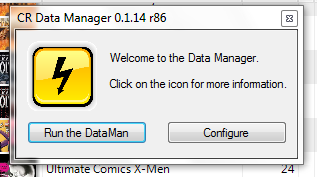
docdoom  
Data Manager for ComicRack

The Manual v3

covering CR Data Manager up to Version 0.2





Contents

[Purpose 3](#_Toc353409876)

[Installation 3](#_Toc353409877)

[Usage 4](#_Toc353409878)

[Setting up the configuration 4](#_Toc353409879)

[Running the Data Manager 5](#_Toc353409880)

[Syntax errors 7](#_Toc353409881)

[Why you should run the ComicVine Scraper before the Data Manager 9](#_Toc353409882)

[Which data can be edited by the Data Manager? 9](#_Toc353409883)

[Fields to use as filter criteria 9](#_Toc353409884)

[Fields that can be modified by the Data Manager 10](#_Toc353409885)

[Advanced features 11](#_Toc353409886)

[Combining Criteria 11](#_Toc353409887)

[Using Modifiers 11](#_Toc353409888)

[List of range modifiers in Criteria (left part of the rule) 12](#_Toc353409889)

[The Number field problem 13](#_Toc353409890)

[List of value modifiers in Value part (right part of the rule) 14](#_Toc353409891)

[Calculated values 15](#_Toc353409892)

[Working with empty values 15](#_Toc353409893)

[Parser directives 16](#_Toc353409894)

[#@ GROUP 16](#_Toc353409895)

[#@ END\_RULES 16](#_Toc353409896)

[Links 17](#_Toc353409897)

[License information 18](#_Toc353409898)

# Purpose

If you are using ComicRack to organize your comic library there is sometimes the need to enter data based on rules. So you might want to set the series group to “Gotham” for all *Batman* books. Or you scraped your data from Comic Vine (using the excellent ComicVine Scraper) and discover that some of the information in the ComicVine database is not correct. So you want to set the publisher information for the first 200 issues of *Walt Disney’s Comics and Stories* to Dell instead of Boom. This plugin will complete all these tasks automatically.

# Installation

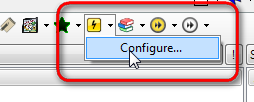
Download the latest version of the Data Manager from [here](http://code.google.com/p/cr-data-manager/downloads/list). In ComicRack then select “Edit -> Preferences -> Scripts”. Click on “Install” and select the file you just downloaded. You have to restart ComicRack afterwards.

**Important:** if you install the Data Manager plug-in over a previous version please **don’t remove the older version** (selecting “Edit -> Preferences -> Scripts -> RC Data Manager -> Remove”). Otherwise you will lose your previous self-defined rules for the Data Manager. Just install the newer version over the old one and you will be fine.

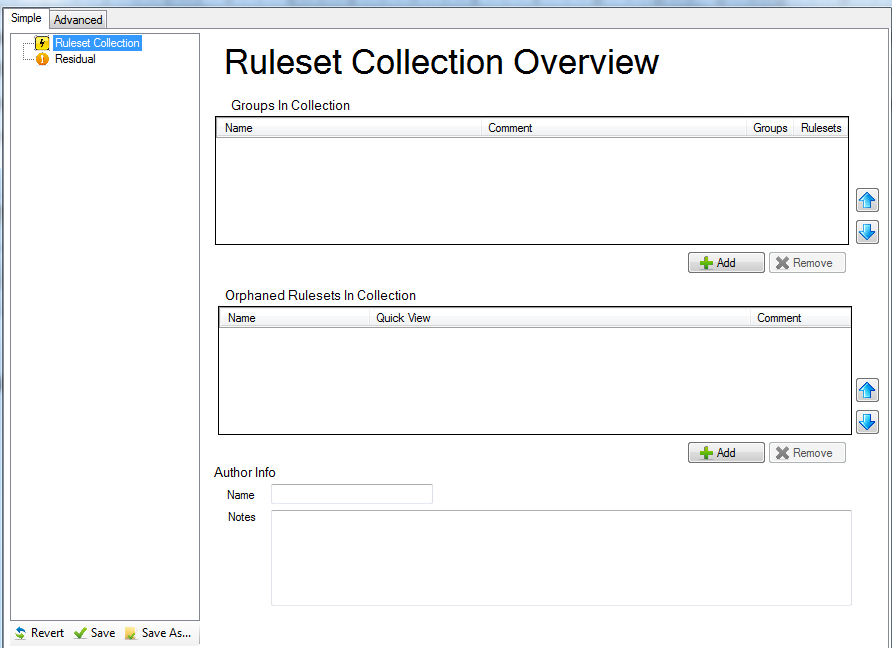
# Usage

## Setting up the configuration

Before you run the Data Manager for the first time you have to define your rules. Start the Data Manager from the context menu in the library browser or from the toolbar in ComicRack (the icon of the Data Manager is the yellow square one with the black flash symbol inside). Click on the small dropdown arrow right to the yellow icon and then click on “Configure”.



When you start the rule editor for the first time you will not see any rules because you have not yet defined any:



In the next steps you will learn how to create some simple but powerful rules.

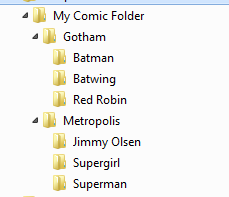
## How-to: Create a few simple (but powerful) rules

### Step 1: What shall your rules do?

At first step you should take a bit of time to think about what your rules should do. Some people want to correct data they scraped from the ComicVine database into their books, others might want to automate the process of preparing their books before they are moved by the Library Organizer.

Let’s start with the last example: A lot of people organize their books by Series Groups. Series Groups group several series with a similar content or main character or whatever. So you might want to use a Series Group “Gotham” for all books that are more or less located in Gotham and a Series Group “Metropolis” for all the Superman books.

This would make it easy to organize your books in a folder structure like this:

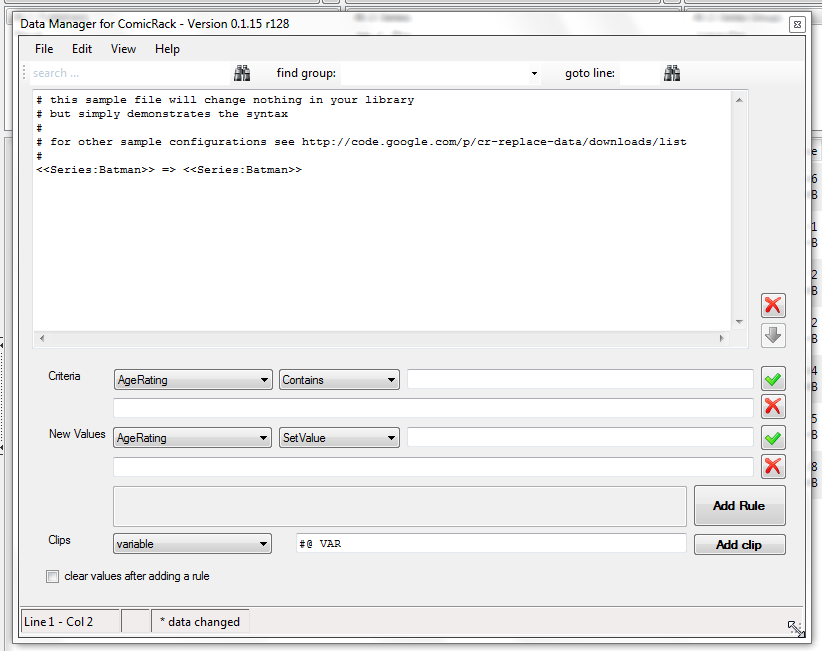


This means that we have to tell the Data Manager that it should shall set Now we know what our rules shall do and we are ready to write our first rules.

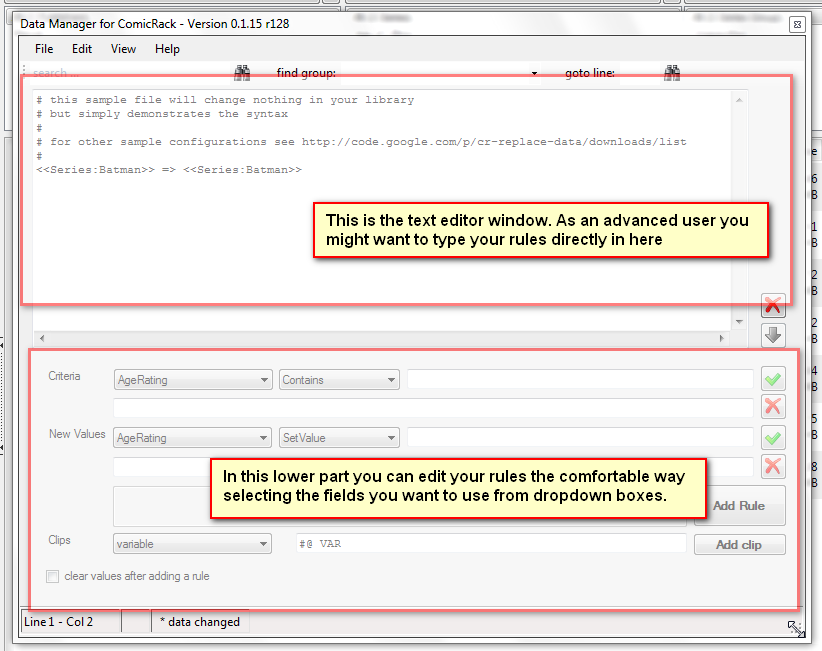
### Step 2: Define your rules

### Step 3: Organize your rules

The configuration editor opens where you can enter the rules which the Data Manager uses later to run bulk changes on selected books in your library (or even the whole library):



The rule editor consists of two parts:



If you are an advanced user of the Data Manager and you already know how to write the rules that will modify your library later (and maybe you don’t like this Windows stuff at all and love the power of Notepad and text only consoles): the upper half of the role editor is your playground! You might type rules like  
<<Series:Superman>> => <<SeriesGroup:Metropolis>>  
directly here. (BTW: this rule means: if a series is named “Superman” the Data Manager should set the value of the field SeriesGroup to “Metropolis”).

But if you like the convenience of a graphical interface and know how to move a mouse across the screen you should consider using the lower part of the rule editor. The next few pages in this manual will demonstrate how you can write very powerful rules to edit your library’s metadata with a few clicks (and a bit of using the keyboard).

## Using the graphical interface of the Data Manager

### Step 1: Create a few simple (but powerful) rules

Dfjk dfdkfj dfkdk dfj dklfj dklfjd fkldj fdklsfj dklsfj slfjdsklfj dsfklj sklfjklfj dsöfl jdfj dlfkjdkflj dfklj dklfj dsklfjdklsfjdslfj dslfj dslf

Some lines in this sample start with a #. Those are comments to make the file easier to read. The other lines (without a # at the beginning) define the rules for what books which data shall be updated. These lines follow the syntax

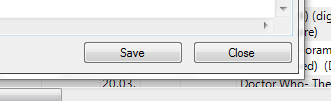
*your rule (which books to modify)*, *followed by a “=>”, followed by your actions (the new data that the script should set in your library)*

So if you want to set the Series Group of all books of any series called ‘Superman’ to ‘Metropolis’ you would simply enter the following line in the configurator window:

<<Series:Superman>> => <<SeriesGroup:Metropolis>>

**Note:** The Data Manager is not limited to such simple rules. The chapter “Advanced Features” later in this manual will show you how to write much more sophisticated rules to be more specific about the books you want to alter.

Once you finished writing your rules click on the “Save” button of the Data Manager Configurator. If you don’t want to save your edits for further use click on “Close”:



**Please do not use a line break inside a rule**. This will not work:

<<Series:X-Men>> =>­  
<<SeriesGroup:X-Men>>

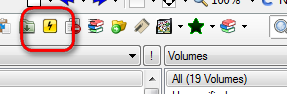
But this is correct and will work:

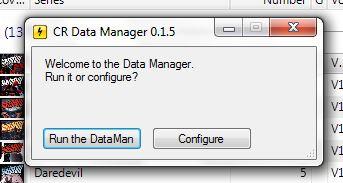
<<Series:X-Men>> => <<SeriesGroup:X-Men>>

## Running the Data Manager

After you edited the configuration you select the books that the script should handle in the ComicRack browser, right click and select Automation -> Data Manager. That’s all.

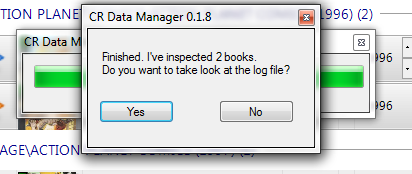
You can also use the icon in the ComicRack toolbar (it is the square yellow symbol with a black flash):



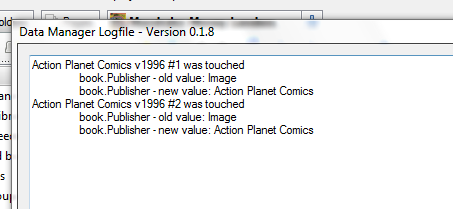


Click on „Run the DataMan” and the Data Manager will edit the selected books in your library according your rules configuration.

Once the Data Manager has finished its work it displays a dialog telling how many books were inspected:



You may click on yes to take a further look what data was altered by the Data Manager. A sample log file will look like this:

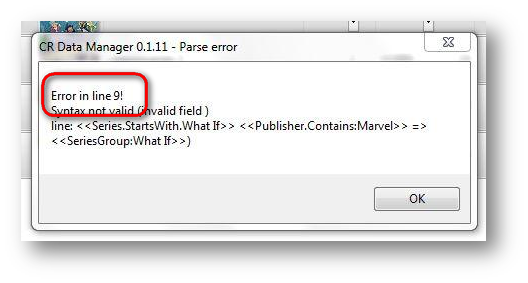


The script will only modify the books you have selected in the ComicRack browser. You don’t have to be afraid that it will affect your whole library.

Also: you can always roll back the modifications to your library done by this script by selecting the “Edit -> Undo” command in ComicRack after you executed this script.

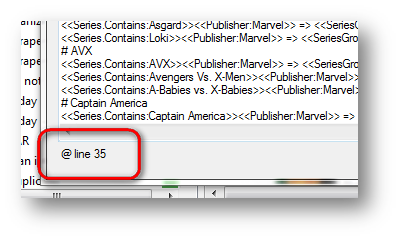
## Syntax errors

Before the Data Manager actually touches your books it will check your rules for any syntax error. When you start the Data Manager to run and receive an error message like this:



If this happens you should run the configurator again and check the line that was mentioned in the dialog above.

**Note:** To make it easier to find that line in the configuration where the syntax error was found the configurator dialog displays the number of the line where your cursor is currently located at bottom left:



## Why you should run the ComicVine Scraper before the Data Manager

You might get unexpected results when you run the Data Manager on books that have not been scraped by the ComicVine scraper or have uncommitted values (so-called “Shadow values”). So it is highly recommended to scrape the books first and afterwards additionally select “Automation -> Commit proposed values”.

Also: you should run the Data Manager on your books **before** you run the Library Organizer. A good rule is to run the standard ComicRack plug-ins in this order:

1. run ComicVineScraper
2. commit proposed values
3. run Duplicates Manager / Scan Info from Filename / More Scan Info from Filename
4. **run Data Manager**
5. run Library Organizer

# Which data can be edited by the Data Manager?

## Fields to use as filter criteria

At the moment you can use these ComicRack fields for your rules:

AlternateNumber

AlternateCount

Series

Volume

Format

Imprint

Publisher

Number

FileDirectory

SeriesGroup

AlternateSeries

Month

Year

MainCharacterOrTeam

FileName

FilePath

Count

Genre

Tags

PageCount

## Fields that can be modified by the Data Manager

At the moment the Data Manager plug-in can be used to set new data in these fields:

Series

Volume

Imprint

Publisher

Format

Number

SeriesGroup

MainCharacterOrTeam

AlternateSeries

Count

Genre

Tags

# Advanced features

## Combining Criteria

The most simple way to define a rule is a rule like this:

<<Series:Batman>> => <<SeriesGroup:Gotham>>

But sometimes you want to be more specific. So maybe you only want to set the series group of the Star Wars series to ‘Marvel Cosmic’ if the publisher is Marvel. In this case you have to enter both criteria (Series is “Star Wars” and Publisher is “Marvel”) in front of the “=>”. The complete rule would be:

<<Series:Star Wars>> <<Publisher:Marvel>> => <<SeriesGroup:Marvel Cosmic>>

**Note:** The criteria on the left side of the rule are always treated like combined with a logical AND operator. That means all criteria must be fulfilled for a rule to be executed.

## Using Modifiers

In the previous example above we have set the series group of all issues of any series named ‘Batman’ to ‘Gotham’. This would edit data in all books where the series is exactly ‘Batman’ but would ignore all series like ‘Batman: The Dark Knight’. You can use modifiers to select books in a more specific way. So if you want to select all books where the word ‘Batman’ is somewhere in the title you would use the modifier ‘Contains’. The phrase

<<Series.Contains:Batman>>

would select not only ‘Batman’, but also ‘Batman and Robin’ and ‘Adventures of Batman’.

The syntax to use a modifier is:

<<Field.Modifier:Criteria>>

### List of range modifiers in Criteria (left part of the rule)

These are the modifier that can be used to limit the books which the Data Manager should touch:

|  |  |  |
| --- | --- | --- |
| **Modifier** | **Purpose** | **Example** |
| Is | select all books where the content of [field] is exactly like [value]  Note: this modifier must not be set: <<Series.Is:X-Men>> delivers the same result as <<Series:X-Men>> | <<Series.Is:X-Men>> selects all books which series is exactly “X-Men”. |
| Range | select all books in a given range | <<Number:Range:1,10>>  selects all books which number is is in the range from 1 to 10 (including). This will only work for the field Number and numerical fields. |
| Not | select all books that would be excluded | <<Publisher.Not:Marvel>> selects all books where the publisher is anything else but not ‘Marvel’ |
| Contains | select all books where the phrase is found somewhere in the field | <<Series.Contains:Batman>> Selects all books where the series contains the word ‘Batman’  <<Series.Contains:Batman Robin>> Selects all books where the exact phrase “Batman Robin” is somewhere in the series, but not where “Batman **and** Robin” is in the series field |
| ContainsNot  (or  NotContains) | Select all book where the phrase is **not** found somewhere in the field **(can only be used with non-numerical fields)** | <<Series.ContainsNot:Batman >>  Selects all books where “Batman” is not found somewhere in the series. |
| ContainsAnyOf | Select all book where any of the values is found somewhere in the field **(can only be used with non-numerical fields, the list of values must be separated by commas)** | <<Series.ContainsAnyOf:Batman,Robin,Joker>>  Selects all books where Batman or Robin or Joker is found somewhere in the series. |
| NotContainsAnyOf | Select all book where any of the values is not found somewhere in the field **(can only be used with non-numerical fields, the list of values must be separated by commas)** | <<Series.NotContainsAnyOf:Superman,Batman>>  Selects all books where neither “Superman” nor “Batman” are found in the series. |
| ContainsAllOf | Select all book where all of the values are found somewhere in the field **(can only be used with non-numerical fields, the list of values must be separated by commas)** | <<Series.ContainsAllOf:Bat,Super>>  Selects all books where “Bat” **and** “Super” are found somewhere in the series. |
| Greater | select all books where the value is greater than the phrase | <<Number.Greater:10>> Selects all books where the number is greater than 10 |
| GreaterEq | select all books where the value is equal or greater than the phrase | <<Number.GreaterEq:10>> selects all books where the number is equal to or greater than 10 |
| Less | select all books where the value is less than the phrase | <<Number.Less:10>> Selects all books where the number is less than 10 |
| LessEq | select all books where the value is equal or greater than the phrase | <<Number.LessEq:10>> selects all books where the number is equal to or less than 10 |
| StartsWith | select all books where the value starts with the phrase | <<Series.StartsWith:Bat>> selects all books where the series name starts with “Bat” (like Batman or Batwing) |
| StartsWithAnyOf | Select all books where the value starts with any of the given phrases **(can only be used with non-numerical fields, the list of values must be separated by commas)** | <<Series.StartsWithAnyOf:Super,Bat>>  Selects all books where Series starts with either ”Super” or “Bat”. |

### The Number field problem

As ComicVine (as does ComicRack) allows any characters in the Number field of an issue (like ½, ∞ and everything unthinkable) it cannot be guaranteed that range modifiers like Range, Less, Greater etc. work correctly with the Number field. If you use a rule like <<Number.Range:0,3>> the Data Manager will find all issues with issue number 0, 0.5, 1, 2,3. But it will not find ½, e.g.

### List of value modifiers in Value part (right part of the rule)

These are the modifier that can be used for special purposes:

|  |  |  |
| --- | --- | --- |
| **Modifier** | **Purpose** | **Example** |
| SetValue | sets the new value  **Note:** Format.SetValue:TPB is same as: Format:TPB | <<Format.SetValue:TPB>> Sets the field Format to the new value “TPB” |
| Calc | perform calculations (see example below) | <<Number.Calc:{Number} - 1>> Replaces the current number of the book with current number minus 1. This modifier will only work in numerical fields (including Number) |
| Add | add a new value to a multi value field (currently: Genre, Tags) | <<Tags.Add:Metropolis>> Adds “Metropolis” as a tag |
| Replace | replace a value in a multi value field (currenty: Genre, Tags) | <<Tags.Replace:Metropolis,Superman>> Removes “Metropolis” and adds “Superman” instead |
| Remove | remove a value from a multi value field (currently: Genre, Tags) | <<Tags.Remove:Metropolis>> Removes “Metropolis” from Tags |

## Calculated values

A very powerful feature of the Data Manager plugin is the use of calculated values.

One of the most notorious incorrect information on ComicVine is the “integration” of the 1999 volume of Amazing Spider-Man into the 1963 volume. How can the DataManager help to enter the correct information into your library?

It is quite simple: First you have to select the incorrectly dated books in your library. The criteria part of your script would be:

<<Series:The Amazing Spider-Man>> <<Volume:1963>> <<Number.Range:442,499>>

Now we have to define what needs to be corrected. We want to change the volume to 1999 and correct the issue numbers to 1 (instead of 442), 2 (instead of 443) etc. This means we have to deduct exactly 441 from the original number. So this would be the right part of your rule (following the ‘=>’):

<<Volume:1999>> <<Number.Calc:**{**Number**}** - 441>>.

The complete line in your configuration file would be:

<<Series:The Amazing Spider-Man>> <<Volume:1963>> <<Number.Range:442,499>> => <<Volume:1999>> <<Number.Calc:{Number} - 441>>

**Note:** If you use field names within the calc modifier you have to enclose them in curly brackets.

## Working with empty values

The Data Manager allows the usage of empty values. You might want to use a rule to set the vaue of SeriesGroup to ‘Avengers’ only if the field MainCharacterOrTeam is empty. For this purpose you write this rule:

<<MainCharacterOrTeam:>> => <<SeriesGroup:Avengers>>

Simply omit the value after the double colon and the script will look only for books where the Main Character field is empty.

With the same logic you can remove values from fields. Suppose you want to remove the values from SeriesGroup where SeriesGroup is ‘Avengers’ you would use

<<SeriesGroup:Avengers>> => <<SeriesGroup:>>

## Parser directives

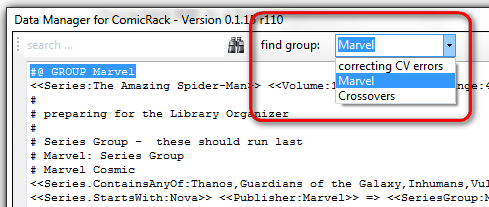
You can control how your rules are run with parser directives. Parser directives start with the characters #@ in a line.

### #@ GROUP

A line starting with:

#@ GROUP

indicates that this is a group header. If you enter a line like “#@ GROUP Marvel” in your rule set you can easily later on find this group in the rule editor with the combo box at the top or the rule editor:



### #@ END\_RULES

If you write this line in your rules:

#@ END\_RULES

then only the rules up to this line will be executed, the remaining lines will be ignored. If you have a rule set like this:

<<Series.Contains:Batman>> => <<SeriesGroup:Gotham>>  
#@ END\_RULES  
<<Series.Contains:Superman>> => <<SeriesGroup:Metropolis>>

then only the first rule <<Series.Contains:Batman>> => <<SeriesGroup:Gotham>> will be executed. The remaining lines will be ignored because of the directive #@ END\_RULES. This can be useful if you want to test a new rule at the beginning of your rule set but don’t want to execute your complete set of rules.

# Links

Discussion about the Data Manager in the ComicRack forum: <http://comicrack.cyolito.com/forum/13-scripts/30963-the-cr-data-manager-enter-data-automatically-based-on-rules-0112>

Downloads: <http://code.google.com/p/cr-data-manager/downloads/list>

Any suggestions for new features and bug reports may be posted here: <http://code.google.com/p/cr-data-manager/issues/list>

The Data Manager has not been created to earn a living. But if you like to honor the effort and help keeping the Data Manager alive and progressing you are kindly invited to donate a few bucks here: <https://www.paypal.com/cgi-bin/webscr?cmd=_s-xclick&hosted_button_id=YYDNBFQKCLAGQ>

# License information

The Data Manager for ComicRack is Copyright 2013 by docdoom.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use the source code and software except in compliance with the License.

You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.