

28 Days of Sitecore Rocks

SUMMARY OF BLOG POSTS BY TREVOR CAMPBELL



Peter Prochazka

Table of Contents

1: 28 Days of Sitecore Rocks: Introduction.....	2
2: 28 Days of Sitecore Rocks Installation and Configuration.....	3
3: 28 Days of Sitecore Rocks: Navigation	6
4: 28 Days of Sitecore Rocks: Advanced Navigation.....	9
5: 28 Days of Sitecore Rocks: Managing Content.....	13
6: 28 Days of Sitecore Rocks: Managing Templates - Part 1.....	16
7: 28 Days of Sitecore Rocks: Managing Templates Part 2	20
8: 28 Days of Sitecore Rocks: Layouts and Sublayouts.....	23
9: 28 Days of Sitecore Rocks: Presentation Layout.....	27
10: 28 Days of Sitecore Rocks: Pipeline Creation.....	32
11: 28 Days of Sitecore Rocks: Publishing.....	34
12: 28 Days of Sitecore Rocks: Job Viewer	37
13: 28 Days of Sitecore Rocks: Log Viewer	39
14: 28 Days of Sitecore Rocks: Media Library.....	42
15: 28 Days of Sitecore Rocks: Package Creation	45
16: 28 Days of Sitecore Rocks: Package Management - Part 1	48
17: 28 Days of Sitecore Rocks: Package Management - Part 2	55
18: 28 Days of Sitecore Rocks: Manage - Part 1	59
19: 28 Days of Sitecore Rocks: Manage - Part 2	62
20: 28 Days of Sitecore Rocks: Manage - Part 3	66
21: 28 Days of Sitecore Rocks: Manage - Part 4	69
22: 28 Days of Sitecore Rocks: Manage - Part 5	73
23: 28 Days of Sitecore Rocks: History Table Viewer and Languages.....	78
24: 28 Days of Sitecore Rocks: Query Analyzer.....	81
25: 28 Days of Sitecore Rocks: Sitecore Installation	85
26: 28 Days of Sitecore Rocks: Hidden Gems - Part 1.....	89
27: 28 Days of Sitecore Rocks: Hidden Gems - Part 2.....	92
28: 28 Days of Sitecore Rocks: Plugin Repository.....	96

1: 28 Days of Sitecore Rocks: Introduction

By [Trevor Campbell](#), February 01, 2013

Sitecore Rocks is an extremely powerful productivity tool that bridges the gap between the Integrated Development Environment (IDE) and a Sitecore web application. Sitecore Rocks aids in many different aspects of development for Sitecore including content manipulation, debugging, performance profiling, and installation evaluation.

Sitecore Rocks has a plethora of features including:

- Sitecore Explorer: View the content tree for multiple websites and multiple databases; delete, edit, or drag and drop multiple items simultaneously.
- Item Editor: Edit the content of Sitecore items directly inside Microsoft Visual Studio - no need to open a web browser. The Item Editor supports multi-edit so that multiple items can be updated in a single operation.
- Search: Fully integrated Sitecore search.
- Media Library: Query-based media library. Upload new media using drag and drop.
- Layout Designer: Use drag and drop to setup presentation for multiple devices.
- Template Designer: Quick and easy configuration of data templates.
- Link Viewer: Quickly locate referrers and references.
- Sitecore Project Management: Connect a Visual Studio project to a Sitecore website and create Sitecore items from aspx, ascx and xsl files with a few clicks. Track file operations (duplicate, copy, move, rename) and update the Sitecore items automatically.
- Log Viewer: Keep track of what is happening on a Sitecore website with a running log.
- Job Viewer: See which jobs are running in the background on a Sitecore website.
- Debug and Trace: View Profile and Trace files.
- Sitecore Query Analyzer (temporary name): scripting CRUD.
- Site Validation: Validate your site against more than 50 rules.
- NuGet: Install NuGet packages that include Sitecore items and files.
- PowerShell: Exposes a Sitecore database as a PowerShell drive.
- Sitecore Rocks is fully extendable using plug-ins that allow you to add new buttons, item editor skins, field types, pipelines and more.

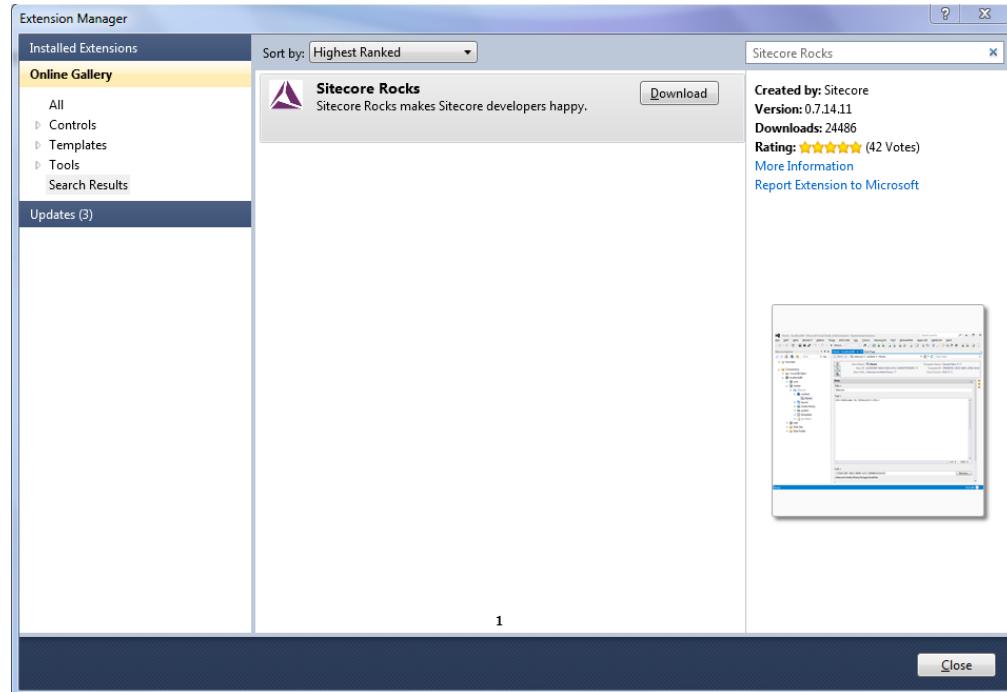
Throughout the month of February I will be publishing a series of blog articles focused on features in Sitecore Rocks and how those features make our lives as developers better. There will be a new article every day for the 28 days of February so be sure to check back daily.

2: 28 Days of Sitecore Rocks Installation and Configuration

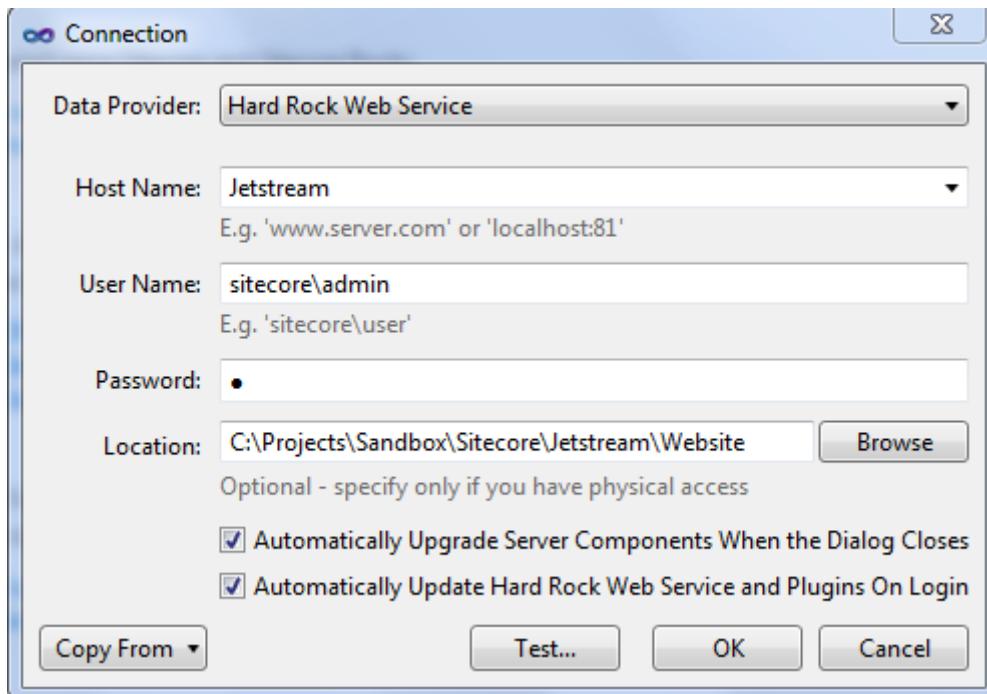
By [Trevor Campbell](#), February 02, 2013

In order to get started using Sitecore Rocks and all its great features we need to start at the beginning, installation and configuration. Thankfully this task is as easy as can be thanks to Visual Studio's Extension Manager.

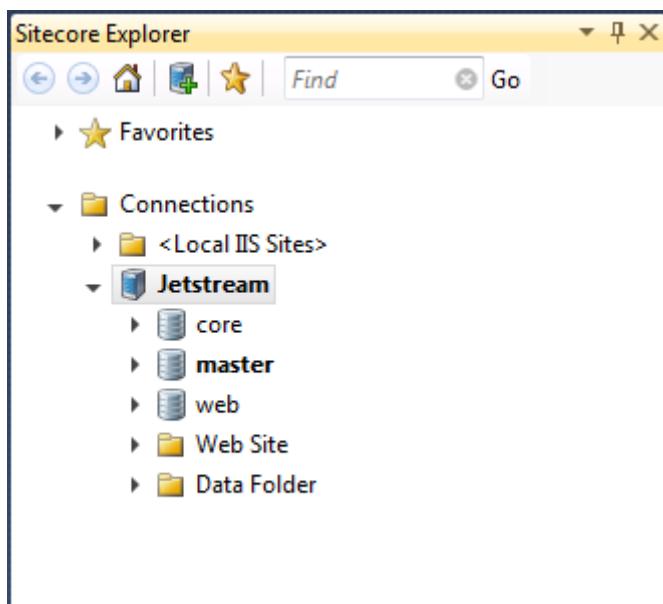
Installation So let's take it from the top. Once you have Visual Studio 2010/2012 started, open the Extension Manager from the Tools menu. Next search the Online Gallery for Sitecore Rocks. This is what you should currently see.



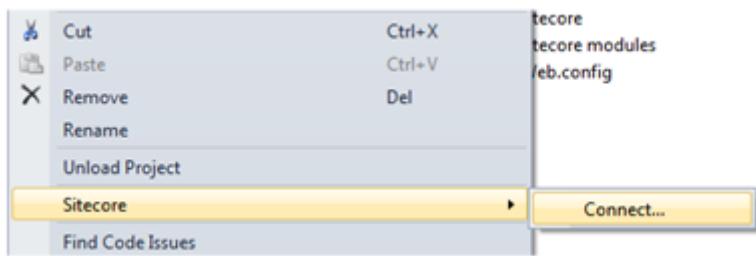
Click the Download button and you'll be guided through the installation. It's a quick couple steps with the usual whatnots. Once complete you will be prompted to restart Visual Studio to complete the installation. Half way there! **Configuration** Lastly we need to make our connections and install the server components. At this point we should have the Sitecore Explorer in the left hand pane of Visual Studio. At the top of the Sitecore Explorer is the "Connect to a Sitecore website" button that looks like a server icon with a plus sign on it; see the following dialog.



The Connection dialog has a few items to discuss. The Data Provider dropdown should have the Hard Rock Web Service selected. The Host Name field can either be used to enter a remote host name or select from the websites on the local machine that are setup in IIS. After entering valid user credentials in the User Name and Password fields, verify the Location field is correct and the “Upgrade Server Components” and “Update Hard Rock Web Service” checkboxes are selected. Lastly, test the connection by clicking the “Test...” button.



We have successfully connected to our local Jetstream instance!



The last step is to connect our Visual Studio project to Sitecore Rocks. Simply open and right click on the project in the Solution Explorer. You should now see a Sitecore menu item with the option to "Connect..." below. Click "Connect..." and we are complete with the configuration steps. Stay tuned as we start to cover the multitude of powerful features Sitecore Rocks has to offer in the following posts of 28 Days of Sitecore Rocks.

3: 28 Days of Sitecore Rocks: Navigation

By [Trevor Campbell](#), February 03, 2013

Now that we have installed and connected to a Sitecore instance it's time to start using the many features of Sitecore Rocks. Let's start exploring by learning how to navigate within Sitecore via Sitecore Rocks. There are numerous ways to navigate Sitecore using Sitecore Rocks. Here are a few of the more commonly used methods for navigation.

Start Page

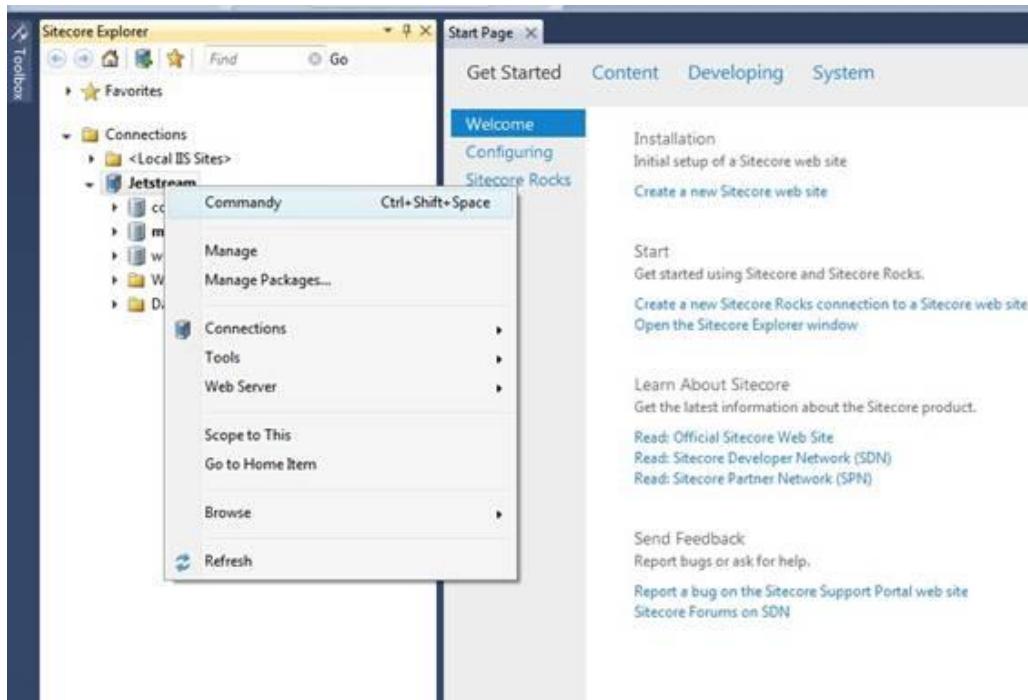
Similar to Visual Studio's start page, Sitecore Rocks' start page contains shortcut links to some of the most commonly used functions. They are categorized under four sections, Get Started, Content, Developing, and System. I highly recommend checking out each category to get an idea of what each is at your disposal.

The screenshot shows the Sitecore Rocks Start Page. At the top, there is a dark header bar with the title "Start Page" and a close button "X". Below the header is a navigation bar with four tabs: "Get Started", "Content", "Developing", and "System". The "Get Started" tab is currently selected and highlighted in blue. On the left side, there is a sidebar with three categories: "Welcome", "Configuring", and "Sitecore Rocks". The "Welcome" category is also highlighted in blue. Under "Welcome", there is a link "Installation" which is described as "Initial setup of a Sitecore web site" and a link "Create a new Sitecore web site". Under "Start", there is a link "Start" described as "Get started using Sitecore and Sitecore Rocks.", a link "Create a new Sitecore Rocks connection to a Sitecore web site", and a link "Open the Sitecore Explorer window". Under "Learn About Sitecore", there is a link "Learn About Sitecore" described as "Get the latest information about the Sitecore product.", a link "Read: Official Sitecore Web Site", a link "Read: Sitecore Developer Network (SDN)", and a link "Read: Sitecore Partner Network (SPN)". Under "Send Feedback", there is a link "Send Feedback" described as "Report bugs or ask for help.", a link "Report a bug on the Sitecore Support Portal web site", and a link "Sitecore Forums on SDN".

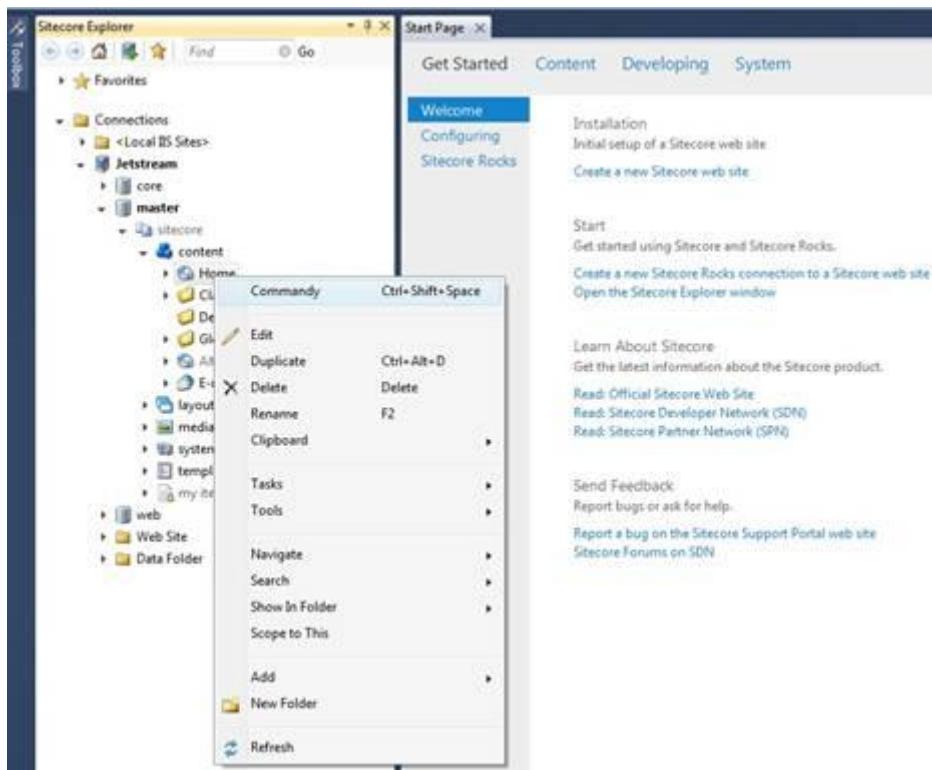
Context Menus

When you first start using Sitecore Rocks the context menus are the most common form of navigation. When right clicking on an item in the Sitecore Explorer you are presented with a plethora of options that are available for the current item.

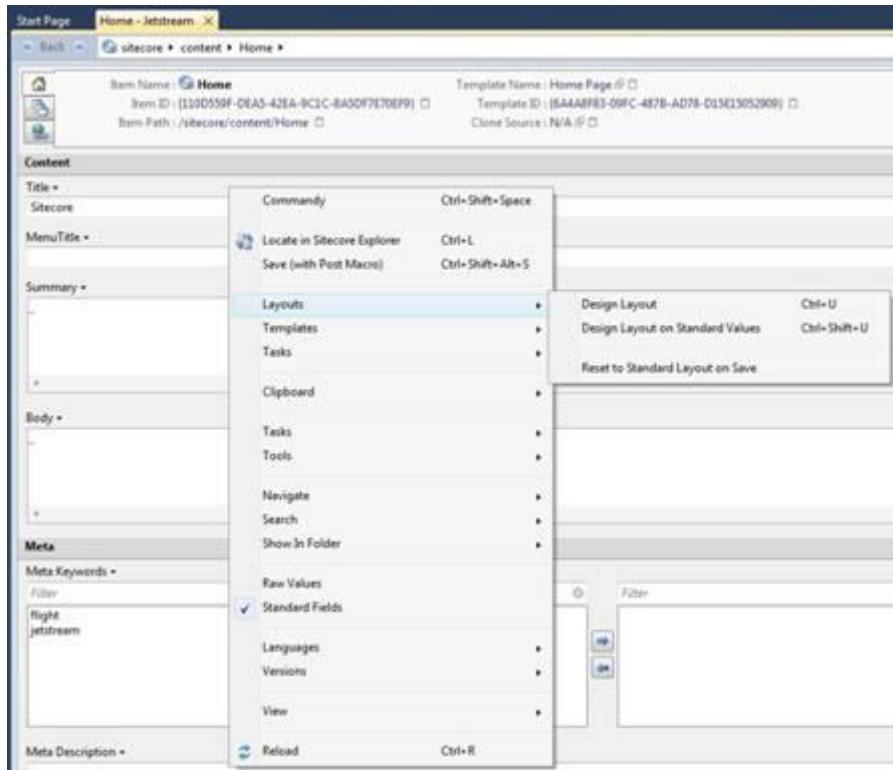
Take for example when we right click on the Jetstream node...



Versus when we right click on the Home item...



Versus when we right click on an open item in the center region.



It takes a little time to learn what options are available and when, but once you figure it out you'll be navigating Sitecore much faster within Visual Studio versus a web browser.

There are some hidden jewels within the context menus, for example:

- Go to Home Item - Navigate quickly to the website's home item from the Sitecore Rocks connection item.
- Browse -> Page Editor - Quickly open the page editor.
- Locate in Sitecore Explorer - Locate the current item in the Sitecore Explorer. This is helpful when navigating via the context menus and you want to find where in the content tree an item is located.
- Layouts -> Design Layout on Standard Values - Fast method to navigate to the presentation details on a template's standard values.
- Navigate -> Same Item in Other Database - Find the current item in another database. This is great for viewing whether an item is published or not.

Simply navigating by using the start page and context menus, you can learn the available options within Sitecore Rocks. When you know what's available then we can start using even more efficient forms of navigating such as Commandy and keyboard shortcuts which we will cover tomorrow.

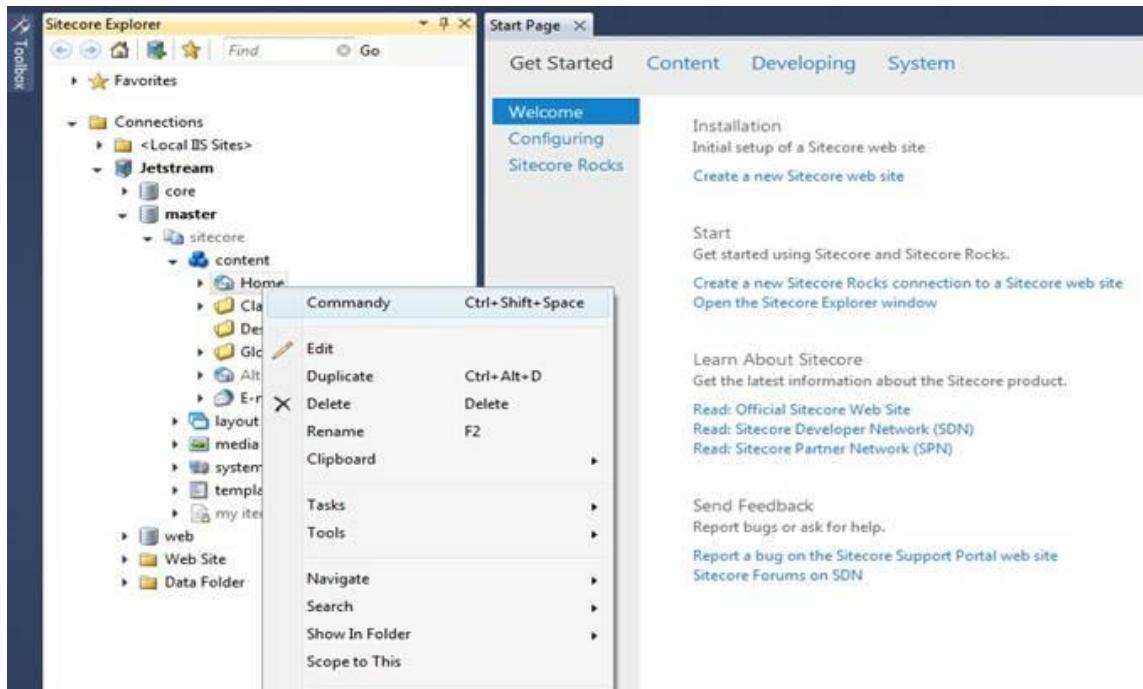
4: 28 Days of Sitecore Rocks: Advanced Navigation

By Trevor Campbell, February 04, 2013

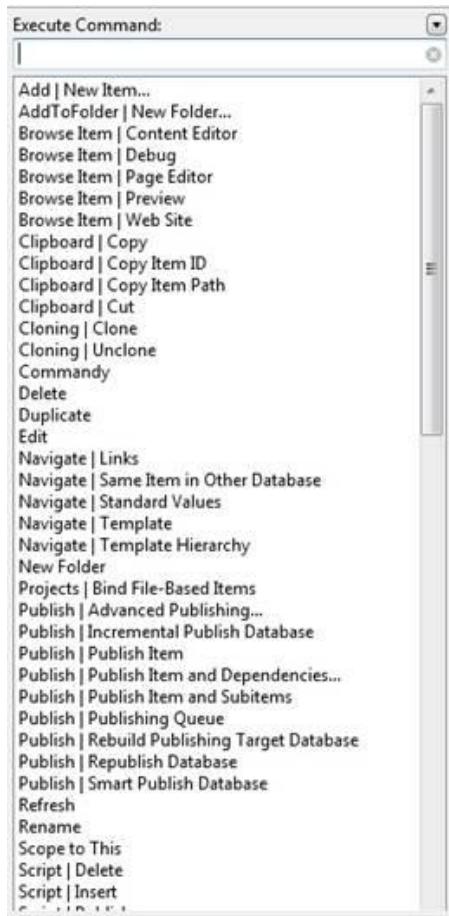
Now that we have an idea of what Sitecore Rocks has to offer let's explore how to get to these features even quicker.

Commandy

Similar to the "Navigate To..." function of Visual Studio, Commandy is a utility for finding functions in Sitecore Rocks. You may have noticed Commandy when navigating using the context menus; it is the first menu item on each and every context menu.

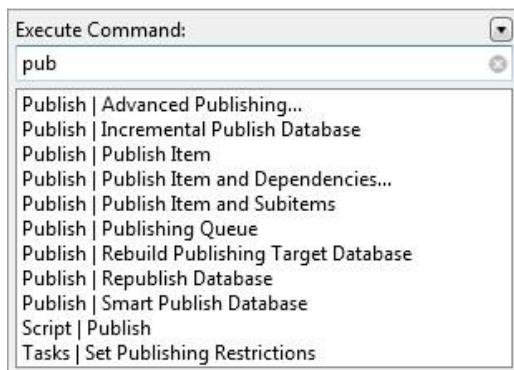


You can either open Commandy using the context menus or by using the "Ctrl + Shift + Space" keyboard shortcut. Once Commandy is open you will notice a long list of commands.



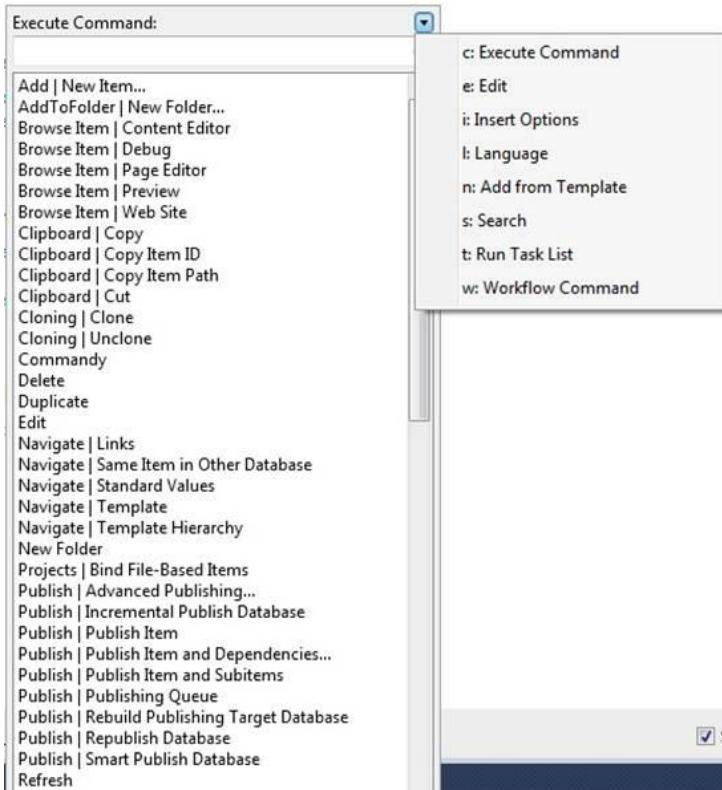
This list contains all of the available commands from the context menus flattened out in a searchable utility.

Need to perform a "Publish Item and Subitems"? Easy, simply start typing "publish" to filter the list of commands and select "Publish | Publish Item and Subitems".

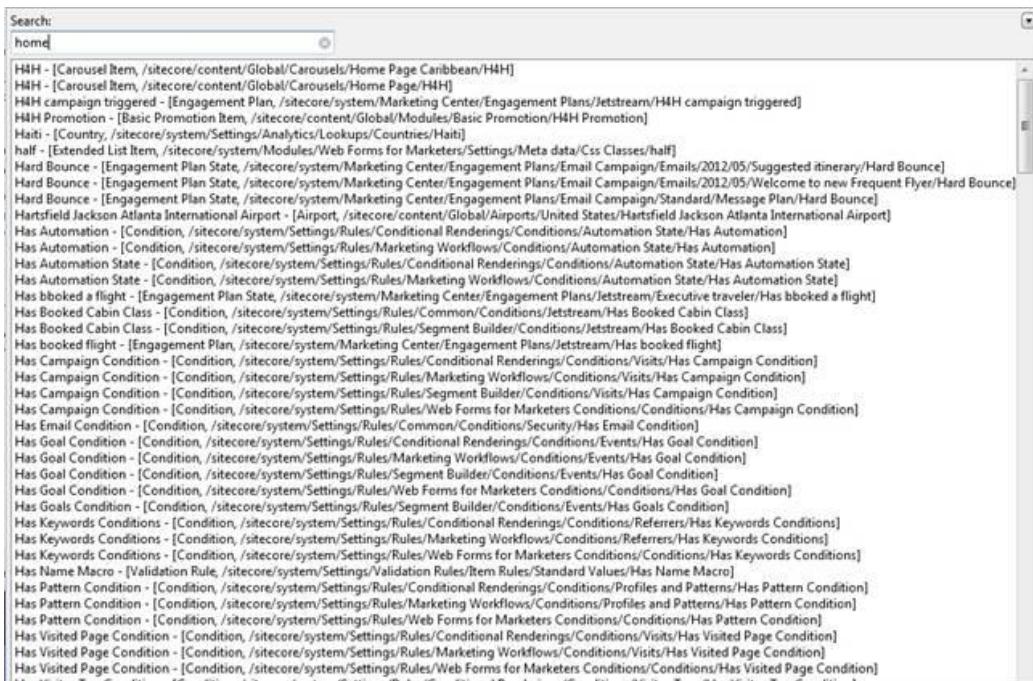


That's it! In just a couple of keystrokes we have performed a publish operation without ever leaving Visual Studio.

You might also notice the drop down menu in the upper right hand corner of Commandy containing even more available options.

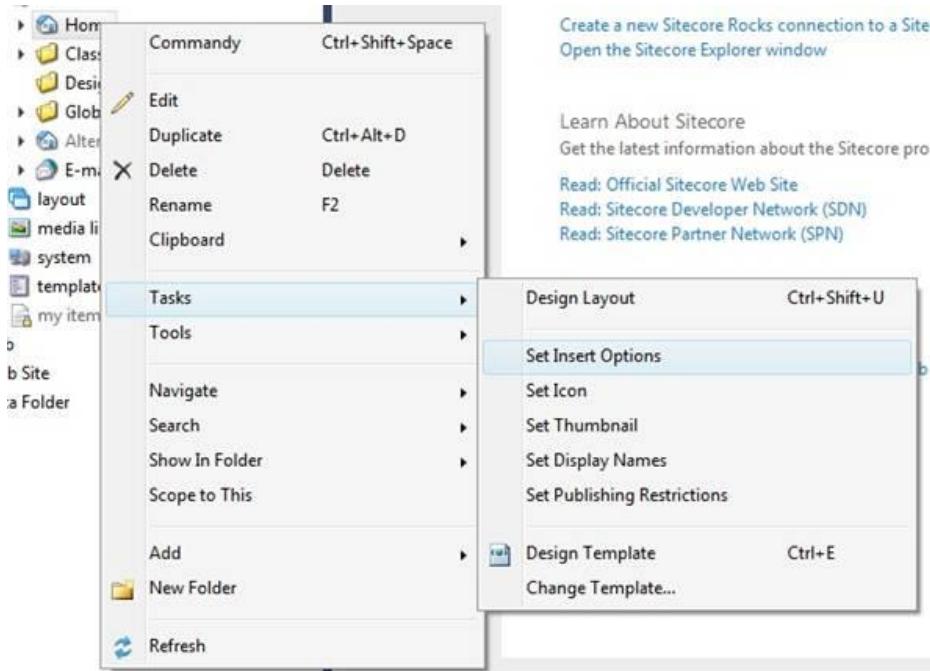


(Tip: I've found the search option to be a quick and easy way to locate items within Sitecore.)



Keyboard Shortcuts

When navigating Sitecore Rocks' context menus you may have also noticed that some of the menu items have keyboard shortcuts available.



If you are more inclined, like myself, to navigate Visual Studio using keyboard shortcuts you will be pleased. When first starting out using Sitecore Rocks I would recommend relying heavily on Commandy until you start identifying which functions you use the most. Once you know which commands you use the most, you can then start learning the keyboard shortcuts and maximize productivity.

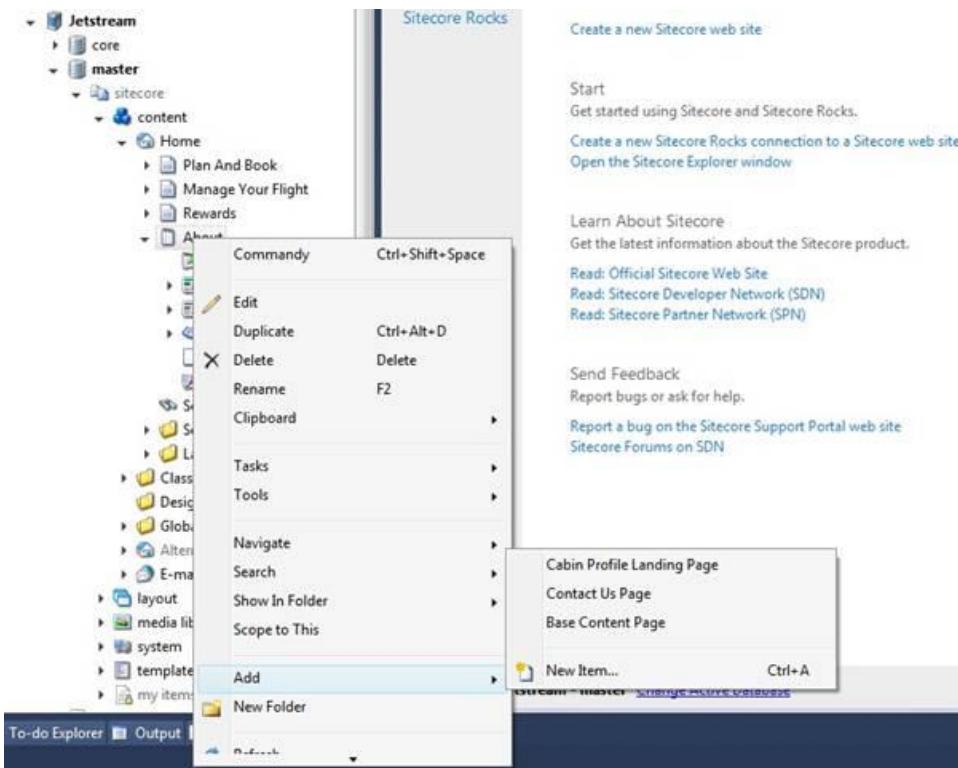
5: 28 Days of Sitecore Rocks: Managing Content

By Trevor Campbell, February 05, 2013

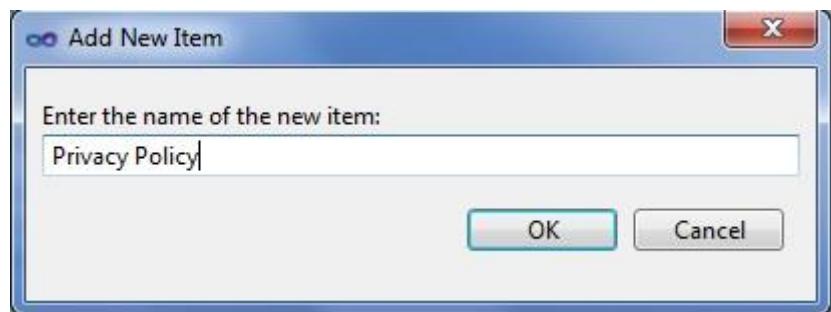
Now that we are familiar with how to navigate and what features are available with Sitecore Rocks let's dive into the features starting with managing content. Creating, editing and deleting items is extremely easy and can be all be done within the comforts of Visual Studio.

Creating New Items

Let's start by creating new content. Navigate to the section in the content tree where you would like to create the new content and right click to open the context menu. In the context menu expand the "Add" menu item to reveal which items are available to be created via the template's insert options.

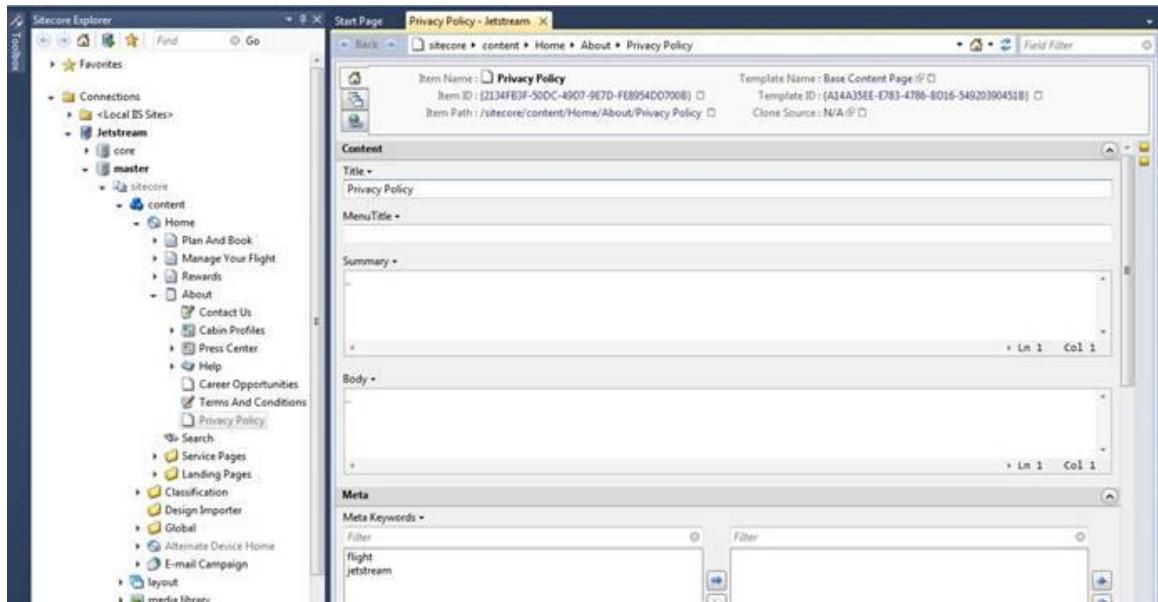


After selecting which type of content to insert, you are prompted to name the new item the same as you would be if creating the item in a Sitecore web application.

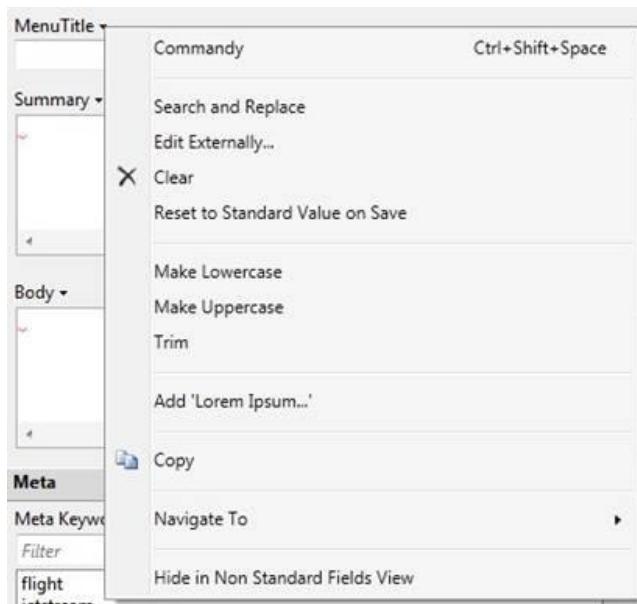


Editing Items

When editing items with Sitecore Rocks, just as with editing files with Visual Studio, a new tab is opened. This is very helpful because you can open and edit multiple items at once by switching back and forth between tabs very quickly.

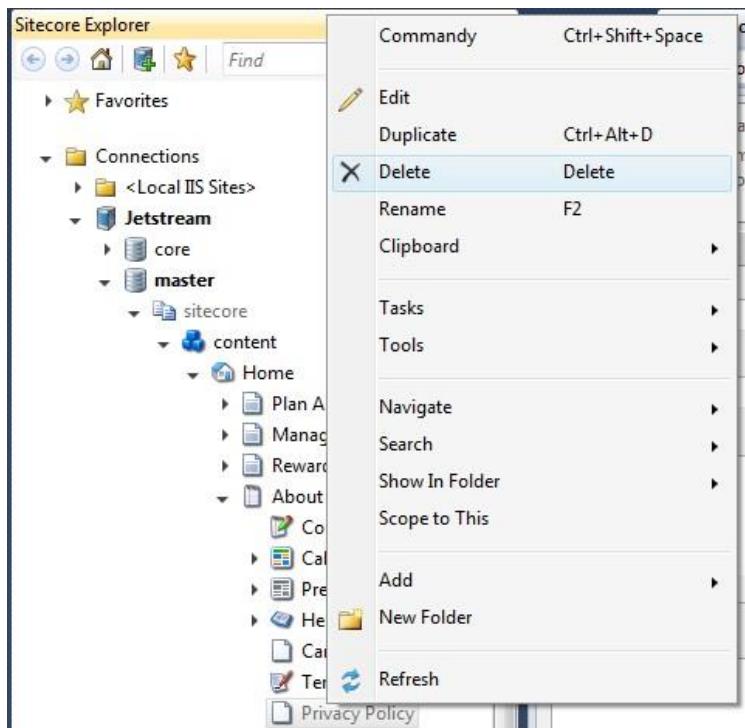


Sitecore Rocks also adds features when editing/adding items for each field via a convenient context menu to the right of the field. "Add 'Lorem Ipsum...'" is one I find helpful when creating new items for development as I don't need to leave Visual Studio to retrieve test content.



Deleting Items

Deleting items can be done either using the context menu or by simply hitting the delete key.



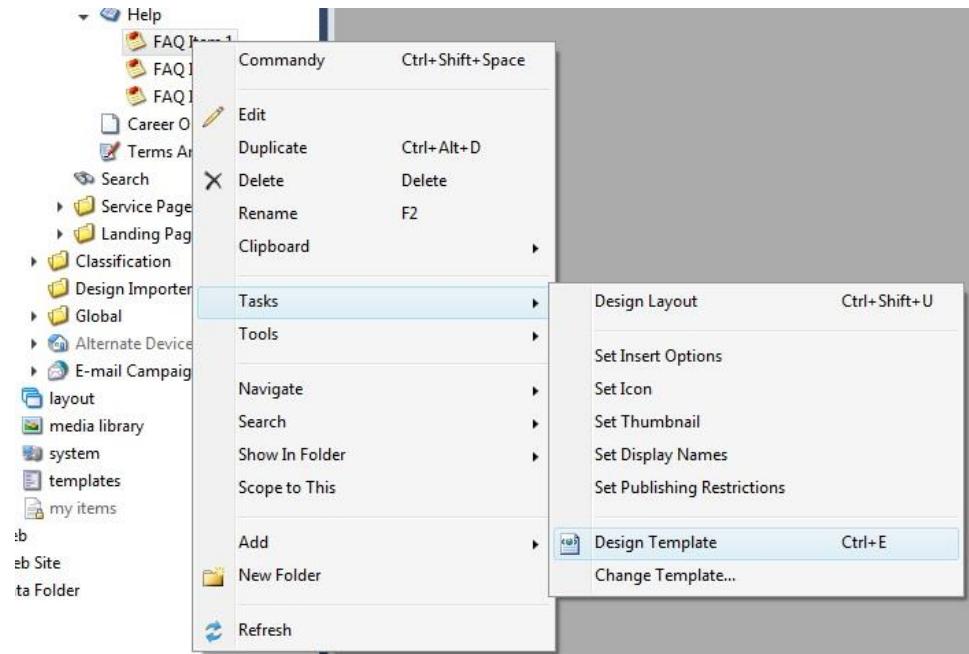
Having the ability to manage content using Sitecore Rocks is a huge time saver and can transform the day to day task of working with items.

6: 28 Days of Sitecore Rocks: Managing Templates - Part 1

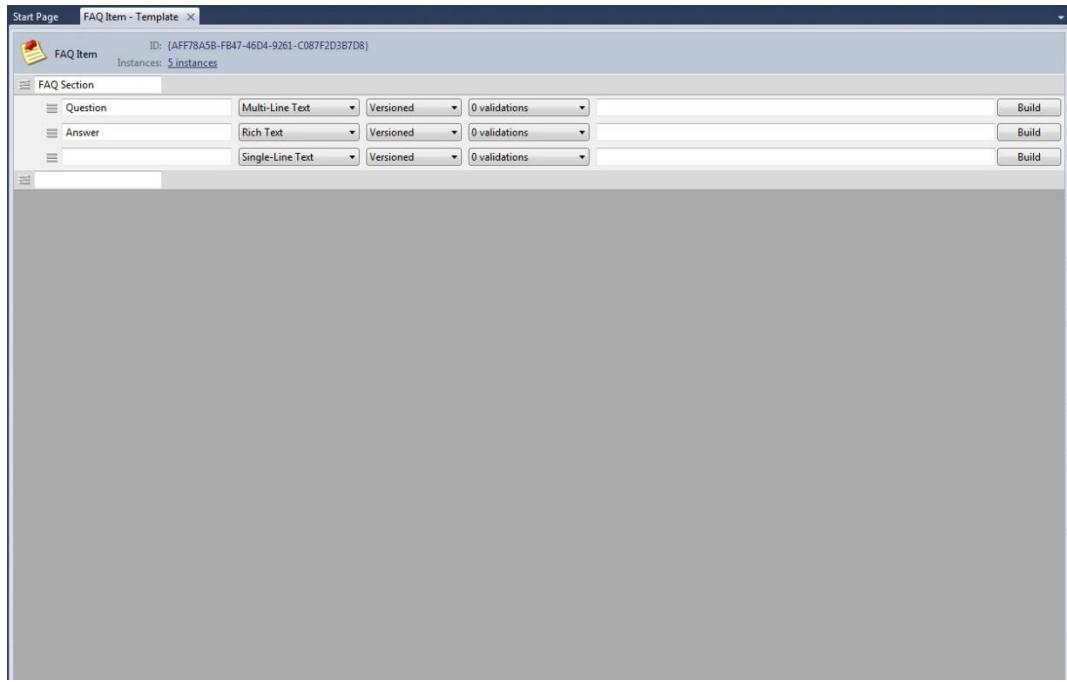
By [Trevor Campbell](#), February 06, 2013

Next in the 28 days of Sitecore Rocks is the wonderful world of template management. This topic will be a two part article due to the amount of features related to template management. The great news is that any one of these features can make a Sitecore developers life easier. So let's get started working with templates in Sitecore Rocks!

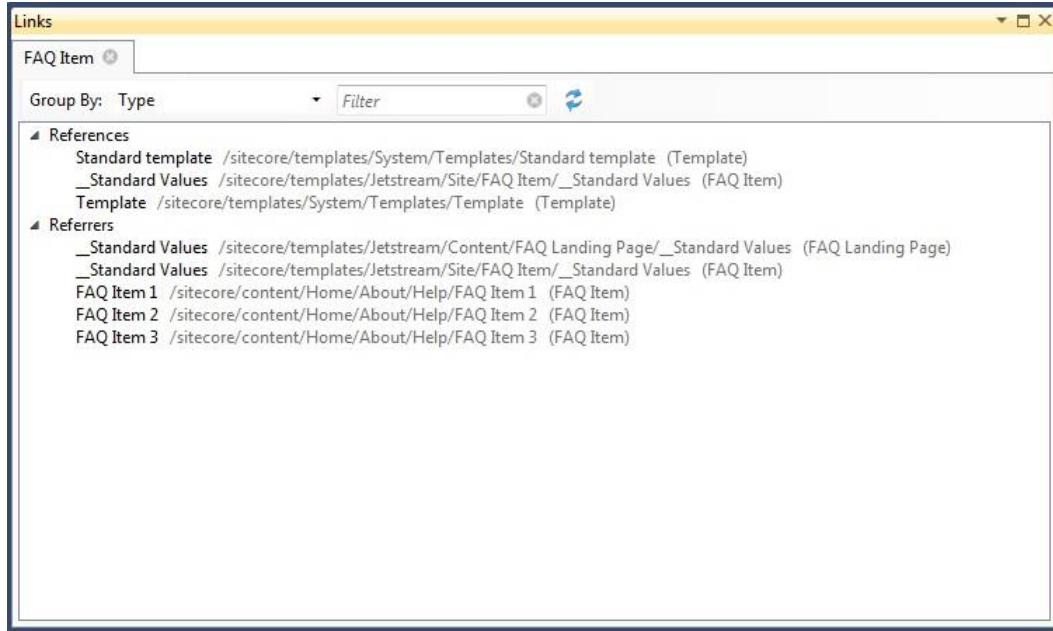
Let's start by opening a template. Navigate to an item in which we would like to edit the template. Open the context menu by right clicking on the item and under "Tasks" you will find "Design Template." We could also use the "Ctrl+E" keyboard shortcut or Command+ to open the template.



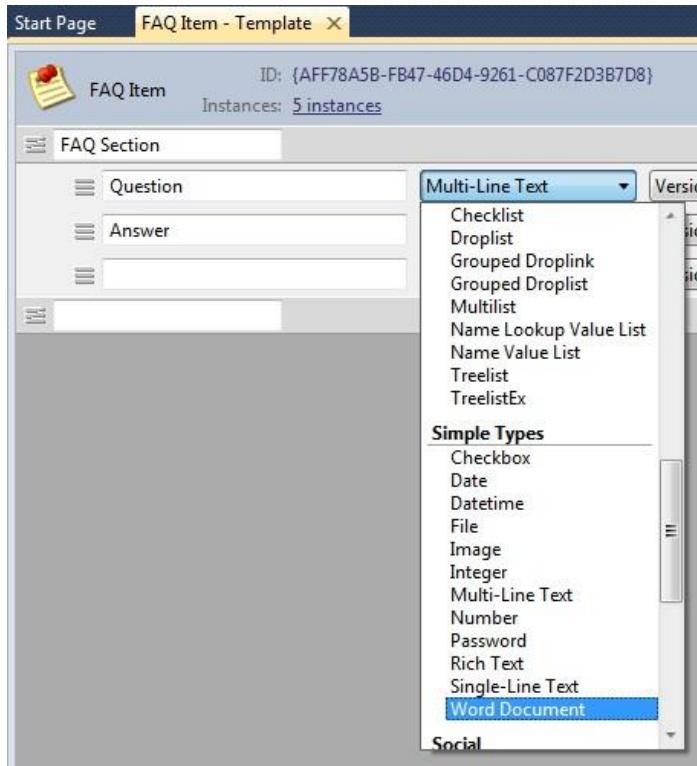
The template opens in a new tab very similar to editing templates in Sitecore via a web browser. There are some noticeable differences however.



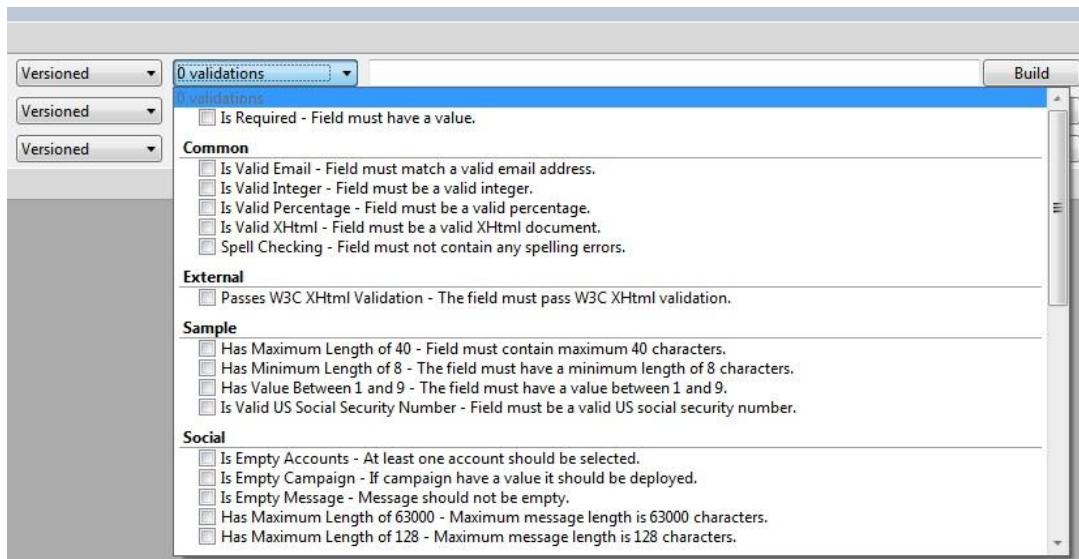
The first difference is the "Instances" link. This link is very helpful because it not only gives the number of items created from this template, but when clicked open also provides the "Links" dialog which gives the references and referrers.



Moving into the template sections and fields reveals a few more interesting bits of information. The field type and versioned drop downs match what we currently have in Sitecore.



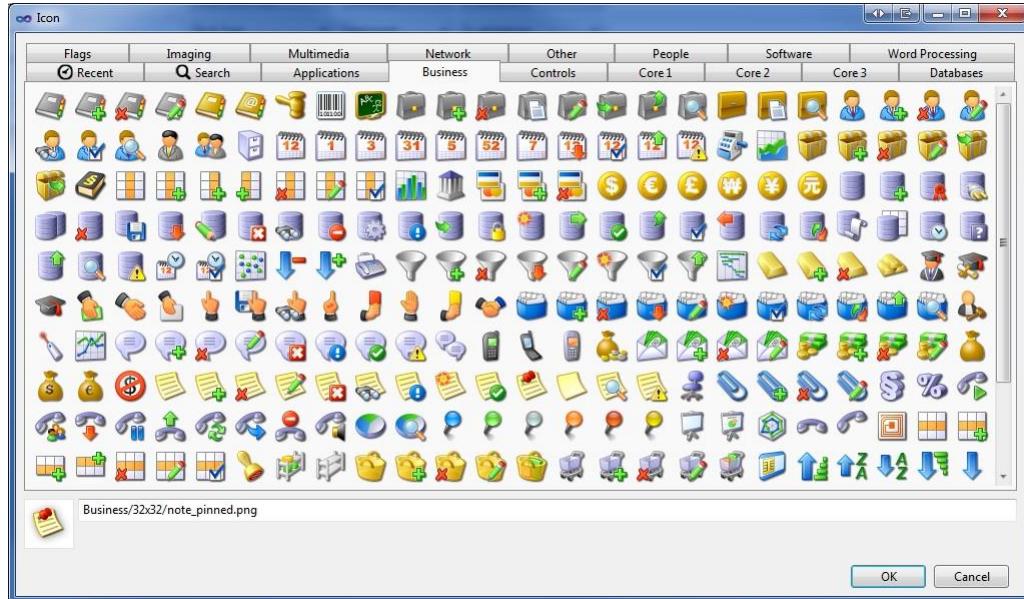
However, the validations drop down is new; opening this drop down reveals that we can set our field validations along with all the other attributes. This approach is much easier than the standard Sitecore method.



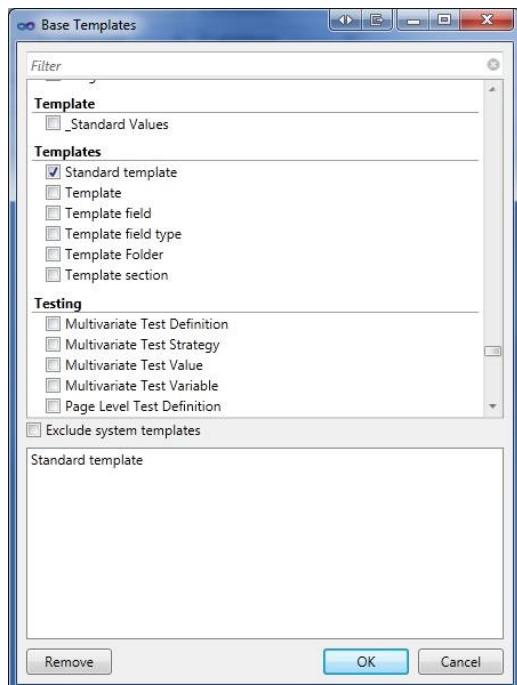
The last piece you will notice associated with fields is the data source textbox and "Build" button. I'll be covering these in part two of managing templates.

There are a few other features associated with template management. If you right click inside the template and open the context menu you will notice a few template specific functions. One of these features is the "Set Icon" function. This function opens a new window where you can

select the icon for the template. This doesn't sound like much but if you've ever associated icons with templates you will appreciate this method.



The other feature in the context menu for the template is the "Base Templates" selection. When using this feature you can quickly select your base templates using the search filter.

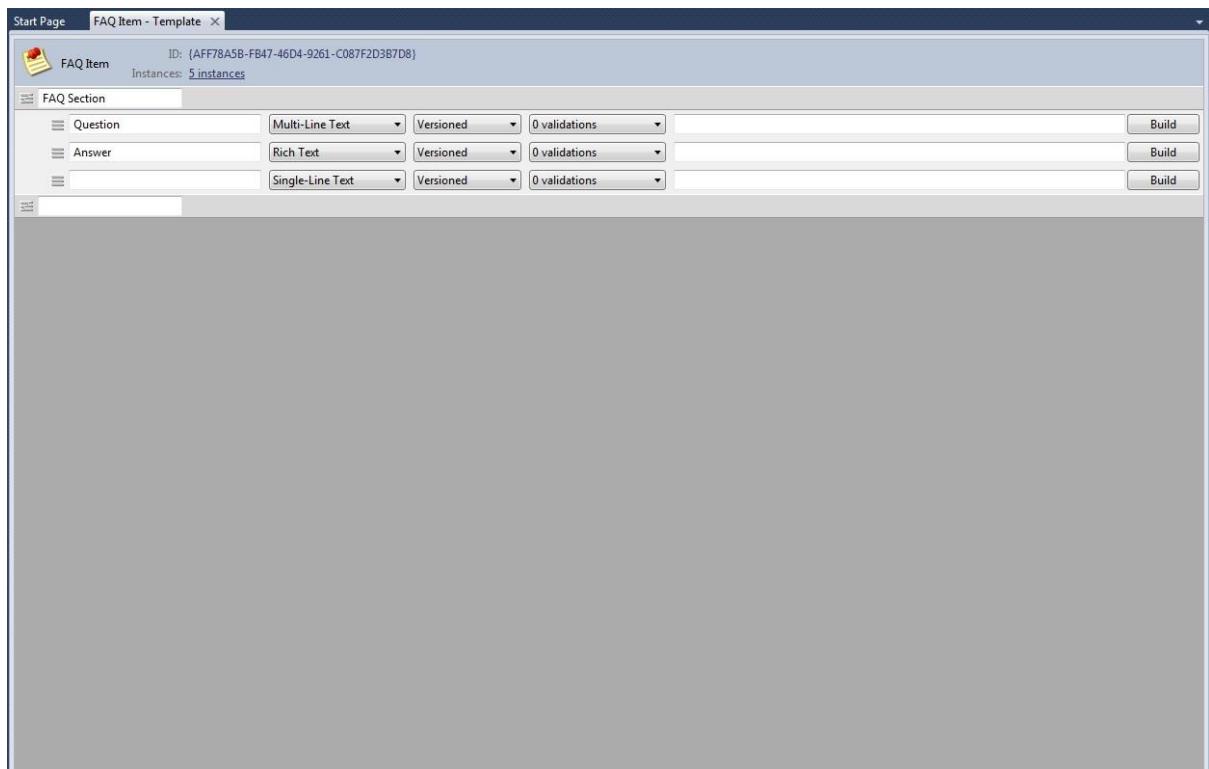


There are a multitude of features in Sitecore Rocks specifically for template management. We'll finish up with template management in part two where we talk about field data sources.

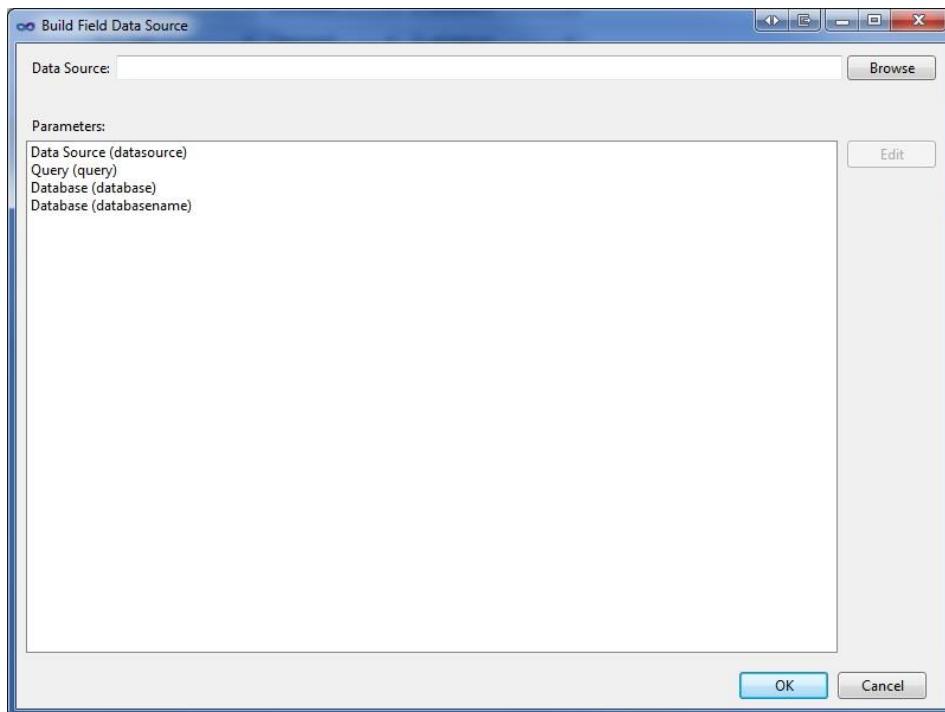
7: 28 Days of Sitecore Rocks: Managing Templates Part 2

By [Trevor Campbell](#), February 07, 2013

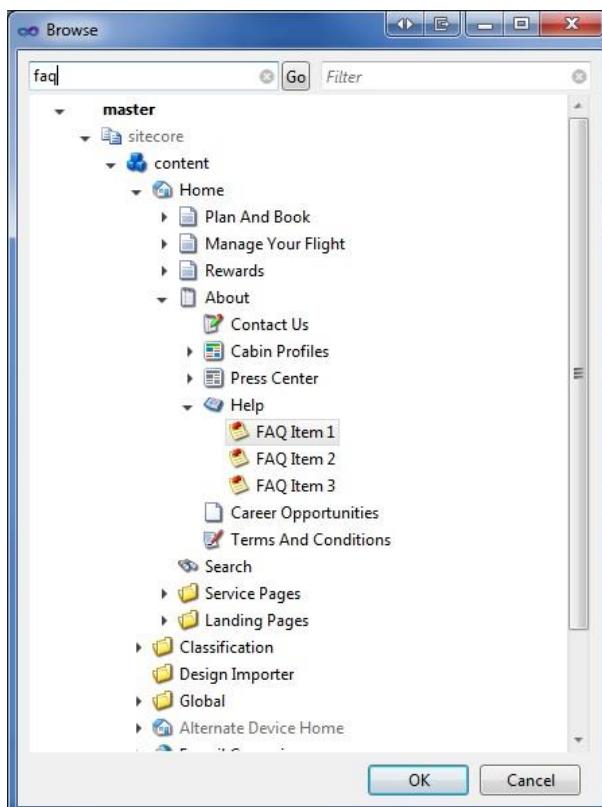
Let's continue with managing templates in Sitecore Rocks by discussing field data sources. The issue with working with data sources in Sitecore is that unless you work with them on a frequent basis, you tend to forget the syntax. If you do forget the data source syntax, then you either navigate away from the current template to find another template with the correct syntax or refer to documentation. Now let's see how Sitecore Rocks makes this an issue of the past. If we don't already have a template open let's do that now.



On the end of our fields is the data source textbox and the "Build" button. We could write our data source query directly into the textbox just as we do in Sitecore now and save the template. However, Sitecore Rocks adds a builder for constructing data sources. Let's click on "Build" to open the builder and investigate what is available.



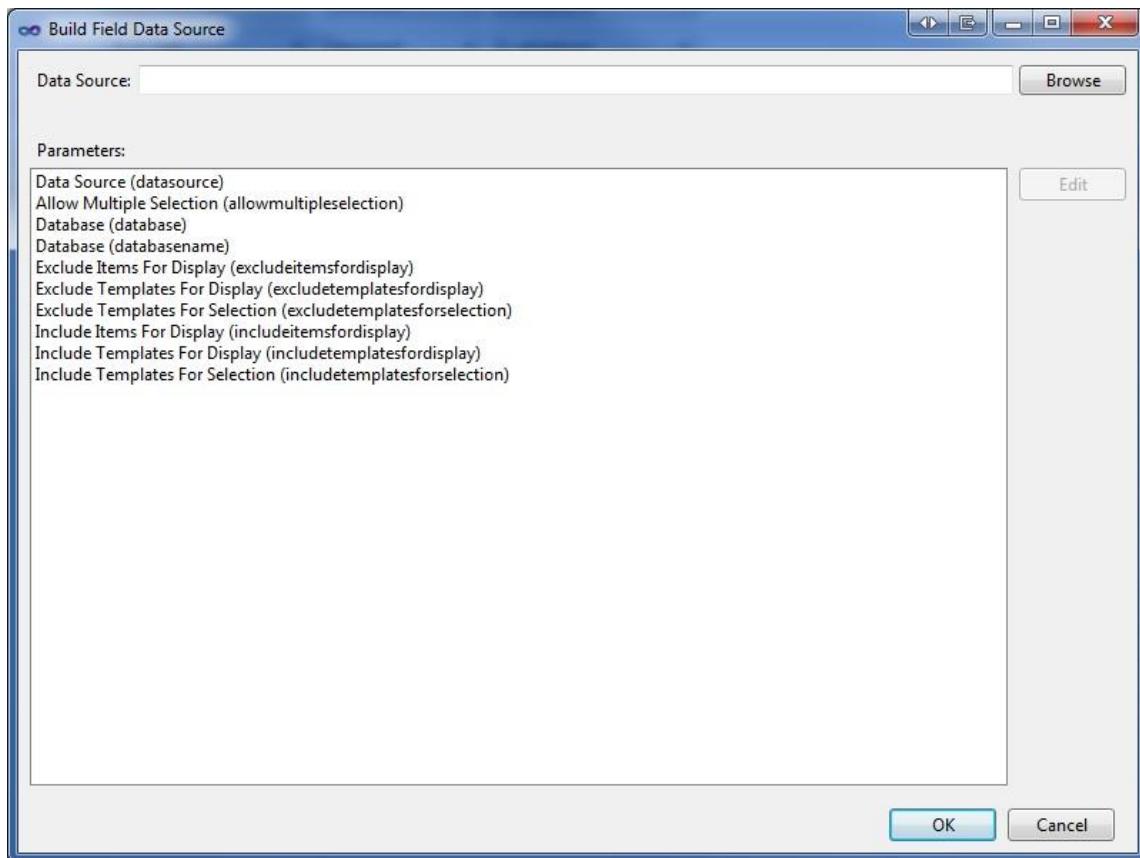
With the builder open you will notice we have several options in the "Parameters" section. Let's start with a simple example of setting a path for the data source. First select the "Data Source (datasource)" option in the "Parameters" section and click the "Edit" button to the far right hand side. This will open a browser dialog to select the item for our data source path.



As with many dialogs in Sitecore Rocks, you can search and filter to quickly find what you're looking for even if it's buried in the content tree. When you find the item you're looking for, simply click the "OK" button and notice that the "Data Source" textbox is now populated.

You may also notice the "Query (query)" option under "Parameters." This option lets you write a query for your data source in the Query Analyzer (more to come on the Query Analyzer in the coming days).

The options available in the "Parameters" section are also dependent on the type of field. Below I've selected a "treelist" field and the options are quite different. Now we have the ability to include and exclude template for display and selection.



Template management in Sitecore Rocks truly makes our lives as developers easier. Putting a series of interfaces around data source creation is a fantastic feature that hopefully someday will make its way into the web interface of Sitecore.

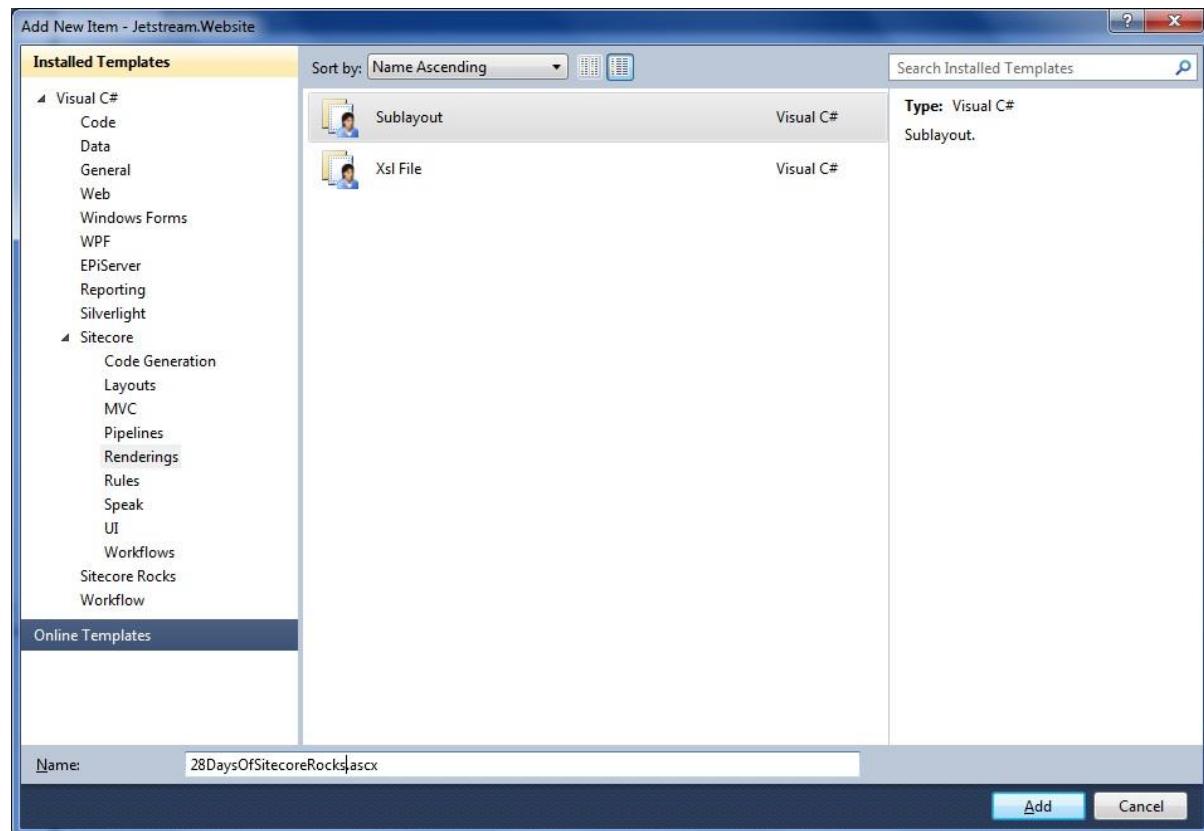
8: 28 Days of Sitecore Rocks: Layouts and Sublayouts

By [Trevor Campbell](#), February 08, 2013

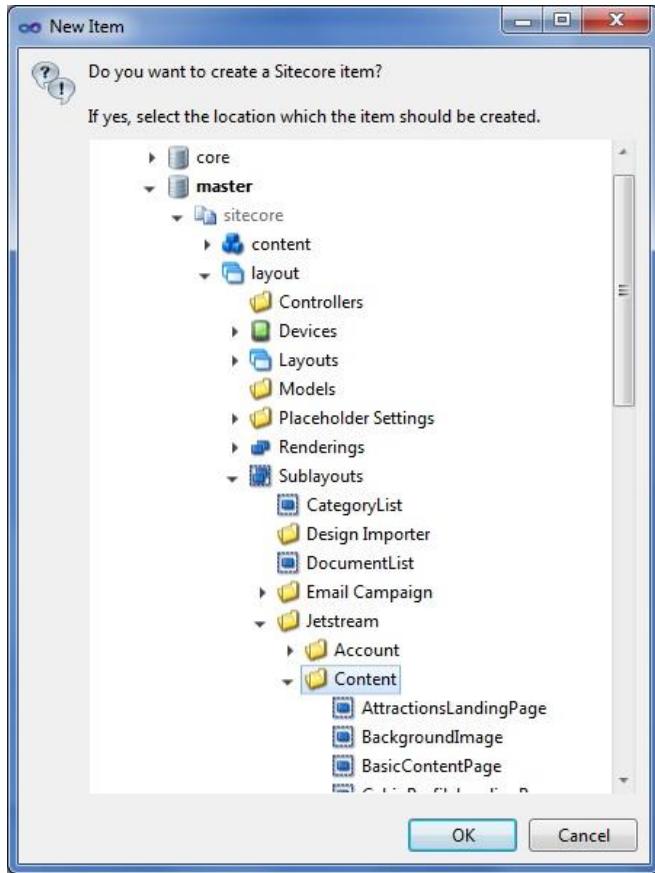
Today we will see how Sitecore Rocks can simplify the creation of Layouts and Sublayouts for developers. Without Sitecore Rocks if a developer wants to create a new layout or sublayout in Sitecore, they would log into the web interface and create one using the Content Editor. This method however doesn't create a code-behind or designer files associated with the layout or sublayout. Let's see how Sitecore Rocks simplifies this for developers.

Create a new Layout / Sublayout

To add a new layout or sublayout, right click the location in Visual Studio's Solution Explorer and select "Add -> New Item." To create a layout, navigate under the Sitecore section to Layouts and select "Layout." To create a new sublayout, navigate under the Sitecore section to Renderings and select "Sublayout." Give your layout or sublayout a name and click the "Add" button.



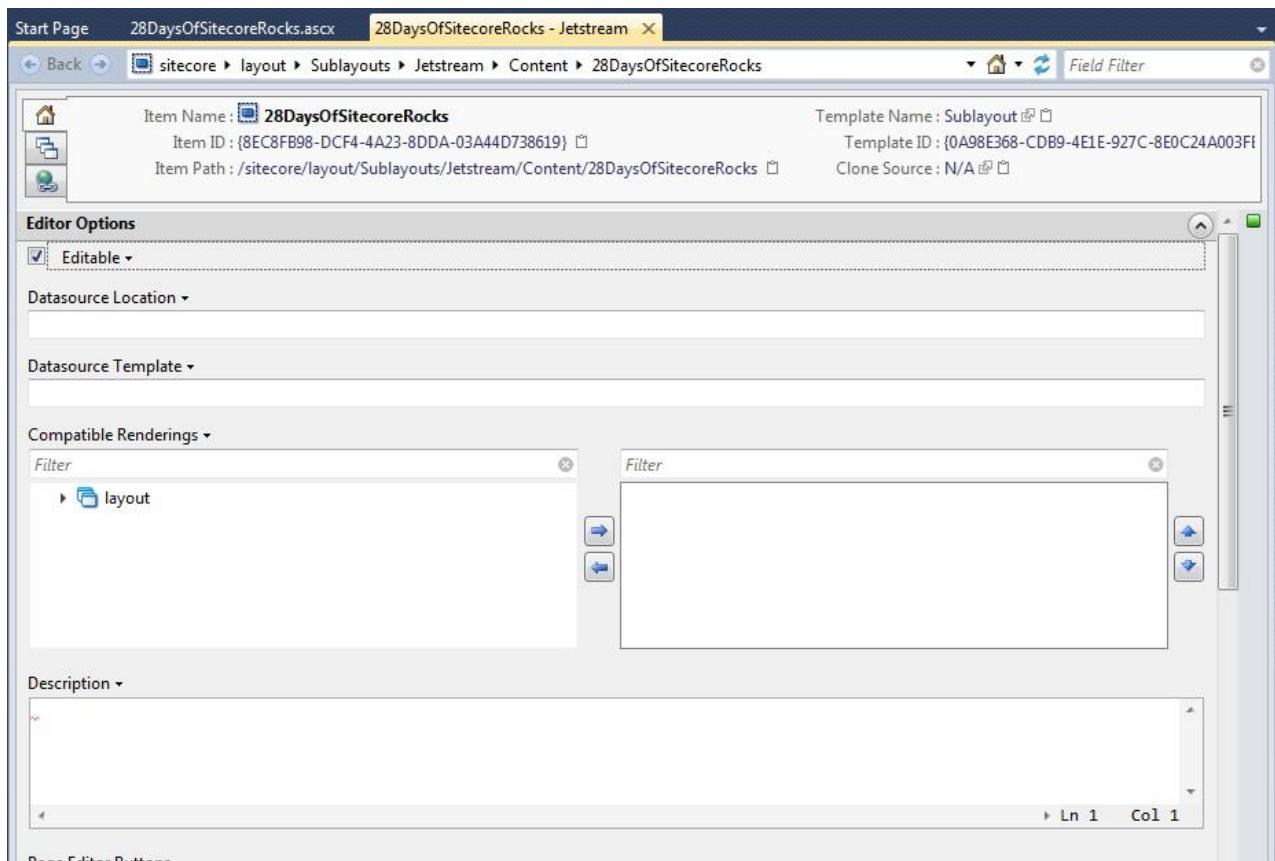
After clicking the add button a new dialog will appear asking if you would like to create a Sitecore item for the new layout or sublayout. Navigate to the location in Sitecore where you would like the item created and click "OK."



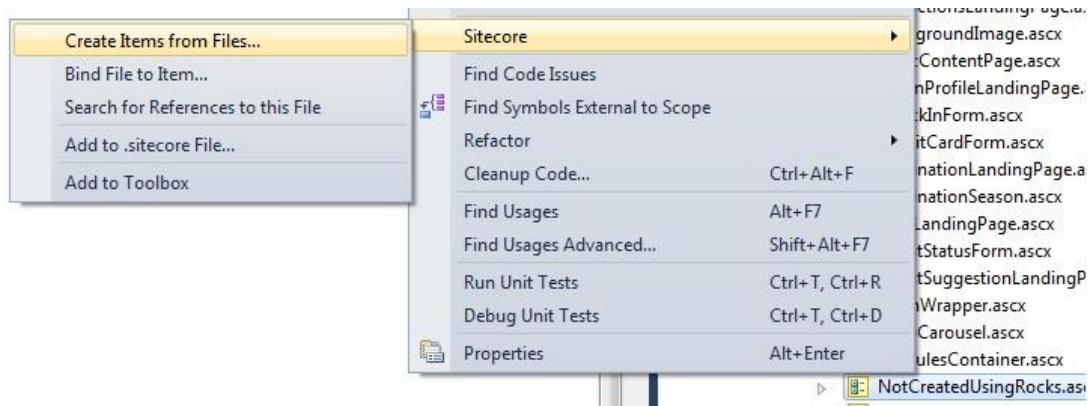
We have successfully created a layout or sublayout both in Sitecore and the file system, and these are associated with one another.

Editing a Layout or Sublayout

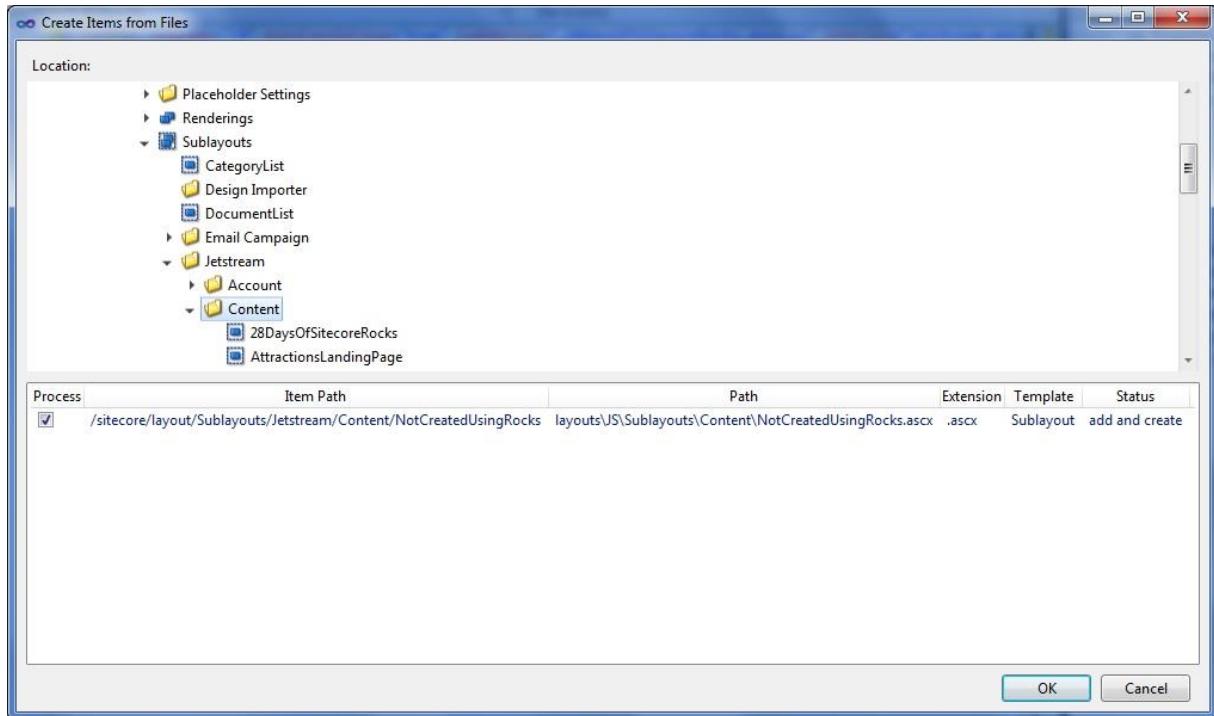
Editing the properties of layout or sublayout items is just as easy as editing content items. This can be done in one of two ways. If you are navigating within the Sitecore Explorer you can locate the item and open it by double clicking. If you are navigating from the Solution Explorer, right click on the layout or sublayout file and under the Sitecore menu select "Edit Item."



If you have a layout or sublayout that was created in Visual Studio but doesn't have an associated item in Sitecore, you can create the item in Sitecore by right clicking the file in the Solution Explorer and under the Sitecore menu select "Create Items from Files."



This will open a dialog prompting where in Sitecore to create the associated item for the layout or sublayout.



As you can see, Sitecore Rocks can improve even the small tasks such as creating layouts and sublayouts.

- Hi Trevor.

I'm getting an error about missing the Sitecore.Rocks.Template.Wizard.dll when attempting to create a sublayout as described. I have loaded the SiteCore Rocks plug-in twice but no luck. Any tips on working through this?

- Michael Prinzo

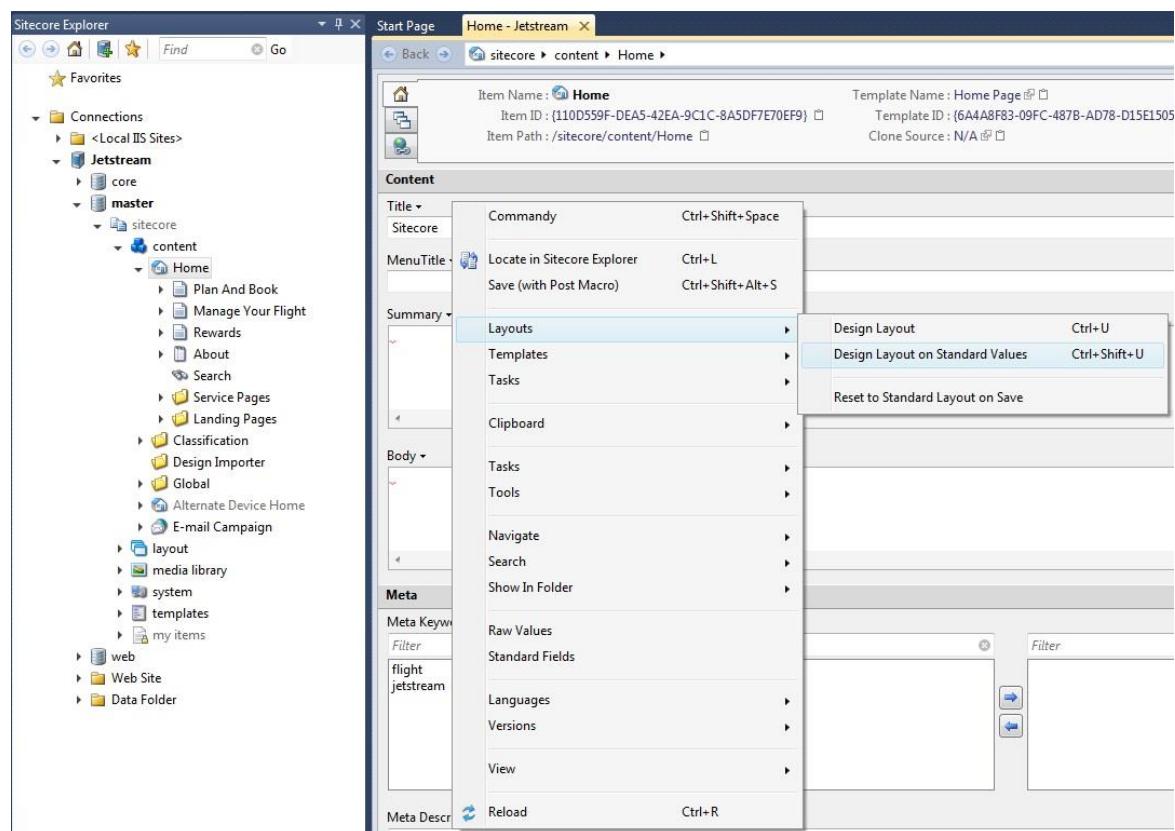
February 14, 2013 at 6:52 AM

- I was getting an error about missing Sitecore.Rocks.Template.Wizard.dll. I was able to find the dll at
c:\Users\[username]\AppData\Local\Microsoft\VisualStudio\10.0\Extensions\Sitecore\Sitecore Rocks\0.7.5\
and copy to:
c:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE\PublicAssemblies\
It fixed my problem.

9: 28 Days of Sitecore Rocks: Presentation Layout

By Trevor Campbell, February 09, 2013

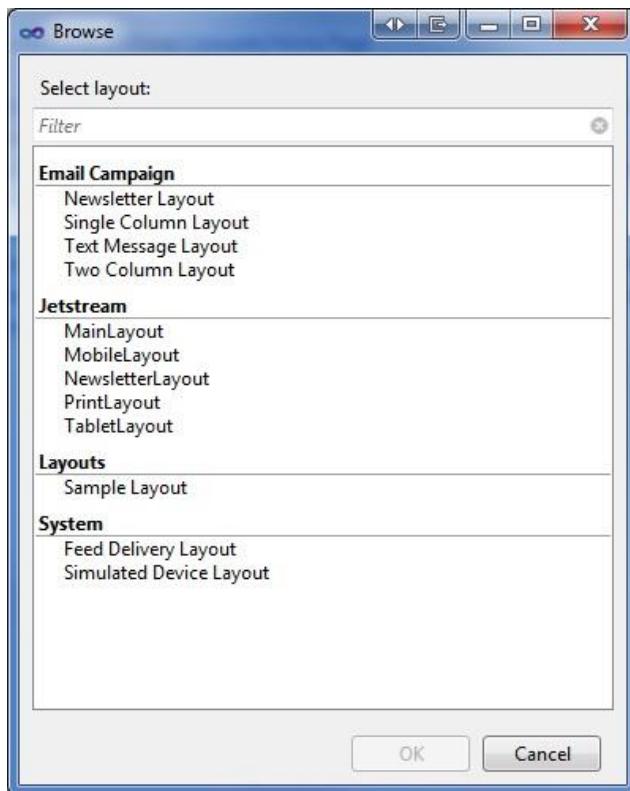
Now that we know how to work with Sitecore templates and how to create layouts / sublayouts the next logical feature to talk about is presentation layout. Sitecore Rocks has a few great features that streamline the management of presentation details. Let's dive in and learn more. First let's start by opening the Design Layout on the Home item. Right clicking inside the item will open the context menu and under the Layouts menu is "Design Layout" and "Design Layout on Standard Values." Since we almost always assign our presentation information on the standard values item of a template, selecting the "Design Layout on Standard Values" will open the presentation details we are going to be working with in most scenarios.



With the Design Layout editor open we can see that it is organized somewhat differently than the web interface version. For starters, the devices are organized into tabs. Also, the data source values are visible on the main page.

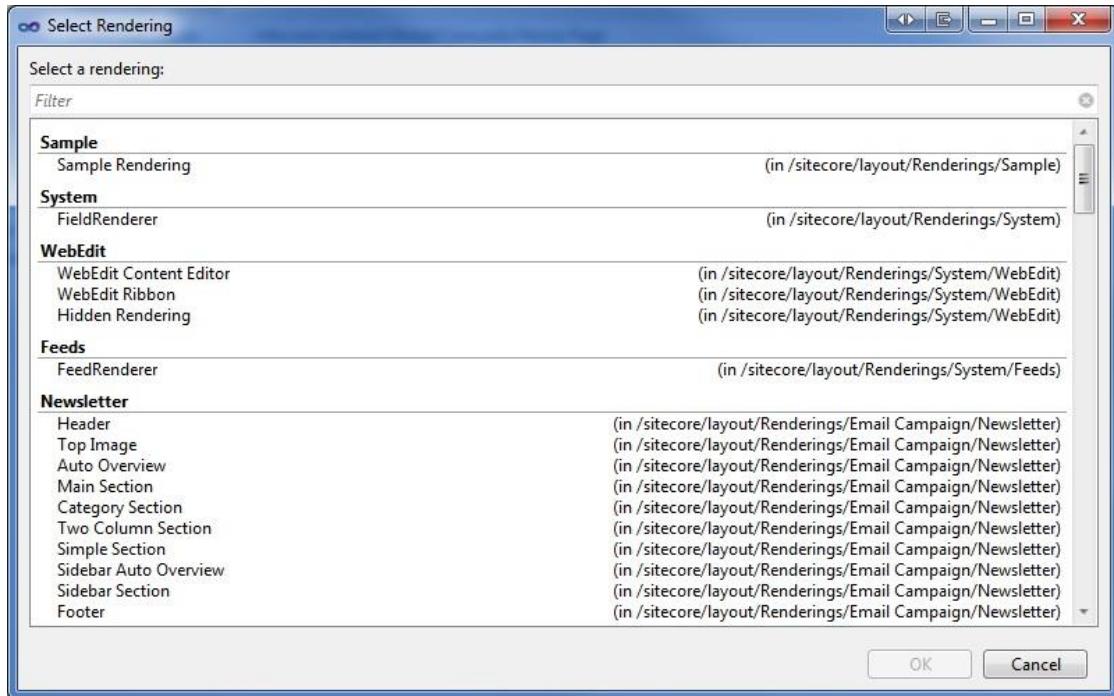
Layout:			
/sitecore/layout/Layouts/Jetstream/MainLayout			
Print			
Feed			
Mobile			
Tablet			
Renderings and Place Holders:			
Id	Rendering	Place Holder	Data Source
ModulesContainer	/main/main_content		
HeroCarousel	/main/top_media	/sitecore/content/Global/Carousels/Home Page	<input type="button" value="Browse..."/>
PhotoWrapper	main		<input type="button" value="Clear"/>
BasicPromotion	main_content_module	/sitecore/content/Global/Modules/Basic Promotion/Jetstream Credit Card small	<input type="button" value="Filter"/>
BasicPromotion	main_content_module	/sitecore/content/Global/Modules/Basic Promotion/Jetstream credit card wide	
BasicPromotion	main_content_module	/sitecore/content/Global/Modules/Basic Promotion/Jetstream Credit Card small	
BasicPromotion	main_content_module	/sitecore/content/Global/Modules/Basic Promotion/Jetstream credit card wide	
BasicDestinationPromotion	main_content_module	/sitecore/content/Global/Modules/Destination Promotion/Chicago Winter Getaway	
DestinationPromotion	main_content_module	/sitecore/content/Global/Modules/Destination Promotion/Chicago Winter Getaway 1	
WideDestinationPromotion	main_content_module	/sitecore/content/Global/Modules/Destination Promotion/Chicago Winter Getaway 1	
PointsPromotion	main_content_module	/sitecore/content/Global/Modules/Basic Promotion/Points Promotion	

Now that we've had a quick tour of the interface, let's start using the Design Layout editor. First let's start by changing the layout of the template. To do so, click the "Browse" button to the far right hand side of the layout textbox. This will open a dialog to select a layout from the currently available layouts in the system.

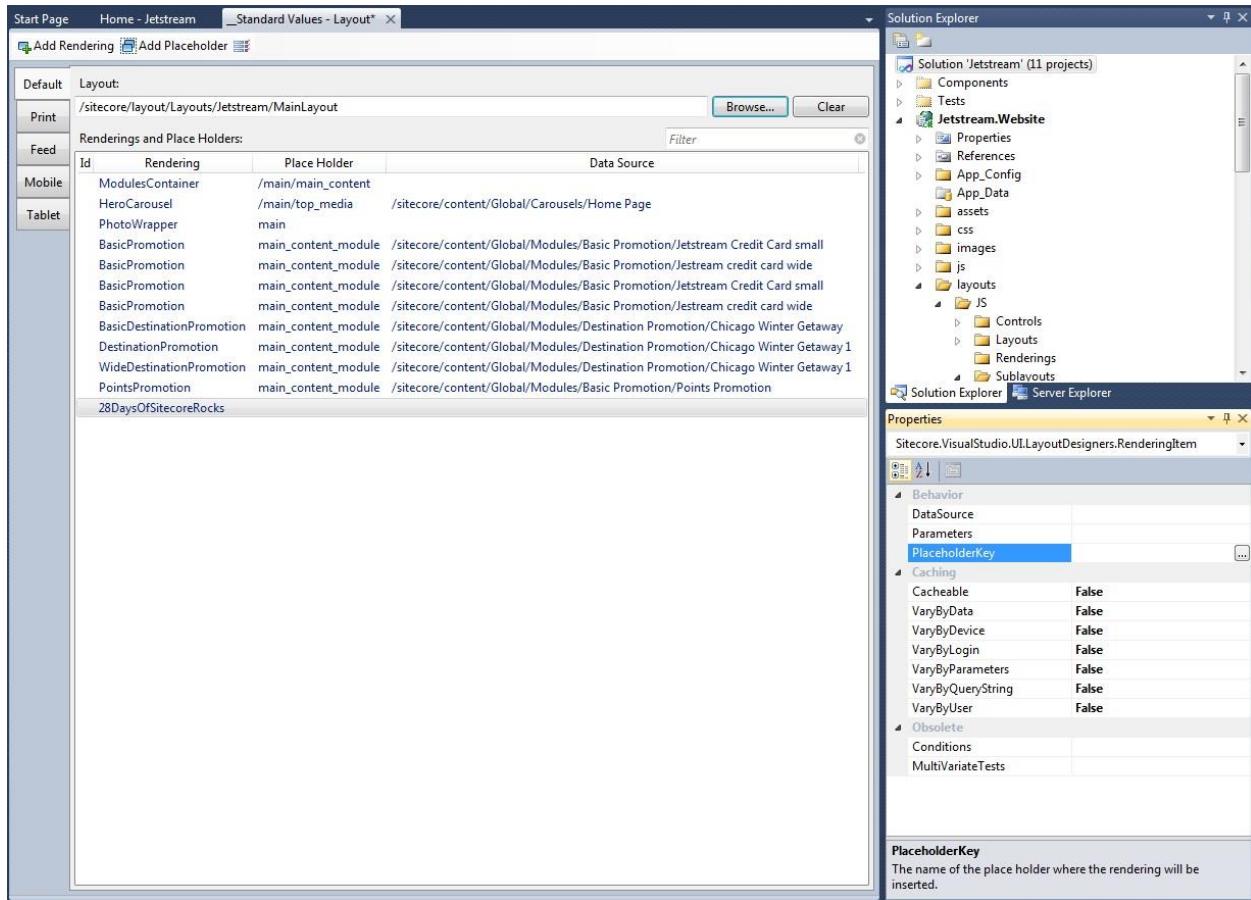


Simply select the desired layout and click "OK." Our layout is now set.

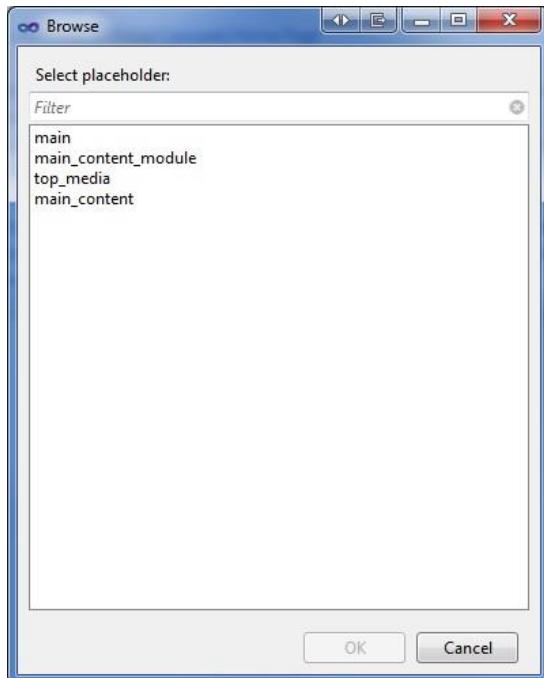
Now let's add a new rendering to the template. To add a new rendering, simply click the "Add Rendering" button at the top of the Design Layout editor. This opens a similar dialog as the layout selection dialog. We typically have many more renderings than layouts so the search filter at the top comes in quite handy to locate the desired rendering.



Wait. What about specifying the placeholder the rendering belongs to? Great question; I'm glad you asked. Next to the "Add Rendering" and "Add Placeholder" buttons is the "Properties" button. Similar to the properties window for code files, this is where we set the attributes for a rendering. Notice the "PlaceholderKey" property. This is the attribute where we set the placeholder.



Clicking the ellipsis located to the far right of the field opens the placeholder selection dialog.



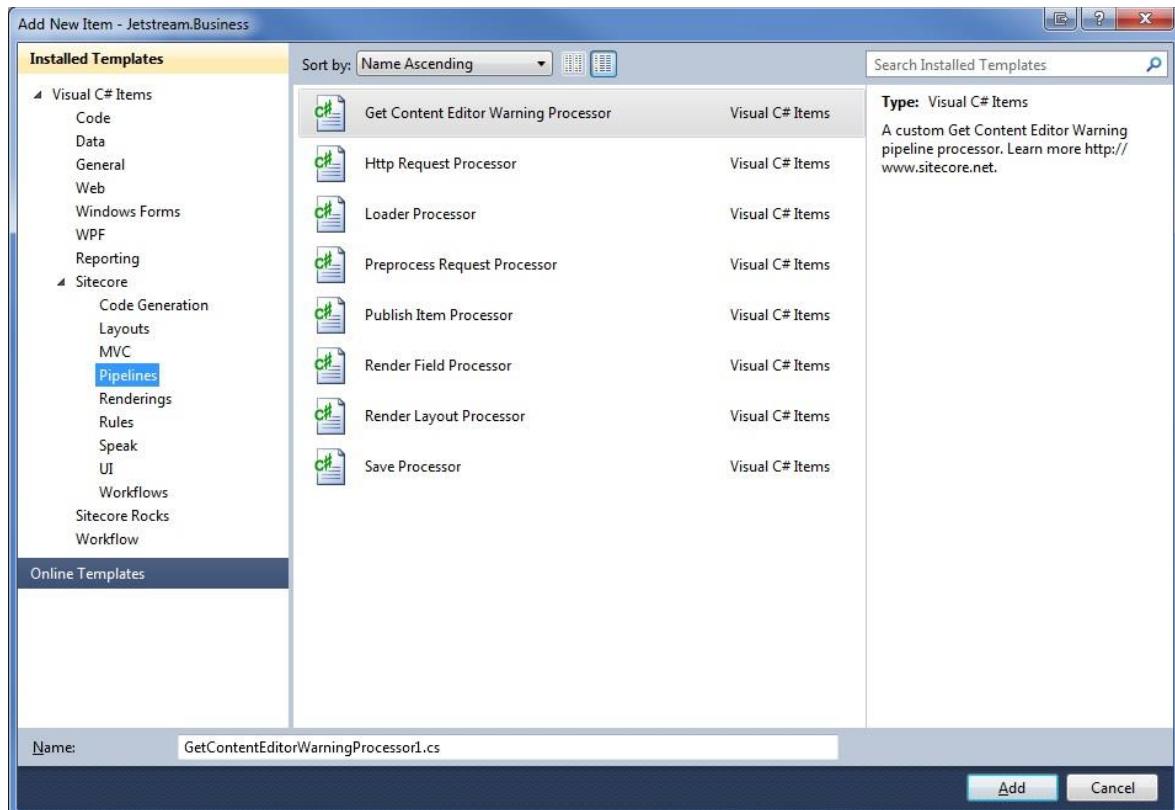
The dialog lists all the placeholders for the selected layout. Simply select the desired placeholder and we are on our way. No more remembering which placeholders are in which layout!

As we've learned, Sitecore Rocks can streamline the process of managing presentation details on a template and make it less prone to errors.

10: 28 Days of Sitecore Rocks: Pipeline Creation

By [Trevor Campbell](#), February 10, 2013

If you've ever created a custom Sitecore pipeline you know that there are a couple components needed for it to properly function. Today we will see how Sitecore Rocks streamlines the creation of new pipelines. To create a new pipeline using Sitecore Rocks, select the location in Visual Studios Solution Explorer where you would like the class created. Next right click and add a new item the same way we did to add a new layout/sublayout. Under the Sitecore section of installed templates is an entire sub-section dedicated to just pipelines!



Select the type of pipeline you are trying to create, give the class file a name and click "Add."

Here are the available pipeline templates:

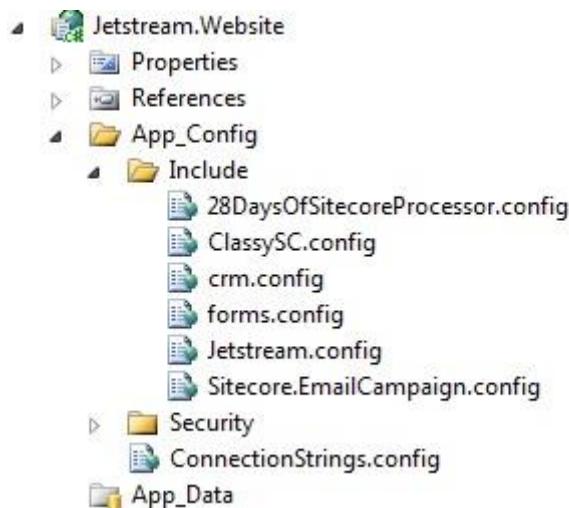
- Get Content Editor Warning Processor
- Http Request Processor
- Loader Processor
- Preprocess Request Processor
- Publish Item Processor
- Render Field Processor
- Render Layout Processor
- Save Processor

If you're looking for more information about Sitecore pipelines, [John West](#) has a [great article](#) all about them.

I've decided to create a new Save Processor in this example. You will notice that the pipeline template creates a new class with the "Process" method defined and the appropriate using statement included.

```
Start Page 28DaysOfSitecoreProcessor.cs* 
using Sitecore;
namespace Jetstream.Business.Pipelines
{
    using Sitecore.Pipelines.Save;
    // TODO: \App_Config\include\28DaysOfSitecoreProcessor.config created automatically when creating _28DaysOfSitecoreProcessor class.
    public class _28DaysOfSitecoreProcessor
    {
        public void Process([NotNull] SaveArgs args)
        {
        }
    }
}
```

You might also notice that the template creates a configuration file in the "App_Config/Include" folder with the same name as our new pipeline.



The configuration file contains an XML transform that inserts our new pipeline processor into the Sitecore saveUI pipeline.

```
Start Page 28DaysOfSitecoreProcessor.cs* 28DaysOfSitecoreProcessor.config* 
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">
  <sitecore>
    <processors>
      <saveUI>
        <processor type="Jetstream.Business.Pipelines._28DaysOfSitecoreProcessor, Jetstream.Business" patch:before="processor[@type='Sitecore.Pipelines.Save.Save, Sitecore.Kernel']" />
      </saveUI>
    </processors>
  </sitecore>
</configuration>
```

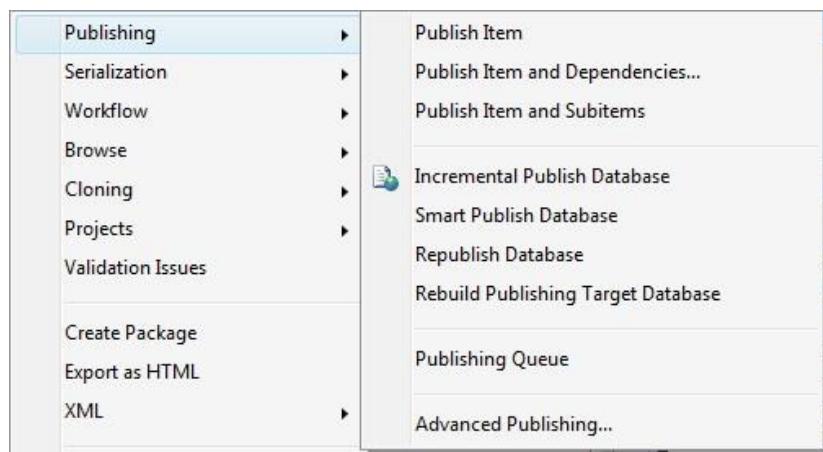
All this was done in one step using the Sitecore Rocks template for creating new pipelines; a simple but very convenient feature!

11: 28 Days of Sitecore Rocks: Publishing

By [Trevor Campbell](#), February 11, 2013

Ever forget to publish an item or better yet forget to publish a dependant item like a template? I've been working with [Sitecore](#) since 2005 and I still occasionally get tripped up by not publishing. I even went to the extent of printing out a Sitecore troubleshooting cheat sheet in which #1 was "Did you publish?" Sitecore Rocks feels your pain and has a couple of features to help.

Let's start by exploring the publishing context menu. Right-click on an item and under the "Tools" you will see the "Publishing" sub-menu.



The top region contains the following publishing options for the current item.

- Publish Item
- Publish Item and Dependencies
- Publish Item and Subitems

The middle region contains the following publishing options for a site publish.

- Incremental Publish Database
- Smart Publish Database
- Republish Database
- Rebuild Publishing Target Database

The bottom region contains a single publishing option.

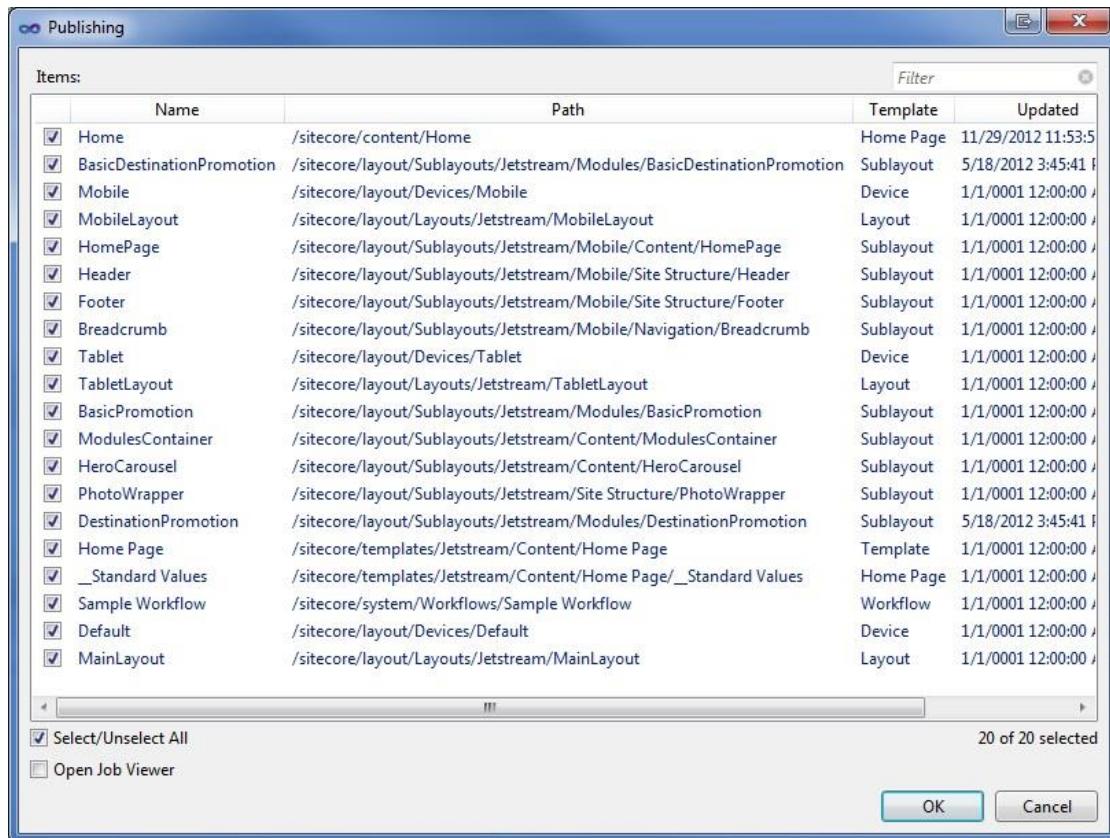
- Advanced Publishing

Let's start by kicking off a "Publish Item." A dialog box appears to indicate you have published the item.



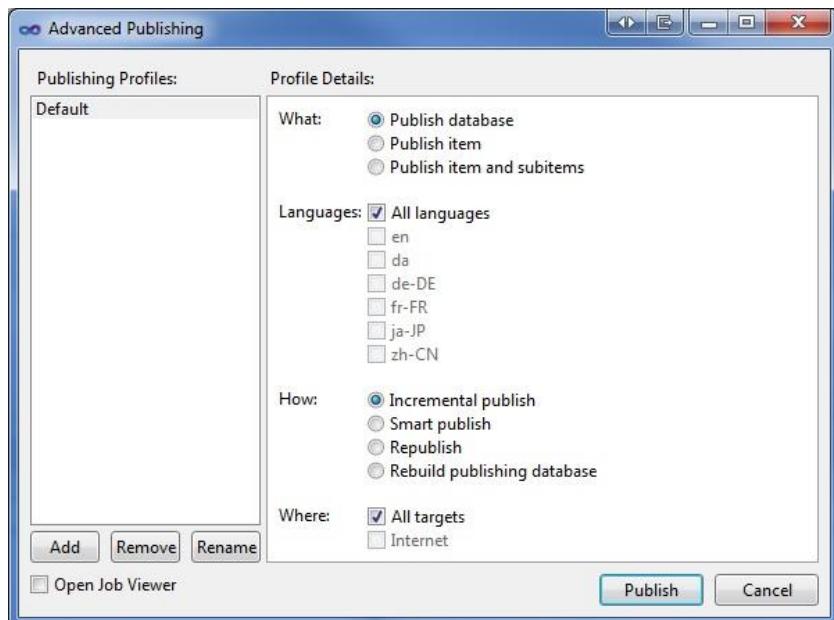
You can select to open the Job Viewer and/or to hide this dialog in the future (more on the Job Viewer tomorrow).

Next let's try a "Publish Item and Dependencies." As the name suggests, the Publish Item and Dependencies will publish the current item and all the items that make up this item. This includes the Sitecore template, layout, sublayouts, devices, media items, etc. The following dialog opens when performing a publish item and dependencies to illustrate.



If you only used Sitecore Rocks for this one feature alone it would be well worth the investment. Of course you would be missing out on so many other features as we have learned.

Lastly let's take a look at the Advanced Publishing feature. When you select this option you are presented with the following dialog.



The Advanced Publishing feature lets you create publishing profiles to quickly execute a publish with a specific configuration. A publishing profile is a collection of publish options such as publishing scope, languages, type and location.

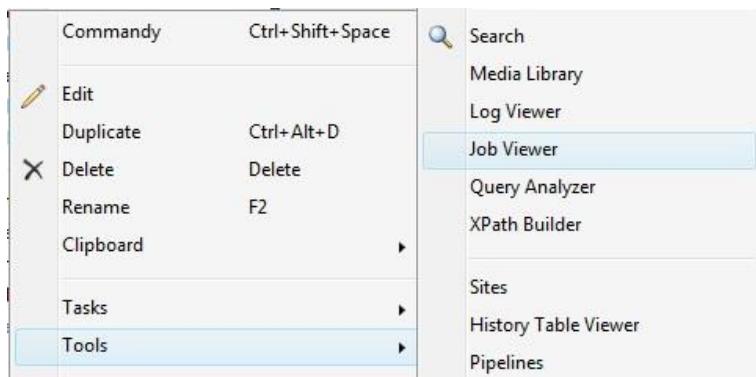
Given the number of times a developer will publish items while developing a site you can imagine the time savings from some of the Sitecore Rocks publishing features.

12: 28 Days of Sitecore Rocks: Job Viewer

By [Trevor Campbell](#), February 12, 2013

Have you ever tried to perform a quick publish of an item and sat at the publishing progress screen waiting for the job to begin? Not knowing that a colleague or a client kicked off a full site republish just moments before. This has happened numerous times to me and I'm guessing I'm not the only one. Let's see how Sitecore Rocks can provide insight into currently running background jobs using the Job Viewer.

To open the Job Viewer, right-click anywhere in the Sitecore Explorer, navigate under the Tools menu and click the Job Viewer item.



The Job Viewer can also be opened from the publishing dialog screen as we mentioned yesterday.



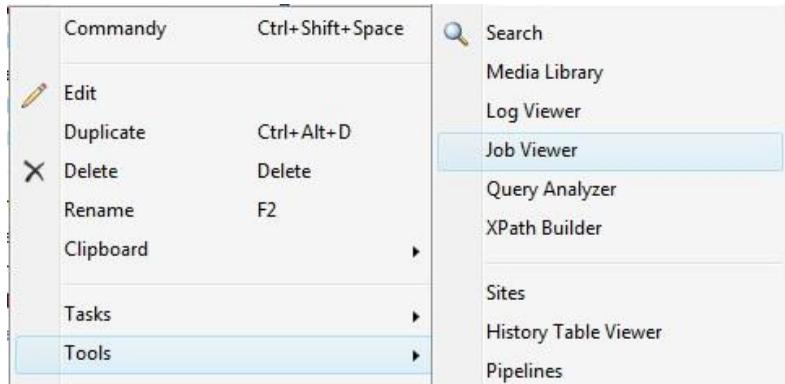
With the Job Viewer open we can see exactly which background jobs have run and which are currently running. In addition to seeing which state a background job is in, we can also see the type, category, queue time, progress and failed. As with many features in Sitecore Rocks we can filter the results to quickly find exactly which jobs we are looking for. We also have the ability to stop and start the updating of the queue if we need to investigate a currently running set of jobs.

Start Page Job Viewer - Jetstream X

Filter 2/7/2013 11:50:45 AM

Name	Category	State	Queue Time	Progress	Failed
Publish	PublishManager	Running	2/7/2013 4:45:49 PM	N/A	False
Publish to 'web'	publish	Running	2/7/2013 4:50:09 PM	2,503	False
Publish to 'web'	publish	Finished	2/7/2013 4:49:57 PM	5,193	False
Publish to 'web'	publish	Finished	2/7/2013 4:48:58 PM	5,193	False
Sitecore.Tasks.DatabaseAgent	schedule	Finished	2/7/2013 4:50:07 PM	N/A	False
UpdateIndex_core	Indexing	Finished	2/7/2013 4:49:59 PM	N/A	False
UpdateIndex_filesystem	Indexing	Finished	2/7/2013 4:49:59 PM	N/A	False
UpdateIndex_master	Indexing	Finished	2/7/2013 4:49:59 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:49:44 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:50:16 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:50:15 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:50:27 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:50:40 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:50:28 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:49:47 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:50:41 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:49:48 PM	N/A	False
UpdateIndex_web	Indexing	Finished	2/7/2013 4:50:12 PM	N/A	False

The Job Viewer has several options if you right-click inside the interface and open the context menu. We have the ability to save the job viewer queue to a file and also have the ability to change the update speed of the Job Viewer itself.



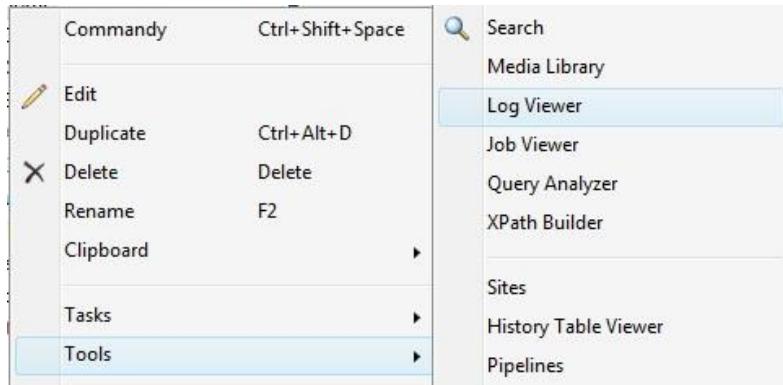
If you've ever wanted to see exactly what background jobs are currently running then the Sitecore Rocks Job Viewer can help. I believe you will find it a helpful tool in the Sitecore Rocks arsenal of features.

13: 28 Days of Sitecore Rocks: Log Viewer

By [Trevor Campbell](#), February 13, 2013

An essential part of working with Sitecore is the ability to interpret the log files in order to troubleshoot issues. The log files are typically the starting point in any Sitecore investigation. The Log Viewer in Sitecore Rocks gives us the ability to work with multiple Sitecore log files and mine the data all from the comforts of Visual Studio.

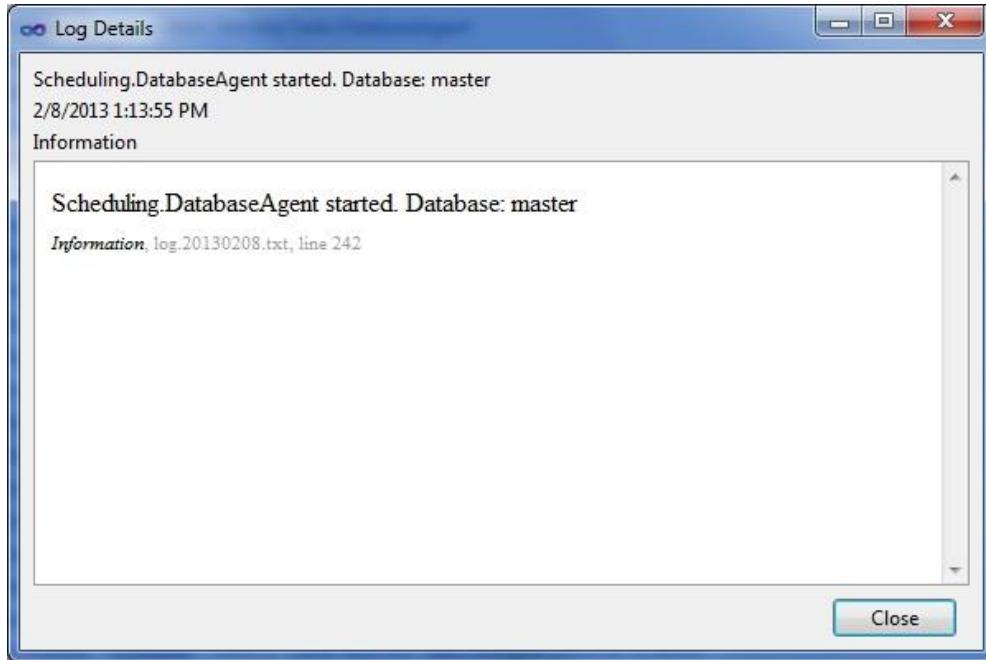
Let's start by opening the Log Viewer. To open the Log Viewer, right-click anywhere in the Sitecore Explorer, navigate under the Tools menu and click the Log Viewer item.



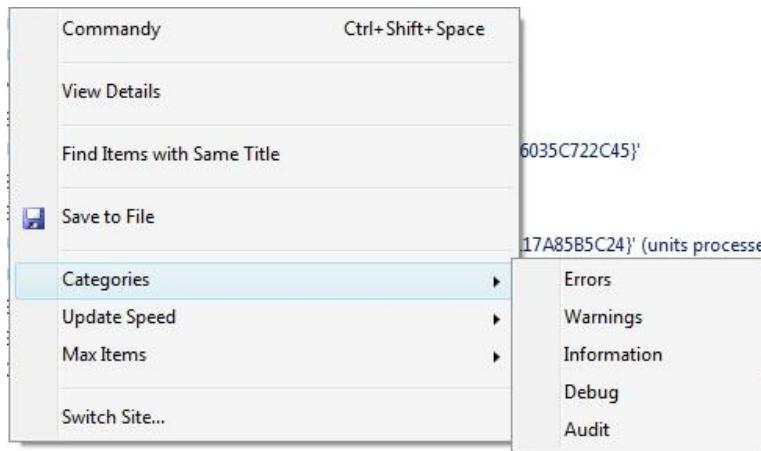
With the Log Viewer open we can see a running stream of the most current log events. Much like the Job Viewer, we have the ability to start and stop the stream of log events. In addition to that we also have the ability to filter the stream using the Include and Exclude fields.

A screenshot of the Log Viewer interface. The window title is "Log Viewer - Jetstream". At the top, there are two filter inputs: "Include: Filter" and "Exclude: Filter", both currently set to "Filter". The timestamp in the top right corner is "2/8/2013 1:14:17 PM". The main area is a table with three columns: Date, Category, and Title. The table lists numerous log entries, all of which are "Information" level logs from the "Scheduler" category. The log entries describe various tasks being added or started by Sitecore agents, such as "DatabaseAgent", "WebDAVOptionsCleanupAgent", "CleanUpFDAObsleteMediaData", "Social.Connector.ScheduledTaskPerformer", "Analytics.Tasks.UpdateReportsSummaryTask", "Analytics.Tasks.SubscriptionTask", "Analytics.Tasks.EmailReportsTask", "CloneNotificationsCleanupAgent", "CounterDumpAgent", "Tasks.CleanupAgent", "PublishAgent", "HtmlCacheClearAgent", "CleanupEventQueue", "CleanupPublishQueue", "CleanupHistory", "CompactClientDataAgent", "TaskDatabaseAgent", "UrlAgent", "DatabaseAgent", and "DatabaseAgent".

To get more information on a single log event, simply double-click the item and a detailed view dialog is opened. In addition to the log message we are also supplied the log category, file and line number. This is helpful when we need to locate which log file a single log event exists in since the Log Viewer parses multiple log files at a time.



The Log Viewer has several options if you right-click inside the interface and open the context menu. If we are looking to find the frequency of a particular event we can select the "Find Items with Same Title."



We can also display log events of a specific category by navigating to the "Categories" menu and selecting the desired category. This is a great way to aggregate all of the error messages in the application spanning multiple log files.

Start Page Log Viewer - Jetstream X

Include: Filter Exclude: Filter Loading...

Date	Category	Title
2/8/2013 1:08:58 PM	Error	The service isn't online.
2/8/2013 1:08:56 PM	Error	The CRM provider couldn't be initialized: The request failed with HTTP status 417: Expectation failed.
2/8/2013 1:08:54 PM	Error	The service isn't online.
2/8/2013 1:08:55 PM	Error	The service isn't online.
2/8/2013 1:08:55 PM	Error	The service isn't online.
2/8/2013 1:08:55 PM	Error	The service isn't online.
2/7/2013 11:50:45 AM	Error	Cannot find dictionary domain for the dictionary entry '/sitecore/system/Settings/Workflow/Check'
2/7/2013 11:49:32 AM	Error	Cannot find dictionary domain for the dictionary entry '/sitecore/system/Settings/Workflow/Check'
2/7/2013 11:48:22 AM	Error	Cannot find dictionary domain for the dictionary entry '/sitecore/system/Settings/Workflow/Check'
2/7/2013 11:46:51 AM	Error	Cannot find dictionary domain for the dictionary entry '/sitecore/system/Settings/Workflow/Check'
2/7/2013 11:45:09 AM	Error	The service isn't online.
2/7/2013 11:45:07 AM	Error	The CRM provider couldn't be initialized: The request failed with HTTP status 417: Expectation failed.
2/7/2013 11:45:07 AM	Error	The service isn't online.
2/7/2013 11:45:07 AM	Error	The service isn't online.
2/7/2013 11:45:06 AM	Error	The service isn't online.
2/7/2013 11:45:06 AM	Error	The service isn't online.
2/7/2013 7:42:14 PM	Error	The service isn't online.
2/7/2013 7:42:12 PM	Error	The CRM provider couldn't be initialized: The request failed with HTTP status 417: Expectation failed.
2/7/2013 7:42:12 PM	Error	The service isn't online.
2/7/2013 7:42:11 PM	Error	The service isn't online.
2/7/2013 7:42:11 PM	Error	The service isn't online.
2/6/2013 11:22:53 AM	Error	The service isn't online.
2/6/2013 11:22:52 AM	Error	The CRM provider couldn't be initialized: The request failed with HTTP status 417: Expectation failed.
2/6/2013 11:22:52 AM	Error	The service isn't online.

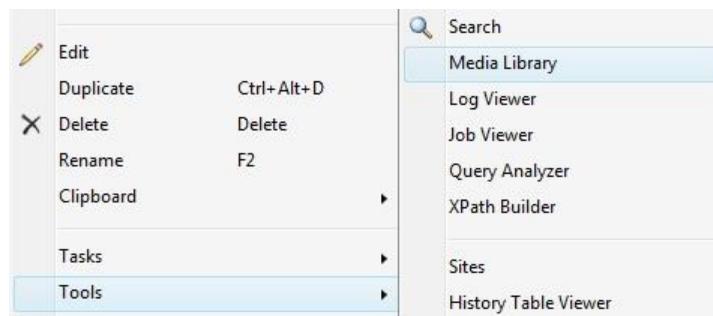
The Log Viewer in Sitecore Rocks is an extremely helpful tool for troubleshooting a Sitecore application. Couple the Log Viewer with the Job Viewer and you can see how debugging an application becomes much more streamlined. All this directly from within Visual Studio!

14: 28 Days of Sitecore Rocks: Media Library

By [Trevor Campbell](#), February 14, 2013

In today's edition of the 28 Days of Sitecore Rocks we will be discussing the Media Library. Ever try to find an item in the Sitecore Media Library, particularly if you didn't create the item? Not the easiest task in Sitecore especially if the Media Library has a large number of items. Thankfully Sitecore Rocks can help! Let's find out how the Sitecore Rocks Media Library can help locate and create new items.

To open the Media Library, right-click anywhere in the Sitecore Explorer, navigate under the Tools menu and click the Media Library item.



Now that we have the Media Library open you will notice the interface is quite different from the web interface. Similar to [Sitecore Item Buckets](#), the Sitecore Rocks interface for the Media Library takes the hierarchical structure of Media Library and flattens it. To locate items in the Media Library we use a combination of searching and filtering. To search on the items in the media library, first select which fields you would like to search on.

The options are:

- Any Searchable Field
- Title
- Keywords
- Description
- Alt Text

Once we have narrowed down which fields to search on we simply need to enter a search term and click "Go."



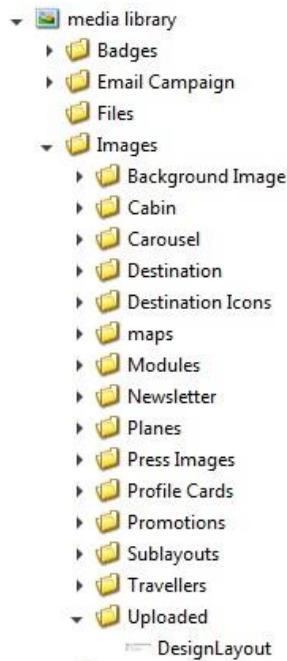
Now that we have a result set to work with, we can further refine our results by filtering. Click the "Filters" button to reveal the filterable attributes.

We can filter on the following attributes:

- Template
- Author
- Path
- Updated

Template	Author	Path	Updated
All Image (140 items) Jpeg (119 items)	All sitecore\admin (174 items) sitecore\MattH (13 items) sitecore\niketa (51 items) sitecore\valc (21 items)	All Badges (6 items) Email Campaign (8 items) Images (150 items) Jetstream Newsletter (16 items)	All 2010 (1 item) 2011 (39 items) 2012 (219 items)

Another great feature in the Sitecore Rocks Media Library is the ability to drag and drop files. To upload a new file into the Media Library, drag it from your local Windows Explorer and drop it directly into the center of the Media Library. The one item to note is that when using drag and drop, the file will be created in the Sitecore Media Library under "Images/Uploaded."



From here we can drag it to a new location in the Media Library tree.

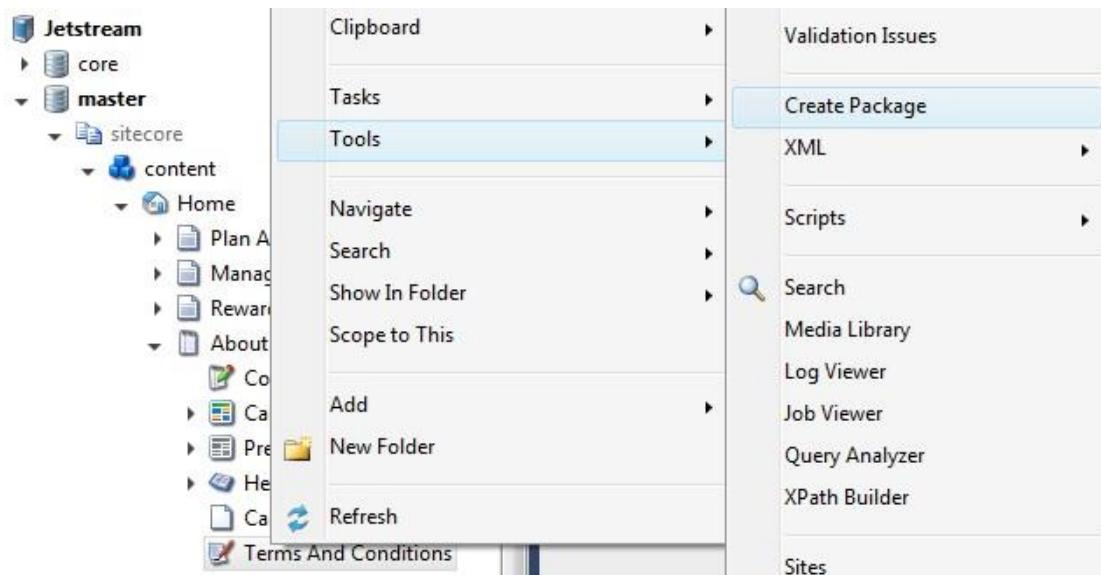
As we have seen in this edition of the 28 Days of Sitecore Rocks, the Media Library in Sitecore Rocks can help locate items using searching and filtering. The Sitecore Media Library can also help create media items with drag and drop.

15: 28 Days of Sitecore Rocks: Package Creation

By [Trevor Campbell](#), February 15, 2013

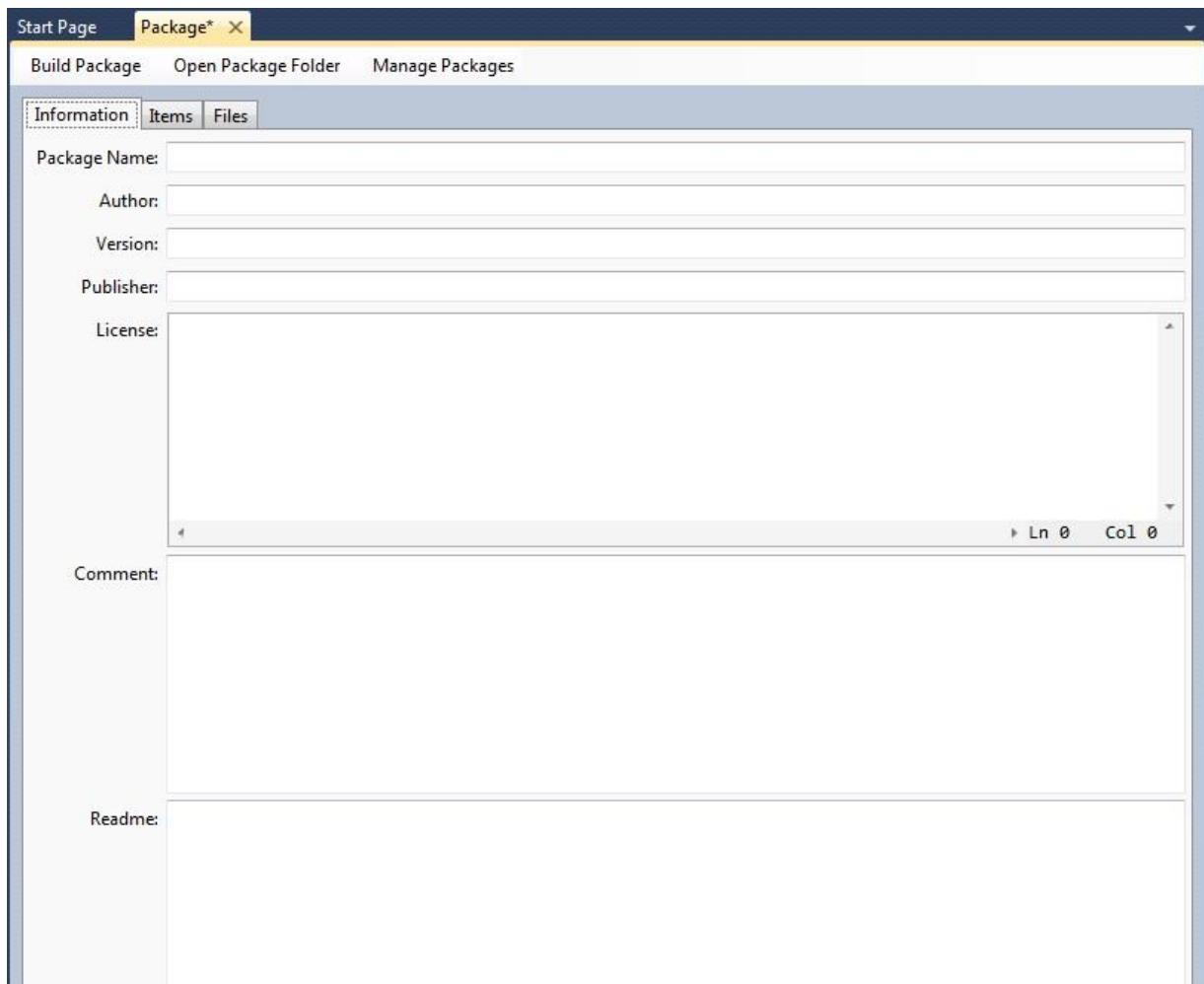
Creating Sitecore packages is a task that all seasoned Sitecore developers are familiar with. Unfortunately we are also familiar with how easy it is to forget an item or a file required for a feature to work correctly. Today we will be discussing how Sitecore Rocks can simplify and streamline the package creation process.

To start the process of creating a Sitecore package let's start by right clicking a Sitecore item that we want to include in a package. In the context menu navigate under the "Tools" menu and select "Create Package."

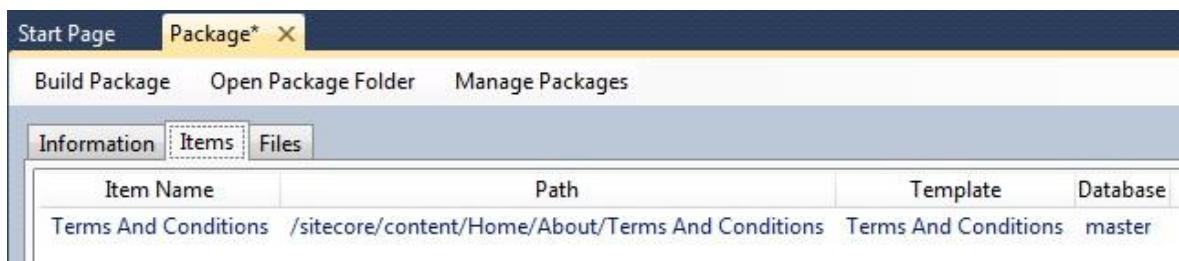


This will open the Sitecore Rocks package creation tool. You will notice that there are three tabs: "Information", "Items", and "Files."

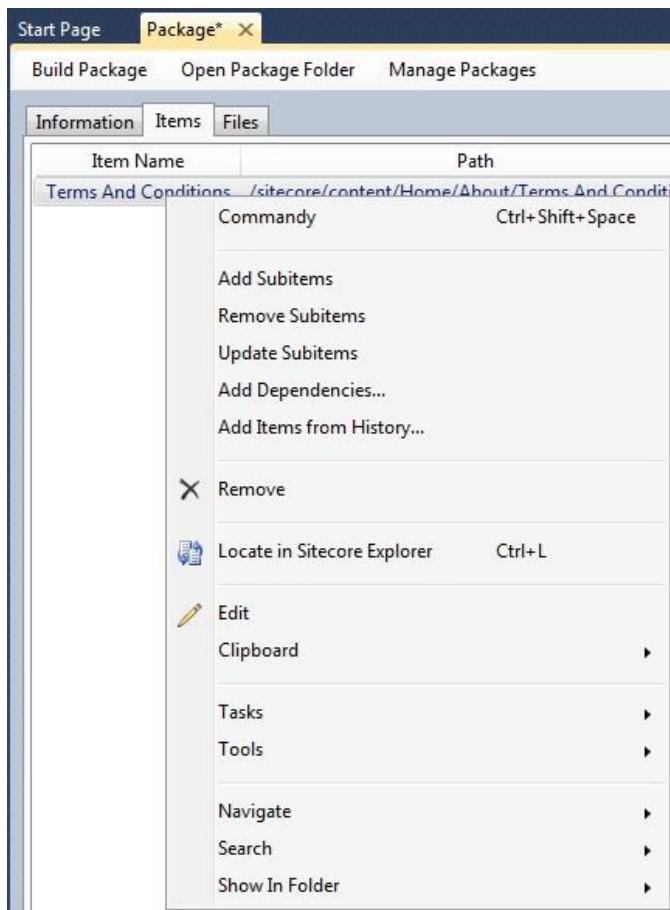
Let's start with the Information tab. If you are familiar with creating packages in the Sitecore web interface then these are fields you have worked with in the past. If you haven't worked with Sitecore packages before, these fields are meta data associated with the package.



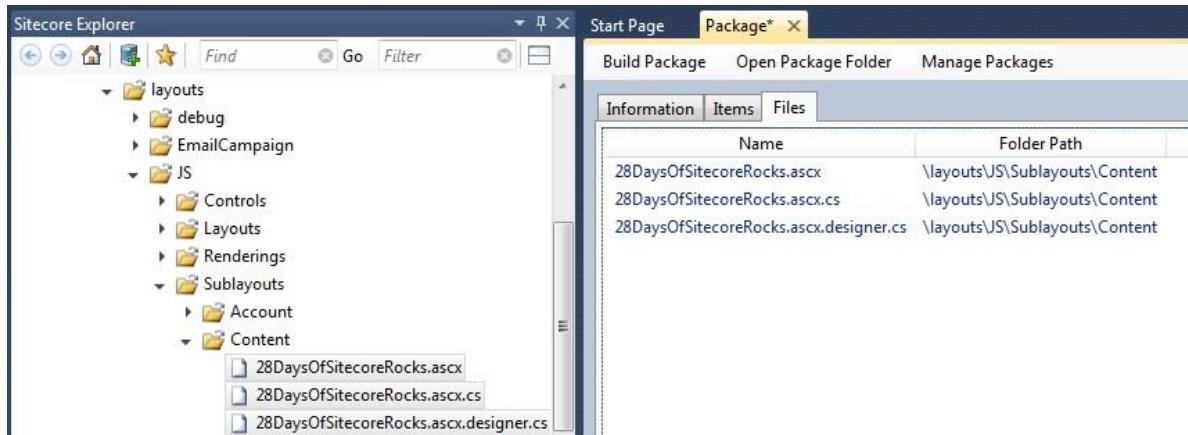
The “Items” tab is where we specify which Sitecore items are to be added to the package. I started creating this package by right clicking on the “Terms and Conditions” item in the content tree which is why we already have an item pre-populated in this package. To add more items to the package simply drag and drop them from the Sitecore Explorer into the package.



Where do we specify to add subitems you ask? Right click the item in the package to open the context menu. Here is where we specify to add the subitems of a particular item. You might also notice the “Add Dependencies” option in the context menu. Similar to the “[Publish Item with Dependencies](#),” this option will include all the dependant Sitecore items needed to deploy. An extremely helpful feature!



The last tab is the "Files" tab. As the name suggests, this is where we add the files to be included in the package. To add files to the package simply drag and drop them from the Sitecore Explorer to the package.



Now that we have added our Sitecore items and files to a package we can build the package to receive its zip file. Clicking on the "Build Package" button at the top of the package will generate the zip file to be installed on another Sitecore instance.

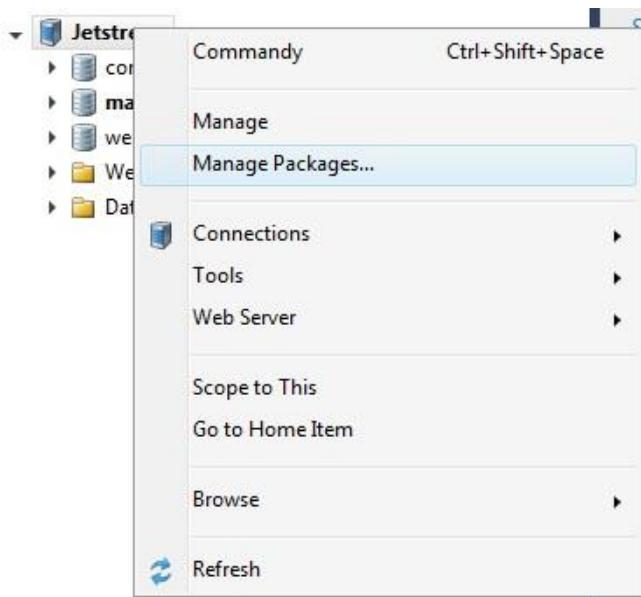
As we've seen, using Sitecore Rocks to create packages can help simplify the package creation process all from within Visual Studio.

16: 28 Days of Sitecore Rocks: Package Management - Part 1

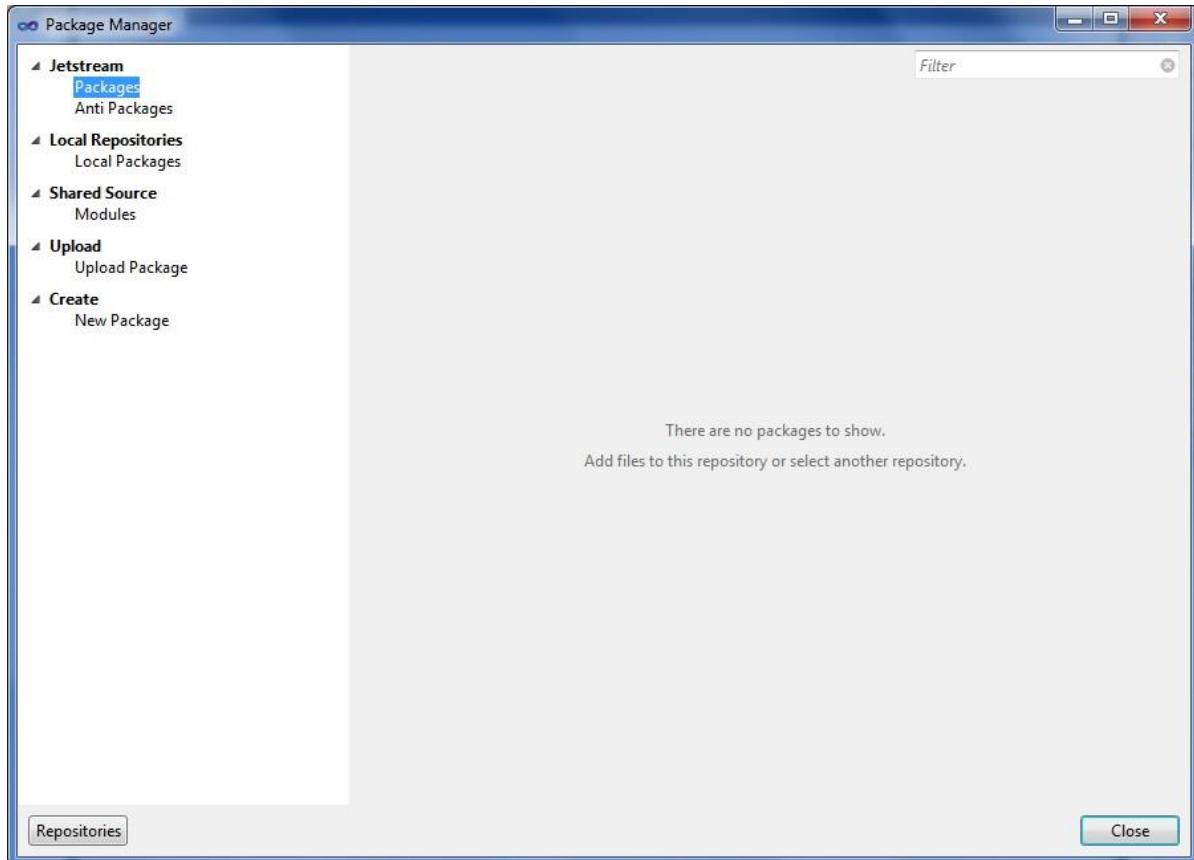
By [Trevor Campbell](#), February 16, 2013

Sitecore Rocks adds some extremely useful features for working with Sitecore packages. Yesterday we created a new Sitecore package using Sitecore Rocks. Today we will install the package we created yesterday and show how the Sitecore Rocks Package Manager can provide more insight into the package installation process.

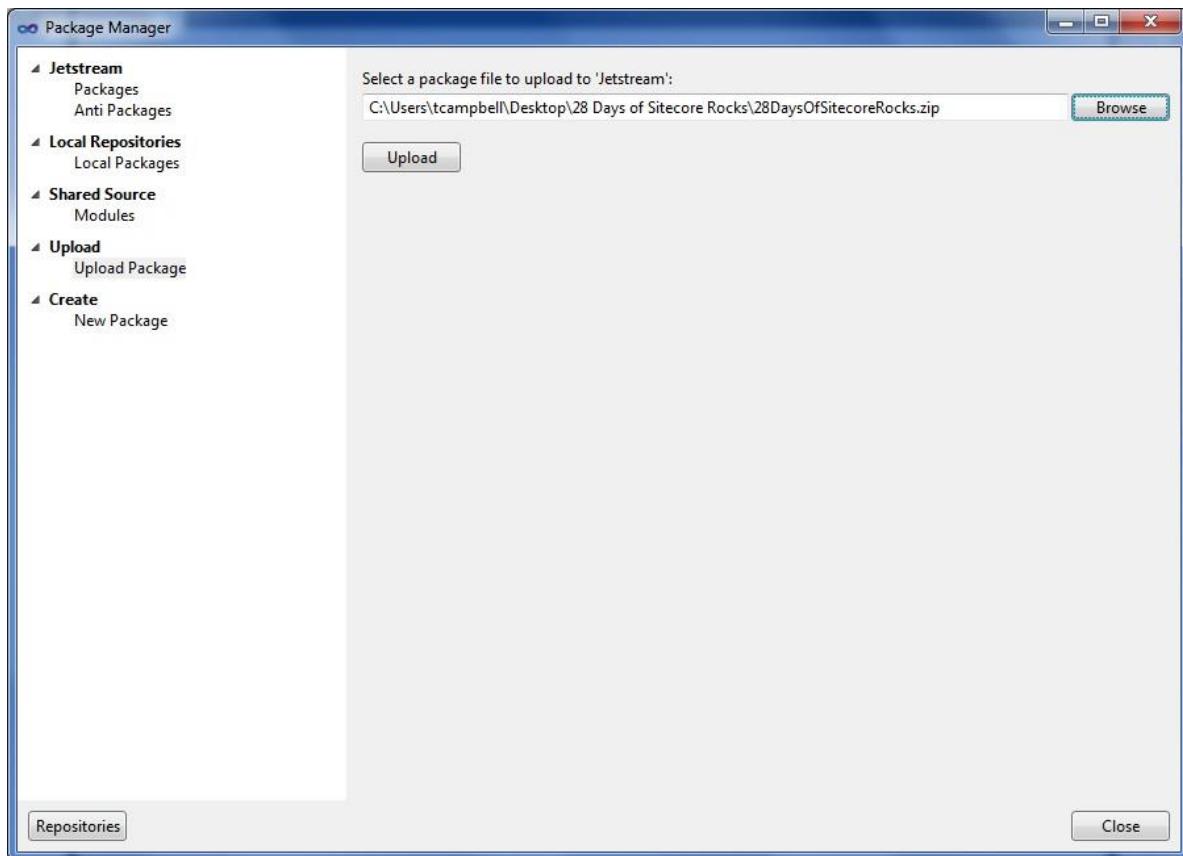
Let's start by opening the Sitecore Rocks Package Manager by right clicking the Sitecore instance, in this case the Jetstream node, in the Sitecore Explorer and selecting the "Manage Packages" menu item.



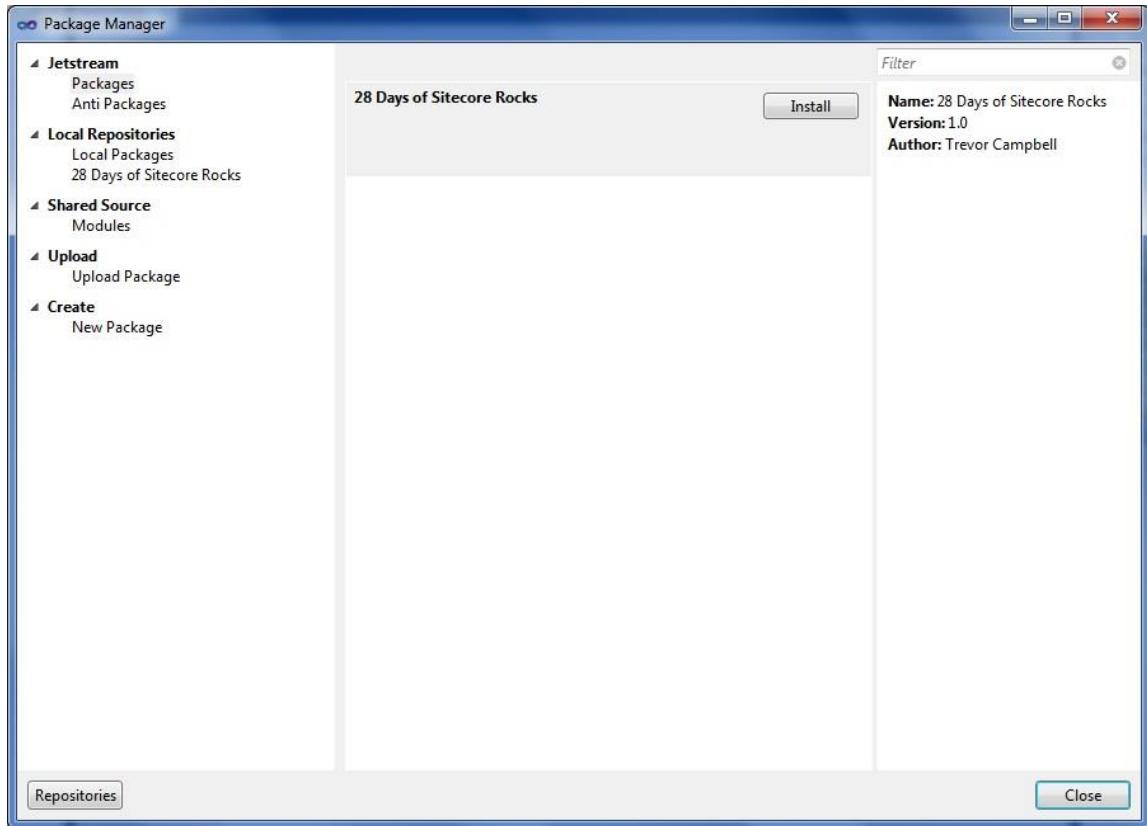
Now that we have the Package Manager open you will notice there are five sections on the left column. The first section is the current instance's collection of packages and anti-packages. You will notice that we currently don't have any packages to install.



Let's start by uploading the package we created yesterday. To upload a new package, select the "Upload Package" option on the left column under the "Upload" section. Clicking the "Browse" button will open a local file explorer window to select a package locally to upload. After locating the package for upload, simply click the "Upload" button.

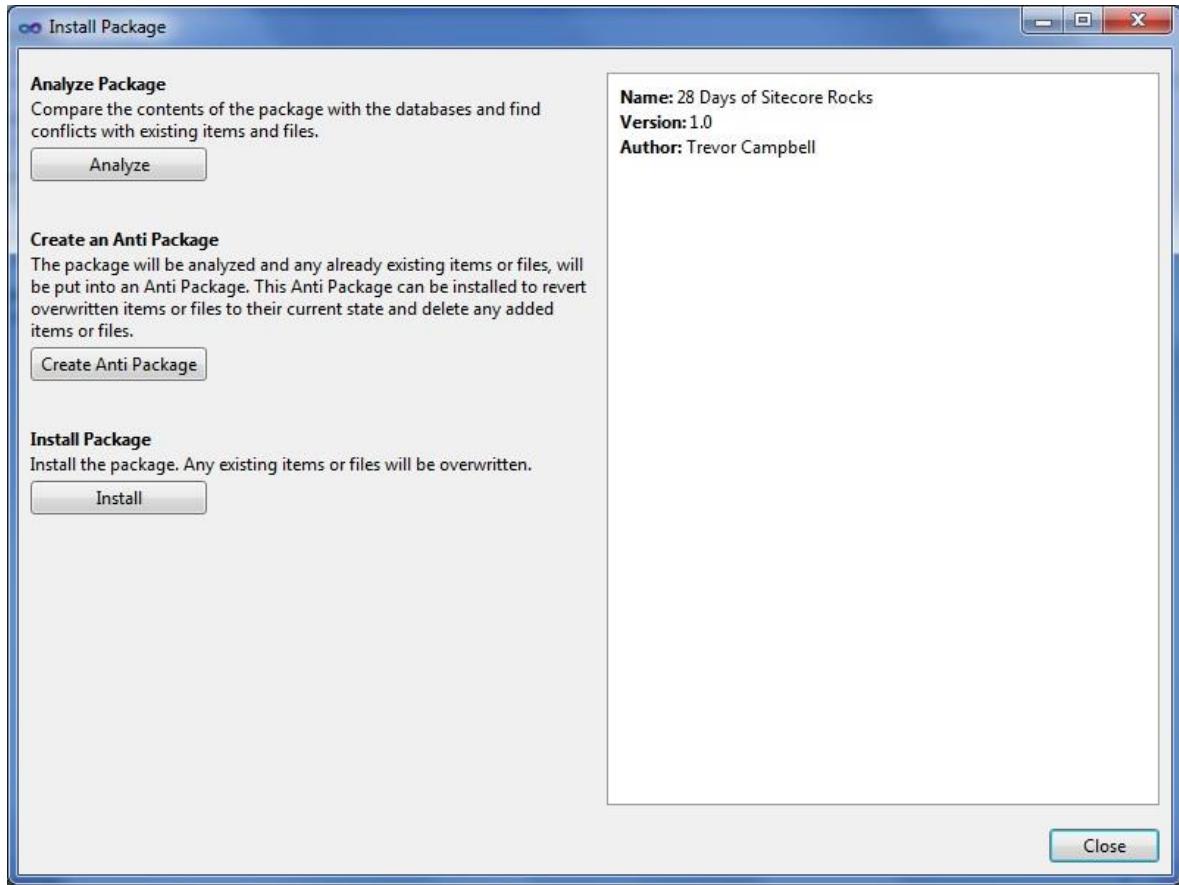


The package will be added to the Jetstream instance packages collection and display the meta data associated with that package on the right column. To start the package installation process, click the "Install" button next to the package to install.

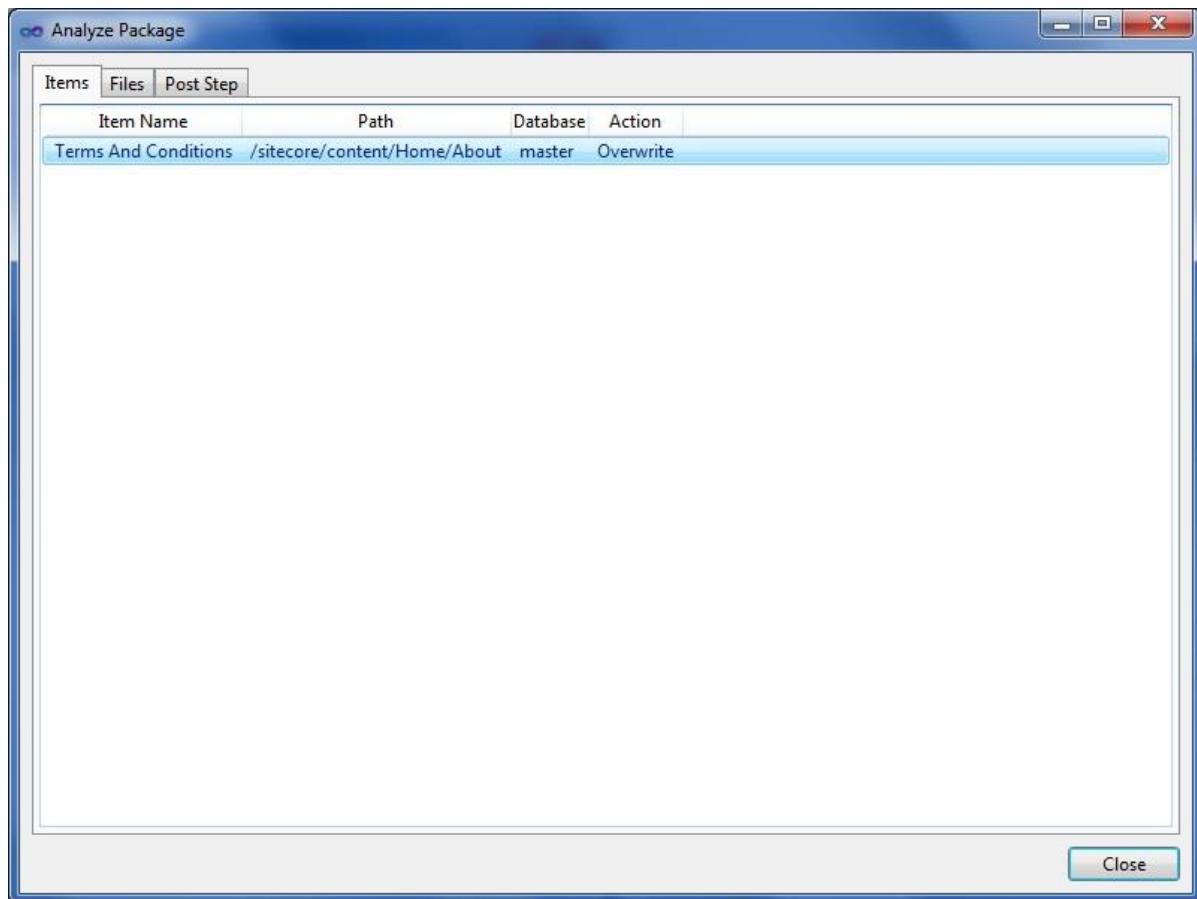


Now that the package installer is open we have three options.

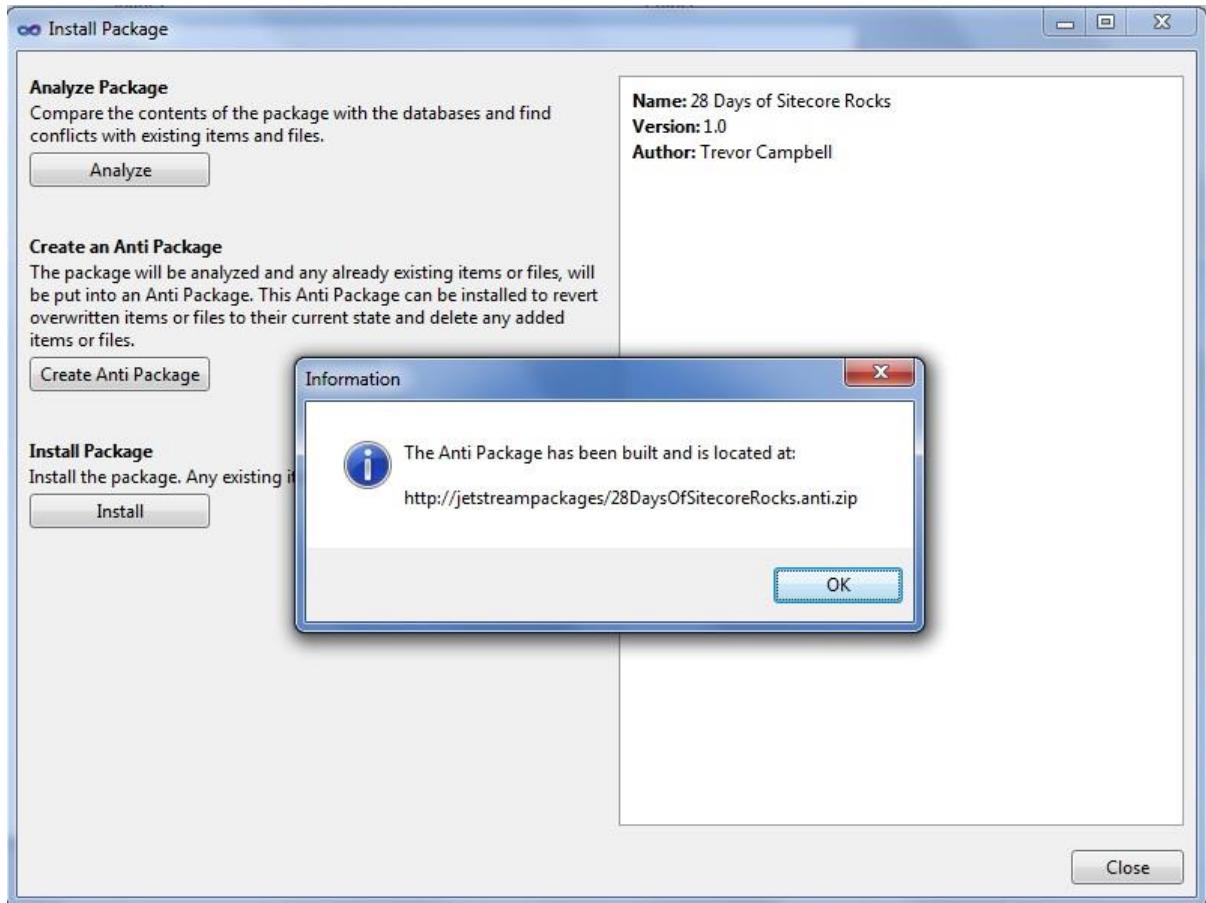
- **Analyze Package** - Compare the items and files with the installation target to help identify potential conflicts before installing the package.
- **Create an Anti Package** - A package's equivalent of a backup.
- **Install Package** - Installs the package.



Let's start by clicking the "Analyze" button to see if we have any conflicts when installing the package. The Analyze Package dialog has three tabs, "Items", "Files" and "Post Step." In our simple scenario you see that we have a single item in the package and it will be overwritten when installed. This is because I am installing the package on the same Sitecore instance that I created the package on.



Next let's create an Anti Package to safeguard our package installation process. To do so, simply click the "Create Anti Package" button and an informational message will appear with the location of the Anti Package.



Lastly, let's install the package by clicking the "Install" button to complete the package installation process.

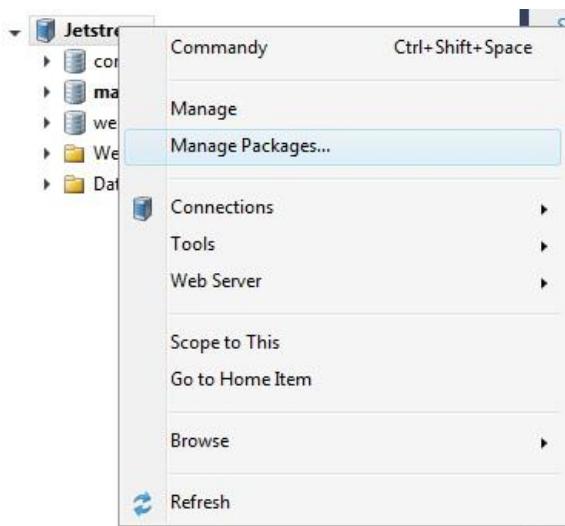
With the addition of package analysis and the ability to create anti packages we can see how using Sitecore Rocks to install packages can be very beneficial.

17: 28 Days of Sitecore Rocks: Package Management - Part 2

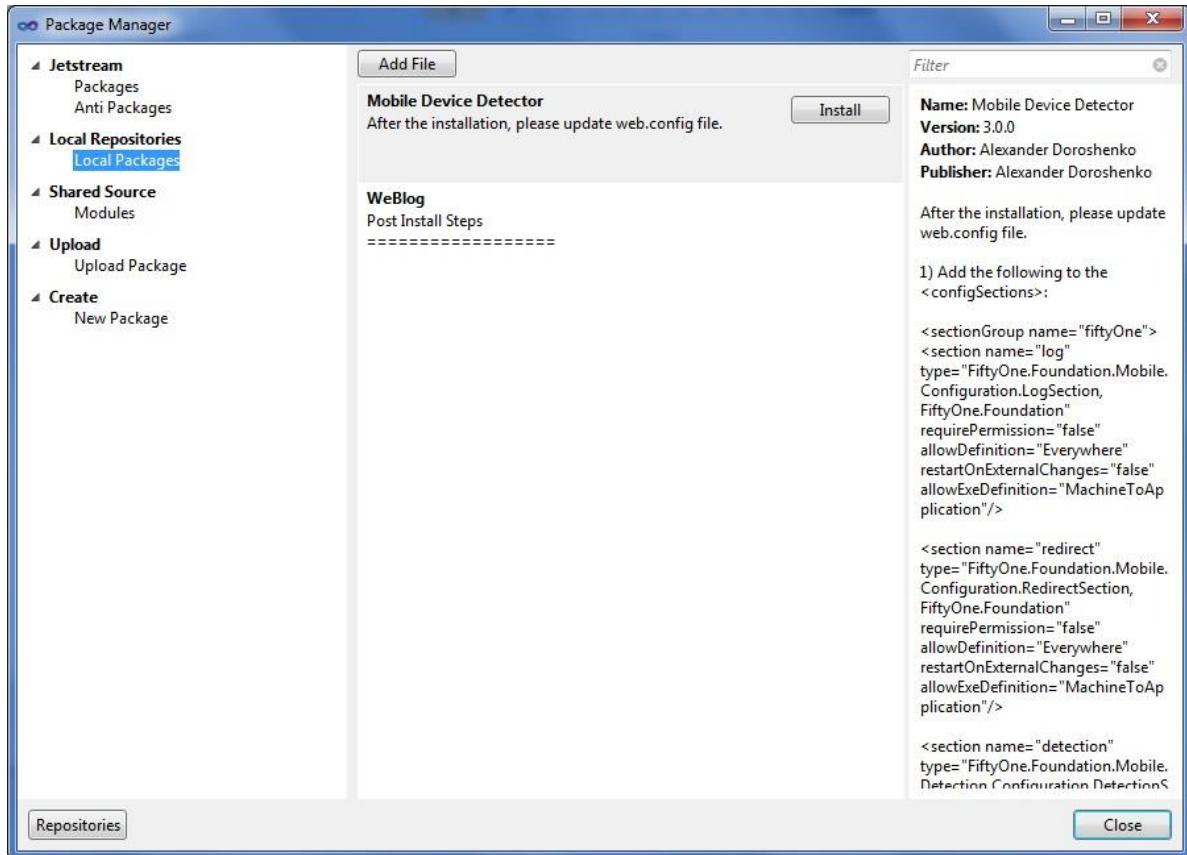
By [Trevor Campbell](#), February 17, 2013

Today we will wrap up our discussion of the Sitecore Rocks Package Manager by covering package repositories and shared source modules. Whether you are building out a local repository of Sitecore modules or downloading from the Sitecore Marketplace, Sitecore Rocks can help you manage your collection of Sitecore modules.

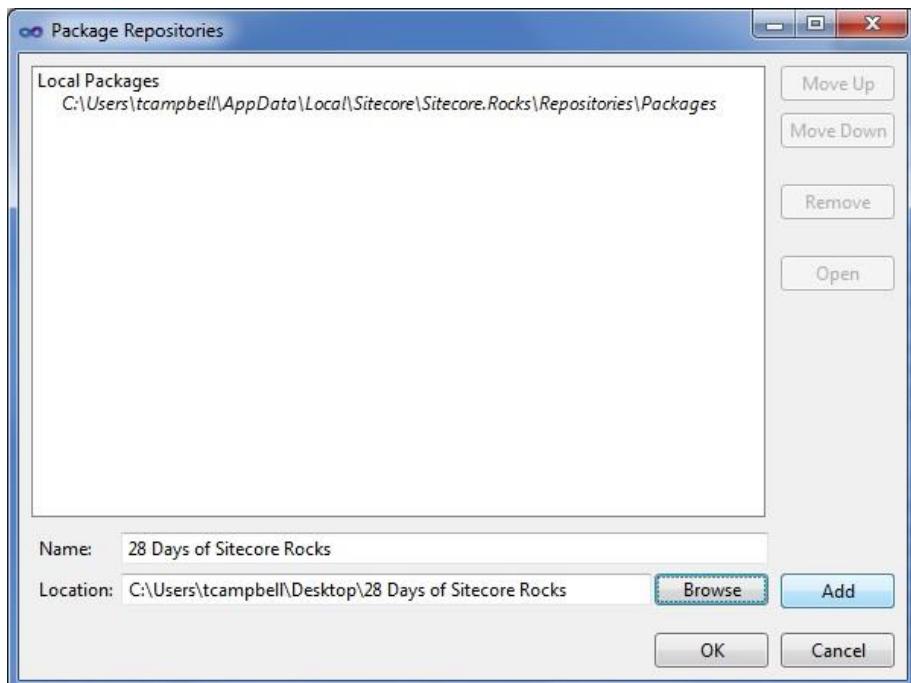
If we don't already have the package manager open, right click the Sitecore instance, in this case the Jetstream node, in the Sitecore Explorer and select the "Manage Packages" menu item.



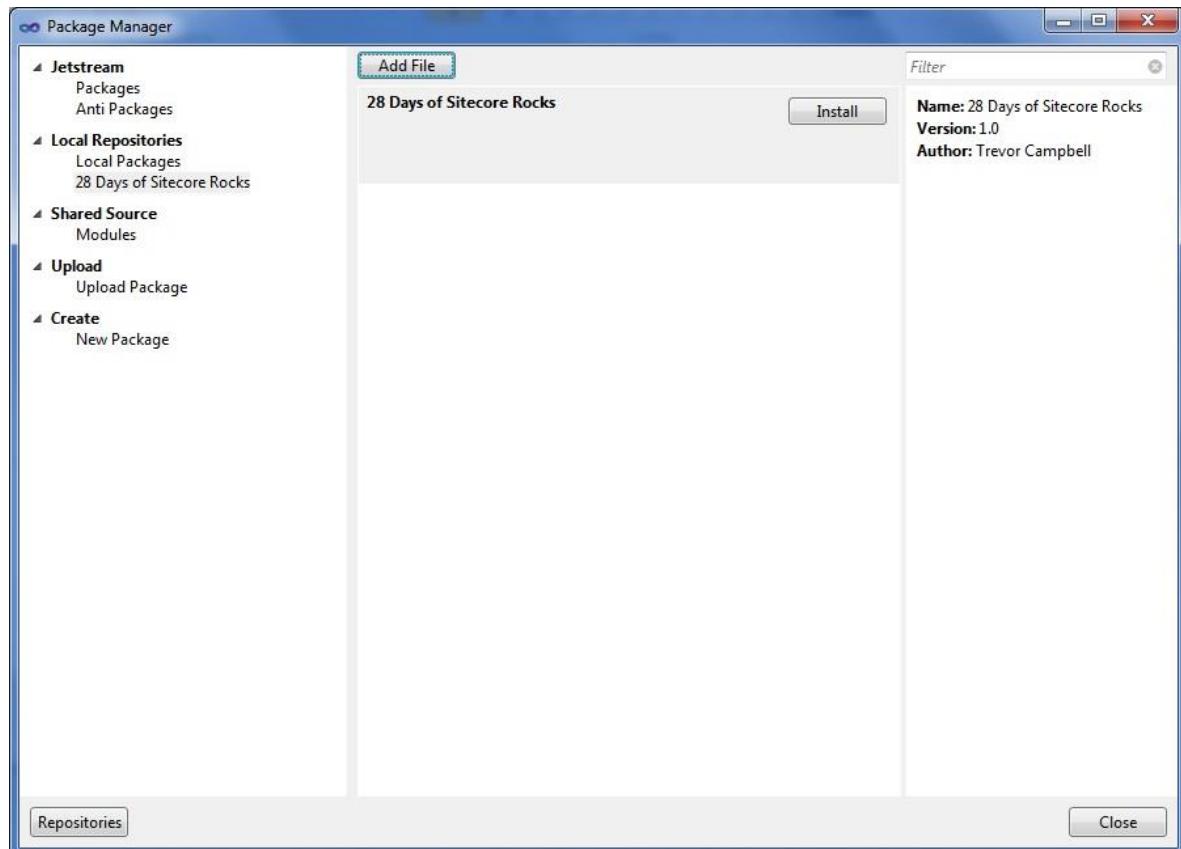
In the Package Manager, select the "Local Packages" item under "Local Repositories" in the left column. When you first navigate to the local packages repository, this repository will be empty. I've added a few packages from the [Sitecore Marketplace](#) by clicking the "Add File" button and uploading the Sitecore packages. From here we can start the package installation process outlined yesterday.



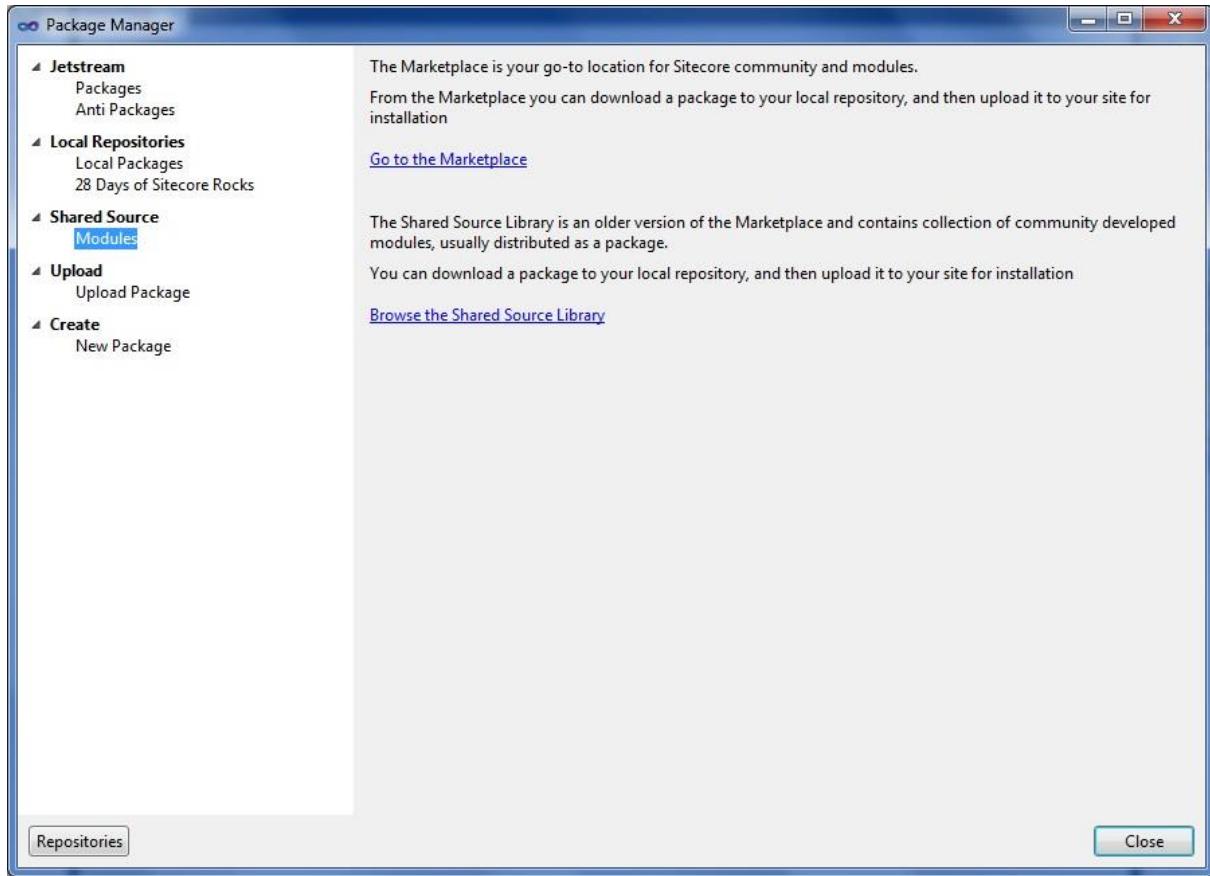
We can also add new repositories by clicking the "Repositories" button at the bottom left corner of the Package Manager. To add a new repository, specify a name and location then click the "Add" button. The location can be a local path or shared network location. A shared network location can be used to aggregate a company repository that can be used by multiple developers for multiple Sitecore instances.



Once we have added our new repository, it will appear under the “Local Repositories” section in the left column.



Under the Shared Source section of the Package Manager you will find information for the Sitecore Marketplace. The Sitecore Marketplace is a great place to find modules developed by other Sitecore developers that want to share their work.



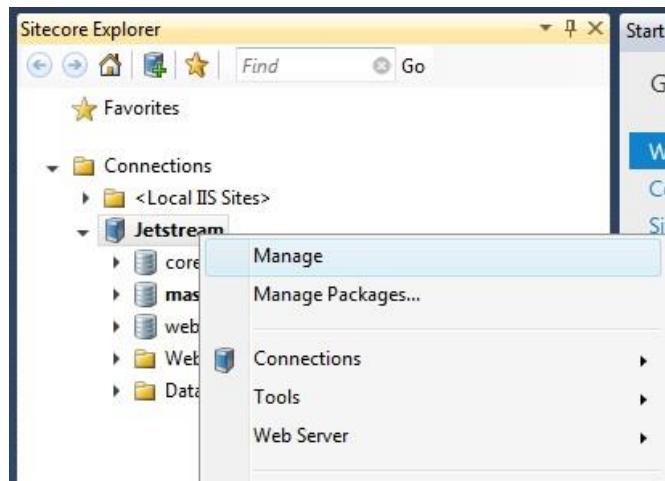
As we have seen over the past two days, the Sitecore Rocks Package Manager is a powerful tool for managing and installing Sitecore modules. Whether you are simply installing a single Sitecore package or building out an entire repository of modules, Sitecore Rocks can help.

18: 28 Days of Sitecore Rocks: Manage - Part 1

By [Trevor Campbell](#), February 18, 2013

Over the next several days we will be diving into the features and functionality of the Sitecore Rocks Manage tool. The Manage tool is a collection of interfaces designed around managing different aspects of a Sitecore web application. In today's post I will be covering the Sites interface which is a visual representation of the sites node in the web.config.

To open the Sitecore Rocks Manage tool, right click the Sitecore instance, in this case the Jetstream node, in the Sitecore Explorer and select the "Manage" menu item.



With the Sitecore Rocks Manage tool open you will notice the different tabs for each interface. The Sites interface is the first tab in the Manage collection. As Sitecore developers, we are all very familiar with the "sites" configuration element in the web.config and this is simply a visual interface for the sites collection. You will notice if you select one of the sites in the top section of the Sites interface that all the properties for that site are displayed in the bottom section. We can use this interface to validate all the different configuration options of an individual site. Also, if you select a single property, you will get some help text at the bottom of the screen describing the property. This is very helpful if you are familiar with a particular property.

Sites Pipelines Commands Temp Files Caches Indexes Statistics Validation

Jetstream

Index	Name	Host	Database	Domain
3	admin			sitecore
4	service			
5	modules_shell		core	sitecore
6	modules_website		web	extranet
7	Alternate	m.Jetstream	web	extranet
8	website		web	extranet
9	scheduler			sitecore
10	system			sitecore
11	publisher			sitecore

Properties:

Caching

Cache Html	<input checked="" type="checkbox"/>
Html Cache Size	10MB
Media Cache Path	
Registry Cache Size	0
View State Cache Size	0
XSL Cache Size	5MB

Configuration

Disable Client Data	<input type="checkbox"/>
Disable XML Controls	<input type="checkbox"/>
Enable Analytics	<input type="checkbox"/>
Enable Preview	<input checked="" type="checkbox"/>
Enable WebEdit	<input checked="" type="checkbox"/>
Enable Workflow	<input type="checkbox"/>
Mode	<input type="checkbox"/>

Html Cache Size
The size of the html cache. Specify the value in bytes or append the value with KB, MB or GB

Using the Sites interface we can also create a new site by right clicking to open the context menu and selecting the "Create New Site" item.

Sites Pipelines Commands Temp Files Caches Indexes Statistics Validation

Jetstream

Index	Name	Host	Database	Domain
3	admin			sitecore
4	service			
5	modules_shell		core	sitecore
6	modules_website		web	extranet
7	Alternate	m.Jetstream	web	extranet
8	website			
9	scheduler			
10	system			
11	publisher			

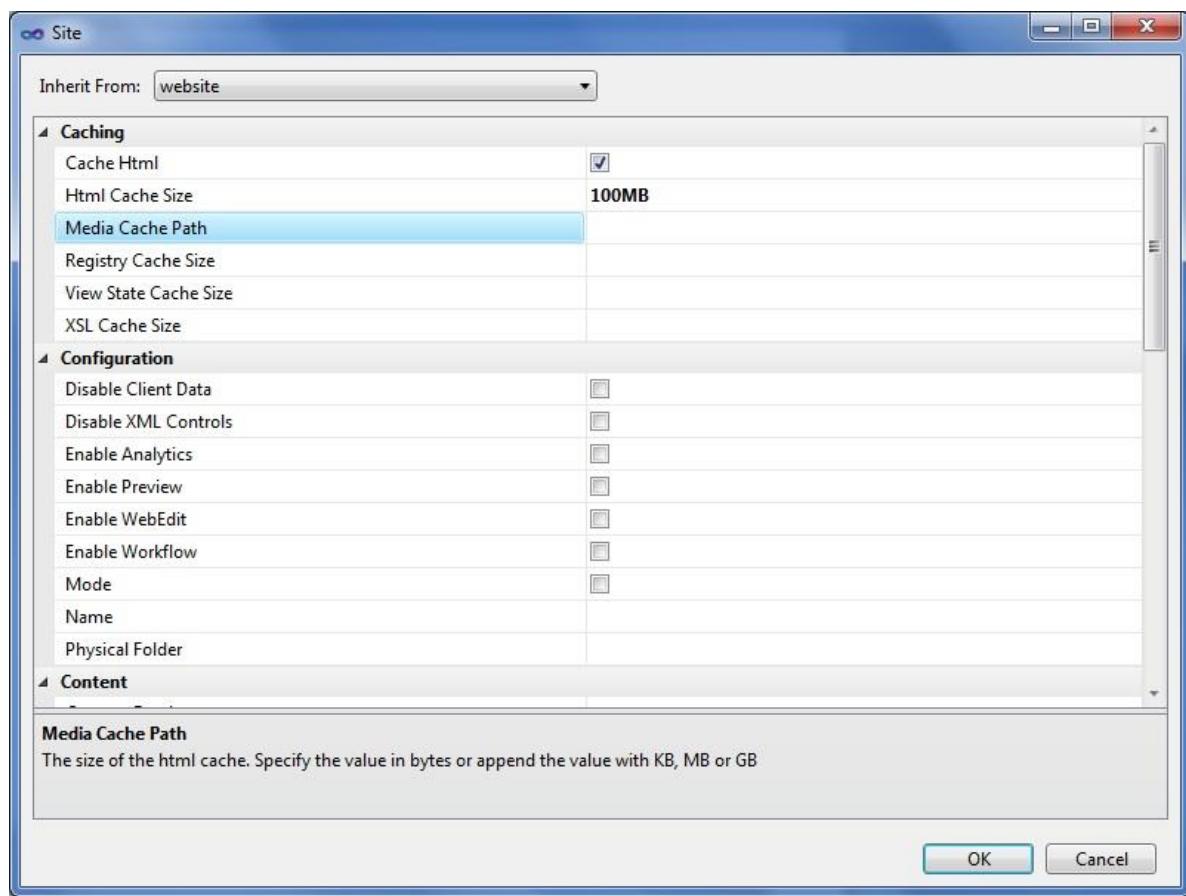
Properties:

Right-clicked row: website

- Create New Site...
- Refresh

Command: Ctrl+Shift+Space

This will open the Create New Site dialog where we can specify a new site and all the configuration properties associated with a Sitecore website.

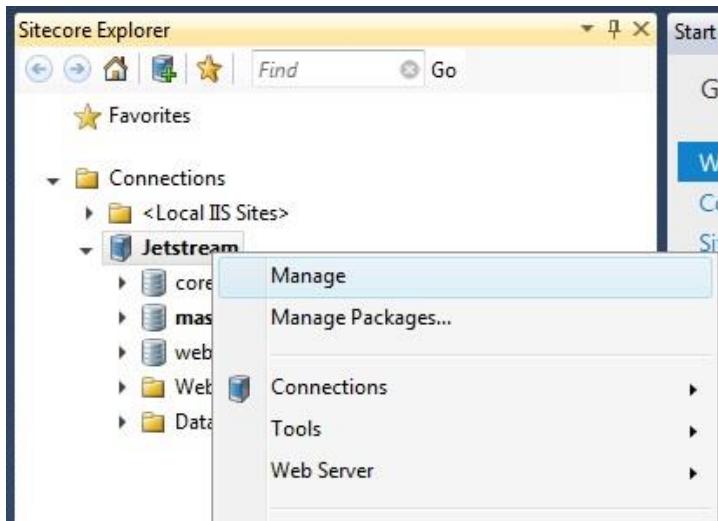


The Sites interface of the Sitecore Rocks Manage tool is very handy to inspect the different configuration properties. Using the Sites interface we can also create new sites.

19: 28 Days of Sitecore Rocks: Manage - Part 2

By Trevor Campbell, February 19, 2013

Let's continue with the Sitecore Rocks Manage tool by discussing the Pipelines, Commands, and Temp Files interfaces. We will see how using the Sitecore Rocks Manage tool can help manage and investigate Sitecore pipelines, commands and temporary files all from within Visual Studio. If we don't already have the Manage tool open, right click the Sitecore instance, in this case the Jetstream node, in the Sitecore Explorer and select the "Manage" menu item.



The second tab in the Manage tool is for the Pipelines interface. This interface can be used to inspect the different Sitecore pipelines and associated processors. When selecting a pipeline in the top section on the interface, the associated processors are loaded in the bottom section showing which order they are executed. Also, if we right click in the processors section it will open the context menu where we have the option to "Create Pipeline Processor" to insert an already defined custom processor.

Sites Pipelines Commands Temp Files Caches Indexes Statistics Validation

Pipeline

- getPageEditorNotifications
- getPageEditorValidators
- getPagePreviewUrl
- getPlaceholderRenderings
- getQueryState
- getRenderingDatasource
- getRenderingPreview
- getSyndicationWarnings
- getTranslation
- group
- healthMonitor
- httpRequestBegin**
- httpRequestEnd
- httpRequestProcessed
- initialize

Processors:

	Class	Assembly
1	Sitecore.Pipelines.PreprocessRequest.CheckIgnoreFlag	Sitecore.Kernel
2	Sitecore.Pipelines.HttpRequest.StartMeasurements	Sitecore.Kernel
3	Sitecore.Analytics.Pipelines.HttpRequest.StartDiagnostics	Sitecore.Analytics
4	Sitecore.Pipelines.HttpRequest.IgnoreList	Sitecore.Kernel
5	Sitecore.Pipelines.HttpRequest.SiteResolver	Sitecore.Kernel
6	Sitecore.DesignImporter.Pipelines.HttpRequest.SiteResolver	Sitecore.DesignImporter
7	Sitecore.Pipelines.HttpRequest.UserResolver	Sitecore.Kernel
8	Sitecore.Modules.EmailCampaign.Core.Pipelines.OnHttpRequestBegin	Sitecore.EmailCampaign
9	Sitecore.SeoToolkit.Pipelines.SetPageMode	Sitecore.SeoToolkit
10	Sitecore.Pipelines.HttpRequest.DatabaseResolver	Sitecore.Kernel
11	Sitecore.Pipelines.HttpRequest.BeginDiagnostics	Sitecore.Kernel
12	Sitecore.SharedSource.MobileDeviceDetector.DeviceResolver	Sitecore.SharedSource.MobileDeviceDetect
13	Sitecore.Pipelines.HttpRequest.LanguageResolver	Sitecore.Kernel

The third tab in the Manage tool is for the Commands interface. With the Commands interface we can see all of the Sitecore commands and their associated type and assembly.

Sites Pipelines Commands Temp Files Caches Indexes Statistics Validation

	Command Name	Type and Assembly
	accessviewer:selectcolumns	Sitecore.Shell.Framework.Commands.AccessViewer.SelectColumns,Sitecore.Kernel
	analytics:reports	Sitecore.Shell.Applications.Analytics.ReportRunner.OpenAnalyticsReports,Sitecore.Client
	analytics:runlookup	Sitecore.Shell.Applications.Analytics.Lookup.RunLookup,Sitecore.Client
	analytics:runvisitoridentification	Sitecore.Shell.Applications.Analytics.VisitorIdentifications.RunVisitorIdentifications,Sitecor
	analytics:status	Sitecore.Shell.Applications.Analytics.Status.Status,Sitecore.Client
	analytics:synchronizedatabase	Sitecore.Shell.Applications.Analytics.SynchronizeDatabase.SynchronizeDatabase,Sitecore.(
	analytics:updatereports	Sitecore.Shell.Applications.Analytics.Reports.Summary.Update,Sitecore.Client
	archive:delete	Sitecore.Shell.Framework.Commands.Archives.Delete,Sitecore.Kernel
	archive:empty	Sitecore.Shell.Framework.Commands.Archives.Empty,Sitecore.Kernel
	archive:refresh	Sitecore.Shell.Framework.Commands.Archives.Refresh,Sitecore.Kernel
	archive:restore	Sitecore.Shell.Framework.Commands.Archives.Restore,Sitecore.Kernel
	archive:showversions	Sitecore.Shell.Framework.Commands.Archives.ShowVersions,Sitecore.Kernel

Right clicking on one of the commands will open the context menu where we can jump to the definition of the command or as seen below, "Search for Usages." The Search for Usages option will open the Query Analyzer with a query of the core and master databases where the command is actually in use.



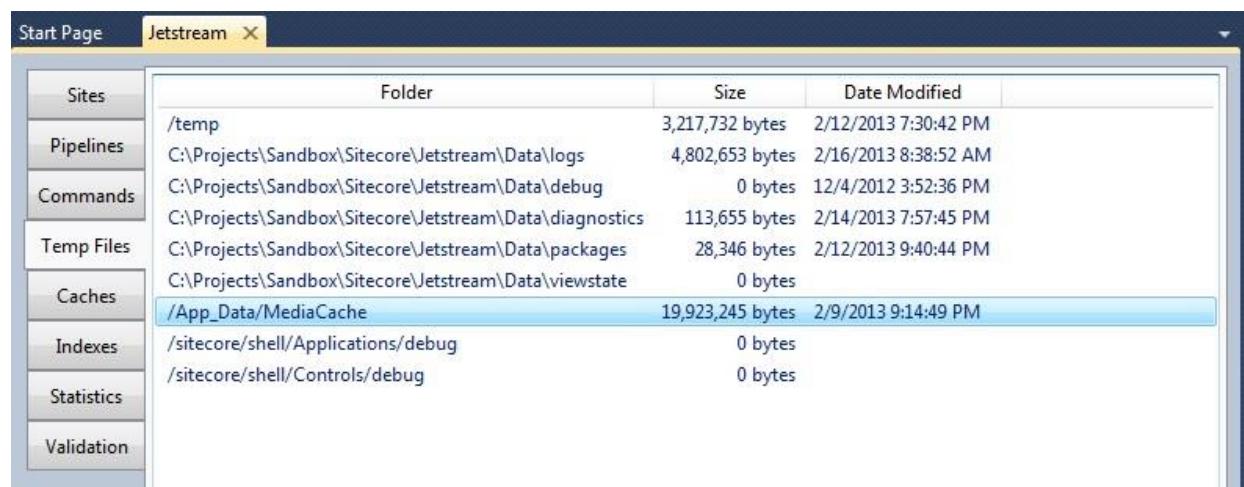
The screenshot shows the Sitecore Query Analyzer interface. The title bar says "Query Analyzer - core [Jetstream]". The toolbar includes "Execute", "Insert Path", "Insert Fields", and "Samples". The query pane contains the following T-SQL:

```
use core;
select * from //*[@Click='accessviewer:selectcolumns' or @Message='accessviewer:selectcolumns'];
use master;
select * from //*[@Click='accessviewer:selectcolumns' or @Message='accessviewer:selectcolumns']
```

The results pane shows a single item in the "Result" tab:

Name	ID	Template Name	Path
Columns	{581C3C60-E7B6-4927-8C91-99630BF38BC8}	Large Button	/sitecore/content/Applications/Security/Access Viewer/Ribbon/Ho

The third tab in the Manage tool is the Temp Files interface. This interface shows us all the temporary files currently on the file system for this Sitecore instance. The interface shows us the location, size and the last modified date of the temporary files.



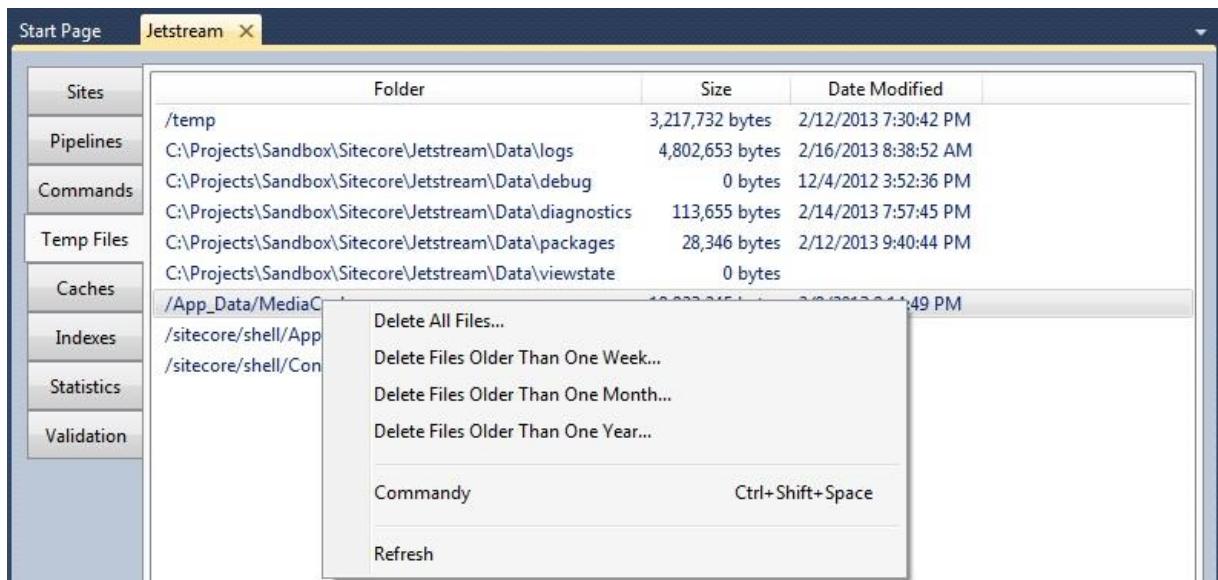
The screenshot shows the Sitecore Temp Files interface. The left sidebar has tabs for "Sites", "Pipelines", "Commands", "Temp Files" (which is selected), "Caches", "Indexes", "Statistics", and "Validation". The main pane displays a table of temporary files:

Folder	Size	Date Modified
/temp	3,217,732 bytes	2/12/2013 7:30:42 PM
C:\Projects\Sandbox\Sitecore\Jetstream\Data\logs	4,802,653 bytes	2/16/2013 8:38:52 AM
C:\Projects\Sandbox\Sitecore\Jetstream\Data\debug	0 bytes	12/4/2012 3:52:36 PM
C:\Projects\Sandbox\Sitecore\Jetstream\Data\diagnostics	113,655 bytes	2/14/2013 7:57:45 PM
C:\Projects\Sandbox\Sitecore\Jetstream\Data\packages	28,346 bytes	2/12/2013 9:40:44 PM
C:\Projects\Sandbox\Sitecore\Jetstream\Data\viewstate	0 bytes	
/App_Data/MediaCache	19,923,245 bytes	2/9/2013 9:14:49 PM
/sitecore/shell/Applications/debug	0 bytes	
/sitecore/shell/Controls/debug	0 bytes	

Right clicking on one of the temporary items will open the context menu where we have several options to clean out temporary files. The options we have are the following:

- Delete All Files
- Delete Files Older Than One Week

- Delete Files Older Than One Month
- Delete Files Older Than One Year



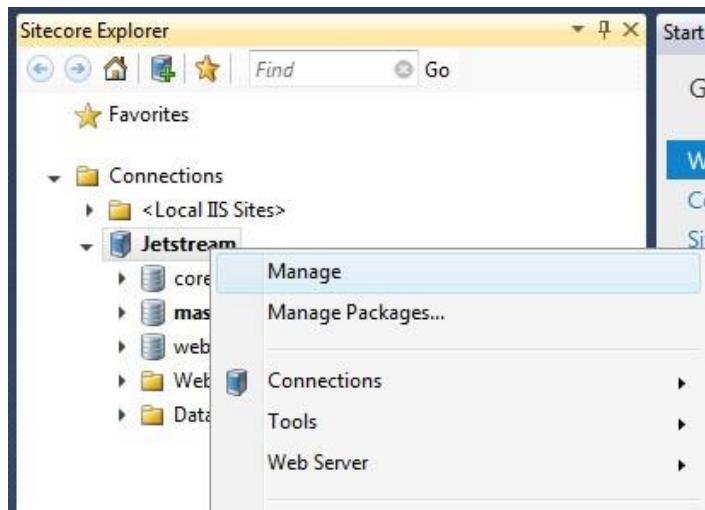
As we have seen today, the Sitecore Rocks Manage tool has several interfaces for working with Sitecore pipelines, commands and temporary files from Visual Studio.

20: 28 Days of Sitecore Rocks: Manage - Part 3

By Trevor Campbell, February 20, 2013

Continuing our discussion of the Sitecore Rocks Manage tool, today we will be covering the Caches interface. Having an understanding of how and what Sitecore Caches is an integral skill when configuring, tuning and troubleshooting performance. The Caches interface will provide us with more information than we've previously had available to us with the Sitecore cache page.

If we don't already have the Manage tool open, right click the Sitecore instance, in this case the Jetstream node, in the Sitecore Explorer and select the "Manage" menu item.



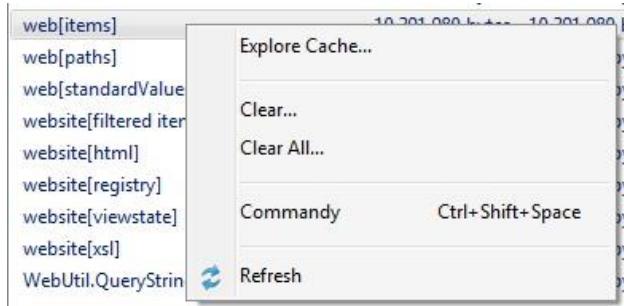
With the Caches interface open we will notice, similar to the Sitecore cache page, each of the caches for the sites currently configured in our instance of Sitecore. The number of columns is what differentiates the Caches interface. Here are the available columns. The bold columns are new.

- Name - Name of the cache. Usually "site [type]." Example: website[items]
- Size - The amount in bytes currently being consumed.
- Delta - The amount in bytes of the delta.
- Count - Number of cache entries.
- Max Size - The maximum size that the cache is allowed to consume before eviction.
- Remaining - The amount of cache available to be consumed before eviction.
- Enabled - Whether the cache is enabled or disabled.
- Scavengable - Boolean value indicating whether the cache is scavengable.
- Priority - The priority level of the cache.

Start Page Jetstream X Web.config

Name	Size	Delta	Count	Max Size	Remaining
AccessResultCache	0 bytes	0 bytes	0	2,097,152 bytes	2,097,152 bytes
admin[filtered items]	0 bytes	0 bytes	0	2,097,152 bytes	2,097,152 bytes
admin[html]	0 bytes	0 bytes	0	0 bytes	0 bytes
admin[registry]	0 bytes	0 bytes	0	5,242,880 bytes	5,242,880 bytes
admin[viewstate]	0 bytes	0 bytes	0	5,242,880 bytes	5,242,880 bytes
admin[xsl]	0 bytes	0 bytes	0	10,485,760 bytes	10,485,760 bytes
Alternate[filtered items]	0 bytes	0 bytes	0	2,097,152 bytes	2,097,152 bytes
Alternate[html]	0 bytes	0 bytes	0	10,485,760 bytes	10,485,760 bytes
Alternate[registry]	0 bytes	0 bytes	0	0 bytes	0 bytes
Alternate[viewstate]	0 bytes	0 bytes	0	0 bytes	0 bytes
Alternate[xsl]	0 bytes	0 bytes	0	5,242,880 bytes	5,242,880 bytes
clientData	0 bytes	0 bytes	0	5,242,880 bytes	5,242,880 bytes
core[blobIDs]	0 bytes	0 bytes	0	512,000 bytes	512,000 bytes
core[data]	1,910,250 bytes	1,910,250 bytes	1,121	20,971,520 bytes	19,061,270 bytes
core[items]	518,090 bytes	518,090 bytes	103	10,485,760 bytes	9,967,670 bytes
core[paths]	738 bytes	738 bytes	7	512,000 bytes	511,262 bytes
core[standardValues]	330 bytes	330 bytes	11	512,000 bytes	511,670 bytes
CRM[Metadata];SC-Demo\svccrmweb:democrm	0 bytes	0 bytes	0	2,097,152 bytes	2,097,152 bytes
DeviceItemsCache	70 bytes	70 bytes	1	10,485,760 bytes	10,485,690 bytes
filesystem[blobIDs]	0 bytes	0 bytes	0	512,000 bytes	512,000 bytes
filesystem[data]	0 bytes	0 bytes	0	0 bytes	0 bytes
filesystem[items]	0 bytes	0 bytes	0	0 bytes	0 bytes
filesystem[paths]	0 bytes	0 bytes	0	0 bytes	0 bytes
filesystem[standardValues]	0 bytes	0 bytes	0	0 bytes	0 bytes
IsUserInRoleCache	0 bytes	0 bytes	0	1,048,576 bytes	1,048,576 bytes
ItemCloningRelations	1,650 bytes	1,650 bytes	33	10,485,760 bytes	10,484,110 bytes
LanguageProvider - Languages	260 bytes	260 bytes	2	102,400 bytes	102,140 bytes
login[filtered items]	0 bytes	0 bytes	0	2,097,152 bytes	2,097,152 bytes
login[html]	0 bytes	0 bytes	0	0 bytes	0 bytes
login[registry]	0 bytes	0 bytes	0	5,242,880 bytes	5,242,880 bytes

What also differentiates the Caches interface is that we have the ability to clear a single cache as opposed to all caches. To do so, simply select the cache you would like to clear and right click to open the context menu. From here we can either clear the single cache or all caches.



Another nice feature of the Caches tool is the ability to explore the caches. To explore a cache, select a cache and open the context menu by right clicking the item and selecting "Explore Cache." The Explore Cache dialog lists out each items key, size, last accessed and data. We can filter the list of items in the cache by entering a value in the filter textbox, as well as evict a single item from a cache by right clicking an item and selecting "Remove."

Explore Cache

Cache Keys in web[items]:

Key	Size	Last Accessed	Data
{02DA2066-BBA4-4F88-80B9-37E545518DF0}en#0	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/content/Home/About/Terms And C
{032C30D5-925F-46E2-87B6-98F072D8F053}da#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Carousel/Page
{032C30D5-925F-46E2-87B6-98F072D8F053}de-DE#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Carousel/Page
{032C30D5-925F-46E2-87B6-98F072D8F053}en#0	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Carousel/Page
{032C30D5-925F-46E2-87B6-98F072D8F053}fr-FR#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Carousel/Page
{032C30D5-925F-46E2-87B6-98F072D8F053}ja-JP#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Carousel/Page
{032C30D5-925F-46E2-87B6-98F072D8F053}zh-CN#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Carousel/Page
{03379AF5-F1AE-4610-B15B-4C7F1032B464}en#0	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/system/Marketing Center/Profiles/Fc
{03665605-A268-4415-9195-191429AF07B0}en#0	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/content/Home/Plan And Book/Book
{03B777A3-51A3-4364-9E80-400D1A259702}da#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Modules/sono
{03B777A3-51A3-4364-9E80-400D1A259702}de-DE#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Modules/sono
{03B777A3-51A3-4364-9E80-400D1A259702}en#0	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Modules/sono
{03B777A3-51A3-4364-9E80-400D1A259702}en#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Modules/sono
{03B777A3-51A3-4364-9E80-400D1A259702}fr-FR#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Modules/sono
{03B777A3-51A3-4364-9E80-400D1A259702}ja-JP#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Modules/sono
{03B777A3-51A3-4364-9E80-400D1A259702}zh-CN#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Modules/sono
{08419A37-F8F3-485D-A287-D462853D2385}en#0	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/layout/Sublayouts/Jetstream/Modul
{08B0E6F1-7EBA-4F03-98B4-94E83A09606A}en#0	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/content/Global/Modules/Destination
{0A98E368-CDB9-4E1E-927C-8E0C24A003FB}en#0	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/templates/System/Layout/Rendering
{0CFBC191-EB13-467A-8891-24AB66DCAE2B}da#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Modules/Mon
{0CFBC191-EB13-467A-8891-24AB66DCAE2B}de-DE#1	5,030 bytes	2/17/2013 12:49:25 PM	/sitecore/media library/Images/Modules/Mon

Filter

Close

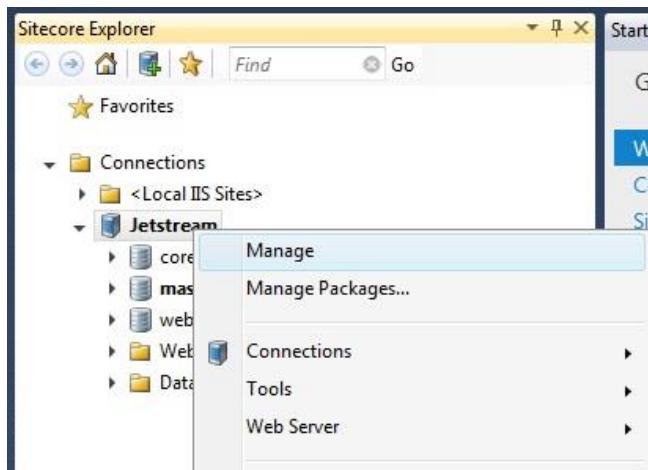
Using the Caches interface of the Sitecore Rocks Manage tool can provide insight into Sitecore caches that was previously unavailable via the standard methods. I would encourage anyone who works with Sitecore to give this interface a try when configuring, tuning or troubleshooting performance.

21: 28 Days of Sitecore Rocks: Manage - Part 4

By Trevor Campbell, February 21, 2013

Today we will be covering two more interfaces in the Sitecore Rocks Manage tool, Indexes and Statistics. Both of these interfaces are extremely helpful during development and later when performance testing. Let's dive in and see what these tools have to offer!

If we don't already have the Manage tool open, right click the Sitecore instance, in this case the Jetstream node, in the Sitecore Explorer and select the "Manage" menu item.



With the Indexes interface open you will see a listing of the indexes on the current instance. In this case you will see that we have two indexes configured, "advanced" and "system." The "advanced" index is specific to this implementation but the "system" index is configured out of the box and is used by Sitecore for searching.

We can rebuild the index by right clicking an index and selecting "Rebuild Index."

Name	Document Count
advanced	15,171
system	26,978

Field Name	Term Count
_created	4,558
_created by	25
_default workflow	2
_display name	220
_editors	1
_help link	1
_hide version	1
_icon	11
_lock	41
_long description	35
_masters	35
_owner	19
_publish	3
_publishing groups	1

When you select an index, the Fields, Documents and Search tabs are populated with data from the selected index. The Fields tab displays all the fields in the index and the number of unique terms associated with each field. You can get more information on the terms by right clicking a field to open the context menu and selecting "Explore Terms." This will list out the terms and the number of documents in the index that are associated with that term.

Term	Document Count
extranet\Anonymous	28
sitecore\admin	6
Sitecore\admin	1
sitecore\Anonymous	4
sitecore\audrey	1
sitecore\azcaser	10

The Documents tab displays the documents in the index based on order found in the index. You can cycle through the documents in the index or skip to a location by entering the number. The results show the location in Sitecore, group, and item name for the selected document.

The last tab is the Search tab and can be used to find documents in the index based on a query. Simply enter the field name to search on, search term and select a query type. In the query results you can locate the item in Sitecore by right clicking a document and selecting "Locate in Sitecore."

The Statistics interface is another very helpful tool for investigating performance issues. This interface is in essence the Sitecore Stats page. They contain the same columns of information as the stats page.

- Name
- Site
- Render Count
- Used Cache
- Average Time
- Average Items
- Max Time
- Max Items
- Total Time
- Total Items
- Last Rendered

Start Page Jetstream X

The screenshot shows the Sitecore Rocks Statistics interface. On the left, there's a sidebar with navigation links: Start Page, Sites, Pipelines (which is selected), Commands, Temp Files, Caches, Indexes, Statistics, and Validation. The main area is a table titled 'Statistics' with the following columns: Name, Site, Render Count, Used Cache, and Average Time. The table lists various Sitecore items and their performance metrics. The 'TitleFR (Text)' item is highlighted with a blue border.

Name	Site	Render Count	Used Cache	Average Time
CarouselByLine (Text)	website	5	0 bytes	0.04 ms
CarouselEditFrame (EditFrame)	website	1	0 bytes	39.96 ms
CarouselImage (Image)	website	15	0 bytes	13.76 ms
CarouselItemEditFrame (EditFrame)	website	5	0 bytes	6.84 ms
CarouselLink (Link)	website	5	0 bytes	6.79 ms
CarouselTitle (Text)	website	5	0 bytes	0.06 ms
DescriptionFR (Text)	website	8	0 bytes	0.35 ms
ImageFR (Image)	website	8	0 bytes	8.65 ms
ImageLinkFR (Link)	website	8	0 bytes	9.24 ms
LinkFR (Link)	website	8	0 bytes	0.38 ms
LinkTitleFR (Text)	website	4	0 bytes	0.05 ms
Placeholder: main	website	1	0 bytes	574.14 ms
Placeholder: main_content	website	1	0 bytes	81.28 ms
Placeholder: main_content_module	website	1	0 bytes	81.23 ms
Placeholder: top_media	website	1	0 bytes	242.91 ms
PriceDetailFR (Text)	website	5	0 bytes	0.04 ms
PriceFR (Text)	website	5	0 bytes	0.05 ms
SocialLinkEF (EditFrame)	website	4	0 bytes	0.04 ms
Sublayout: /layouts/JS/Sublayouts/Content/28DaysOfSitecoreRocks.ascx	website	1	0 bytes	0.00 ms
Sublayout: /layouts/JS/Sublayouts/Content/HeroCarousel.ascx	website	1	0 bytes	242.90 ms
Sublayout: /layouts/JS/Sublayouts/Content/ModulesContainer.ascx	website	1	0 bytes	81.25 ms
Sublayout: /layouts/JS/Sublayouts/Modules/BasicDestinationPromotion.ascx	website	4	0 bytes	4.90 ms
Sublayout: /layouts/JS/Sublayouts/Modules/BasicPromotion.ascx	website	3	0 bytes	18.62 ms
Sublayout: /layouts/JS/Sublayouts/Modules/DestinationPromotion.ascx	website	1	0 bytes	5.71 ms
Sublayout: /layouts/JS/Sublayouts/SiteStructure/Footer.ascx	website	1	0 bytes	208.78 ms
Sublayout: /layouts/JS/Sublayouts/SiteStructure/Header.ascx	website	1	0 bytes	39.48 ms
Sublayout: /layouts/JS/Sublayouts/SiteStructure/PhotoWrapper.ascx	website	1	0 bytes	572.95 ms
TitleFR (Text)	website	8	0 bytes	0.06 ms
VisitorIdentification1 (VisitorIdentification)	website	1	0 bytes	20.21 ms

The advantage of using the Sitecore Rocks Statistics interface over the stats page is that we never need to leave Visual Studio to view the statistics information.

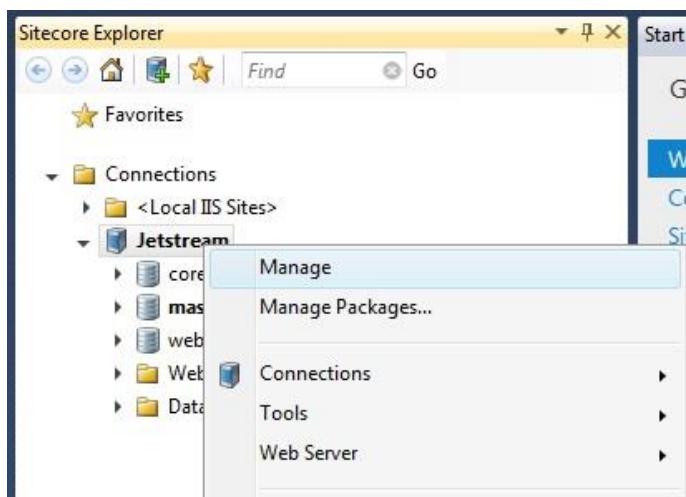
The Indexes and Statistics interfaces of the Sitecore Rocks Manage tool, as we've seen, are valuable tools when developing a Sitecore website and when performance tuning. Using these tools in conjunction with Sitecore best practices can have a dramatic impact on the performance of your implementation.

22: 28 Days of Sitecore Rocks: Manage - Part 5

By [Trevor Campbell](#), February 22, 2013

Today we will be wrapping up the Sitecore Rocks Manage tool by discussing the final interface, Validation. This interface is one of my favorite features in all of Sitecore Rocks. Part of my role as a Technical Lead is evaluating Sitecore implementations developed both internally and externally. Over time I've compiled an exhaustive list of items that I check manually when performing an evaluation. The Validation interface has allowed me to automate a large portion of these items. Let's get started and see how!

If we don't already have the Manage tool open, right click the Sitecore instance, in this case the Jetstream node, in the Sitecore Explorer and select the "Manage" menu item.



The first time you open the Validation interface you are prompted to run the assessment. This typically takes a minute or so depending on the size of the database and file system. When the assessment is complete the resulting report is categorized with an indication of the number of issues found in each.

The screenshot shows the Jetstream application window. The left sidebar has a navigation menu with items: Start Page, Sites, Pipelines, Commands, Temp Files, Caches, Indexes, Statistics, and Validation. The Validation item is highlighted with a red box. The main content area has a header with buttons: Re-run Assessment, Filter Validations, Edit Custom Validations, and Select Databases and Languages. Below this is a timestamp: Last Updated: 2/1/2013 1:17:55 PM. A filter input field with a clear button is also present. The main content area displays a tree view of validation categories under the Validation tab, with the 'Validation (568)' node expanded. Other collapsed categories include Configuration (11), Fields (1), Items (6), Licensing (1), Performance (2), Presentation (110), Security (64), Templates (10), and Configuration (11) again.

Expanding a category will display the issues in that category. The issues are color coded by the four severities: Errors, Warnings, Suggestions and Hints. Each issue has a title, problem, solution and even sometimes a button to fix the issue.

The screenshot shows the Jetstream application window with the Validation category expanded. The main content area displays five validation issues under the Configuration (11) category:

- Configure mail server in web.config** (Severity: Error)

The mail server settings in web.config are missing.
It is recommended to set the "MailServer", "MailServerPort", "MailServerUserName" and "MailServerPassword" settings in web.config.
- Enable server specific properties for indexing** (Severity: Warning)

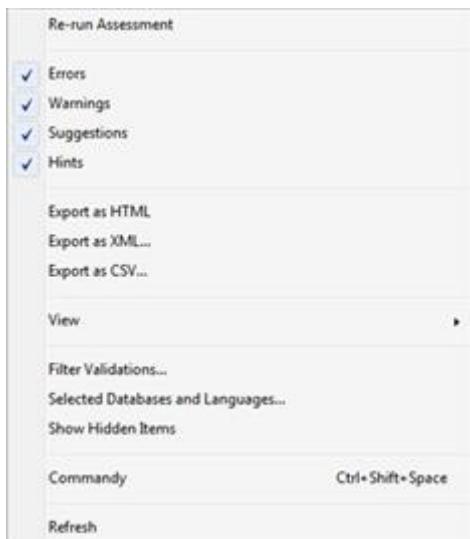
When indexing is enabled, it is also recommended that server specific properties are enabled.
To enable server specific properties, set the setting "Indexing.ServerSpecificProperties" to "true" in the web.config.
- Referenced types in web.config** (Severity: Error)

The referenced type "Sitecore.Pipelines.GetItemCommands.ShellCommands, Sitecore.Kernel" does not exist. It is referenced from /sitecore[database="SqlServer"]/processors/getItemCommands/processor [mode="on" type="Sitecore.Pipelines.GetItemCommands.ShellCommands, Sitecore.Kernel"].
Either correct the reference or remove it.
- Referenced types in web.config** (Severity: Error)

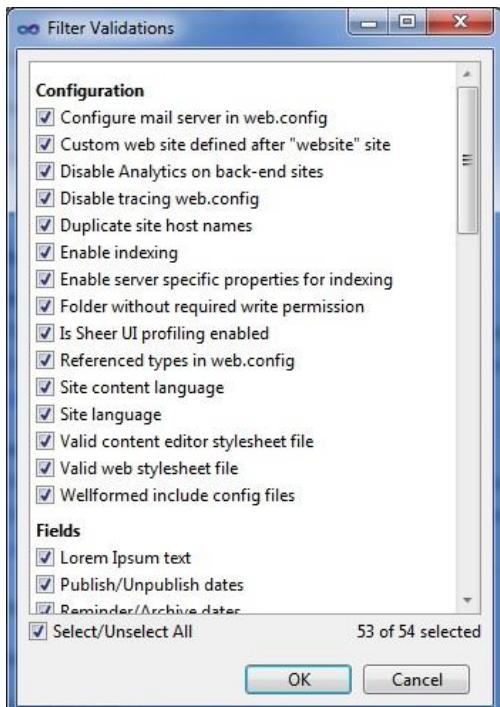
The referenced type "Sitecore.Pipelines.GetItemCommands.StaticCommands, Sitecore.Kernel" does not exist. It is referenced from /sitecore[database="SqlServer"]/processors/getItemCommands/processor [mode="on" type="Sitecore.Pipelines.GetItemCommands.StaticCommands, Sitecore.Kernel"].
Either correct the reference or remove it.
- Referenced types in web.config** (Severity: Error)

The referenced type "Sitecore.Pipelines.GetPageEditorNotifications.GetWorkflowNotification, Sitecore.Kernel, Version=6, Culture=neutral" does not exist. It is referenced from /sitecore[database="SqlServer"]/pipelines/getPageEditorNotifications/processor [type="Sitecore.Pipelines.GetPageEditorNotifications.GetWorkflowNotification, Sitecore.Kernel, Version=6, Culture=neutral"].

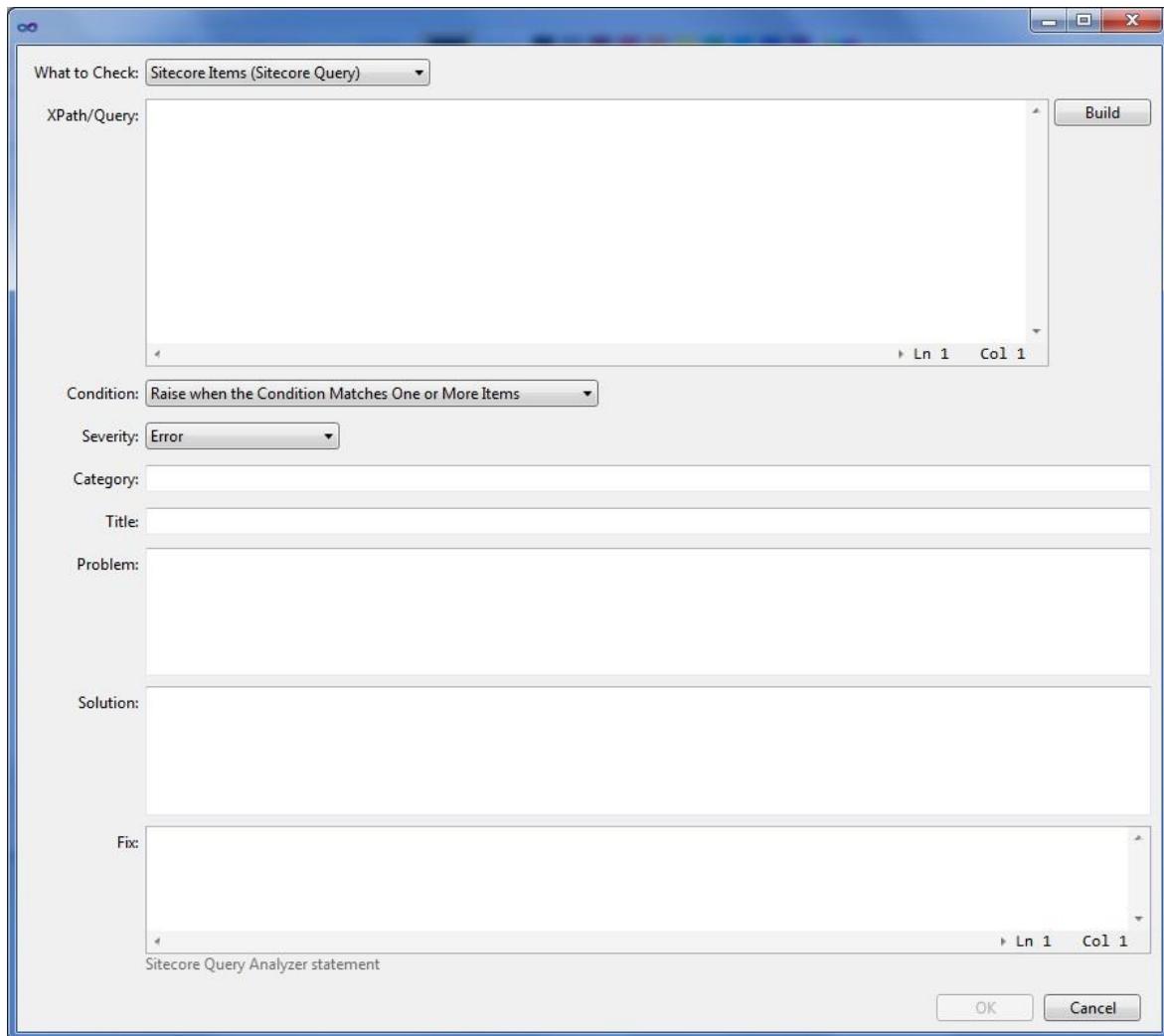
When we open the context menu on the results we have the option to only show issues of a selected severity. We also have the ability to export the results of the assessment to HTML, XML, or CSV.



If we select the "Filter Validations" in either the context menu or at the top of the Validations interface, we can see the full list of validations and select to filter out a validation if wanted.



Here's the best part. We can write our own custom validations and run them along with the out of the box validations! To create our own custom validations, click the "Edit Custom Validations" button at the top of the Validations interface to open the Custom Validations dialog. Here we can add our own validations by clicking the "Add" button or edit an existing custom validation by clicking "Edit." This opens the custom validation editor where we can create/edit our custom validation.



We have the ability to check the following by either writing a Sitecore query or an Xpath query:

- Sitecore Items (Sitecore Query)
- Sitecore Items (Xpath)
- Web.config (Xpath)
- Expanded web.config (Xpath)
- Web Site File System (Xpath)
- Data Folder File System (Xpath)

Once we have written our check criteria we can define a severity, category, title, problem, solution and even a fix. That's it! Now we have validations we can run on our solution to validate anything we choose.

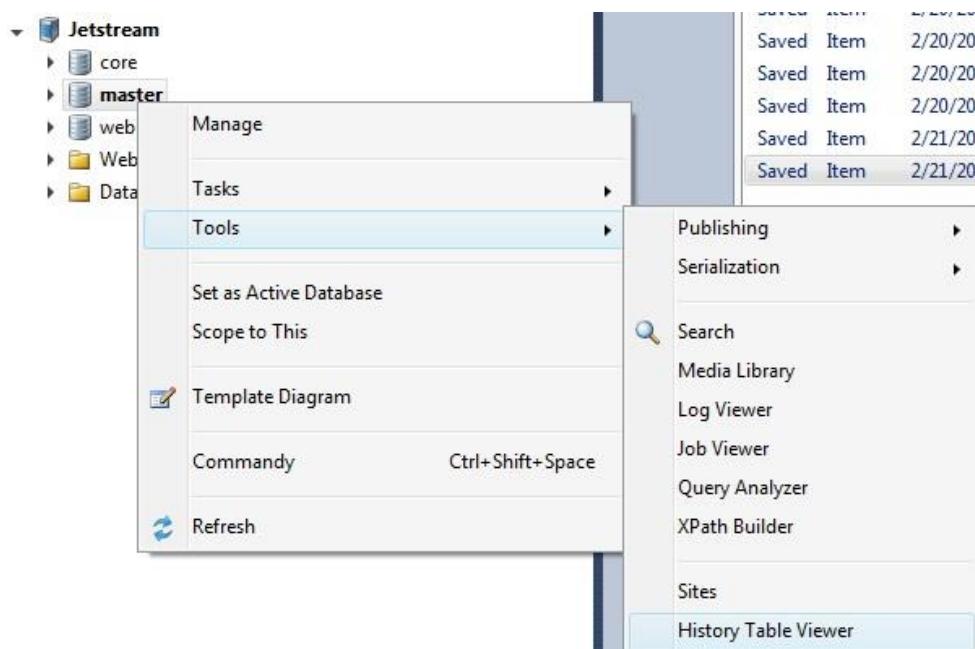
The Validations interface of the Sitecore Rocks Manage tool is extremely helpful for validating a Sitecore implementation for best practices. It can be used at any point during the software development lifecycle to validate an implementation as well.

23: 28 Days of Sitecore Rocks: History Table Viewer and Languages

By [Trevor Campbell](#), February 23, 2013

Today's topic in the 28 Days of Sitecore Rocks is the History Table Viewer and Languages tools. The History Table Viewer is a specialized tool to investigate content history. The Language tool is also a specialized tool that aids in the management of Sitecore languages.

Let's start with the History Table Viewer. To open the History Table Viewer, right-click anywhere in the Sitecore Explorer, navigate under the Tools menu and click the History Table Viewer item.



The History Table Viewer, as the name implies, is a tool to view the Sitecore History database table in Visual Studio. The Sitecore History table is used to tell what, when and who changed or added something in Sitecore. Using this tool we can filter the results to narrow down a particular update. Each result has the following information.

- **Action** - Created, Added Version, Saved, Deleted, etc.
- **Category** - The Category of the action.
- **Date Time** - Date and Time of action.
- **Item ID** - The Sitecore Item ID.
- **Language** - The Sitecore Language of the item.
- **Version** - The Sitecore Version of the Item.
- **User Name** - The Sitecore user who performed the action.
- **Task** - The Task performed.
- **Path** - The Path in Sitecore to the Item.
- **Additional Info** - Any Additional Information about the action.

Jetstream X

Languages	Date Range:	2/20/2013 12:39 PM	to	2/21/2013 12:39 PM	Filter		
History	Action	Category	Date Time	Item ID	Language	Version	User Name
	Saved	Item	2/20/2013 1:46:55 PM	{AB0B6514-5FF0-4A20-A41C-46805826875E}	en	2	sitecore\Anonymous
	Saved	Item	2/20/2013 1:46:55 PM	{D74D54C5-82CA-4B6D-BFFC-9CCFC51C649F}	en	1	sitecore\Anonymous
	Saved	Item	2/20/2013 1:46:55 PM	{D69FD6AE-45A4-4E5-A221-79D596FD0DFC}	en	1	sitecore\Anonymous
	Saved	Item	2/20/2013 1:46:55 PM	{121D3524-1428-4AB5-8733-53CB87C4DEB6}	en	1	sitecore\Anonymous
	Saved	Item	2/20/2013 1:46:56 PM	{C6AAD871-B8B8-483D-8F7A-871BE6D6D8BF}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:05:45 AM	{121D3524-1428-4AB5-8733-53CB87C4DEB6}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:05:45 AM	{C6AAD871-B8B8-483D-8F7A-871BE6D6D8BF}	en	1	sitecore\Anonymous

We can also filter the result set by a date range if we know an approximate date/time in which we are looking for an update.

Jetstream X

Languages	Date Range:	2/20/2013 12:39 PM	to	2/21/2013 11:45 PM	Filter		
History	Action	Category	Date Time	Item ID	Language	Version	User Name
	Saved	Item	2/20/2013 1:46:55 PM	{AB0B6514-5FF0-4A20-A41C-46805826875E}	en	2	sitecore\Anonymous
	Saved	Item	2/20/2013 1:46:55 PM	{D74D54C5-82CA-4B6D-BFFC-9CCFC51C649F}	en	1	sitecore\Anonymous
	Saved	Item	2/20/2013 1:46:55 PM	{D69FD6AE-45A4-4E5-A221-79D596FD0DFC}	en	1	sitecore\Anonymous
	Saved	Item	2/20/2013 1:46:55 PM	{121D3524-1428-4AB5-8733-53CB87C4DEB6}	en	1	sitecore\Anonymous
	Saved	Item	2/20/2013 1:46:56 PM	{C6AAD871-B8B8-483D-8F7A-871BE6D6D8BF}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:05:45 AM	{121D3524-1428-4AB5-8733-53CB87C4DEB6}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:05:45 AM	{C6AAD871-B8B8-483D-8F7A-871BE6D6D8BF}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:49:15 PM	{AB0B6514-5FF0-4A20-A41C-46805826875E}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:49:15 PM	{D74D54C5-82CA-4B6D-BFFC-9CCFC51C649F}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:49:15 PM	{D0FE3508-709B-45C2-8BE5-09DC96E4BA90}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:49:16 PM	{D69FD6AE-45A4-4E5-A221-79D596FD0DFC}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:49:16 PM	{121D3524-1428-4AB5-8733-53CB87C4DEB6}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:49:16 PM	{C6AAD871-B8B8-483D-8F7A-871BE6D6D8BF}	en	1	sitecore\Anonymous
	Saved	Item	2/21/2013 12:54:29 PM	{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}	en	1	sitecore\admin

The other tab in this tool is the Languages tool. It's a very straightforward interface that displays the current languages enabled in Sitecore.

Jetstream X

Languages	Name	Display Name	Filter
History	da	Danish : dansk	
	de-DE	German (Germany) : Deutsch (Deutschland)	
	en	English : English	
	fr-FR	French (France) : français (France)	
	ja-JP	Japanese (Japan) : 日本語 (日本)	
	zh-CN	Chinese (Simplified, PRC) : 中文(中华人民共和国)	

We can add a new language by simply right clicking the Languages tool to open the context menu. Here we can pull the information from a redefined language in Sitecore or create a new language from scratch.



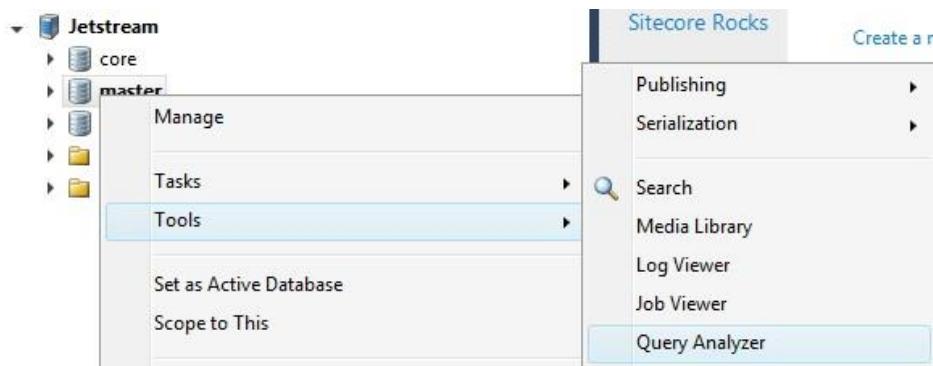
The History Table Viewer and Languages tools provide insight into two distinct functions of Sitecore. They both can aid in the development effort of a Sitecore site from different aspects.

24: 28 Days of Sitecore Rocks: Query Analyzer

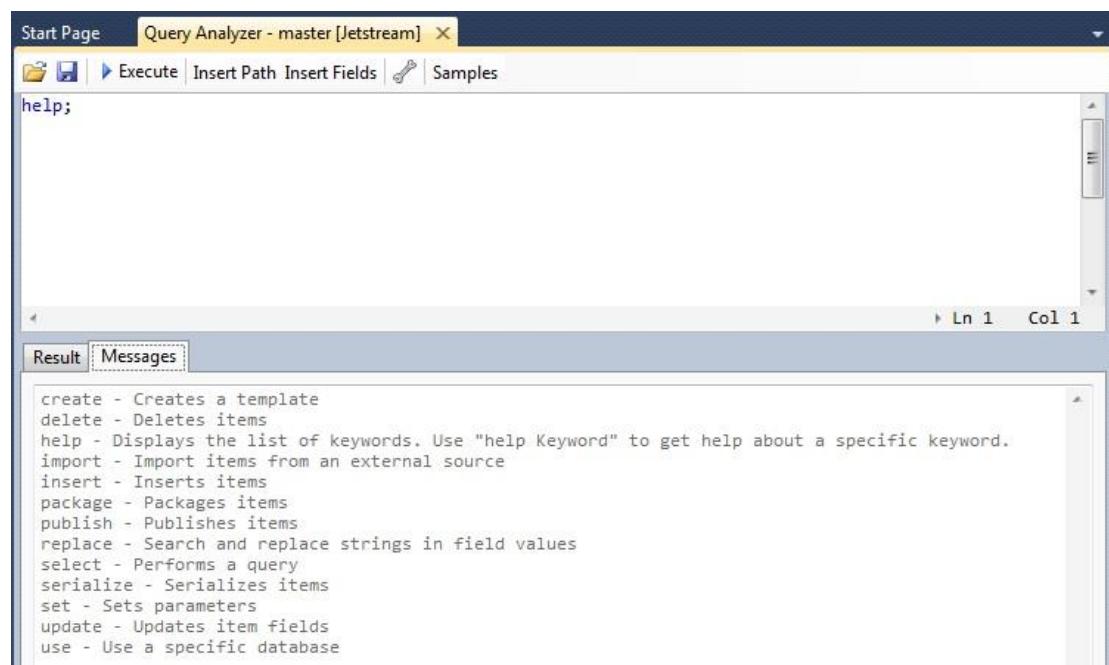
By Trevor Campbell, February 24, 2013

Of all the features in Sitecore Rocks I would say that the Query Analyzer is in the top three. Due to the sheer power of the Query Analyzer, it's hard to write only a single blog article about it. Therefore, this article will get you started using the Query Analyzer and provide a few resources to help along the way.

Let's start by opening the Query Analyzer. Right click on a database, navigate under the Tools menu and select "Query Analyzer" from the menu.

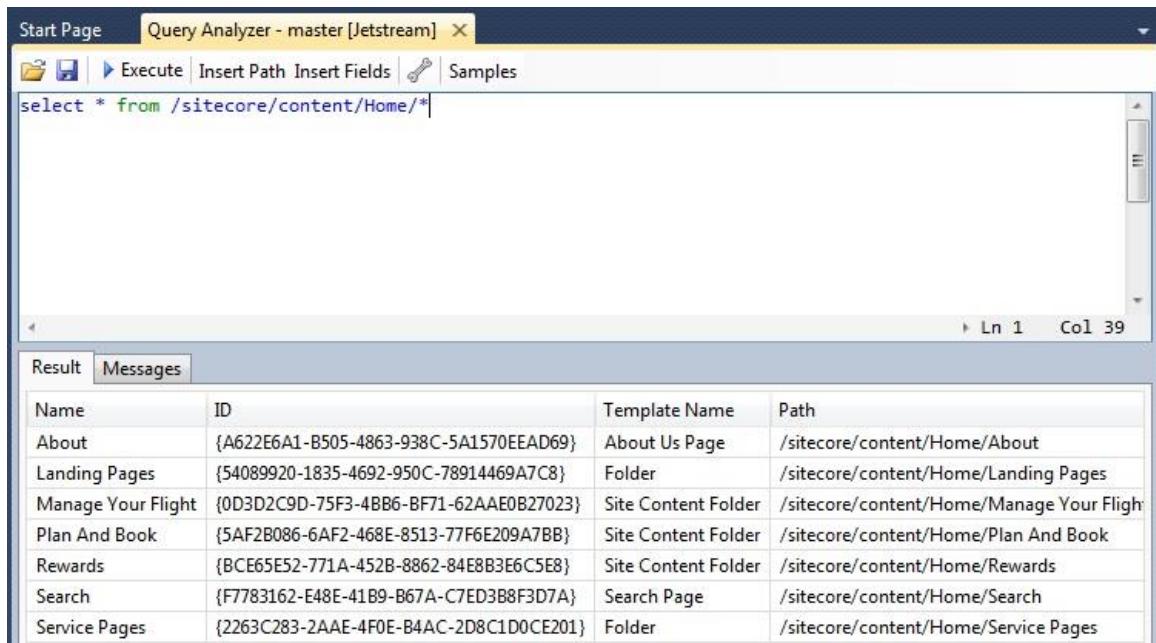


If you've never worked with the Query Analyzer before, the best way to get familiar with what's available is to start with the help commands. Simply enter "help" for the query and execute. This will return the list of commands available at our disposal.



We can dive deeper in the commands help by appending a command to the help query. For example, if you enter "help select" and execute this query you will see all the options and syntax for the select command.

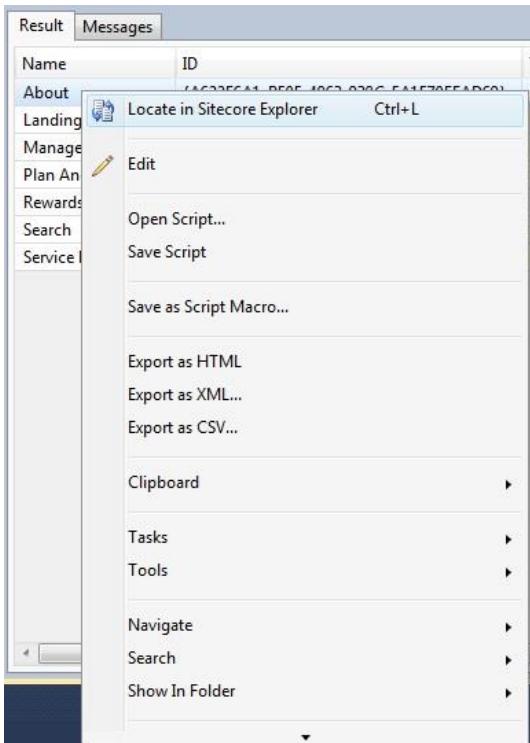
Now that we have a basic idea what commands are available let's execute some queries and see the results. If you are intimately familiar with the Sitecore Query syntax then you will most likely be able to jump right in and query away. If not, there are a couple good resources to help get you familiar. The first resource is the "Samples" button at the top of the Query Analyzer. This will take you to the Sitecore Rocks wiki page for [Sitecore Query Analyzer Samples](#). The second great resource is John West's blog article titled "[Sitecore Rocks Query Analyzer Ingredients for the Sitecore ASP.NET CMS](#)".



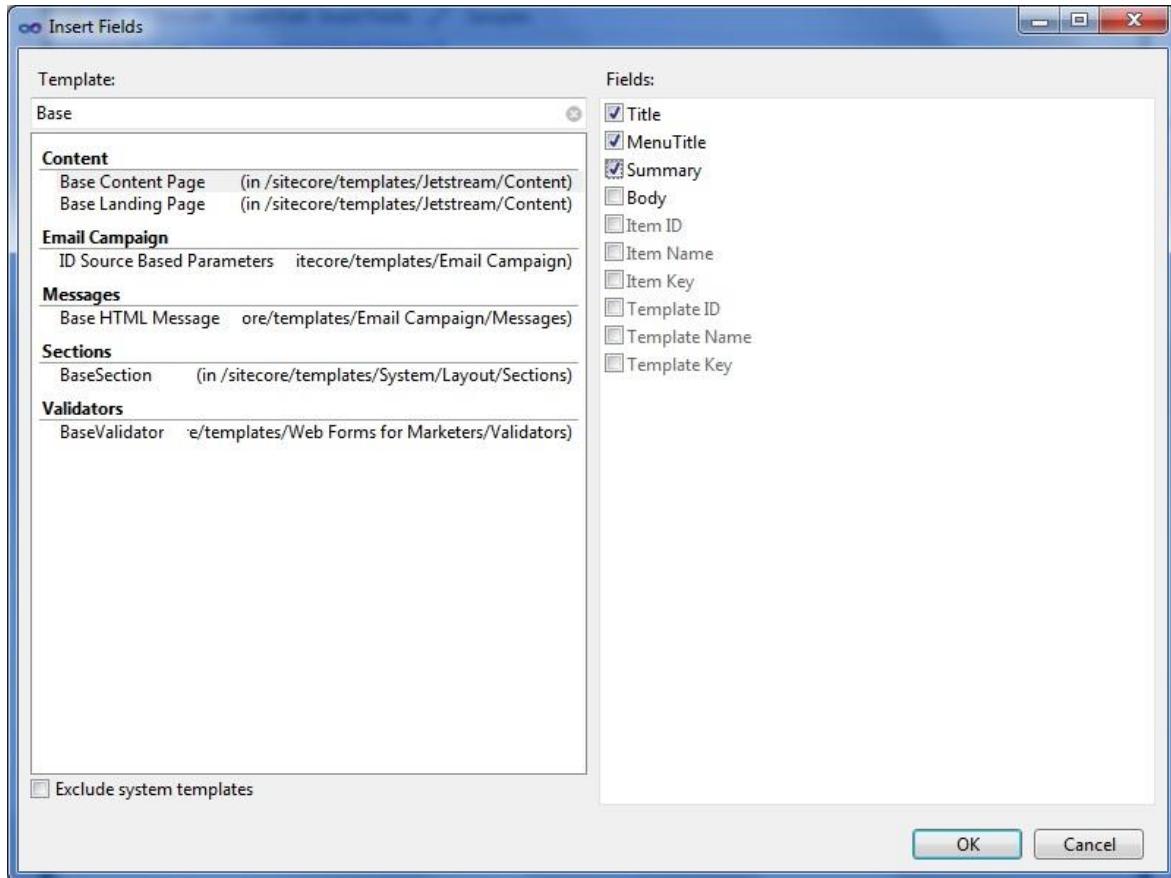
The screenshot shows the Sitecore Query Analyzer window. At the top, there is a toolbar with icons for Start Page, Query Analyzer - master [Jetstream], Execute, Insert Path, Insert Fields, and Samples. Below the toolbar, a query editor window contains the SQL-like query: `select * from /sitecore/content/Home/*`. In the bottom half of the window, there is a table titled "Result" with the following data:

Name	ID	Template Name	Path
About	{A622E6A1-B505-4863-938C-5A1570EEAD69}	About Us Page	/sitecore/content/Home/About
Landing Pages	{54089920-1835-4692-950C-78914469A7C8}	Folder	/sitecore/content/Home/Landing Pages
Manage Your Flight	{0D3D2C9D-75F3-4BB6-BF71-62AAE0B27023}	Site Content Folder	/sitecore/content/Home/Manage Your Flight
Plan And Book	{5AF2B086-6AF2-468E-8513-77F6E209A7BB}	Site Content Folder	/sitecore/content/Home/Plan And Book
Rewards	{BCE65E52-771A-452B-8862-84E8B3E6C5E8}	Site Content Folder	/sitecore/content/Home/Rewards
Search	{F7783162-E48E-41B9-B67A-C7ED3B8F3D7A}	Search Page	/sitecore/content/Home/Search
Service Pages	{2263C283-2AAE-4F0E-B4AC-2D8C1D0CE201}	Folder	/sitecore/content/Home/Service Pages

After running a query in the Query Analyzer we have several options with the results. If we open the context menu on a result item we can navigate to the item by selecting the "Locate in Sitecore Explorer" menu item. We can also export the results of our query to HTML, XML, or CSV formats.



Want to return certain fields in your results? No problem. If you know the name of the field off the top of your head then you can write it out with an ampersand (@) in front to return just that field. If you don't remember the exact name of the field, click the "Insert Fields" button at the top of the Query Analyzer and filter by Template.



The advantage of selecting specific fields is that you can edit the field's content directly in the Query Analyzer results! When you tab off of the field that you edited, the item is automatically saved.

Title	MenuTitle	Summary
About		Welcome to our world!

Additionally, if you don't know the path for your query, you can use the "Insert Path" dialog to navigate the Sitecore tree and select the exact path needed.

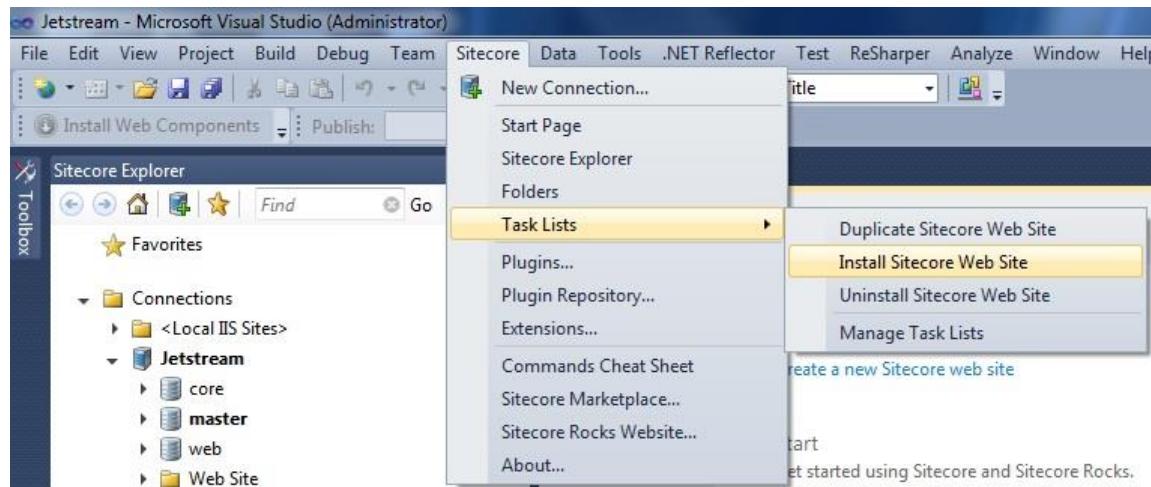
Once you have worked with the Sitecore Rocks Query Analyzer you will find a vast number of usages and question how you ever worked without it. I highly recommend every developer uses this tool and never look back!

25: 28 Days of Sitecore Rocks: Sitecore Installation

By Trevor Campbell, February 25, 2013

In today's edition of the 28 Days of Sitecore Rocks we will be covering the Sitecore Installation tool. This tool automates the installation of a Sitecore instance and other tasks associated with setting up a Sitecore web site. We will also see how to extend the installation tool to automate custom tasks.

To open the Sitecore Installation tool, click the "Sitecore" menu item in the menu bar and under the "Task Lists" menu item click the "Install Sitecore Web Site."

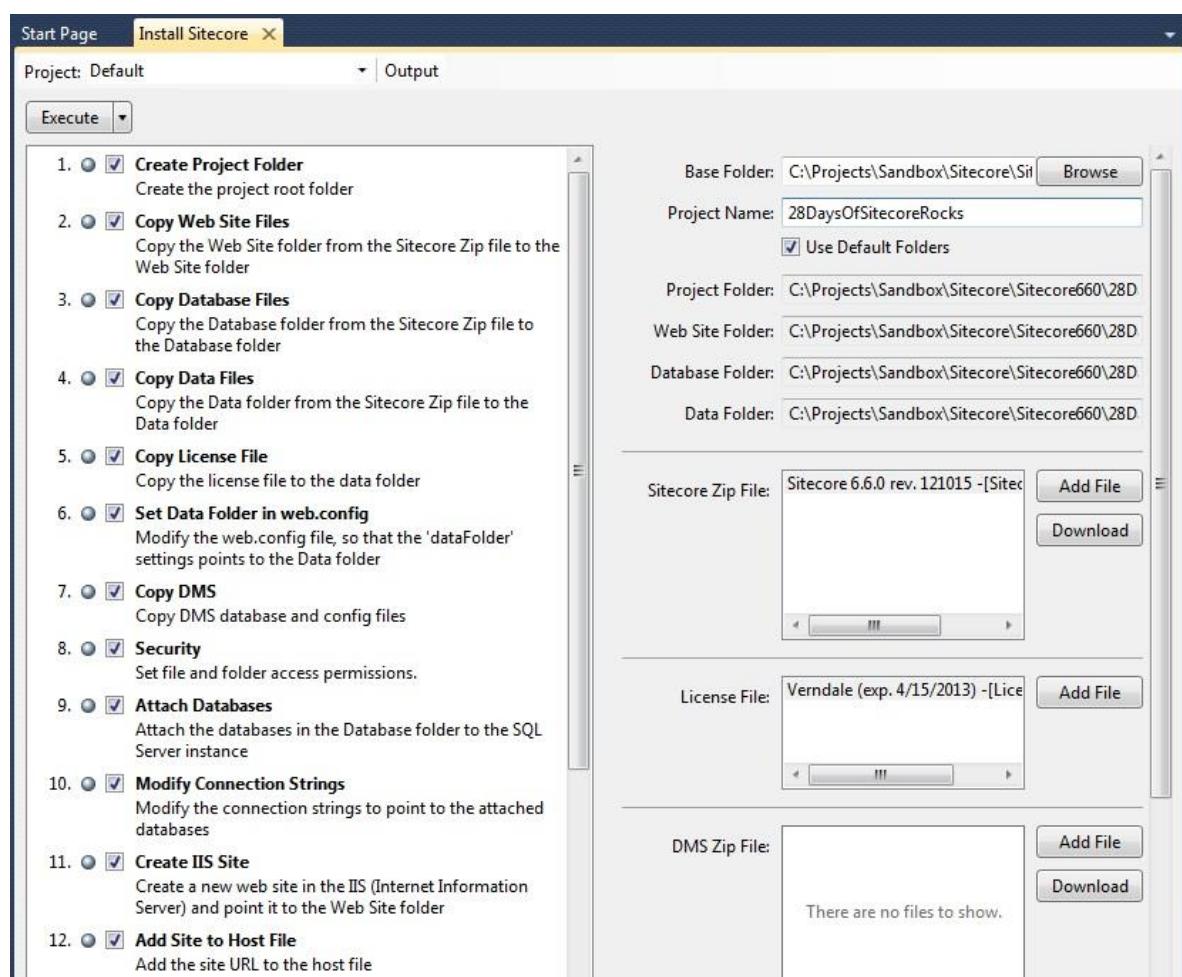


With the Sitecore Installation tool open we can see all the tasks that the installation tool can perform. They are as follows:

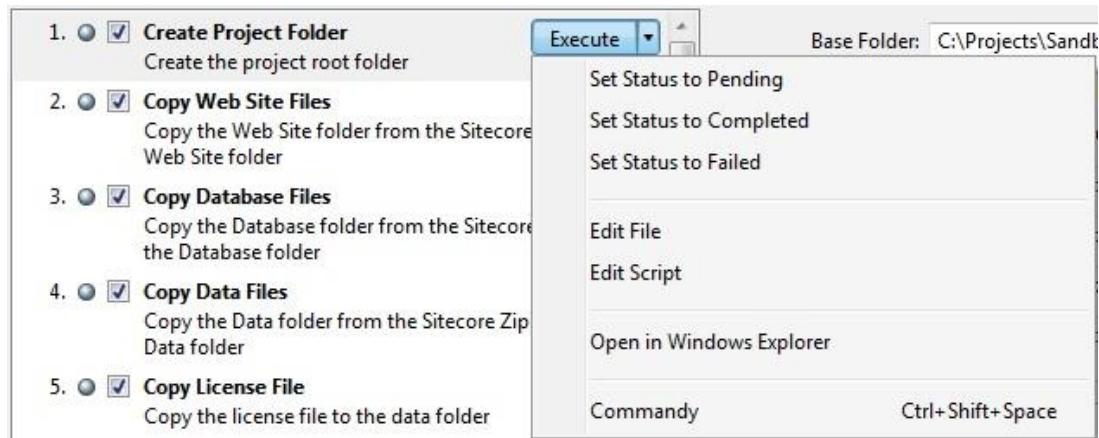
- **Create Project Folder** - Task that creates the project folder.
- **Copy Web Site Files** - Task that copies the web site files to the appropriate location specified.
- **Copy Database Files** - Task that copies the database files to the appropriate location specified.
- **Copy Data Files** - Task that copies the data files to the appropriate location specified.
- **Copy License File** - Task that copies the supplied Sitecore license file to the data folder.
- **Set Data Folder in web.config** - Task that sets the data folder location in the web.config.
- **Copy DMS** - Task that copies the supplied Sitecore DMS files to the appropriate location.
- **Security** - Task that sets the file and folder permissions.
- **Attach Databases** - Task that attaches the databases in the database folder to the specified SQL Server instance.
- **Modify Connection Strings** - Task that modifies the connection strings to the newly created databases.
- **Create IIS Site** - Task that creates an application pool and web site in IIS.
- **Add Site to Host File** - Task that inserts the new IIS web site in the hosts file.
- **Install Sitecore Rocks** - Task that installs Sitecore Rocks.
- **Create Connection** - Task that creates a Sitecore Rocks connection.
- **Install Packages** - Task that installs the packages specified.
- **Warm Up** - Task that makes a request to the web site to start up the system.
- **Open Sitecore** - Task that opens the web site in a web browser.

All we need to do is supply some basic information and the installation tool will do the heavy lifting.

- **Base Folder** - The base folder is where all the project files will be installed.
- **Project Name** - The name of the project.
- **Project Folder** - The path where the project will be created.
- **Web Site Folder** - The path where the Web Site will be created.
- **Database Folder** - The path where the Database files will be created.
- **Data Folder** - The path where the Data folder will be created.
- **Sitecore Zip File** - The Sitecore zip file from SDN.
- **License File** - The Sitecore license file.
- **DMS Zip File** - The Sitecore DMS zip file.
- **SQL Server Instance** - The SQL Server instance name.
- **SQL User Name** - The SQL Server user name.
- **SQL User Password** - The SQL Server password.
- **Packages** - Which packages are to be installed.



Each task has a pull down menu that lets us change the status of the task from pending, completed or failed. It also gives us the ability to edit the PowerShell Script associated with the task.



Selecting the "Edit File" option from the tasks pull down menu will open the task XML file. This file contains meta data to display the task in the Installation tool.

```
<?xml version="1.0" encoding="UTF-8"?>
- <task>
  <header>Add Site to Host File</header>
  <text>Add the site URL to the host file</text>
  <tasklist>Install Sitecore</tasklist>
  <priority>1700</priority>
</task>
```

Selecting the "Edit Script" option from the tasks pull down menu will open the PowerShell script file. Here we can make any modifications required.

```
$projectName = $parameters['projectName'];
$exists = Select-String "C:\WINDOWS\system32\drivers\etc\hosts" -Pattern "$projectName" -quiet;
if ($exists -eq $False)
{
    Add-Content -Path "C:\WINDOWS\system32\drivers\etc\hosts" -value "`r`n127.0.0.1 $projectName";
    Write-Output "Added $ projectName to host file";
}
else
{
    Write-Output "$ projectName already exists in host file. Skipping.";
}
```

We can also add custom tasks to the Installation tool by creating an XML file and PowerShell script in the Sitecore Rocks task lists folder. This folder is typically located in the following location.

C:\Users\[USERNAME]\AppData\Local\Sitecore\Sitecore.Rocks\Repositories\TaskLists\InstallSitecore

An easy way to get started with creating custom tasks is to check out the existing tasks and modify as required.

The Sitecore Rocks Installation tool is one of my favorite tools for automating the installation of a Sitecore instance. The Sitecore Rocks Installation tool can be used to get a Sitecore instance running very quickly to test a new module, build a proof of concept or start a new project.

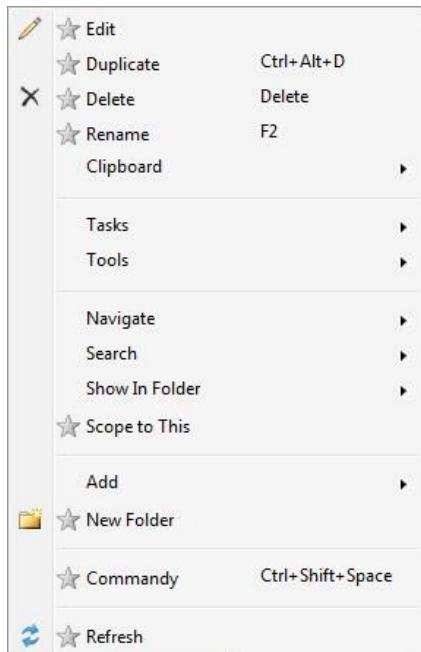
26: 28 Days of Sitecore Rocks: Hidden Gems - Part 1

By [Trevor Campbell](#), February 26, 2013

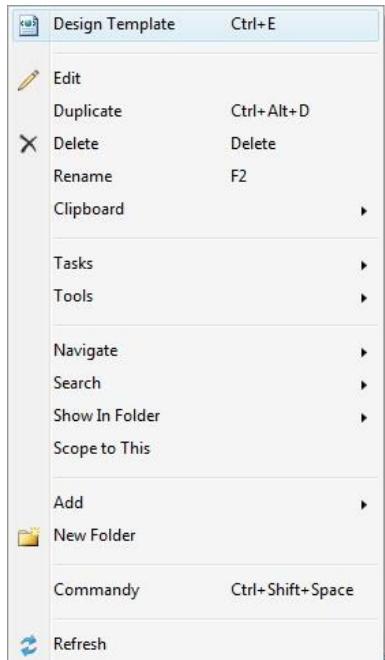
Over the past twenty-five days we have covered many of the major features of Sitecore Rocks, but there are also numerous minor features that can be real time savers. Today we will be discussing three of these "hidden gem" features: Favorites, My Items, and Adding Multiple Items.

Favorites

One of the little known hidden gems of Sitecore Rocks is Favorites. If you are the type of person who works exclusively with the context menus then you will find favorites extremely valuable. Favorites in Sitecore Rocks gives you the ability to mark a menu item and have it display at the very top of the context menu even if it's buried in a sub menu. To make a menu item a favorite, hold down the ctrl + shift keys and right click an item to open the context menu. What you should see is the context menu with gray stars next to the menu items.

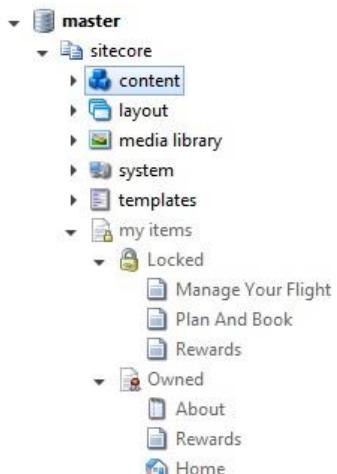


Clicking one of the gray stars will mark the menu item as a favorite and bring it to the top of the menu next time you open it normally. This can be a real time saver for items that are frequently accessed and are in sub menus.



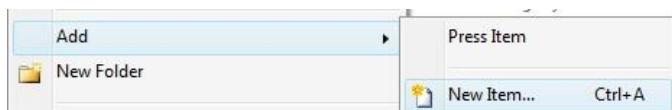
My Items

The My Items section of the Sitecore Rocks tree may not be hidden, but can be useful nonetheless. The My Items section can show which items we have locked and which items we own.

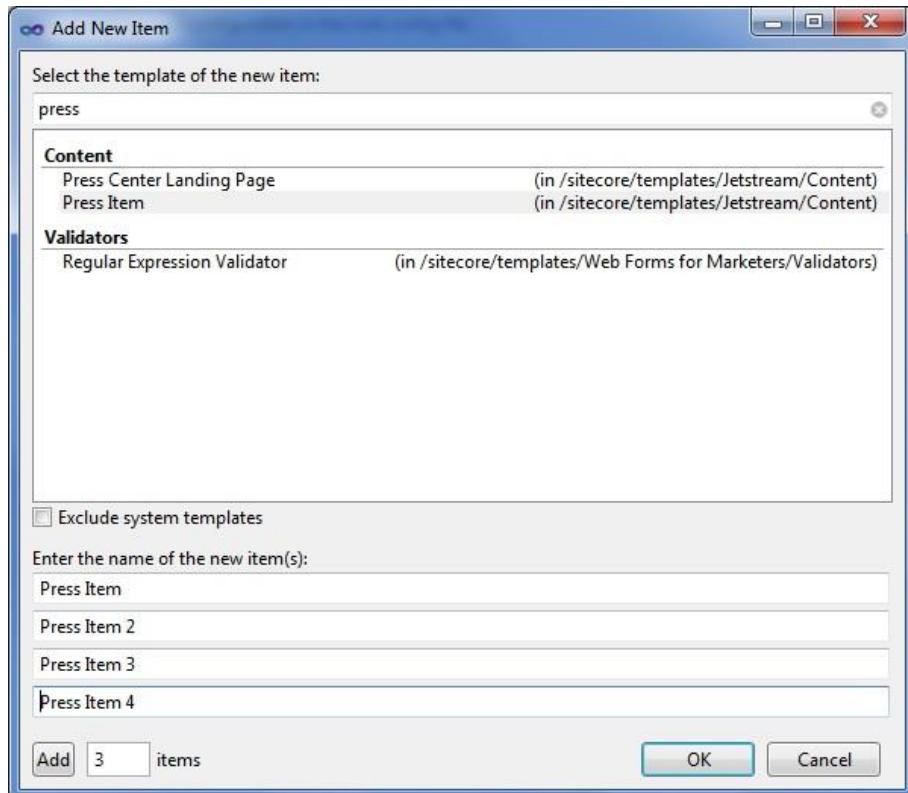


Adding Multiple Items

Ever wish there was a way to create multiple items of the same template? Maybe you need to test the functionality of a paging control and the thought of creating a block of items doesn't sound like the best use of your time? Luckily for us Sitecore Rocks can help. To create multiple items, right click the location where we would like to create the items and under the "Add" menu, click "New Item" instead of the insert option.



This will open the Add New Item dialog where we can select the template of the new item we want to create. To create multiple items, enter the number of items you would like to create in the textbox down in the far left hand corner and click the add button. This will add a line for an item name for each item to be created.



After clicking "OK" the multiple items will be created with the item names specified!

The Favorites, My Items and Adding Multiple Items are a few of the lesser known features of Sitecore Rocks but they still provide great value to developers. These "hidden gems" will hopefully come in handy for you as they have for me!

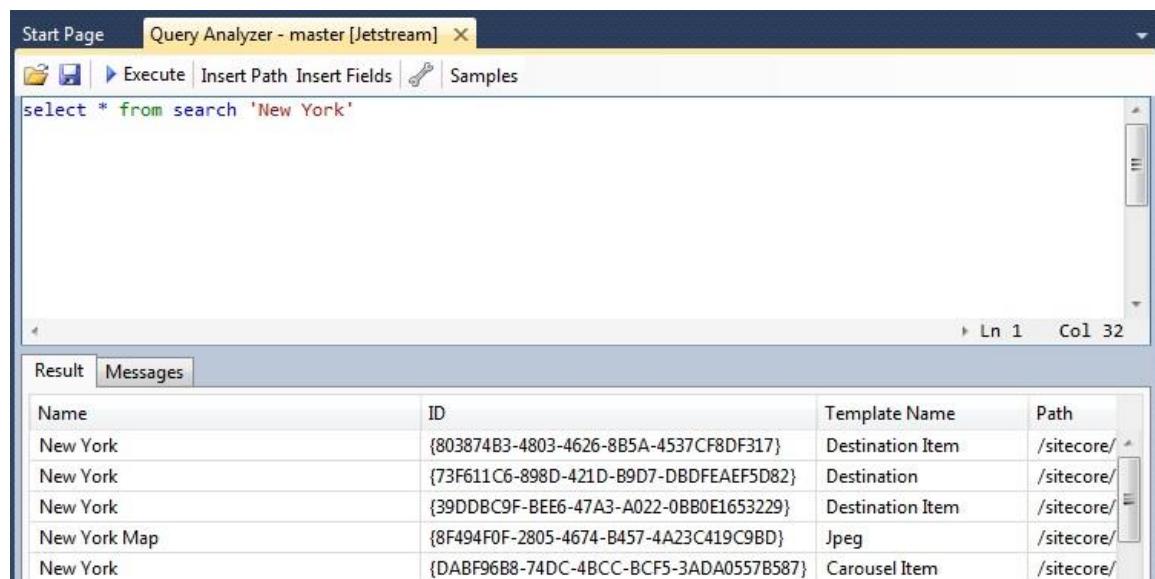
27: 28 Days of Sitecore Rocks: Hidden Gems - Part 2

By [Trevor Campbell](#), February 27, 2013

Let's continue discussing a few more of Sitecore Rocks' hidden gems. Today we will be covering searching Lucene indexes in the Query Analyzer, Search and Replace, and the Expanded web.config features of Sitecore Rocks.

Searching Lucene in Query Analyzer

One of the undocumented features in the Query Analyzer is the ability to search our Lucene indexes. To search Lucene indexes simply add the "search" keyword after the "from" statement. This will return results from which we can navigate to the corresponding item in Sitecore.



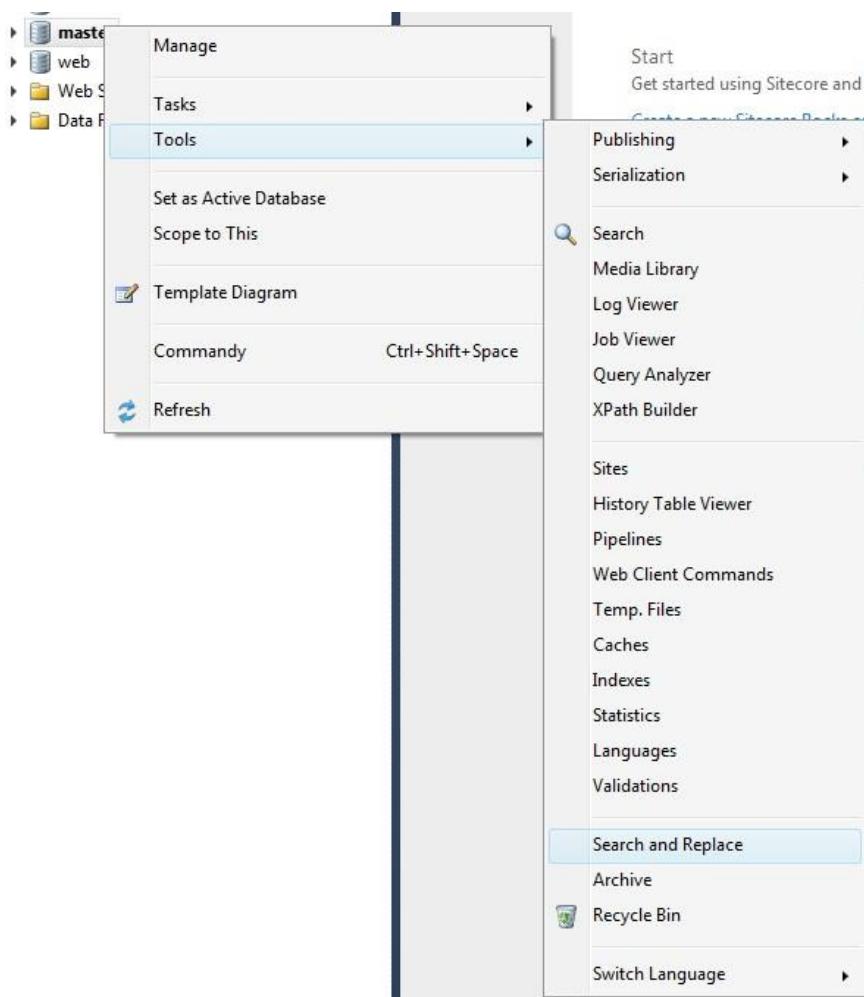
The screenshot shows the Sitecore Query Analyzer interface. The title bar says "Query Analyzer - master [Jetstream]". The toolbar includes "Start Page", "Execute", "Insert Path", "Insert Fields", "Samples", and a magnifying glass icon. The query window contains the SQL-like command: "select * from search 'New York'". The results window shows a table with the following data:

Name	ID	Template Name	Path
New York	{803874B3-4803-4626-8B5A-4537CF8DF317}	Destination Item	/sitecore/
New York	{73F611C6-898D-421D-B9D7-DBDFFAEF5D82}	Destination	/sitecore/
New York	{39DDBC9F-BEE6-47A3-A022-0BB0E1653229}	Destination Item	/sitecore/
New York Map	{8F494F0F-2805-4674-B457-4A23C419C9BD}	Jpeg	/sitecore/
New York	{DABF96B8-74DC-4BCC-BCF5-3ADA0557B587}	Carousel Item	/sitecore/

Search and Replace

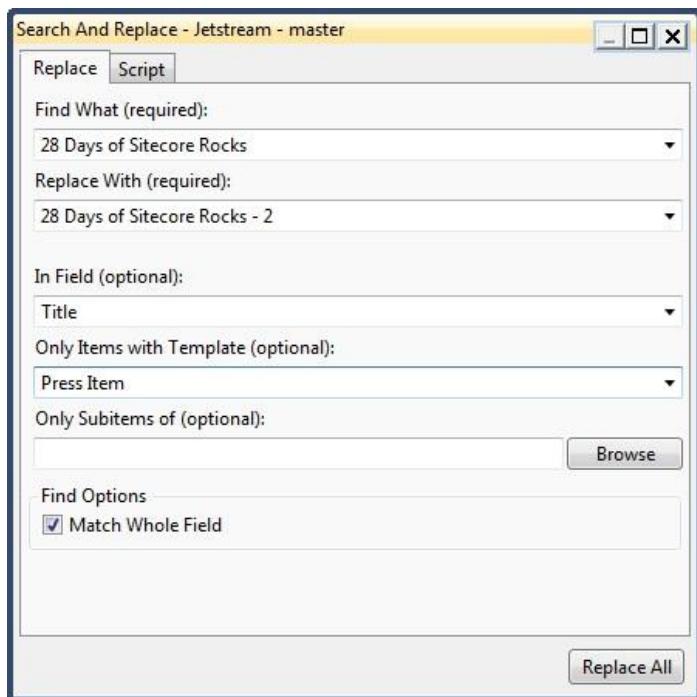
Another hidden gem in Sitecore Rocks is the Search and Replace function. The Search and Replace function gives us the ability to locate content in a Sitecore database and replace it with other content.

To open the Search and Replace screen, right click on a Sitecore database in the Sitecore Explorer and under "Tools" is the "Search and Replace" menu item.



/Community/Technical Blogs/Trevor Campbell 2/srfeb2701

The Search and Replace dialog is pretty straightforward, simply enter the text that you want to find and the text to replace it with. Optional parameters include which field, template and sub-items to search and replace on.



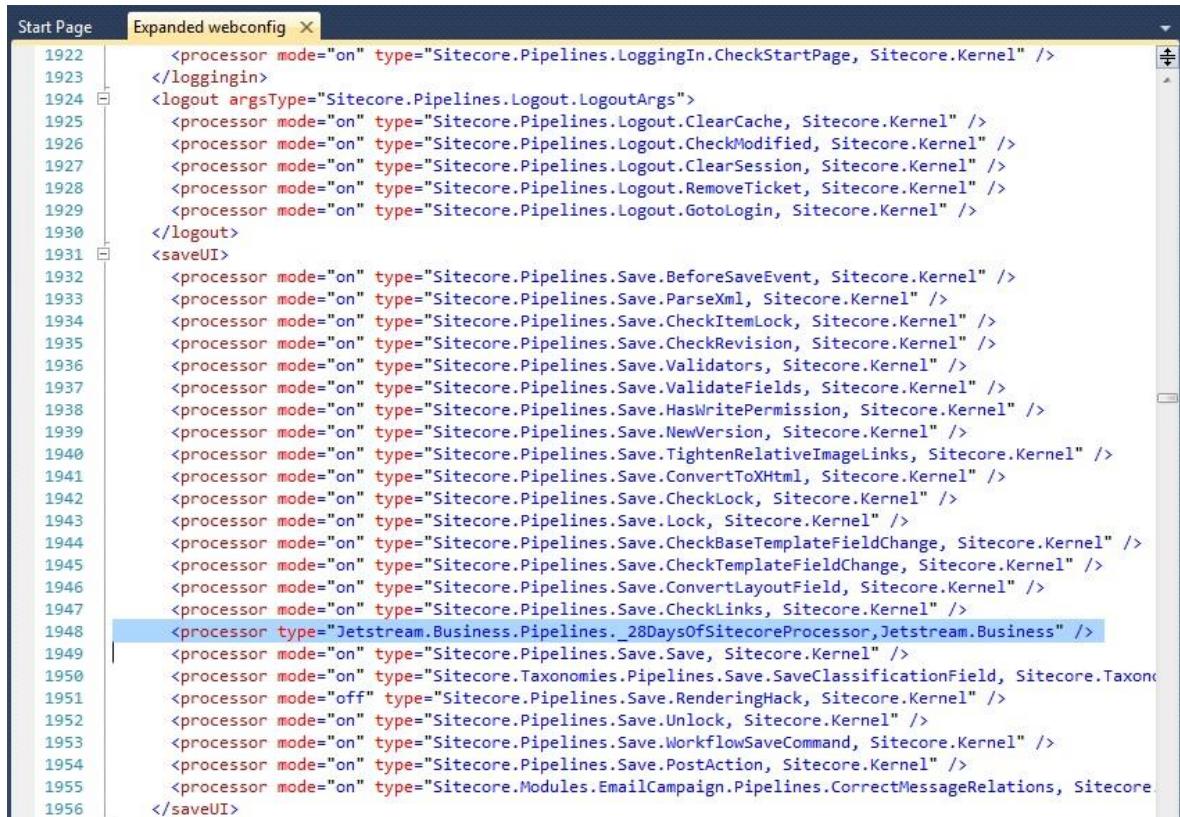
Behind the scenes the dialog is constructing a query for the Sitecore Rocks Query Analyzer to execute.

```
Search And Replace - Jetstream - master
Replace Script
replace "28 Days of Sitecore Rocks" with "28 Days of Sitecore Rocks - 2"
in #@Title#
from //*[#@Title# = "28 Days of Sitecore Rocks" and @@templatename="Press Item"]
```

Replace All

Expanded web.config

The Expanded web.config feature gives us the ability to see all the settings across each configuration file aggregated in a single location. If you are familiar with the ShowConfig page in Sitecore then this is along the same lines. It is a very helpful feature when you need to view all the configuration elements in relation to one another.



The screenshot shows the Sitecore Rocks IDE interface with the "Start Page" selected. A tab titled "Expanded webconfig" is active. The code editor displays a large portion of the Sitecore web.config file, specifically the `<configuration>` section. The code is numbered from 1922 to 1956. A blue rectangular selection highlights the following processor definition:

```
<processor mode="on" type="Jetstream.Business.Pipelines._28DaysOfSitecoreProcessor, Jetstream.Business" />
```

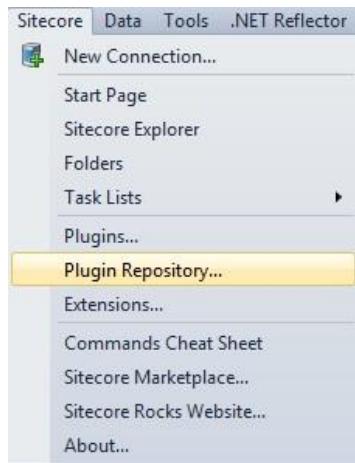
The Search Lucene in Query Analyzer, Find and Replace, and Expanded web.config are a few of the lesser known features of Sitecore Rocks but they still provide great value to developers. These "hidden gems" will hopefully come in handy for you as they have for me!

28: 28 Days of Sitecore Rocks: Plugin Repository

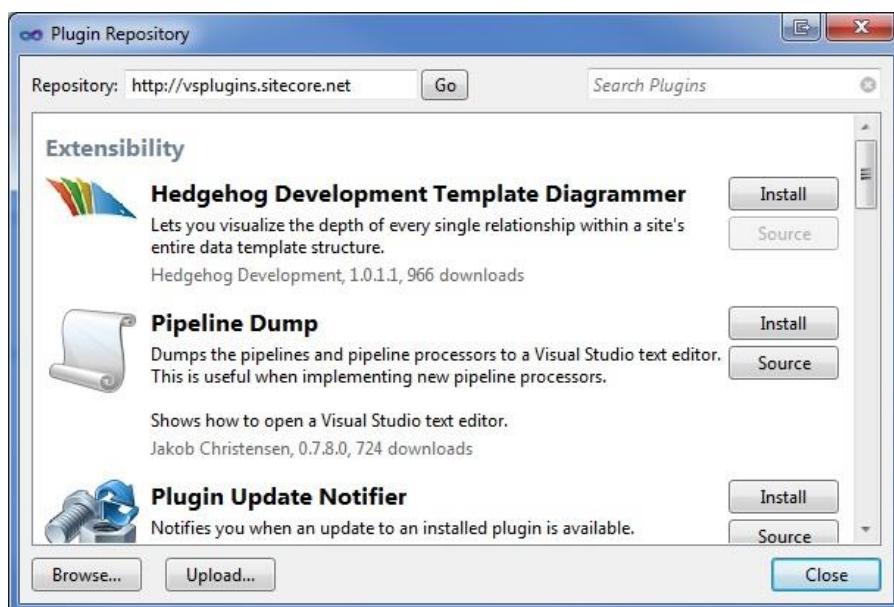
By [Trevor Campbell](#), February 28, 2013

On the final day of the 28 Days of Sitecore Rocks we will be covering the Plugin Repository. Similar to the extensibility of the Sitecore CMS, Sitecore Rocks can also be extended to great lengths. Didn't find a piece of functionality you expected to see in Sitecore Rocks? The first place to look is the Sitecore Rocks Plugin Repository.

To open the Plugin Repository, click the "Sitecore" menu item in the menu bar and click the "Plugin Repository."



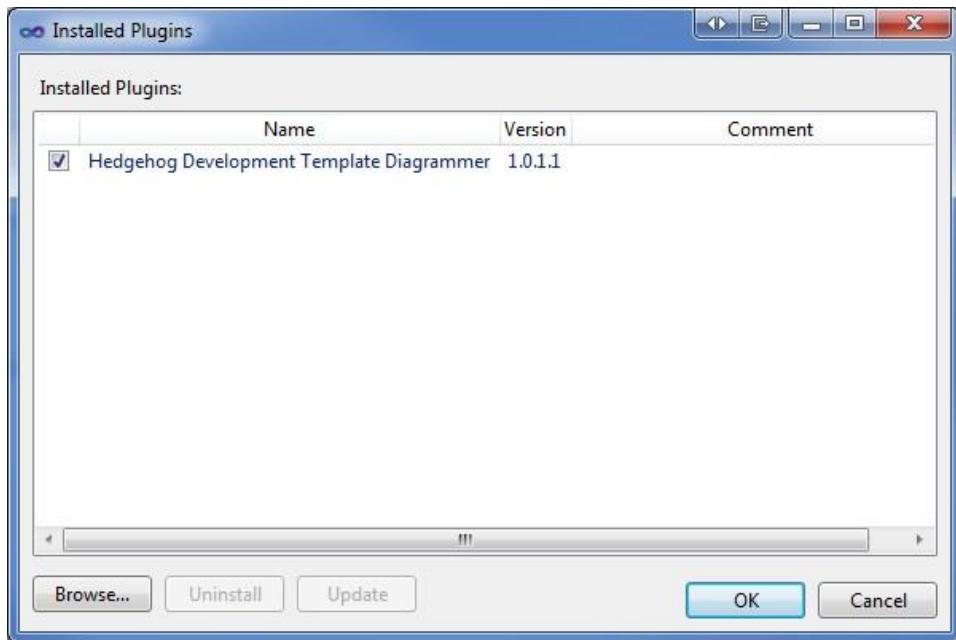
The Plugin Repository interface is simple and straightforward. The plugins are grouped by category and each plugin provides a title, description, developer, version and number of downloads as well as a button to install the plugin. We also have the option to search the repository if that is your preferred method of navigation.



The plugins are also available from the [Sitecore Rocks website](#) on the [plugins page](#).

Clicking the install button will start the plugin installation process which is typically fairly quick. Depending on the type of plugin being installed you may need to update the Sitecore Rocks server components to enable the plugin's functionality.

Once we have installed a number of plugins, we can manage the collection of installed plugins using the Installed Plugins tool. Using the Installed Plugins tool we can get information about uninstall and update currently installed plugins.



The Sitecore Rocks Plugin Repository is a great tool to extend the functionality of Sitecore Rocks. I highly recommend at very least having a look at the available plugins to see if there is anything of interest.

This concludes the 28 Days of Sitecore Rocks! Hopefully over the past 28 days we have learned how Sitecore Rocks can dramatically improve Sitecore development, and all from within Visual Studio.

If there are any other Sitecore Rocks topics you would like more information about, or that I didn't cover, leave a comment below.