HTML page in memory and substitutes the default CSS link to the CSS link provided by the plug-in before rendering it using WebKit. A CSS example:

```
body
{
    font-family: 'Lucida Grande', LucidaGrande, Lucida, sans-serif;
    font-size: 13 point;

    background-color: white;
    padding: 0;
    margin: 0;
}

div
{
    display: block;
    padding-bottom: 0em;
    margin-bottom: 1em;
    margin-left: 1em;
    margin-right: 1em;
}
```

```
div.book-definition {
    padding: 0;
    margin: 0;
}

div.empty-message
{
    font-style: italic;
    padding: 1em;
}

div.multiple-message
{
    font-style: italic;
    padding: 1em;
}

p.booklist-item
{
    padding-left: 20px;
}

div.booklist-title
{
    font-style: italic;
    font-weight: bold;
    margin: 0;
    background: #;
    border-bottom: #ffffff;
}

div.booklist-authors
{
    padding-left: 30px;
}

div.book-title
{
    font-size: larger;
    background: #005FD7;
    padding: 0.5em;
    margin: 0 0 1em;
    border-bottom: 1px solid #000080;
    color: #ffffff;
    font-weight: bold;
    text-shadow: 0 1px 2px black;
}
```

```
div.book-summary
{
    margin-top: 1em;
    text-align: justify;
}
```

Both the CSS and text file are distributed in a bundle. In addition to these files, the bundle contains a file called `Info.plist` that provides the main application with details such as the name of the plug-in and the its type. For instance:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>BooksPlug-inName</key>
    <string>Default Display</string>
    <key>BooksPlug-inType</key>
    <string>Display</string>
    <key>CFBundleDevelopmentRegion</key>
    <string>English</string>
    <key>CFBundleIdentifier</key>
    <string>net.aetherial.books.DefaultDisplay</string>
    <key>CFBundleInfoDictionaryVersion</key>
    <string>6.0</string>
    <key>CFBundlePackageType</key>
    <string>APPL</string>
    <key>CFBundleSignature</key>
    <string>????</string>
    <key>CFBundleVersion</key>
    <string>1.0</string>
</dict>
</plist>
```

If you're comfortable editing these text files, you are perfectly capable of writing your own display plug-ins.

Building your own plug-in

The first recommended step in building your own plug-in is to copy a plug-in that already works and modify that. You can find examples of these online at

<http://www.aetherial.net/books-plugins/>

Or you may look within the main Books application bundle. Copy the plug-in to your desktop and then modify the `Info.plist` file by exploring the contents of the plug-in using the Finder. You should change only the name of the plug-in.

Once this is finished, copy the modified plug-in to a folder in

```
~/Library/Application Support/Books/Plugins
```

and launch Books. When you go to the preferences window, you should be able to select your modified plug-in.

Developing and testing your plug-in is also simple. Copy the three HTML example files to your desktop and replace the `-csslink-` string with a path pointing to your own CSS file. There are three HTML files you need test:

`empty.html` – Displayed when no book records are selected.

`multiple.html` – Displayed when more than one book record is selected.

`book.html` – Displayed when only one book record is selected.

If you look at the `multiple.html` and `book.html` file, you'll notice that there is not actually any book information present. In `multiple.html`, the application replaces `-booklist-` with text like

```
<p class="booklist-item">
  <div class="booklist-title">Book Title</div>
  <div class="booklist-authors">Joe Author</div>
</p>
<p class="booklist-item">
  <div class="booklist-title">Book Title Two</div>
  <div class="booklist-authors">Jim Author</div>
</p>
<p class="booklist-item">
  <div class="booklist-title">Book Title Six</div>
  <div class="booklist-authors">Joe Writer</div>
</p>
```

These elements need to be styled in your CSS file.

The `book.html` file includes more information in this form:

```
<body>
<div class="book-definition">
  <div class="book-title">Book Title</div>
  <div class="book-authors">Joe Author</div>
  <div class="book-summary">
```

```
        This<br />
        Is<br />
        A Multi-Line<br />
        Summary.
    </div>
    <div class="book-otherField">Other field</div>

    ...

</body>
```

So, if you want a particular field to appear differently than the other <div>'s, you'll need to add a definition in the CSS file.

To test your CSS, load the HTML in Safari and see how it looks. Try it with a variety of data and a variety of collections of fields. The <div>'s in the HTML will be generated in the order in which they appear in the comma-delimited text file or in the order that the user selects. If you wish to always have an element (like the title) positioned in a particular area, consider using `position: absolute` in your CSS file.

Once you're satisfied, save the files in the plug-in bundle and drop it in the plug-ins folder at

```
~/Library/Application Support/Books/Plugins
```

Then see how it looks in the application.

Distributing your plug-in

When you're confident that your plug-in is well-tested and looks good, you may wish to distribute it to other Books users. There's a couple of ways to get the word out:

1. Send an e-mail to books@aetherial.net. This is the best way to get it listed on the Books Plug-In weblog at

```
http://www.aetherial.net/books-plugins/
```

2. Send a note to the Books Users e-mail list. Details about the list are available at

```
http://groups.yahoo.com/group/books\_users/
```

If the plug-in would be useful to a large enough number of Books users, it may be included in the main distribution.

Frequently asked questions

Q: Where are the frequently asked questions?

A: This is a 1.0 document. No one has had the opportunity to ask any questions, much less ask any questions frequently.

Feedback

If you have any questions, comments, or suggestions, please send them to

`books@aetherial.net`

Document changes

February 10, 2006 (1.0): Initial version