



# docdoom's crDataMan - The Manual

## Contents

[Contents](#)

[Purpose](#)

[Installation](#)

[Usage](#)

[Setting up the configuration](#)

[Running the Data Manager](#)

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

[Fields to use as rules](#)

[Fields that can be modified by the script](#)

[Advanced features](#)

[List of modifiers](#)

[Calculated values](#)

[Handling empty values](#)

## Purpose

If you are using ComicRack to organize your comic library there is sometimes the need to enter data depending 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 Disneys Comics and Stories to Dell instead of Boom. This plugin will complete all these tasks automatically.

## Installation

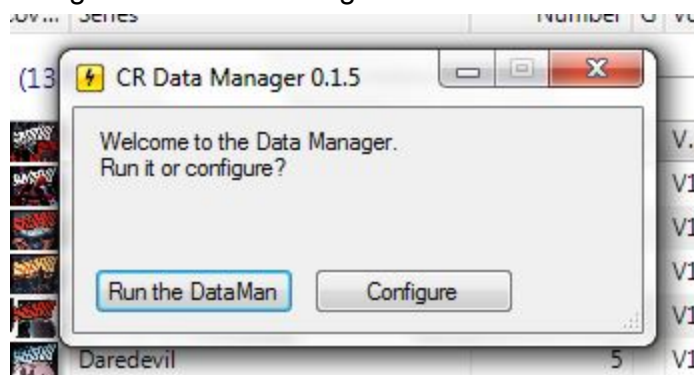
Download the latest version of the Data Manager from [here](#) (select the file with the crplugin extension). In ComicRack then select “Edit -> Preferences -> Scripts”. Click on “Install” and select the file you just downloaded. You have to restart ComicRack afterwards.

**Note:** 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'll be fine.

## Usage

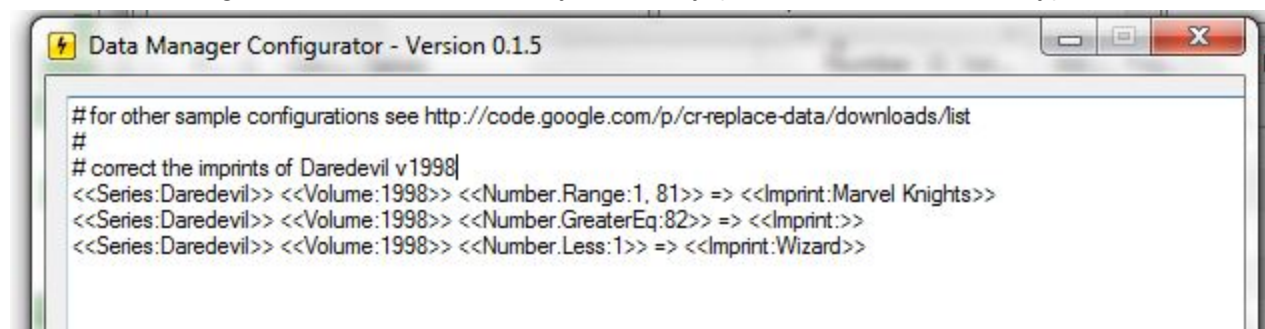
### Setting up the configuration

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. In the dialog box click on “Configure”:



and select “Configure”.

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):



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 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 to your configuration file:

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

If you want to set the MainCharacterOrTeam field of all Marvel books with ‘Avengers’ in series name to ‘Avengers’ you would add this line to your configuration file:

```
<<Series.Contains:Avengers>> <<Publisher.Contains:Marvel>> => <<MainCharacterOrTeam:Avengers>>
```

If you have finished writing your rules click on the “Save and exit” button of the Data Manager Configurator. If you don’t want to save your edits for further use click on “Cancel without saving”:



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

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

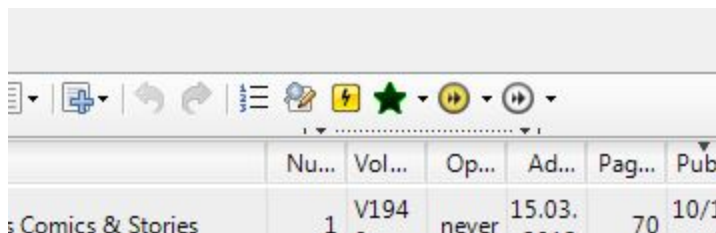
But this is correct and will work:

```
<<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 flash):



**Note:** the Data Manager is most effective if you run it on books that you have already scraped with the CR ComicVine Scraper. A good rule is:

*First:* run ComicVineScraper

*Second:* run Data Manager

*Third:* run Library Organizer

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.

## Which data can be edited by the Data Manager?

### Fields to use as rules

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

Series,  
Volume,  
Format (**new in version 0.1.3**)  
Imprint,  
Publisher,  
Number,  
FileDirectory,  
SeriesGroup  
AlternateSeries (**new in version 0.1.3**)  
Month,  
Year,  
MainCharacterOrTeam  
FileName (**new in version 0.1.3**)  
FilePath (**new in version 0.1.3**)  
Count (**new in version 0.1.3**)  
Genre (**new in version 0.1.7**)

### Fields that can be modified by the script

At the moment the script can be used to set new data in these fields:

Series  
Volume  
Imprint,  
Publisher  
Format (**new with version 0.1.3**)  
Number  
SeriesGroup  
MainCharacterOrTeam  
AlternateSeries (**new in version 0.1.3**)  
Count (**new in version 0.1.3**)  
Genre (**new in version 0.1.7**)

## Advanced features

The most simple way to define a rule is this

<<Series:Batman>> => <<SeriesGroup: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 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 modifiers

These are the modifiers that can be used:

Modifier	Purpose	Example
Range	select all books in a given range	<<Number.Range:1,10>> selects all books which number 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	<<Series.Contains:Batman>> Selects all books where the series contains the word 'Batman'
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	<<Number.LessEq:10>>

	value is equal or greater than the phrase	selects all books where the number is equal to or less than 10
<b>StartsWith (new in version 0.1.3)</b>	select all books where the value starts with the phrase	<code>&lt;&lt;Series.StartsWith:Bat&gt;&gt;</code> selects all books where the series name starts with “Bat” (like Batman or Batwing)

## 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>>.
```

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

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>>
```

## Handling empty values

The Data Manager allows the usage of empty values. You might want to use a rule to set the value 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:>>
```