R & E Foster & Associates Pty Ltd
ABN: 42 129 335 020
info@refactored.com.au
Phone: +61-3-5282-4246

Refactored Content Views Plugin 0.3

Installation and Users Guide

Contents

Introduction

The **Refactored Content Views** package is essentially a content views (number of times viewed) counter. The current functionality offered by this package includes:

- Optional Data Type that allows for configuring view counters with various categories and the ability to instruct Macros etc. to “hide” the view Count yet still increment it.
- Optional incrementing when displaying the view count (useful when you want to display the view count in a content listing, for example)
- Example Razor Script and Macro.
- Library methods to manipulate the counters and retrieve details as an XML fragment for use with XSLT.

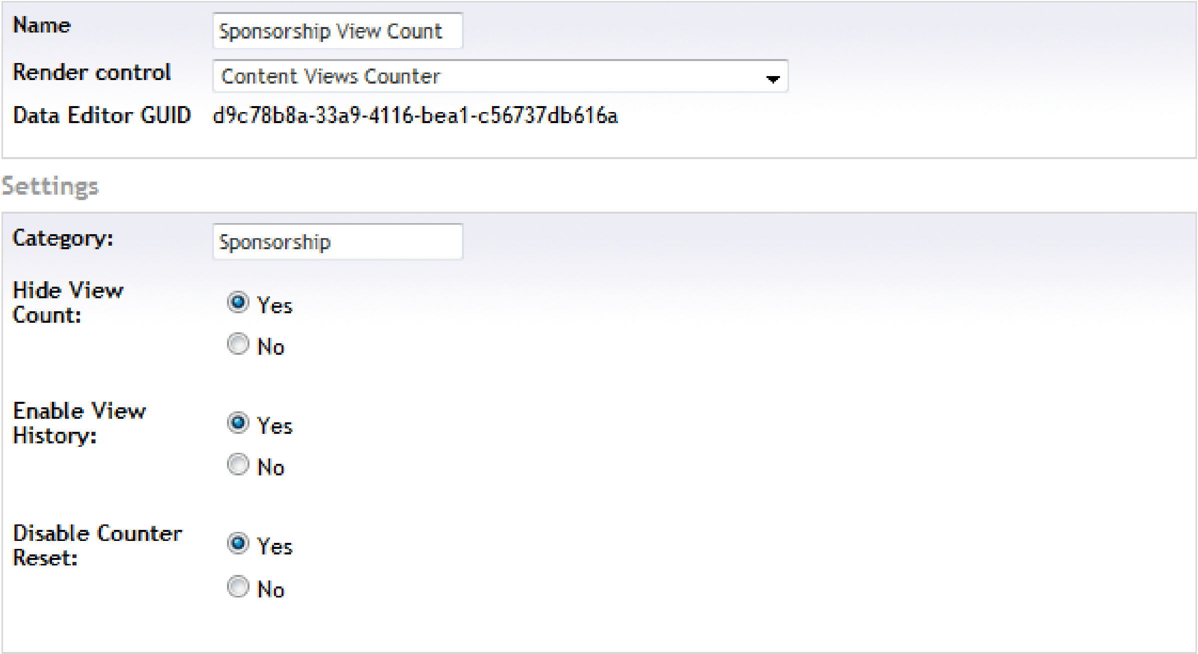
Basic Usage.

To simply retrieve and/or increment the counter for a specific content item, call the following library method. The category and increment parameters are optional, with default values shown in *italics*:

ViewCount.GetViewCount(nodeId, category: *“<empty string>”*, increment: *false*);

There is no requirement to configure a DataType; supplying the node id of any valid Content-based node (Member, Document, Media, etc.) will create the Views record in the database if it doesn’t exist. However configuring and using a DataType will allow you to control the advanced features of the counter.

Data Type Parameters

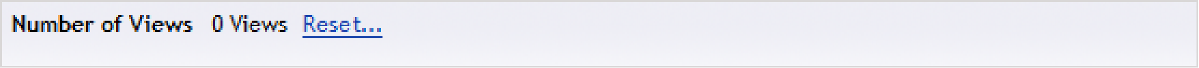


Name	Sponsorship View Count
Render control	Content Views Counter
Data Editor GUID	d9c78b8a-33a9-4116-bea1-c56737db616a

Settings

Category:	Sponsorship
Hide View Count:	● Yes ○ No
Enable View History:	● Yes ○ No
Disable Counter Reset:	● Yes ○ No

- **Category** – Specifying a different category for multiple DataTypes allows you to differentiate between multiple View Counts in a single content item. You can then render the content in different views and have a different View Count for each rendering.
- **Hide View Count** – Allows you to control (in conjunction with the API and Razor or XSLT macros, for example) whether to hide or show the view count at a Data Type level.
- **Enable View History** – Turns on recording of View Count History data including the time the view was incremented. Also recorded is Reset command events. This data is stored in the refViewCountHistory table and persists even if the current view count is reset. This is off by default.
- **Disable Counter Reset** – Turning this on disables the Reset action on Content configured with a View Count DataType:



Number of Views 0 Views Reset...

Out of the Box

Example Razor Script – PageViews.cshtml

The macro associated with this Script is “Page Views” and has the following Parameters:

- Category (text)
- Increment (bool)

```
@inherits umbraco.MacroEngines.DynamicNodeContext
@using umbraco.MacroEngines;
@using umbraco.NodeFactory;
@using Refactored.UmbracoViewCounter;

@if (!ViewCount.HideCounter(Model.Id, category: Parameter.Category)) {
  <span># Views:@ViewCount.GetViewCount(@Model.Id, category: @Parameter.Category,
increment: @Parameter.Increment == "1").ToString("N0")</span>
} else {
  ViewCount.Increment(Model.Id, category: Parameter.Category);
}
```

Default DataType – View Count

Default settings are no Category, No History, Show View Count and Enable Counter Reset.

Comments and Suggestions

If you have any comments or suggestions regarding this document or the package it describes, you can contact the author here:

Robert Foster

Email: rfoster@refoster.com.au
Blog: http://refactored.com.au/blog