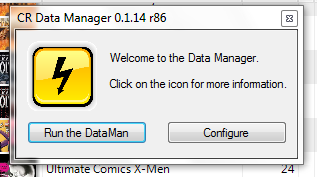
**docdoom & T3KN0Gh057**  
Data Manager for ComicRack

The Manual v3

covering CR Data Manager up to Version 1.0





Contents

[Purpose 3](#_Toc354440019)

[Installation 3](#_Toc354440020)

[Usage 4](#_Toc354440021)

[Setting up the configuration 4](#_Toc354440022)

[How-to: Create a few simple (but powerful) rules 5](#_Toc354440023)

[Step 1: What shall your rules do? 5](#_Toc354440024)

[Step 2: Organize your rules 6](#_Toc354440025)

[Step 3: Build your rules 8](#_Toc354440026)

[Running the Data Manager 11](#_Toc354440027)

[Why you should run the ComicVine Scraper before the Data Manager 12](#_Toc354440028)

[Which data can be edited by the Data Manager? 12](#_Toc354440029)

[Fields to use as filter criteria 12](#_Toc354440030)

[Fields that can be modified by the Data Manager 13](#_Toc354440031)

[Advanced features 14](#_Toc354440032)

[Combining Criteria 14](#_Toc354440033)

[Using Modifiers 14](#_Toc354440034)

[List of range modifiers in Criteria 14](#_Toc354440035)

[The Number field problem 17](#_Toc354440036)

[List of value modifiers in Value part (right part of the rule) 17](#_Toc354440037)

[Calculated values 18](#_Toc354440038)

[Working with empty values 18](#_Toc354440039)

[#todo: Parser directives 19](#_Toc354440040)

[Links 20](#_Toc354440041)

[License information 21](#_Toc354440042)

# 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

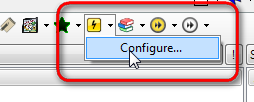
Download the latest version of the Data Manager from [here](http://code.google.com/p/cr-data-manager/downloads/list). In ComicRack then select “Edit -> Preferences -> Scripts”. Click on “Install” and select the file you just downloaded. You have to restart ComicRack afterwards.

**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.

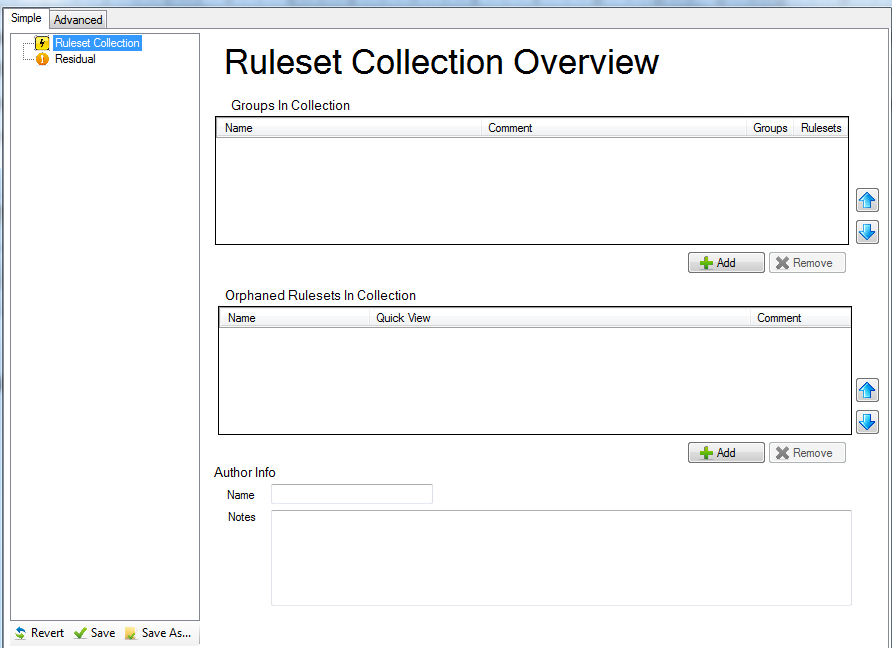
# Usage

## 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 very 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.

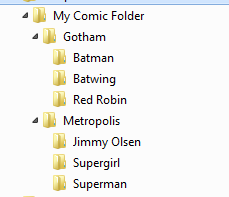
## How-to: Create a few simple (but powerful) rules

### 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.

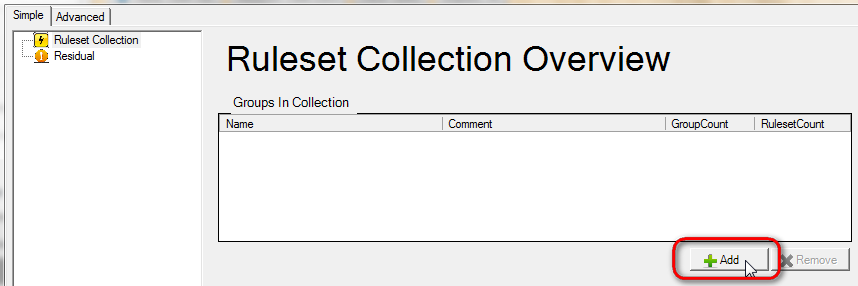
This would make it easy to organize your books in folder structures like these:



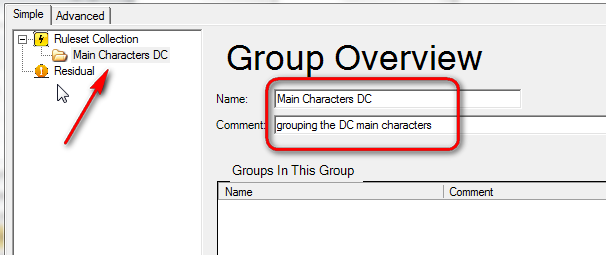
In the first example this means that we have to tell the Data Manager that it shall set the Series Group to a value that depends on the name of the series. In the second example we would like to organize our books by the main character. The next step demonstrates how to prepare your books so that they can easily later on organized by the Library Organizer in this main character structure.

### Step 2: Organize your rules

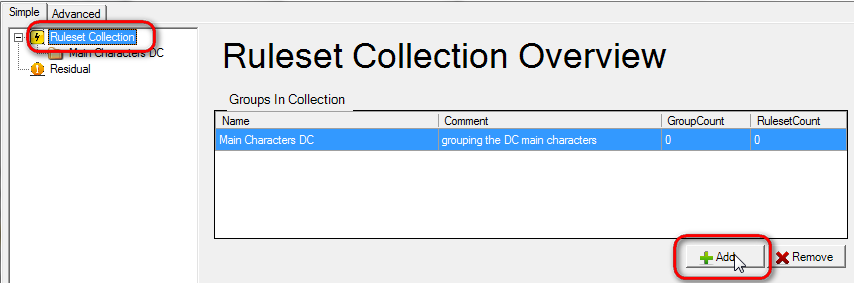
Now that we have planned what we want to do we are ready to create our first set of rules. Open the rule editor as described above. As you may want to create several Series Group rules it is a good practice to create a group for all those main character rules. This is because this makes it much easier to navigate through your set of rules, especially when you have a very large set of rules.

To create a group for the Main Characters Group rules click on “Add” below the “Groups in Collection” window: 

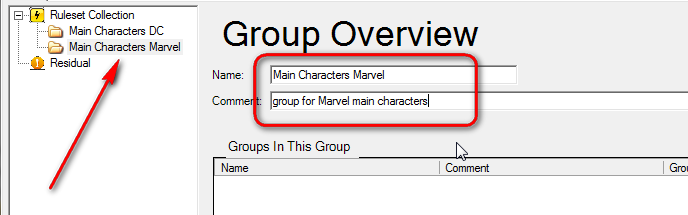
In the next panel add a name and optional comment for the group:



Now take a look at the treeview on the left and see that it already has your new group in it!

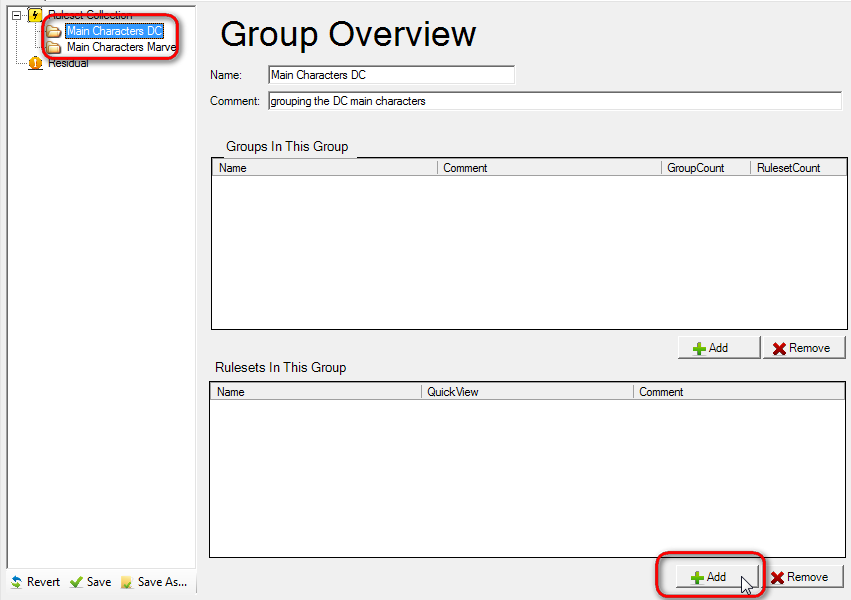
As you might also have some Marvel books in your collection you might also want to create a group for Marvel main characters. Quite easy: select the node “Ruleset Collection” in the treeview and click on “Add” again:

Again enter a group name and optionally a description and you have the structure where you can build your rules:

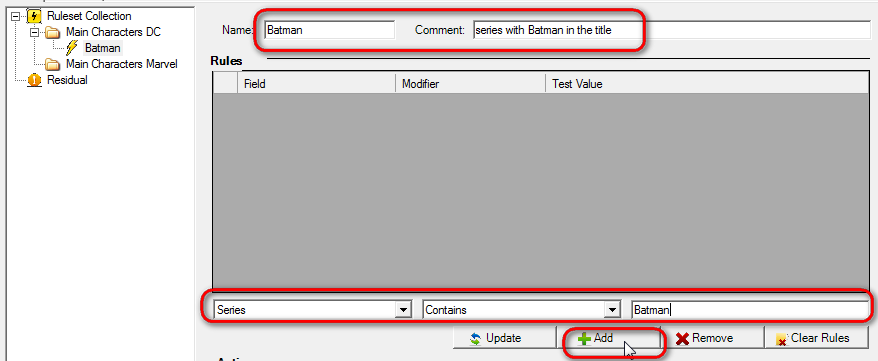


### Step 3: Build your rules

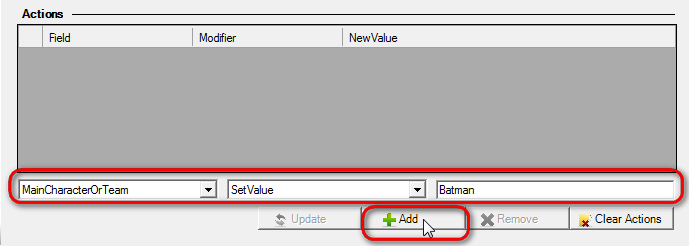
Now we want to build our rules. First we want to define that all books that have “Batman” in the series title should get “Batman” as the main character. For that select the node “Main Characters DC” and click on “Add” below the panel “Rulesets in this group”:



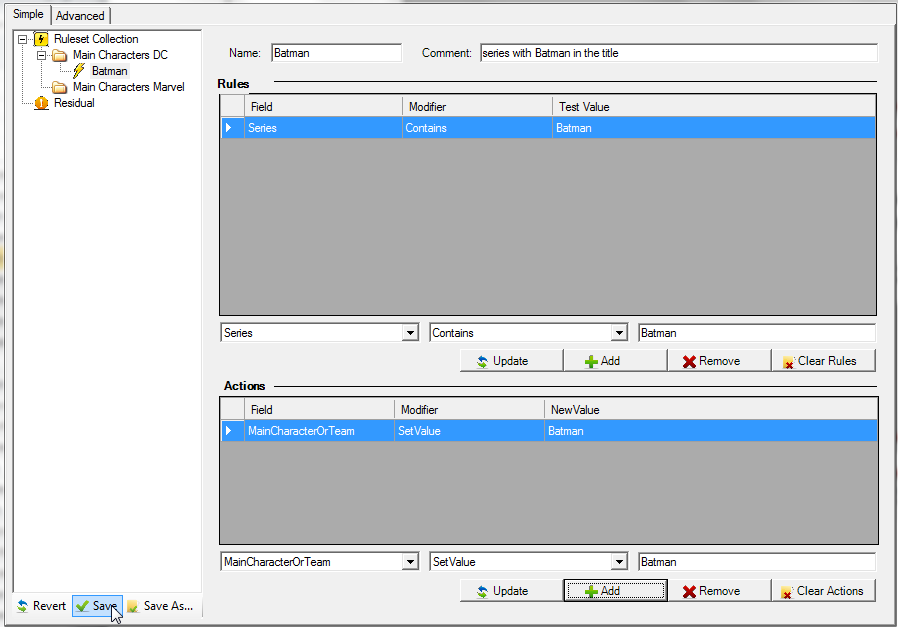
In the upper panel enter a name and optional comment for your rule and define your rule below the panel. The rule definition here means: all books with a series that contains the word “Batman” somewhere in the title. Finally click on “Add”:



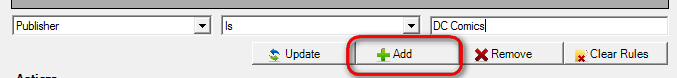
Now we define what shall happen with these books. Below the “Actions” panel you define what value will be set (in this case “MainCharacterOrTeam” shall be set to “Batman”). Click on “Add” afterwards:



If your rule editor looks now like this you are finished with your first rule:



You can also add more criteria to one rule: maybe you want to be sure that **only** the Batman books should be handled by this rule that are published by DC Comics, simply select the Batman rule in the treeview and add this rule:



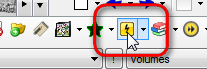
That’s all. Even with this very small rule set you could later on set the MainCharacter field for hundreds of books to “Batman” (where it fits) with one mouse click. Of course you can add hundreds of groups and thousands of rules to help you make organizing your library a lot easier.

This was a very simple example but should give you an idea how to build rules with the Data Manager.

Later on in this manual all fields and modifiers are listed that are currently allowed.

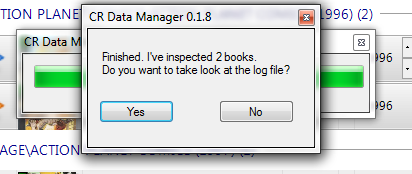
## Running the Data Manager

After you edited the configuration you select the books that the script should handle in the ComicRack browser just click icon in the ComicRack toolbar (it is the square yellow symbol with a black flash):

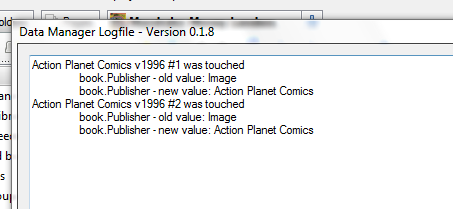


Once you clicked on it the Data Manager will edit the selected books in your library according your rules configuration on the fly.

When 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 Data Manager 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.

## 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:

# todo: list all allowed fields

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:

# todo: list all allowed fields

Series

Volume

Imprint

Publisher

Format

Number

SeriesGroup

MainCharacterOrTeam

AlternateSeries

Count

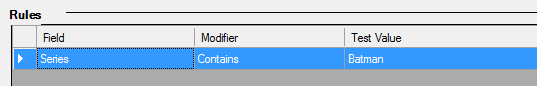
Genre

Tags

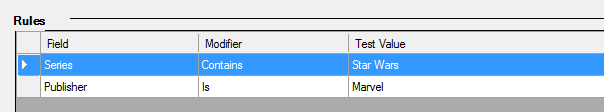
# Advanced features

## Combining Criteria

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



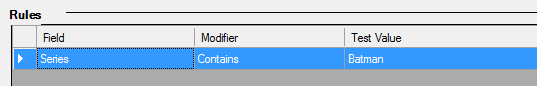
But sometimes you want to be more specific. So maybe you only want to set the series group of the Star Wars series to ‘Marvel Cosmic’ if the publisher is Marvel. In this case you have to enter both criteria (Series is “Star Wars” and Publisher is “Marvel”) in front of the “=>”. The complete rule would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



**Note:** These 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 rule

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

There are a lot more modifiers available to be more specific.

## List of range modifiers in Criteria

# todo: list all available modifiers

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

# todo: screenshot (Number.Range:1,11)

the Data Manager will find all issues with issue number 0, 0.5, 1, 2,3. But it will not find the roman numeric literal XI .

## 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

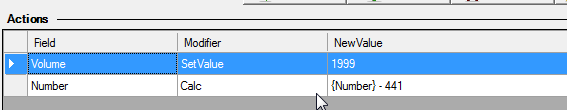
A very powerful feature of the Data Manager plugin is the use of calculated values.

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:

# todo: screen shot (<<Series:The Amazing Spider-Man>> <<Volume:1963>> <<Number.Range:442,499>>)

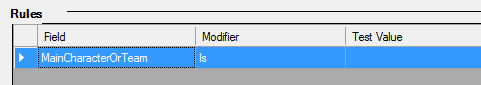
Now we have to define what needs to be corrected. We want to change the volume to 1999 and correct the issue numbers to 1 (instead of 442), 2 (instead of 443) etc. This means we have to deduct exactly 441 from the original number. So this would be the “Actions” part of your rule:

:

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

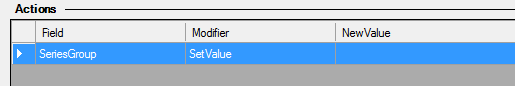
## Working with empty values

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



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



## #todo: Parser directives

END\_RULES: Especially for testing.

# 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.