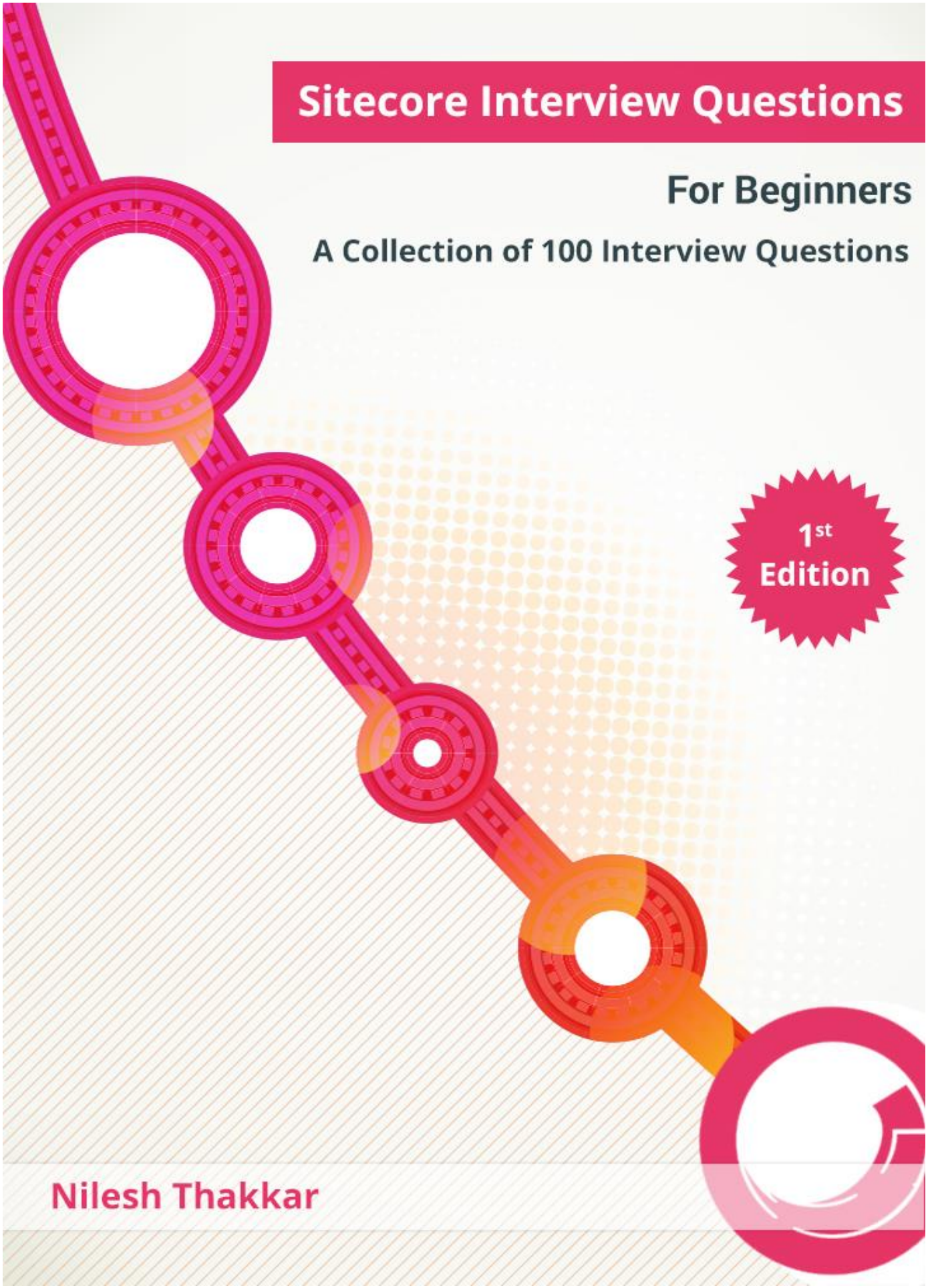
Sitecore Interview Questions

For Beginners

A Collection of 100 Interview Questions



Nilesh Thakkar

Contents

1) What is Sitecore? 7
2) Why should you use Sitecore? 7
3) How can you setup Sitecore? 7
4) Which are different versions of Sitecore? 7
5) First look at Sitecore? 8
6) What is Desktop? 10
7) What is Content Editor? 10
8) What is Page Editor? 10
9) How many databases are associated with Sitecore? 11
10) What is core database? 11
11) What is master database? 11
12) What is web database? 12
13) What is content author? 12
14) What is Item in Sitecore? 12
15) What is Template in Sitecore? 12
16) What is publishing? 12
17) Which are different types of publishing? 12
18) What is Staging Environment? 13
19) What is Production Environment? 13
20) What is versioning in Sitecore? 14
21) Which are different types of view modes in Sitecore? 14
22) What is standard values in Sitecore? 14
23) What is standard fields? 14
24) What is raw values? 14
25) Which field contains item sort value? 17
26) What is Layout in Sitecore? 17
27) Which are different types of rendering in Sitecore? 17
28) What is difference between droplist and droplink field type? 18
29) What is pipeline? 19
30) What is processor? 19
31) Under which pipeline Sitecore context gets defined? 19
32) How many processors are there under <httpRequestBegin> pipeline? 20

33) What is command? 20
34) Which are the steps to add a new command in Sitecore? 20
35) Which options are provided by Sitecore interface to move items from one instance to another instance? 20
36) What is difference between Package and Serialization? 22
37) Which are the options to move items from master database to web database? 24
38) What is difference between Publish and Transfer? 25
39) What is scheduler? How can you define scheduler in Sitecore? 26
40) What is publishing target? Where can you define publishing targets in Sitecore? 28
41) What is workflow? How can you check current workflow status of an item? 30
42) Which are different tokens supported by Sitecore? 33
43) How can you check particular item is referred by which other items in Sitecore? 33
44) How can you add help text for particular field in Sitecore? 34
45) What is display name? Why one should use it? 35
46) Which characters are not supported by default for item name in Sitecore? 36
47) How can you configure log folder path of your Sitecore site? 37
48) Where does Sitecore store all packages? How can you configure this package folder path? 37
49) Which are different types of caching in Sitecore? 37
50) Where can you add a new site in Sitecore? 39
51) Where should you overrides default configuration in Sitecore? 42
52) Which are different admin tools provided by Sitecore? 42
53) Which search mechanism is used by Sitecore default? 46
54) Which are other search mechanism industry is using in Sitecore? 46
55) Which are different types of HTML Editor profiles available in Sitecore? 47
56) Where can you configure default HTML Editor profile in Sitecore? 48
57) From where can you configure caching of rendering controls? 48
58) How to get current item in Sitecore? 50
59) How to get home page item in Sitecore? 50
60) How to get string field value in Sitecore? 50
61) How to get other field value in Sitecore? 50
62) How can you get all child items of item which has particular template? 50
63) If you have image field in item and you want to get that image field URL in order to set on your image control, how can you achieve it using Sitecore API? 51
64) How can you get reference of database in Sitecore? 51

65) How to retrieve a template by ID in Sitecore? 51
66) How to retrieve a template by name in Sitecore? 51
67) How to switch database from content editor? 51
68) How to retrieve the content database reference in Sitecore? 52
69) Which is method provided in Sitecore to write output to HTML outstream? 52
70) How to retrieve current device that is getting used by user? 52
71) What is inline editing in Sitecore? 53
72) How to display/render field value to support inline editing? 53
73) How to access Sitecore page in edit, preview and normal mode using URL? 54
74) How retrieve selected item from Sitecore Droplink field? 54
75) How retrieve selected item from Sitecore Generic link field? 54
76) How retrieve selected item from Sitecore Multilist field? 54
77) Suppose you want to check that which items are referring particular item and you want to retrieve list of these referrer items. How can you get it? 55
78) What is difference between item.Fields["Title"].HasValue and item.Fields["Title"].Value? 55
79) Which are different types of validations in Sitecore? 55
80) Which of the validations getting applied by Sitecore in Page Editor? 56
81) What is event queue and how does it work? 56
82) Where are all events stored in event queue? 56
83) What is CM server and CD server? 56
84) What is Sitecore DMS? 57
85) Which are three ways to trigger goal in sitecore DMS? 64
86) What is shared and unversioned fields? 66
87) Which are new features in Sitecore 7? 66
88) What is item bucket and what is advantage of it? 67
89) How to configure image quality when you resize image using querystring parameter like w (width) and as (allow stretch)? 67
90) How to reset engagement profile value in Sitecore DMS? 67
91) What is the best way to play with querystring in Sitecore? 67
92) How to change height, width and other parameters of image dynamically in Sitecore? 68
93) When should you use Sitecore Clean Up Database feature? 68
94) How to add a new CSS class in rich text editor dropdown? 69
95) How to hide an item in Sitecore? 69
96) How to see hidden items in Sitecore? 70

97) Which of the following Sitecore DLLs cannot be decompiled? 70
98) How to report issue to Sitecore team? 70
99) Where to find third party tools or modules about Sitecore? 70
100) From where can you get more help about Sitecore? 70

Disclaimer

This book is presented solely for educational purposes. This book is designed to provide information on Sitecore CMS only. This information is provided with the knowledge that the author do not offer any legal or other professional advice. In the case of a need for any such expertise consult with the appropriate professional. This book does not contain all information available on the subject. This book has not been created to be specific to any individual's or organizations' situation or needs. Every effort has been made to make this book as accurate as possible. However, there may be typographical and or content errors. Therefore, this book should serve only as a general guide and not as the ultimate source of subject information. This book contains information that might be dated and is intended only to educate and entertain. The book may include information by third party blogs or websites. Third Party materials comprise of the products and opinions expressed by their owners. As such, the authors of this book do not assume responsibility or liability for any Third Party Material or opinions. The author shall have no liability or responsibility to any person or entity regarding any loss or damage incurred, or alleged to have incurred, directly or indirectly, by the information contained in this book. You hereby agree to be bound by this disclaimer.

Thanks To,

Sitecore Community

What is Sitecore?

Sitecore is very flexible Content Management System (CMS). It is one of the leading enterprise-level content management systems (CMS) using which you can create enterprise website and intranet portal website.

Sitecore was found in 2001 and it is built on Microsoft .NET platform. Sitecore also allows deployment via Microsoft Azure.

Why should you use Sitecore?

Here are the top features of Sitecore CMS that separates it from other CMS:

- ➔ It provides out-of-the-box flexibility and is highly customizable
- ➔ With Sitecore you can create multi-language site with little effort
- ➔ Provide personalized user experience and content variation features
- ➔ E-commerce Services
- ➔ It has out-of-the-box engagement automation and engagement analytics
- ➔ Email Campaign Manager
- ➔ Integration with Cutting Edge Technology
- ➔ Highly Scalable
- ➔ Improved Search Engine Optimization
- ➔ User based and Role based Security
- ➔ Over 350 free Sitecore Module available to use
- ➔ Growing community

How can you setup Sitecore?

Initially Sitecore was providing free express edition to developers for hands on which is now no longer available. You need to download Sitecore from Sitecore Developer Network (SDN) - http://bit.ly/VtJaeo. Please note that you will need license file to install Sitecore CMS.

Which are different versions of Sitecore?

Sitecore Version	Year
Sitecore 5	2007
Sitecore 6.0	2008
Sitecore 6.1	2009
Sitecore 6.2	2009
Sitecore 6.3	2010
Sitecore 6.4	2010
Sitecore 6.5	2011

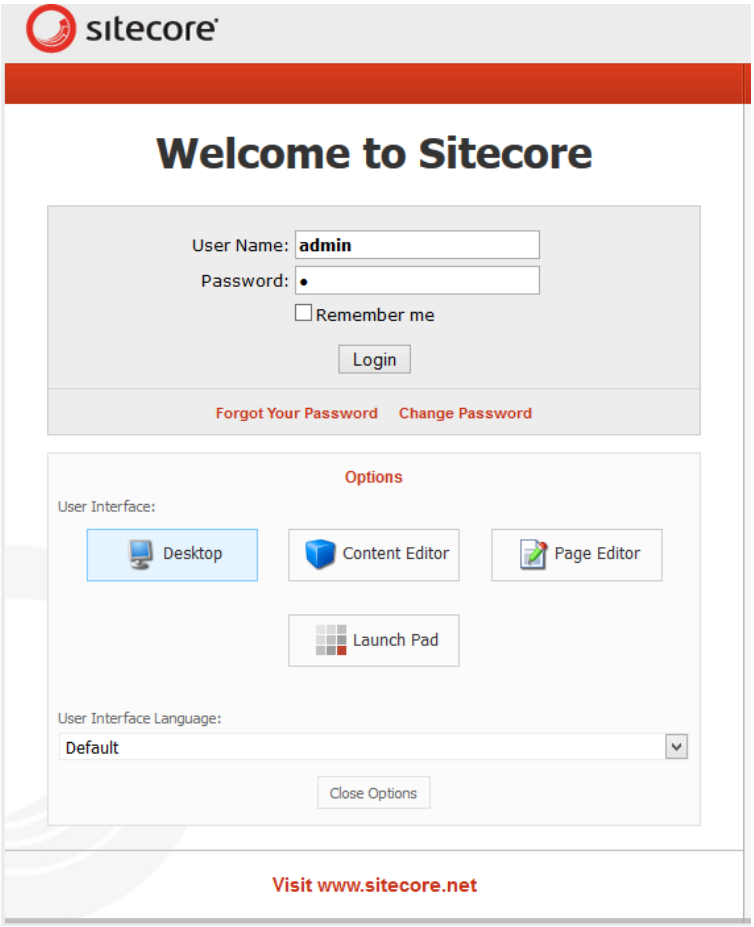
Sitecore 6.6	2012
Sitecore 7.0	2013
Sitecore 7.1	2013
Sitecore 7.2	2014
Sitecore 7.5	2014
Sitecore 8.0	Coming soon

First look at Sitecore?

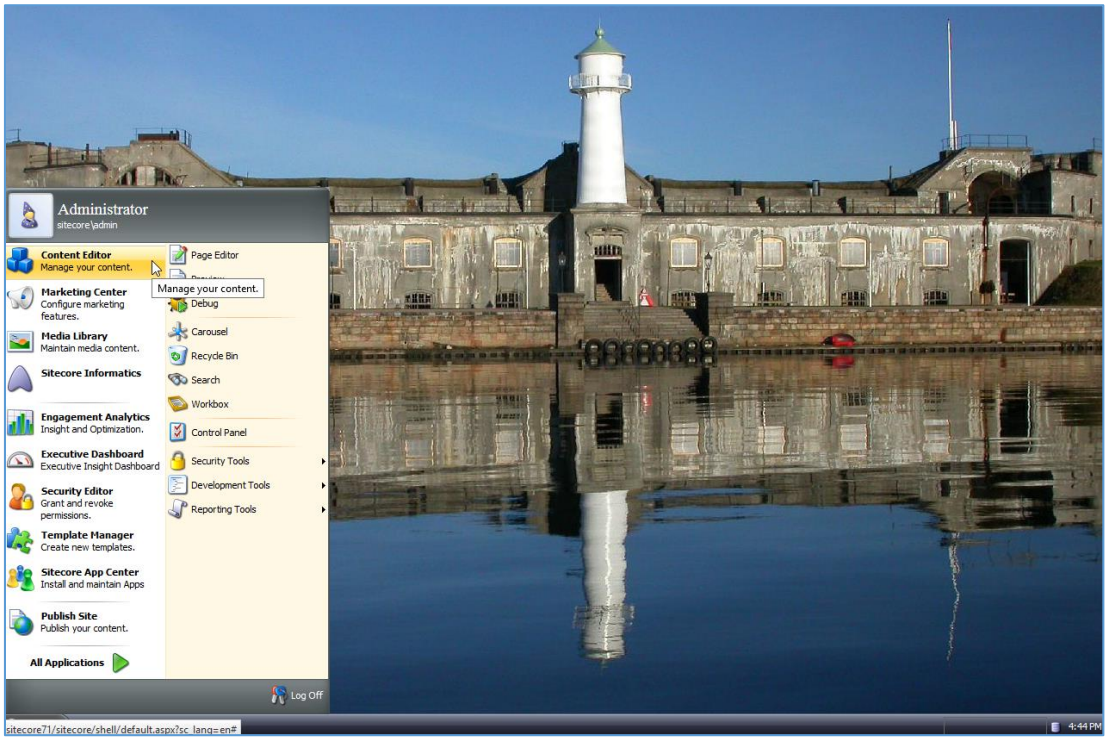
Sitecore user interface is very similar in look and feel to windows desktop, which makes it easy for new users to pick up and learn Sitecore.

There are three ways you can login to Sitecore – Desktop, Content Editor and Page Editor.

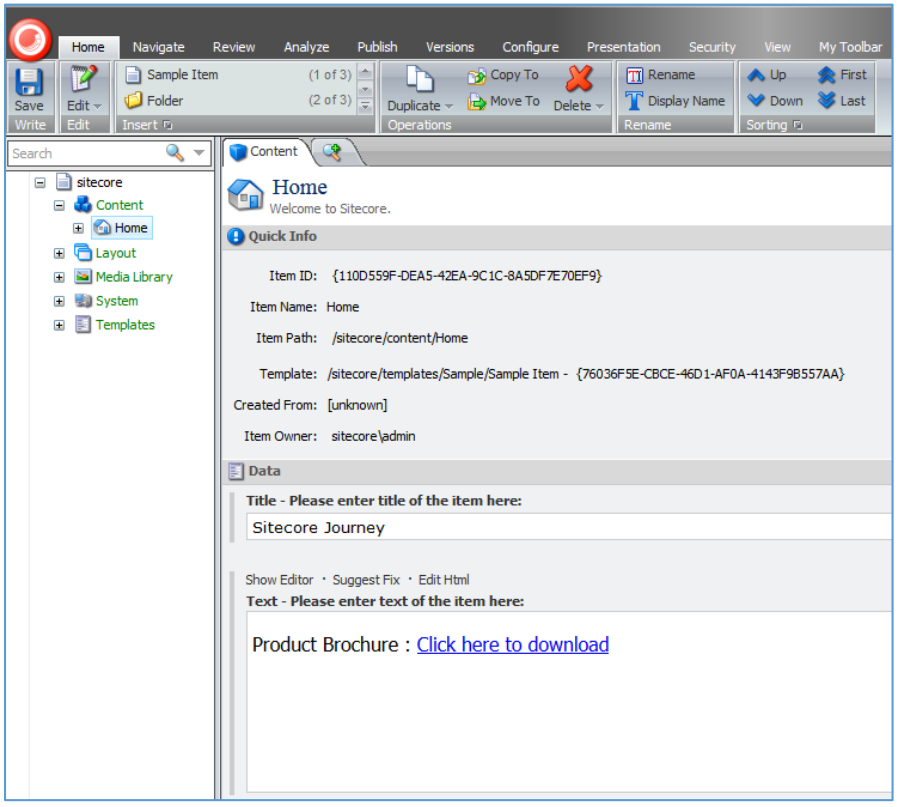
Sitecore login screen:



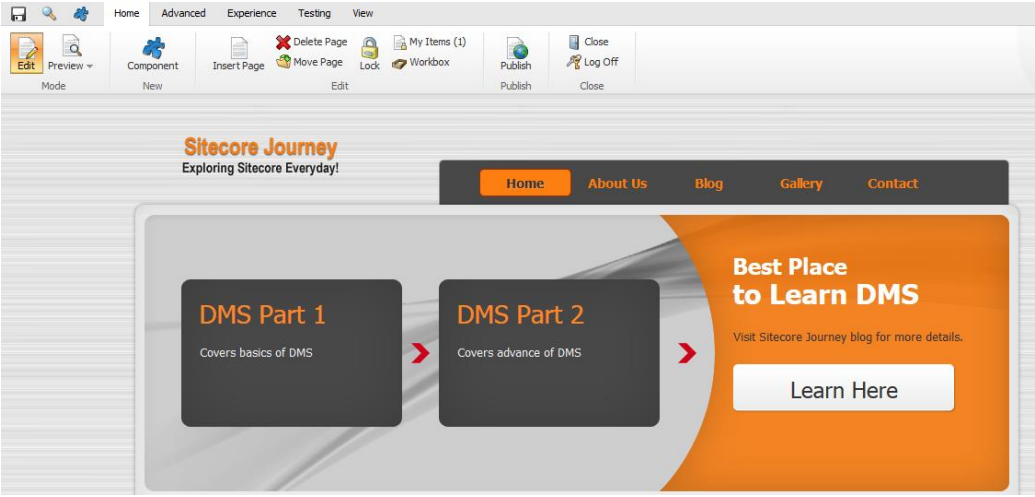
Sitecore Desktop screen:



Sitecore Content Editor screen:



Sitecore Page Editor screen:



What is Desktop?

When you are login using Sitecore Desktop you will find rich set of features using which you can

- ➔ Create and Install packages
- ➔ View files using File Explorer
- ➔ Clean up database using Control Panel
- ➔ Change the desktop background
- ➔ Change the application options
- ➔ Many more

Sitecore Desktop user interface has many feature options compare to Content Editor user interface.

What is Content Editor?

The Content Editor is designed for more **experienced** content authors who feel comfortable working with a computer, Microsoft Word and familiar with Sitecore functionality to add, edit or remove content of website using rich set of features.

Content Editor Interface has less feature options compare to Desktop interface but have more feature options compare to Page Editor Interface.

What is Page Editor?

Content authors are generally focused on the quality and accuracy of the material displayed on the Web site and not on the design, style, and layout of the site. For this reason most content

authors prefer to work in such interface that is designed to meet their needs and not those of the developers and designers who create the Web sites.

With this mind, Sitecore CMS allows content authors to edit and create items directly on the Web page with the Page Editor.

The Page Editor is the simplest of the editing tools that Sitecore contains. It is designed for **inexperienced** content editors who edit and write content items directly on the page. It is a WYSIWYG editor (what you see is what you get) and limits the amount of functionality that is presented to the user.

When you log in to the Content Editor, the items that make up your Web site are presented to you in a very different way than they are in the Page Editor. In the Page Editor, you see the items as they are presented on the Web site.

How many databases are associated with Sitecore?

There are three databased associated with Sitecore:

1) Core database
2) Master database
3) Web database

What is core database?

This core database has two purposes:

1) The core database contains all Sitecore settings. It contains definitions for the Sitecore user interface (Content Editor, Page Editor, etc.). This is where you can add context menu option to the Content Editor or a new ribbon button to the Page Editor.
2) It contains the ASP.NET membership tables that drive authentication and security.

What is master database?

The master database is the authoring database - it contains all versions of content. Whenever a new piece of content is created, edited or deleted it is stored here. This is the database that content authors interact with, and is the default database used by the Sitecore Content Editor.

What is web database?

The web database contains the live content of the website. When a content editor publishes some content then it is copied from the master database to the web database. It is subset of the master database and optimized for size and speed.

What is content author?

As per wiki - A website **content writer** or **content author** is a person who specializes in providing relevant content for websites. Every website has a specific target audience and requires a different type and level of content. Content should contain words (key words) that attract and retain users on a website. Content written specifically for a website should concentrate on a specific topic. It should further be easy to read, offering the information in easy to understand clusters or laid out in bullet points.

What is Item in Sitecore?

An item is a record in database. Items are basic building block of a Sitecore Site. An item may represent any kind of information, e.g. a piece of content, a media file, a layout etc. Items always have a name and ID that uniquely identifies the item within the database. Items have a template that defines which fields the item contains. An item represent a single version of piece of content is a single language.
An item can be retrieved from a database using **Items**.

What is Template in Sitecore?

Sitecore uses data templates to define structures of item. Templates contain fields to represent individual elements. Each data template field has a type, such as Text, Image, Checkbox and various other field types.

What is publishing?

By this time you are now aware with the fact that whenever you add, edit and delete any item in Sitecore, it will be stored in master database. Now in order to move this change to live website, you need to use publishing. Publishing is a process which will help you to copy updated items from master database to web database.

Which are different types of publishing?

1) Republish

As name implies, it will publish every item no matter whether it is changed or not. It is intended to be used when you are publishing a new site first time. This is most time consuming publishing method as it blindly publish all items.

2) Smart Publish

This method works smartly by comparing each item in the master database with the item in web database. Sitecore maintains revision number for each item which gets changed whenever the item gets updated. By comparing this revision number with web database, it will create a list of updated items changes and will publish only those items changes. Even though this method is comparing each item, it is much faster than republish method.

3) Incremental Publish

Every time an item is changed, it is added to the publishing queue. This applies both to changes made through the Sitecore user interface and changes made programmatically. Doing an incremental publish will only publish the items in the publishing queue. Therefore only items that has been changed will be published and Sitecore does not have to do any comparisons to figure out which items has been changed. This way of publishing is therefore by far the fastest. Republish and smart publish do not use the publishing queue.

What is Staging Environment?

As a best Sitecore practice, there should be separate Sitecore instance where all changes should be deployed before they are moved to live site (production environment). This Sitecore instance is treated as staging environment.
A meaningful URL is given to stage environment like: http://stag.yoursite.com

Once user is happy with changes on staging environment, deployment is then roll-out to production environment.

What is Production Environment?

Like staging Sitecore instance, there should be separate Sitecore instance for production environment. Production environment is nothing but a live site which is getting used by outside world. If this Sitecore instance gets down, that means your live website is down.

Respective live URL of above stage URL http://stag.yoursite.com would be http://www.yoursite.com

What is versioning in Sitecore?

In Sitecore content authors have the ability to make a piece of content version able. Each version of an item begins as a replica of the original or of another version and are all stored in Sitecore. Previous versions operate individually and can be published as the active version of a content item instead of the most recent version. At any time an active version can be rolled-back to a previous version and vice versa.

Which are different types of view modes in Sitecore?

1) **Preview mode:**
2) **Page-Editor mode:**
3) **Normal mode:**

What is standard values in Sitecore?

Standard values were first introduced in Sitecore 5.3 to overcome some of the issues experienced with the predecessors of standard values – "Masters". In Sitecore 6 masters was completely removed and the concept of standard values took over.

Standard values are a way of having default or fallback values for fields in Sitecore, meaning that when items are created, you can specify a field value that should be used by default. This does not only account for custom fields you build, but also standard fields in Sitecore such as presentations and insert options. This means that you can specify a value on the standard values, and when you create a new item which inherits from this template, it will by default use the values specified on the standard values.

What is standard fields?

Sitecore provides few common fields for all items, these fields are called standard fields.
These fields can help you to:

- ➔ Set icon of your Sitecore item
- ➔ Check last item update status
- ➔ Check the current status of item's workflow
- ➔ And many more…

What is raw values?

Sitecore always stores field values as plain text in the appropriate Sitecore database (in the Value column of the Fields table).

In Sitecore, there are two ways of getting data out of a field – the raw value or the rendered value.

A field's raw value is how the data is stored in the underlying database – for simple fields such as Single Line Text, there is no difference between the rendered and raw values; it's just plain text.
This is not the case for more complex field types, such as **General Link, Image**, or **Rich Text.**

Take the raw value of a General Link field, for instance – if you enable raw values in the Content Editor (by clicking the **View** tab and ticking **Raw values**) and look at an item with a General Link field, you will see that the value stored in the database is **custom XML**:

<link linktype="internal" url="/Home/History/Old activities" target="" id="{CB2BC023-C9CF-4086-811E-0952ADFA5AF9}" />

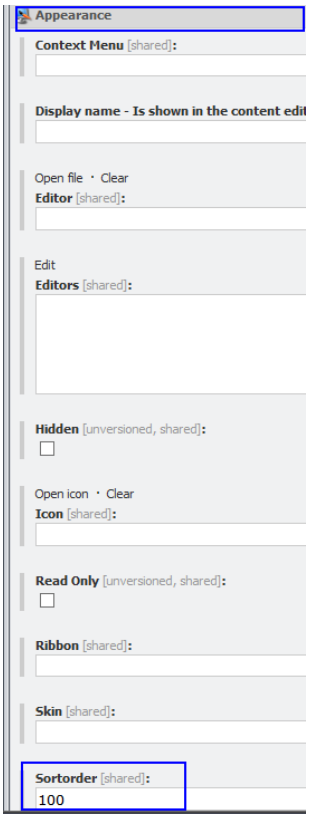
The following table provides an example of each field type, as the value is stored in the Sitecore database.

Field Type	**Example**
checkbox	1
checklist	{E00D9E00-2582-4022-88AE-151CE3CD7C41}|{858D25B3-B8A6-4ED1-B6F8-B52C4CC0EFDF}
date	20051003T120000
datetime	20051003T163256
file	/images/extimages/outh gif
html	This is some example HTML content.
icon	/sitecore/shell/Themes/standard/carousel.png
image	<image alt="Alternate text" width="150" height="150" hspace="4" vspace="4" mediaid="{539587C7-1F36-4D94-A10D-FFFAF079B746}" mediapath="/images/hist_gunung100px jpg" showineditor="1" usethumbnail="1" src="/upload/images/hist_gunung100px.jpg"/>
internal link	/sitecore/content/Home/Sample Forms
layout	<r ><d id="{FE5D7FDF-89C0-4D99-9AA3-B5FBD009C9F3}" l="{14030E9F-CE92-49C6-AD87-7D49B50E42EA}"><r id="{493B3A83-0FA7-4484-8FC9-4680991CF743}" ph="content" ds="" par="Param1=Value1" cac="1" vbd="1" vbu="1" /></d><d id="{46D2F427-4CE5-4E1F-BA10-EF3636F43534}" l="{14030E9F-CE92-49C6-AD87-7D49B50E42EA}"><r id="{5D104030-657E-48DA-B032-0F9BB1A3C671}" ph="content" ds="" par="" /><r id="{E0E16B09-0176-4756-978C-63459B19F8ED}" ph="content" ds="/sitecore/content/Home/Other selectable document"

	par="Param1=Paramvalue1&Param2=Paramvalue2" cac="1" vbdev="1" vbqs="1" /></d></r>
link	<link text="Sitecore" linktype="external" url="http://www.sitecore.net" anchor="" title="go to Sitecore" class="Normal" target="_blank" />
lookup	{858D25B3-B8A6-4ED1-B6F8-B52C4CC0EFDF}
memo	This is a test.
multilist	{632511FF-5E1D-4820-8379-CA5923328603}|{E00D9E00-2582-4022-88AE-151CE3CD7C41}
password	asdjlk
reference	{0D45476E-F67C-4F14-9B4E-C26D53F6EC48}
security	extranet|{A365D840-553C-4E76-9DFD-AC3BB58E80A1}|8||extranet|{24918F00-C53F-40D9-A8EB-DD25EBB131F9}|8448||
server file	/global.aspx
text	Text field
tree	{DB290BF5-0D3E-41FA-8C0F-DDBBE4EEF68B}
tristate	1

Which field contains item sort value?

Sortorder field under Appearance section contains item sort value.
Decrease sort value to move up and increase sort value to move down item in tree.



What is Layout in Sitecore?

Whenever user sends any request to Sitecore through a URL, Sitecore resolves that item by generating visual representation of that item. Layout provide the basic structure of this visual representation. Layout is container which contains different component of the page.

Technically speaking, your .ASPX page is layout and .ASCX, .CS or .XSLT file is component (user control).

Which are different types of rendering in Sitecore?

There are basically three types of rendering used by Sitecore developers.

1) Sublayout Rendering (mostly used)

    a. This is developer's favorite and most used rendering method. Sublayout is nothing but .ASCX user control.

2) Web Control Rendering (average used)

a. ASP.NET web controls are created to use this rendering. Create .CS file for this rendering.

3) XSL Rendering (rarely used)

a. Very few programmers are using this rendering as writing XSLT program is bit difficult and it has limited features. Create .XSLT file to use XSL rendering.

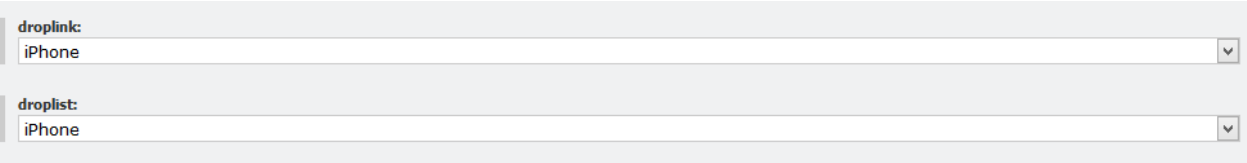
What is difference between droplist and droplink field type?

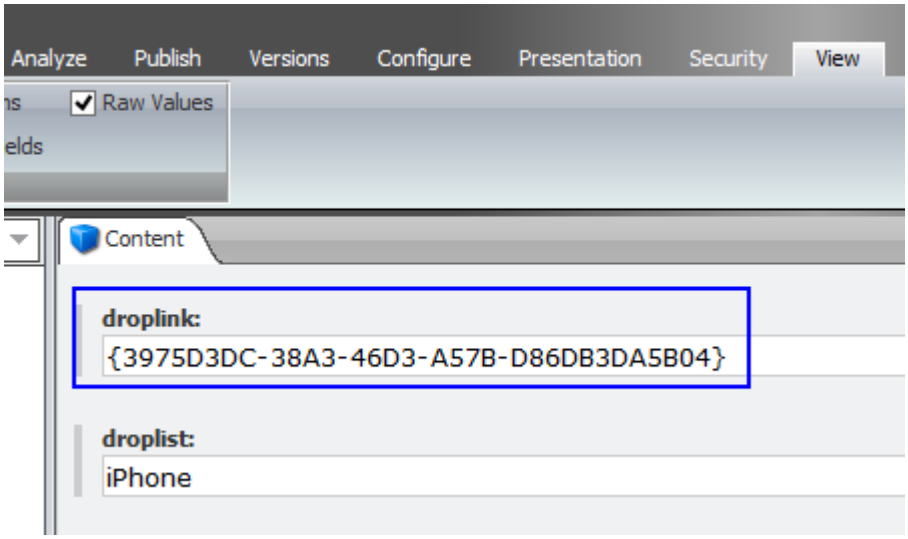
By the time you reach to this question, you must be knowing that Sitecore always stores field values as plain text in the appropriate Sitecore database (in the Value column of the Fields table).

Sitecore has two different types of drop-down lookup fields that are available: Droplist and Droplink. Both of them function essentially the same way: they point to a Sitecore data item and list its children in a drop-down list for a content editor. This is a great tool because it allows you (as a data designer) to build a nice set of lookup values and not worry that editors will fat-finger these values.

But why are there the two, and what's the difference between them?

Droplist data type only stores the string value of the item that was chosen by the content editor, while **Droplink** stores **the GUID** of the item that was chosen by content editor.





So you may be wondering – so what?

This will cause a problem when you want to read selected item's field value. If you use droplist, you are not going to get selected item and will get only selected item's string value. But if you use droplink, you can easily get selected item and then can read any field value of that selected item.

What is pipeline?

A lot of things happens when you request a page in Sitecore which is handled by pipelines. Pipelines define a sequence of processor that implement different functions such as handling page requests, to uploading files and saving items through the UI. If you have a series of tasks, which need to be performed to accomplish a task, then a pipeline may be the way to go.

Each processor in a pipeline contains a method named Process() that accepts a single argument and returns void. This method should return immediately if the processing context is not relevant to the processor. A processor can abort the pipeline, preventing Sitecore from invoking subsequent processors.

Sitecore separates the pipelines into two groups: those defined within the /configuration/sitecore/pipelines and /configuration/sitecore/processors elements in the web.config file. In general, those defined within the /configuration/sitecore/processors define pipelines that operate for UI requests and can interact with the user. Those defined with the /configuration/sitecore/pipelines tend to define system processes.

What is processor?

Each step in pipeline is called a 'processor'. The sequence of processors within a pipeline is defined in web.config file. Each Processor contains a unique operation and all these steps together create a pipeline.

<pipelines /> define system process such as initialize Sitecore application, load content editor warning. <processors /> define UI process such as copy item, delete item, drag item and so on.

Under which pipeline Sitecore context gets defined?

<httpRequestBegin> under configuration/sitecore/pipelines in web.config pipeline defines Sitecore context.

How many processors are there under <httpRequestBegin> pipeline?

By default there are 19 processors under <httpRequestBegin> pipeline but it may vary based on different modules you have installed in Sitecore.
For example: If you have installed Sitecore DMS then you will find 21 processors under <httpRequestBegin> pipeline.

What is command?

When you click any button within the Sitecore interface, the respective command gets called. Typically Sitecore commands are used to map actions to c# code. This mapping is defined in /App_Config/Commands.config file.

If you want to add your own commands you can either edit /App_Config/Commands.config file or setup a patch file in /App_Config/Include.

The Core database contains the command definition.

Which are the steps to add a new command in Sitecore?

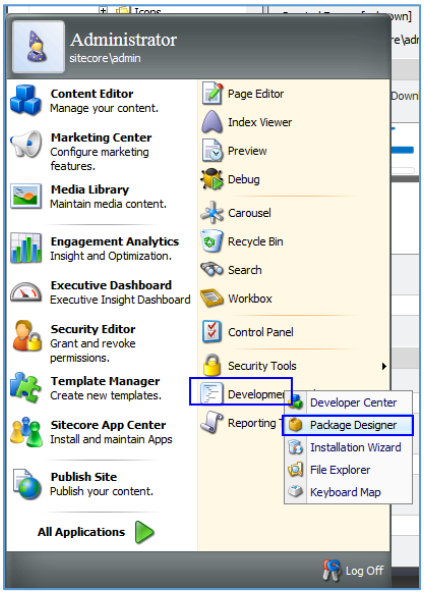
- ➔ Create a class that inherits from **Sitecore.Shell.Framework.Commands.Command** class
- ➔ Add your command either at /App_Config/Commands.config file or setup a patch file in /App_Config/Include.
- ➔ Add your command definition in core database

Which options are provided by Sitecore interface to move items from one instance to another instance?

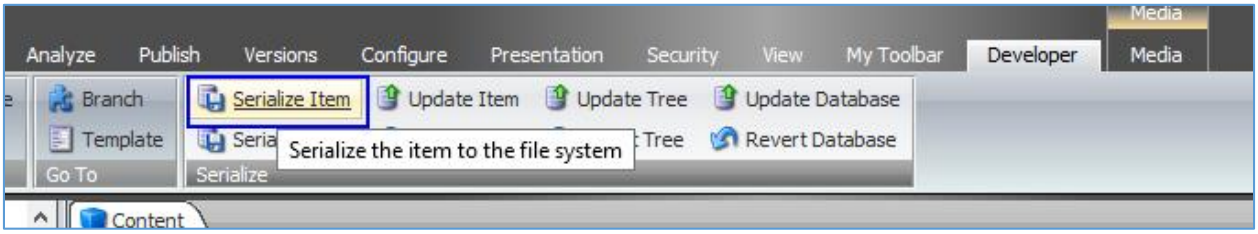
Once you are ready with your Sitecore website, and you want to give it to QA people to test on their Sitecore instance () – how can you achieve this?
In short, how to move your Sitecore items from Development environment to QA environment?

There are two options to achieve this requirement:

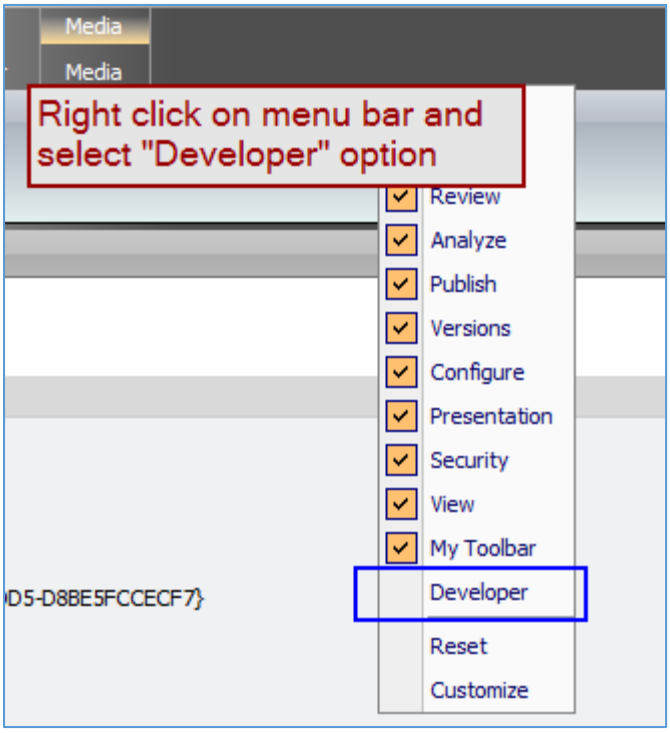
1) **Package**



2) **Serialization**



If you are not seeing “Developer” option in menu bar, then configure it as shown in below screen capture.



If you have question which is the good choice – Package or Serialization? Keep reading this book, it will be covered in next set of questions.

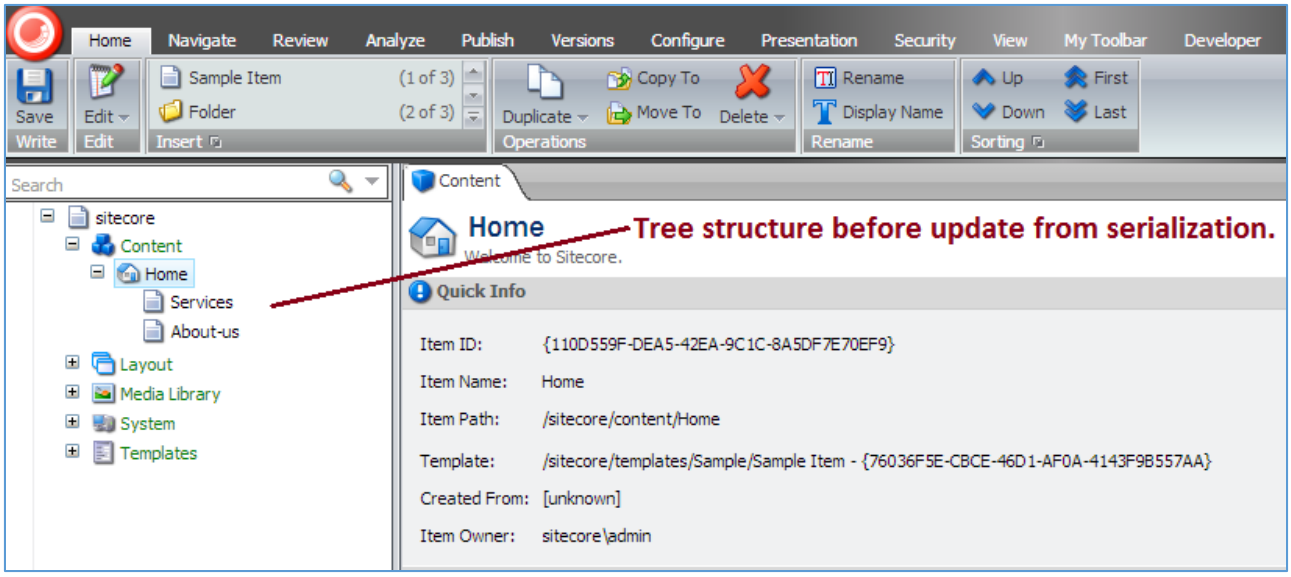
What is difference between Package and Serialization?

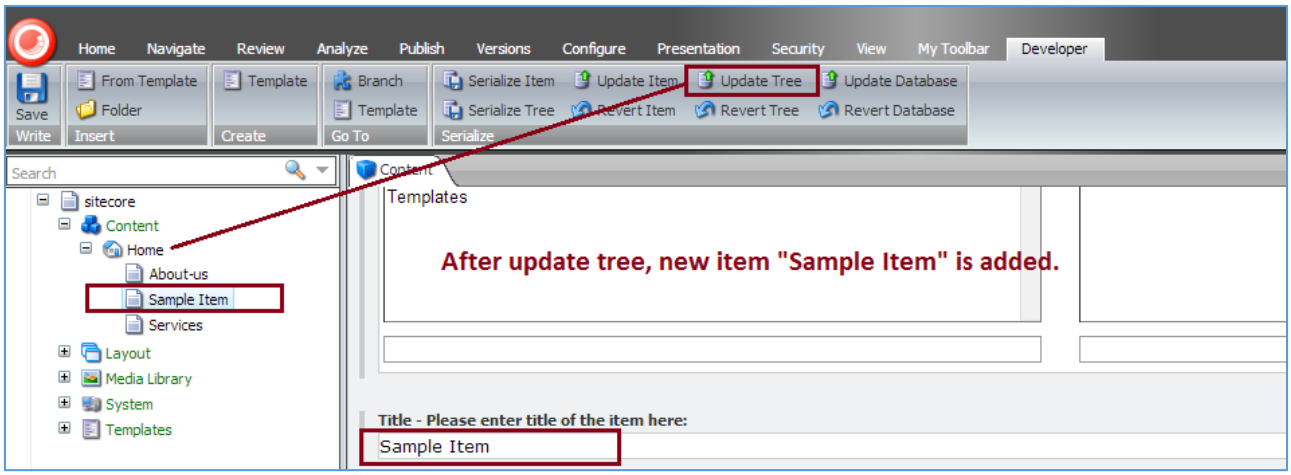
1) Sitecore Package:

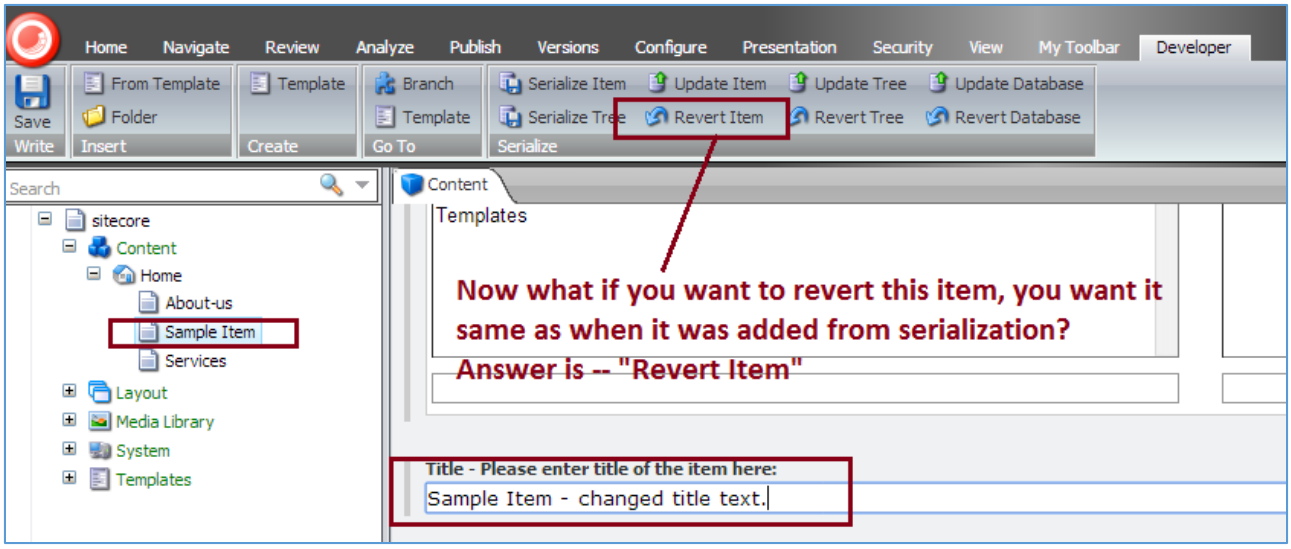
- There is size limit of 2 GB for package creation in sitecore, so you need to create multiple packages instead of creating one package if your deployment exceeds 2 GB size.
- After installing package, if you have changed few items and now you want to revert these items back to original version then there is no way to achieve it.
- For example: If your package contains 50 new items, then you will have 50 new items in your sitecore instance after installing this package. Now let say you have changed 10 of these 50 items and you want to revert these 10 items back to previous version (same version when you installed package), then there is no such mechanism in package installation which will help you to revert these specific 10 items.

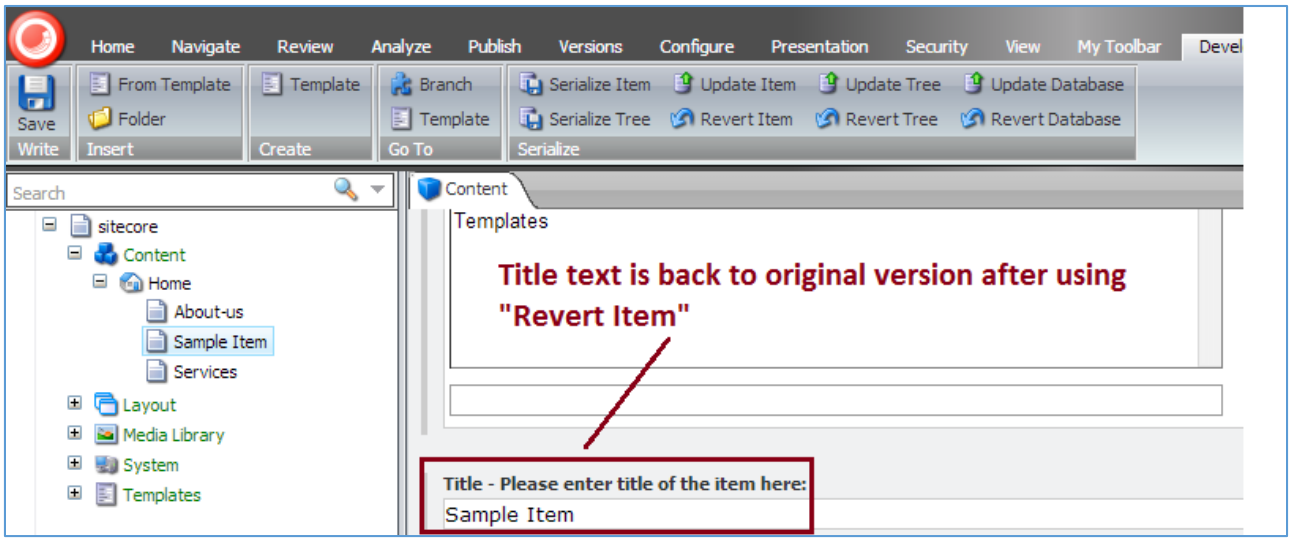
2) Sitecore Serialization:

- There is no size limit of serialization items in sitecore.
- You have option to revert your installation using revert options (e.g. revert tree, revert items) in serialization.


Home
Navigate
Review
Analyze
Publish
Versions
Configure
Presentation
Security
View
My Toolbar
Developer
Save
Edit
Sample Item
Folder
(1 of 3)
(2 of 3)
Duplicate
Copy To
Move To
Delete
Rename
Display Name
Up
Down
First
Last
Write
Edit
Insert
Operations
Rename
Sorting
Search
sitecore
Content
Home
Services
About-us
Layout
Media Library
System
Templates
Content
Home
Welcome to Sitecore.
Tree structure before update from serialization.
Quick Info
Item ID: {110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}
Item Name: Home
Item Path: /sitecore/content/Home
Template: /sitecore/templates/Sample/Sample Item - {76036F5E-CBCE-46D1-AF0A-4143F9B557AA}
Created From: [unknown]
Item Owner: sitecore\admin


Home
Navigate
Review
Analyze
Publish
Versions
Configure
Presentation
Security
View
My Toolbar
Developer
Save
From Template
Folder
Template
Branch
Template
Serialize Item
Update Item
Update Tree
Update Database
Serialize Tree
Revert Item
Revert Tree
Revert Database
Write
Insert
Create
Go To
Serialize
Search
sitecore
Content
Home
About-us
Sample Item
Services
Layout
Media Library
System
Templates
Content
Templates
After update tree, new item "Sample Item" is added.
Title - Please enter title of the item here:
Sample Item





So it's better to use "Serialization" than "Package" - lesson I learnt.

Note: Always use firefox browser while installing items from "package" or "serialization" because Internet Explorer gets time-out issue while installing a big package.

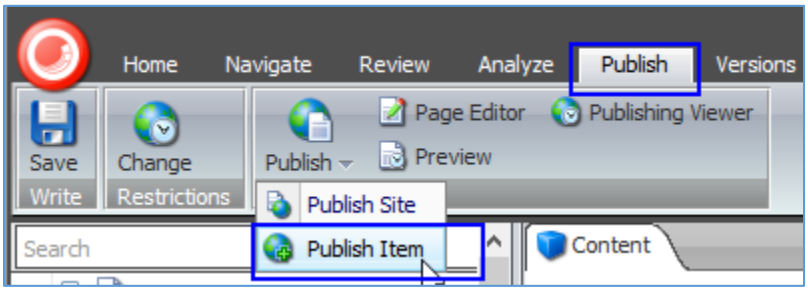
Which are the options to move items from master database to web database?

By the time you are reading this question, you may be knowing that in real world there are two Sitecore environments.

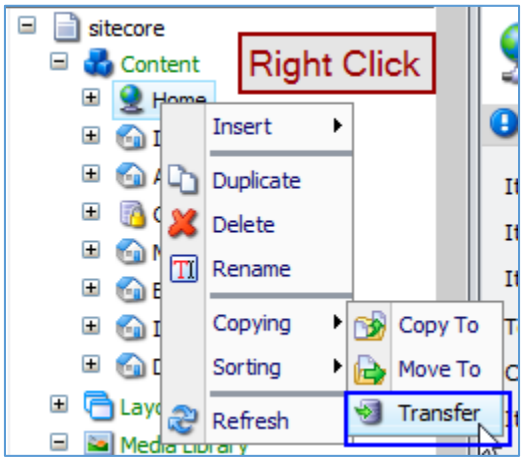
1) Staging Environment which points to master database
2) Live/Production Environment which points to web database

In order to move Sitecore items from staging environment (master database) to live environment (web database), there are two options available in Sitecore:

1) Publish (recommended) – you should **always use publish option** if you want to move item(s) from staging environment (master database) to production environment (web database).



2) Transfer (rarely used in specific scenarios) – you should **refrain** from transfer option if you want to move item(s) from staging environment (master database) to production environment (web database).



Then in which circumstances you should use transfer option? What's the difference between publish and transfer? Why it is recommended to use publish option?

Read the next question to find this answer.

What is difference between Publish and Transfer?

Whenever any item is published in Sitecore, there are couple of tasks getting executed in background. One of these tasks are clearing cache. That means whenever any item is published, Sitecore clear the cache of that item so that you will not find any caching issue on live website.

On the opposite side, there is no cache clear job performed while using transfer option. That means when you transfer any item(s), it will simply move item(s) from master database to web database. So it's your responsibility to clear the cache manually by using cache clear tool.

Now let's talk about circumstances in which you can use transfer option. If you have published tons of items which will take hours of time to get complete and suddenly you come across a situation where you want publish single/few item(s) instantly. You cannot publish this item until publishing queue completes current running publishing items (tons of item). Even if you publish that urgently required item, it will be queued and will be published last. In such situation, you can take advantage of transfer option to move instantly required item from stage environment (master database) to live environment (web database). This will push your item to live environment instantly although you need to clear the cache manually by using cache clear tool.

If you don't know what is cache clear tool, then don't worry we will discuss about Sitecore cache and cache clear tool in upcoming questions.

What is scheduler? How can you define scheduler in Sitecore?

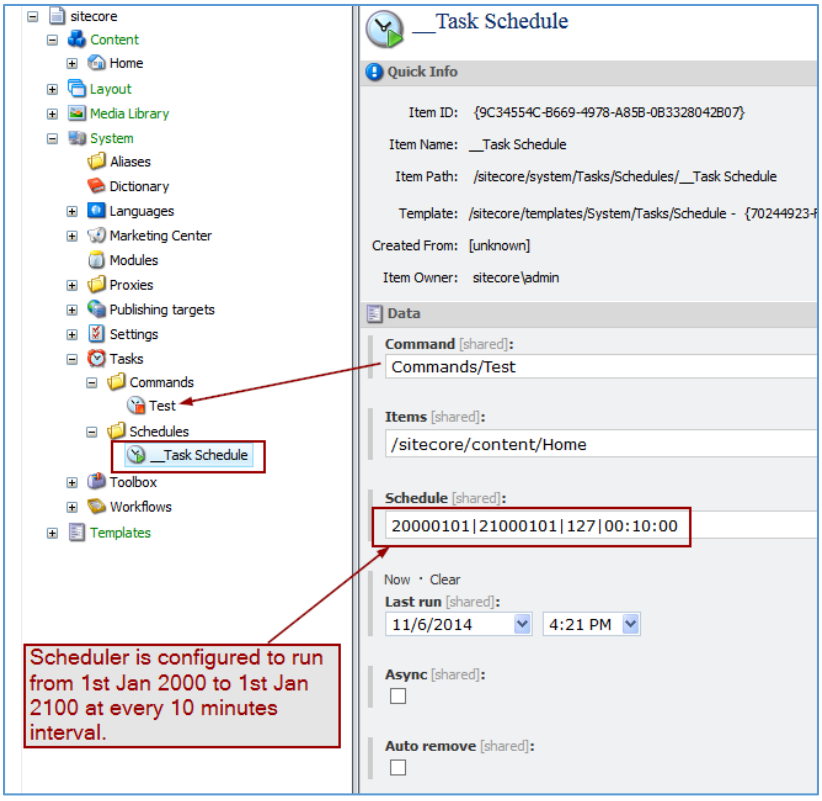
As name implies, scheduler is task that gets executed at some interval. There are three ways to schedule process in Sitecore CMS.

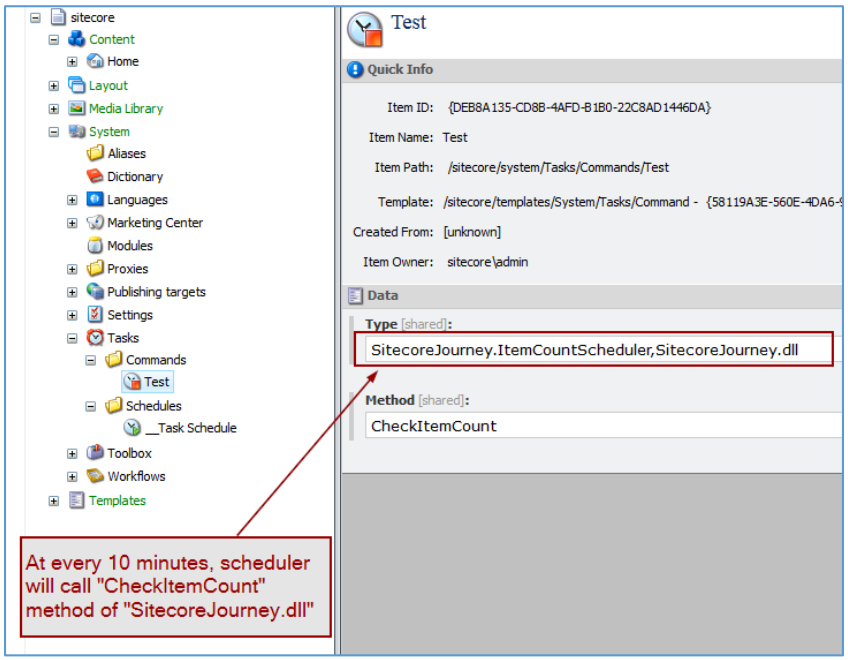
1) Configure agent in web.config

   This is very straightforward and the most common way to create a scheduler in Sitecore. Using this approach will restart ASP.NET worker process as it requires to change web.config file.

2) Create schedule tasks in Sitecore database

   In this option, you can define scheduler in Sitecore database under /sitecore/system/Tasks/Schedules. This option is user-friendly as you can configure scheduling option from CMS directly. This approach would not restart ASP.NET worker process as you don't require to change web.config file

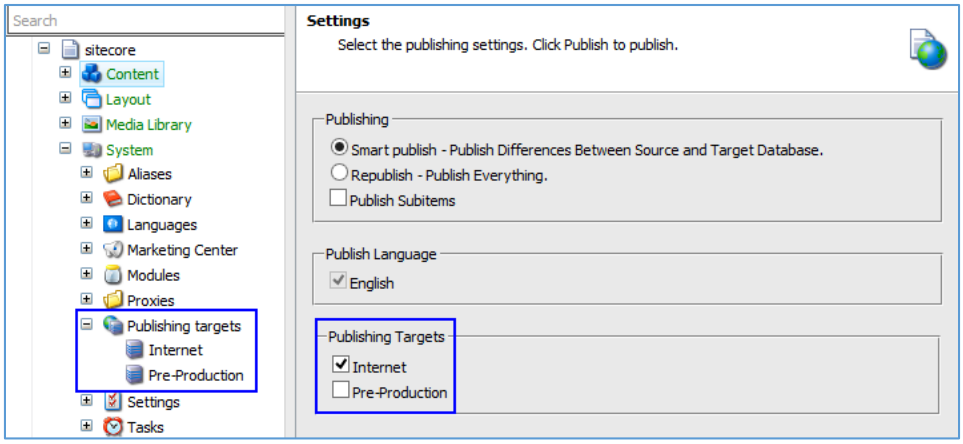




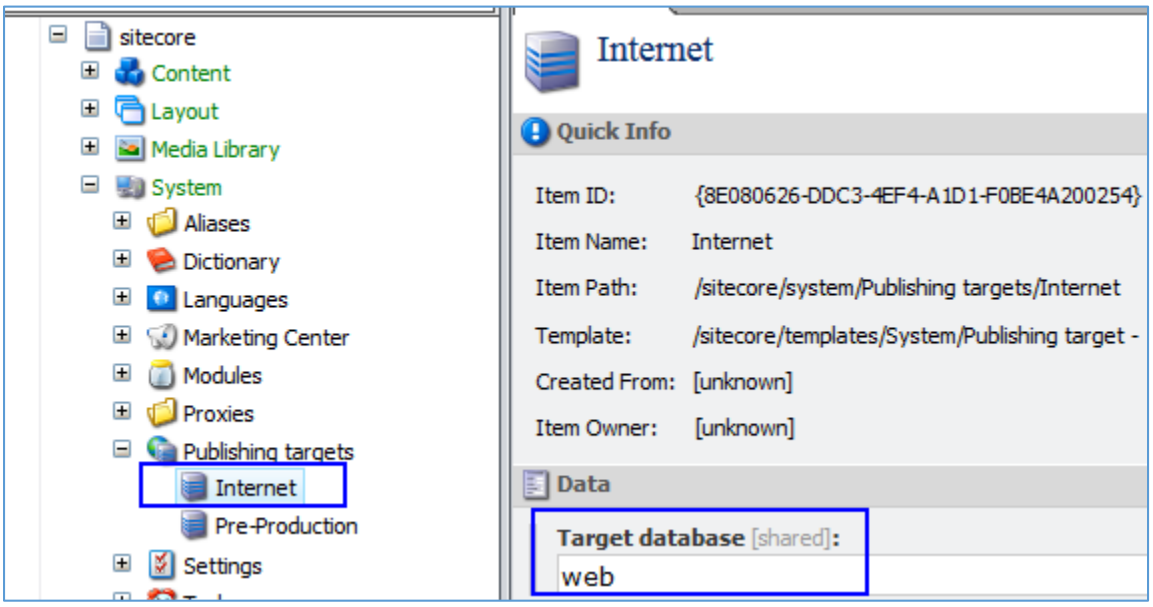
3) Create Window Task Scheduler (window service) to call web service in Sitecore

What is publishing target? Where can you define publishing targets in Sitecore?

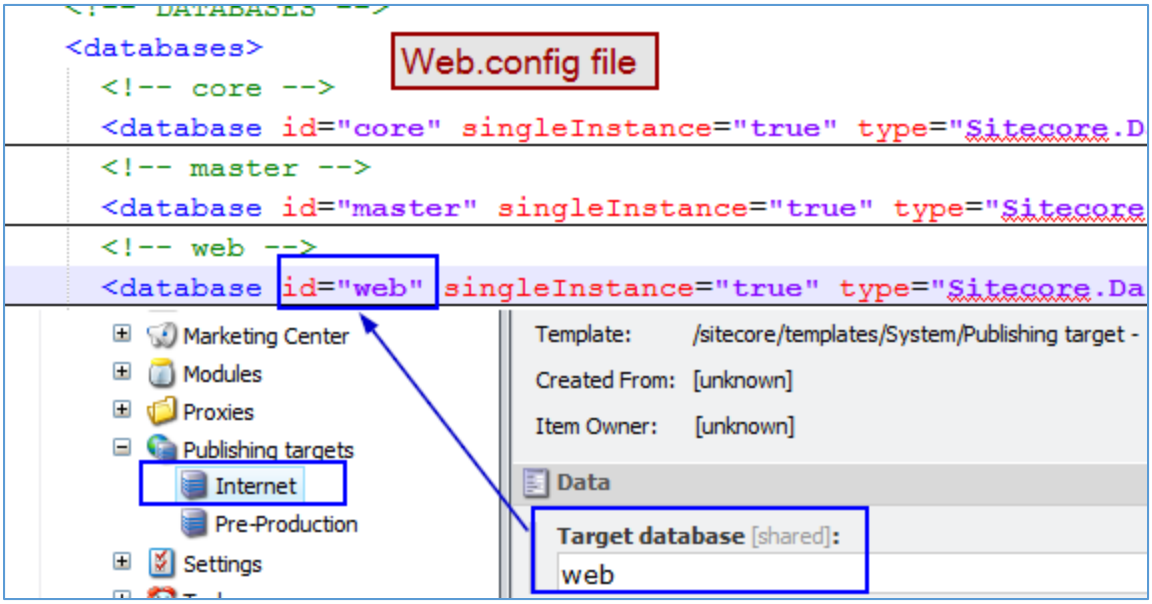
In order to move items from master database to live (web) database, we use publish option. If you have got chance to observe publishing window, you may have seen **publishing target** section at bottom of the window.



So from where these checkbox values are coming? They are coming from publishing target configuration.



As per the above screens “Internet” publishing target is pointing to “web” database. And the “web” database definition is configured in web.config file.

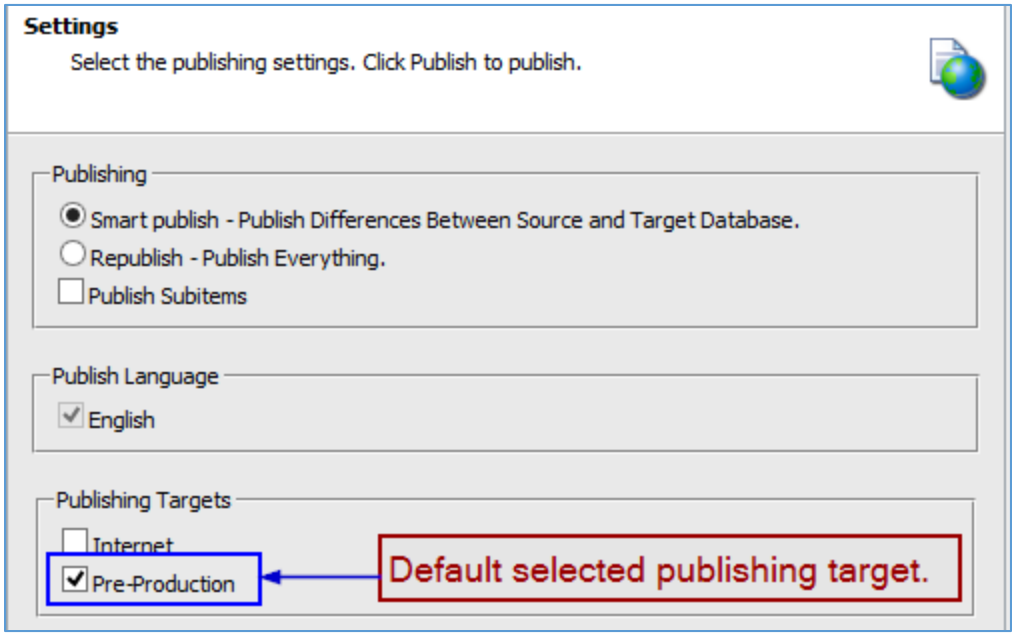


You can configure more than one publishing target database as per your need. It is worth to note that master database contains all versions items while publishing target database (e.g. web) contains only single latest version of each language item.

If you have multiple publishing targets then you will see all of them in publishing window as checkbox items. In case you want to set any of these publishing target database as your default target database then you can configure it via **DefaultPublishingTargets** setting in web.config file.

<setting name="DefaultPublishingTargets" value="Pre-Production" />

```
<!--  DEFAULT PUBLISHING TARGETS
      Pipe separated list of default publishing targets.
      Targets in this list will be preselected in the Publish dialog.
-->
<setting name="DefaultPublishingTargets" value="Pre-Production" />
```



What is workflow? How can you check current workflow status of an item?

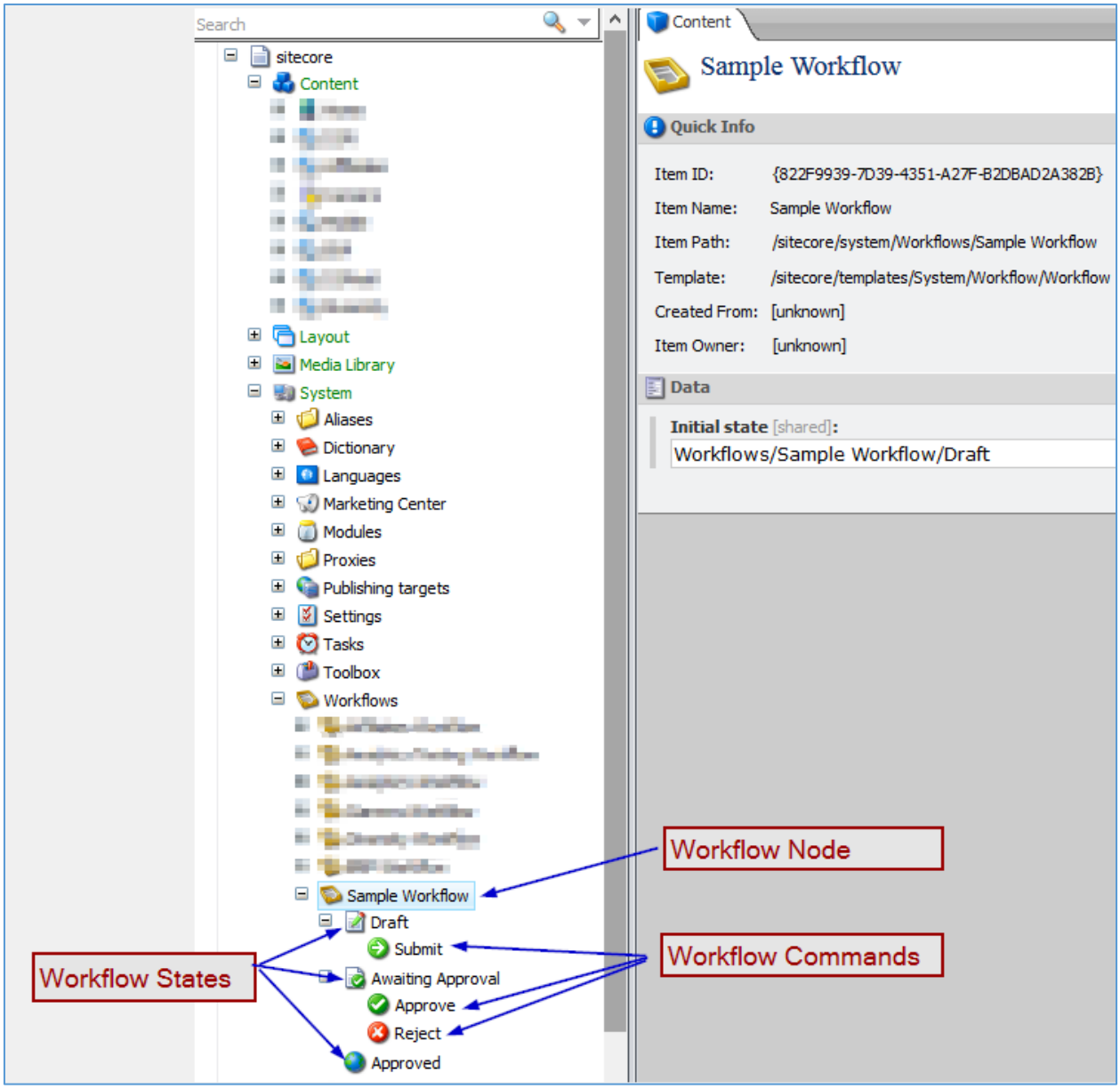
Sitecore is famous for its flexible architecture. You can achieve many requirements without doing any code and by doing some configuration only. One of such eye caching feature is workflow.

Workflow is controllable way by which you can define series of tasks to add, review and publish content.

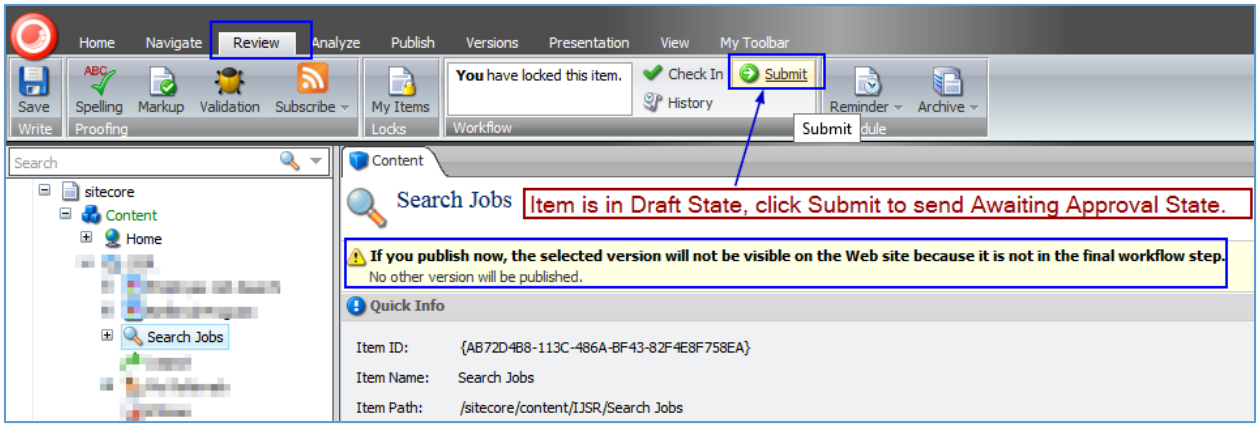
In every CMS there are broadly three types of users:

1) Content Author/Editor – responsible for adding and updating content only. This type of users should not be allowed to publish the content.
2) Content Reviewer – responsible for reviewing content that is submitted by author. This type of user should not be allowed to publish the content.
3) Content Publisher – responsible for publishing content approved by reviewer.

In order to achieve above series of task, you can take advantage of Sitecore Workflow feature. You can find workflow at /sitecore/system/Workflows in Sitecore content tree. Below figure shows an example of sample workflow.



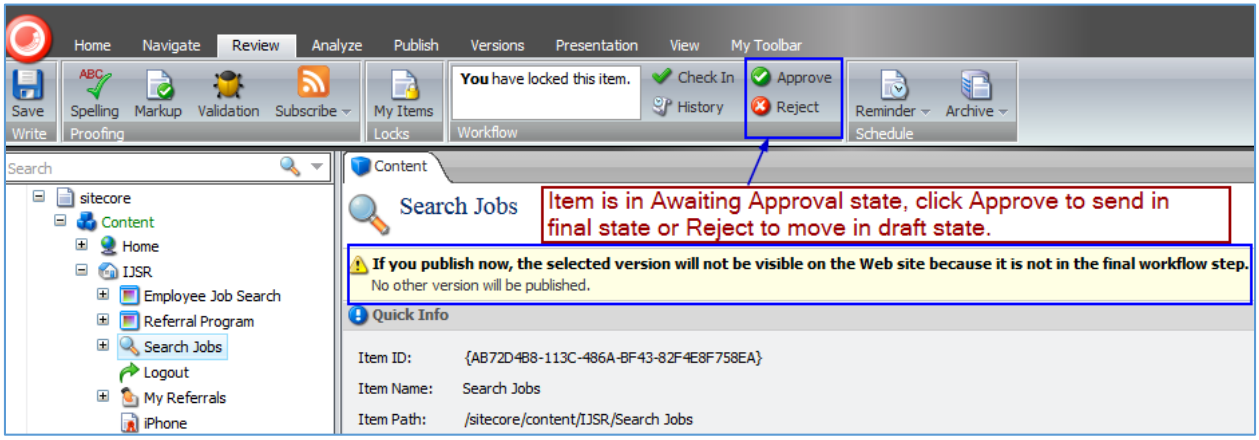
Draft State: The draft state is initial state and it will be assigned to item whenever user creates or edit the content. Sitecore item cannot be published if it is in draft state.



Awaiting Approval State: Once content editing is completed and item is ready to review, user can submit that item to reviewer. This submission action will move item from draft state to awaiting approval state. Sitecore item cannot be published if it is in awaiting approval state.

Approved State: This is the last state of item.
There are two types of actions can be executed under this state:

1) Approve: this action will move Sitecore item to final state and item can be publish now.
2) Reject: this action will move Sitecore item to draft state again



At any point of item if you want to know current state of item you can either check workflow section fields of item or by looking into workbox.

Which are different tokens supported by Sitecore?

When you create an item, Sitecore replaces token with corresponding value of that item.

For example: if you have specified $name token at “Title” field in standard value, then Sitecore will replace “Title” field with item name that you provided while creating item.

Here is the list of tokens supported by Sitecore:

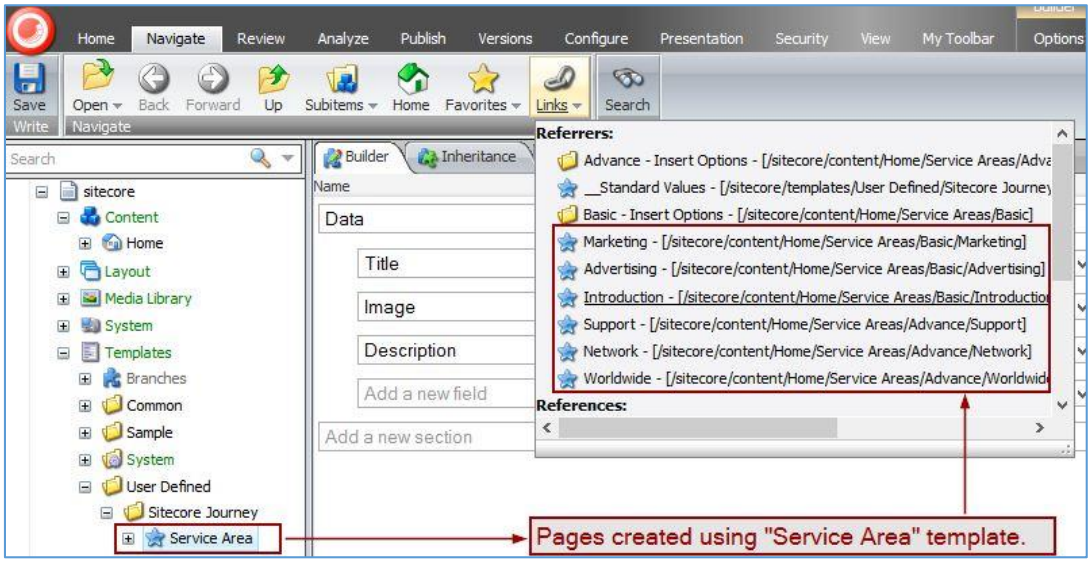
- **$name**: The name of the item
- **$id**: The ID of the item
- **$parentid**: The ID of the parent of the item
- **$parentname**: The name of the parent of the item
- **$date**: The system date (yyyyMMdd)
- **$time**: The system time (HHmmss)
- **$now**: The date and time (yyyyMMddTHHmmss)

How can you check particular item is referred by which other items in Sitecore?

In your day to day Sitecore work you may come across a situation where you want to know how many items are referring particular item.

For example you are working on template item and you want to know that how many items are created using this template or how many items are using this template?

Below screen shot will guide you how to get referrers list.

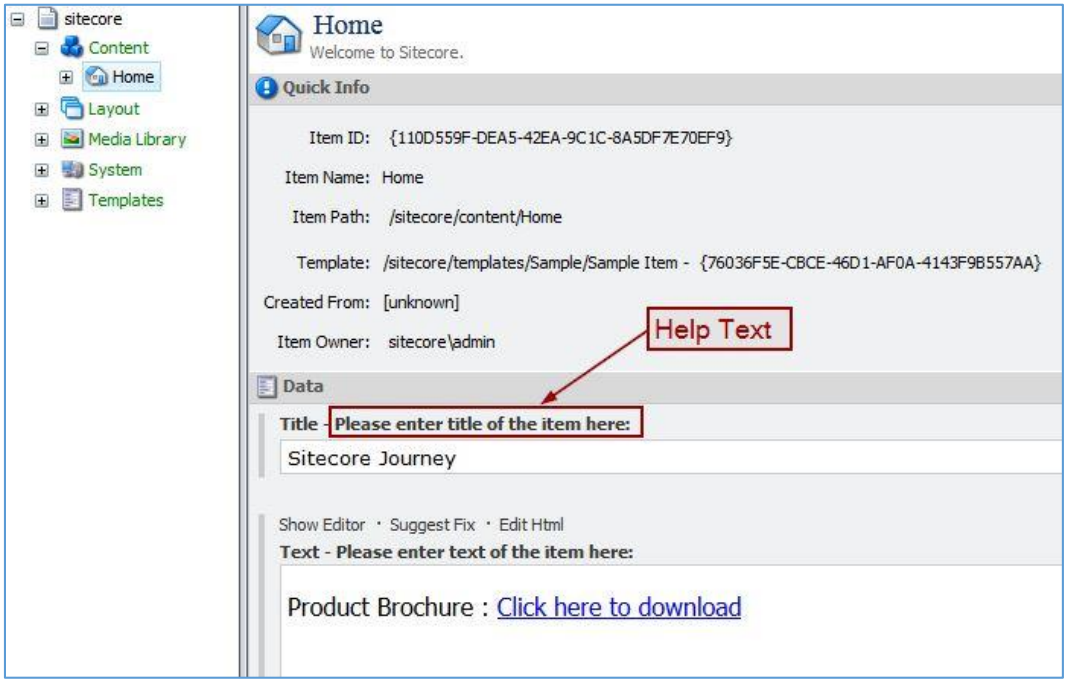


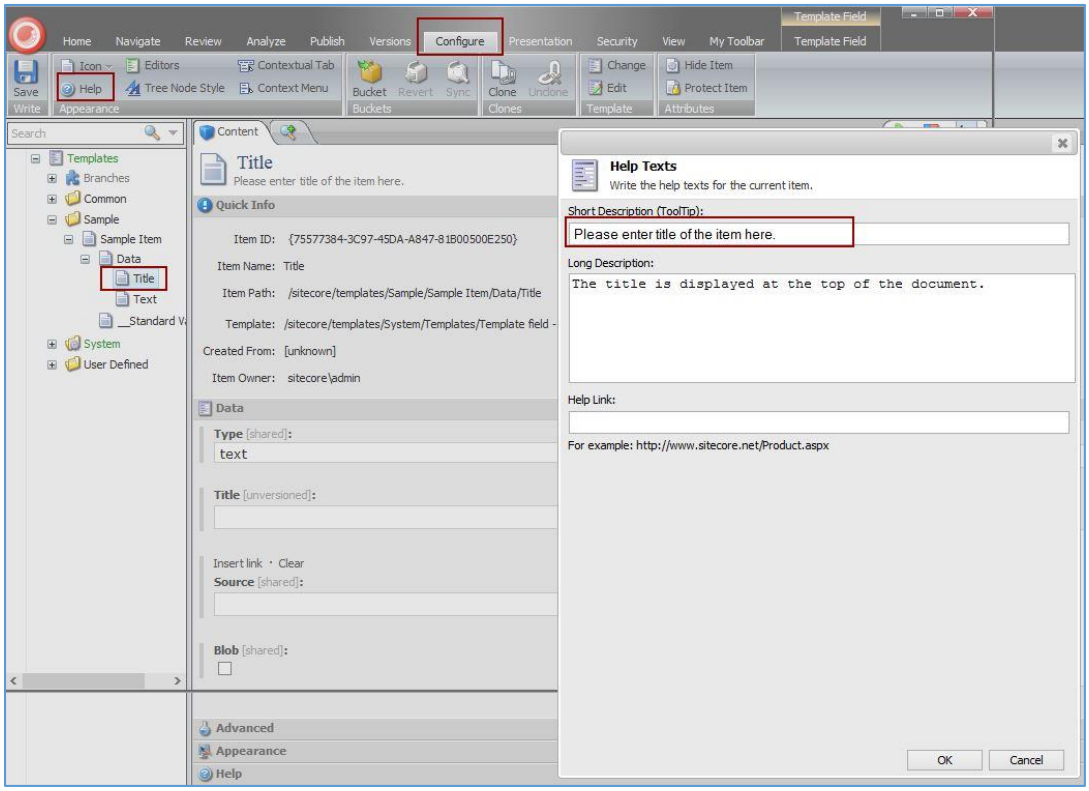
If want to check this for multiple items then it will be time consuming job to select every item and check links for all those items. For that you can download “Sitecore Informatics” module from Sitecore Market place which will show you all your site templates and items created using those templates.

How can you add help text for particular field in Sitecore?

Many times you want to provide some help text to user to understand the purpose of particular field. You want to convey message to user what should be value of that field. This can be accomplished by configuring help text for that field.

Below screen-shots will help you how to configure help text for any field.

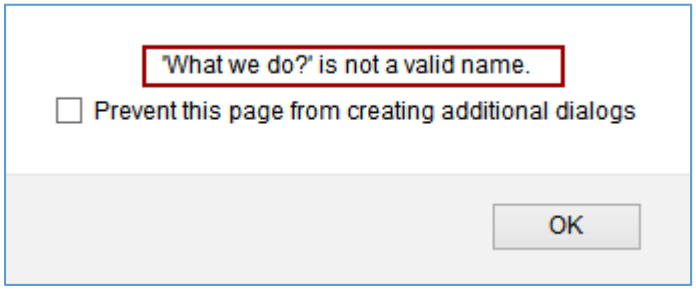




What is display name? Why one should use it?

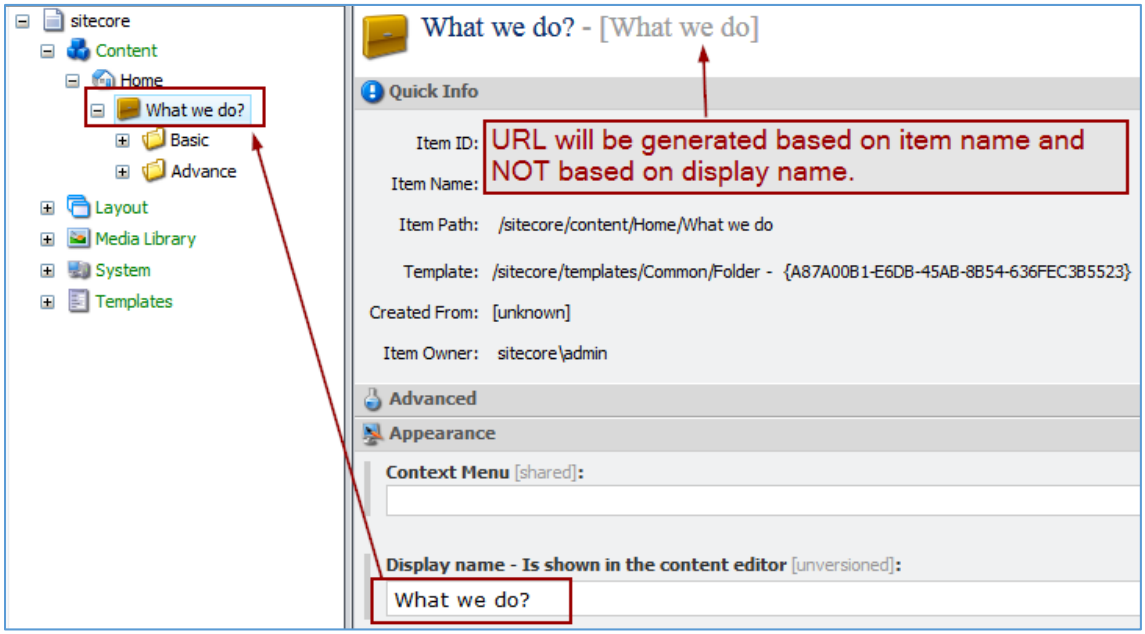
Whenever you create an item in Sitecore, you want to give user-friendly name so that content author can easily understand the purpose of that item. But do you know there are few restrictions in item name, you cannot add special characters in Sitecore item name.

For example – if you want to create an item “What we do?” Sitecore will not allow it as it contains special character “?”



So how can you achieve it?

The answer is – use **Display Name** field. Once you create item by following item name criteria. You can then change display field of that item by including characters which are not allowed while creating item.



Display name (e.g. “Who we are?”) will be shown in CMS user interface such content tree in place of item name (e.g. “Who we are”).

The advantages of using display name fields are:

- ➔ Display name can include characters not allowed in item names.
- ➔ Display names can vary by language while item names cannot.

It is worth to note that Sitecore constructs the default URL of an item based on its name not based on display name.

Which characters are not supported by default for item name in Sitecore?

By default Sitecore allows alphanumeric, $ and – character in item name. You cannot use any other special characters in item name.
In case if you want to allow some special character in item name, then you need to change **ItemNameValidation** setting value in web.config file.

```
<!--  ITEM NAME VALIDATION
      Regular expression for validating item names
-->
<setting name="ItemNameValidation" value="^[\w\*\$][\w\s\-\$]*(\(\d{1,}\)){0,1}$"/>
```

How can you configure log folder path of your Sitecore site?

Sitecore uses log4net API to log all the events and information. By default Sitecore create date wise logs in log folder.

```
<appender name="LogFileAppender" type="log4net.Appender.SitecoreLogFileAppender, Sitecore.Logging">
  <file value="$(dataFolder)/logs/log.{date}.txt"/>
  <appendToFile value="true"/>
  <layout type="log4net.Layout.PatternLayout">
    <conversionPattern value="%4t %d{ABSOLUTE} %-5p %m%n"/>
  </layout>
</appender>
```

You will find this logs folder under dataFolder variable path. dataFolder is the variable that defines exact path of your logs folder.

```
<sitecore database="SqlServer">
  <sc.variable name="dataFolder" value="C:\Inetpub\wwwroot\Sitecore71\Data\"/>
  <sc.variable name="mediaFolder" value="/upload"/>
  <sc.variable name="tempFolder" value="/temp"/>
```

Where does Sitecore store all packages? How can you configure this package folder path?

As we have seen earlier, Sitecore stores all the packages under dataFolder variable path. . dataFolder is the variable that defines exact path of your logs folder.

```
<!--  PACKAGE PATH
      Specifies the path where packages are located.
-->
<setting name="PackagePath" value="$(dataFolder)/packages"/>
```

```
<sitecore database="SqlServer">
  <sc.variable name="dataFolder" value="C:\Inetpub\wwwroot\Sitecore71\Data\"/>
  <sc.variable name="mediaFolder" value="/upload"/>
  <sc.variable name="tempFolder" value="/temp"/>
```

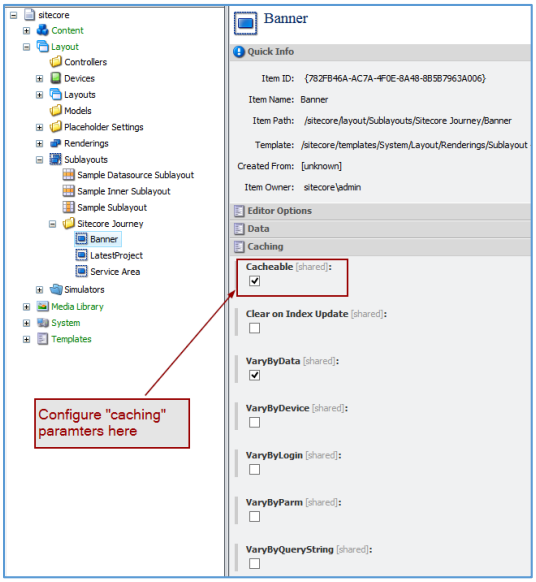
Which are different types of caching in Sitecore?

Sitecore provides a number of caching layers. Each cache serves as a key/value collection. These caching layers are:

1) HTML Cache

It caches actual HTML output generated from rendering and sublayout.

It can be configured from **caching section** of rendering controls.



Each website can be configured to enable or disable HTML cache along with allowed HTML cache size in web.config file under <sites> section.

```
<site name = "website" virtualFolder="/" physicalFolder="/"
rootPath="/sitecore/content " startItem="/Home" database="web"
domain="extranet" cacheHtml="true" htmlCacheSize="10MB"
enablePreview="true" enableWebEdit="true" enableDebugger="true"
disableClientData="false" />
```

2) Item Cache

Sitecore items are cached under this cache layer. This cache contains objects of the Sitecore class Sitecore.Data.Items.Item, which are the one you use in your code most of the time. Whenever any sitecore item is requested, it will be served from item cache and if it is not there then it will be served from data cache (next layer of cache) and will populate cache in item cache layer. So that whenever next request for same item is made, it will be directly served from item cache.

3) Data Cache

Data cache contains items of the Sitecore class **ItemInformation**. Whenever any request is made, data is pulled from data cache. But if it is not present there it will be pulled from prefetch cache and then will be put into data cache. So that whenever next request for same item is made, it will be directly served from data cache.

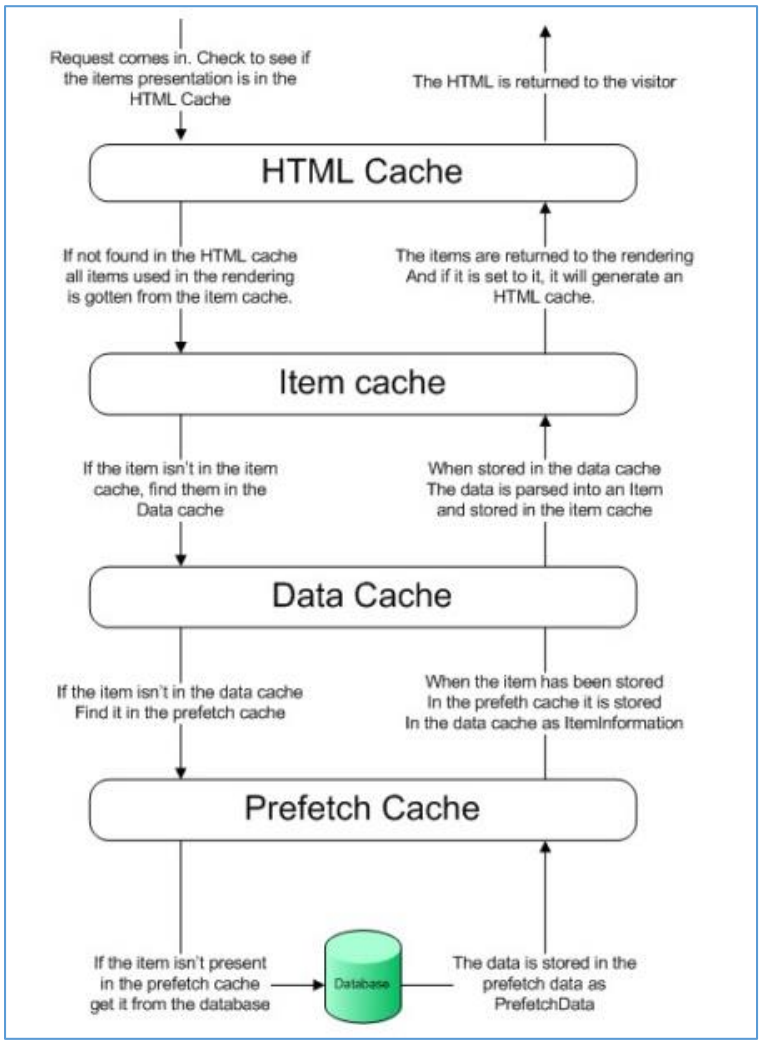
Data cache can be configured in web.config file under Caching.DefaultDataCacheSize setting:

```
<setting name = "Caching.DefaultDataCacheSize" value="10MB" />
```

4) Prefetch Cache

There is prefetch cache for each database. The cache contains items of Sitecore class PrefetchData which is same like data pulled from database. Prefetch cache load the cache at Sitecore start up as per specified in App_Config/Prefetch folder.

Below diargram illustrates how all these are actually working.



Where can you add a new site in Sitecore?

Creating a new site in Sitecore instance (e.g. http://local.sitecore7.com) involves three steps process:

Step-1:

Configure your site hostname in Sitecore Instance. There are two ways to add new site in your Sitecore instance.

1) Add <site> element under <sites> section in web.config file

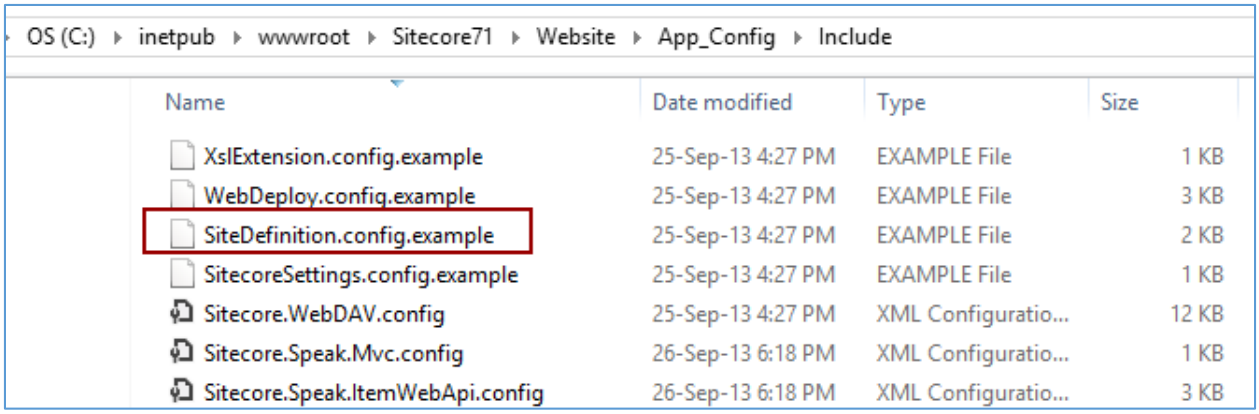
```
<sites>
  <site name="shell" virtualFolder="/sitecore/shell" physicalFolder="/sitecore/shell" rootPath="/sitecore/content" startIte
  "/home" language="en" database="core" domain="sitecore" loginPage="/sitecore/login" content="master" contentStartItem=
  "/Home" enableWorkflow="true" enableAnalytics="false" analyticsDefinitions="content" xmlControlPage=
  "/sitecore/shell/default.aspx" browserTitle="Sitecore" htmlCacheSize="2MB" registryCacheSize="3MB" viewStateCacheSize=
  "200KB" xslCacheSize="5MB" disableBrowserCaching="true"/>
  <site name="login" virtualFolder="/sitecore/login" physicalFolder="/sitecore/login" enableAnalytics="false" database="cor
  domain="sitecore" disableXmlControls="true"/>
  <site name="admin" virtualFolder="/sitecore/admin" physicalFolder="/sitecore/admin" enableAnalytics="false" enableWorkflo
  "true" domain="sitecore" loginPage="/sitecore/admin/login.aspx"/>
  <site name="service" virtualFolder="/sitecore/service" physicalFolder="/sitecore/service"/>
  <site name="modules_shell" virtualFolder="/sitecore modules/shell" physicalFolder="/sitecore modules/shell" rootPath=
  "/sitecore/content" startItem="/home" language="en" database="core" domain="sitecore" content="master" enableAnalytics=
  "false" enableWorkflow="true"/>
  <site name="modules_website" virtualFolder="/sitecore modules/web" physicalFolder="/sitecore modules/web" rootPath=
  "/sitecore/content" startItem="/home" language="en" database="web" domain="extranet" allowDebug="true" cacheHtml="true"/>
  <site name="website" virtualFolder="/" physicalFolder="/" rootPath="/sitecore/content" startItem="/home" database="web"
  domain="extranet" allowDebug="true" cacheHtml="true" htmlCacheSize="10MB" registryCacheSize="0" viewStateCacheSize="0"
  xslCacheSize="5MB" filteredItemsCacheSize="2MB" enablePreview="true" enableWebEdit="true" enableDebugger="true"
  disableClientData="false" hostName="local.sitecore7.com"/>
  <site name="scheduler" enableAnalytics="false" domain="sitecore"/>
  <site name="system" enableAnalytics="false" domain="sitecore"/>
  <site name="publisher" domain="sitecore" enableAnalytics="false" enableWorkflow="true"/>
</sites>
```

2) Add <site> element under <sites> section in SiteDefintion.config file

SiteDefinition.config.example

```
<!--

Purpose: This include file adds a new site definition

To enable this, rename this file so that it has a ".config" extension and
change all the parameters to suit your own scenario

Notice how "patch:before" is used to insert the site definition BEFORE the
existing <site name="website" ...> element

You can use "patch:before" and "patch:after" as an attribute of an inserted
element to specify an insertion point for the new element. Both accept an
XPath relative to the parent node of the inserted element.

-->
<configuration xmlns:patch="http://www.sitecore.net/xmlconfig/">
  <sitecore>
    <sites>
      <site name="mysite" patch:before="site[@name='website']"
            virtualFolder="/"
            physicalFolder="/"
            rootPath="/sitecore/content"
            startItem="/home"
            database="master"
            domain="extranet"
            allowDebug="true"
            cacheHtml="true"
            htmlCacheSize="10MB"
            enablePreview="true"
            enableWebEdit="true"
            enableDebugger="true"
            hostName="local.sitecore7.com"
            disableClientData="false"/>
    </sites>
  </sitecore>
</configuration>
```

You will find SiteDefintion.config.example file at \Website\App_Config\Include\ folder. You should remove .example extension in order to use it.

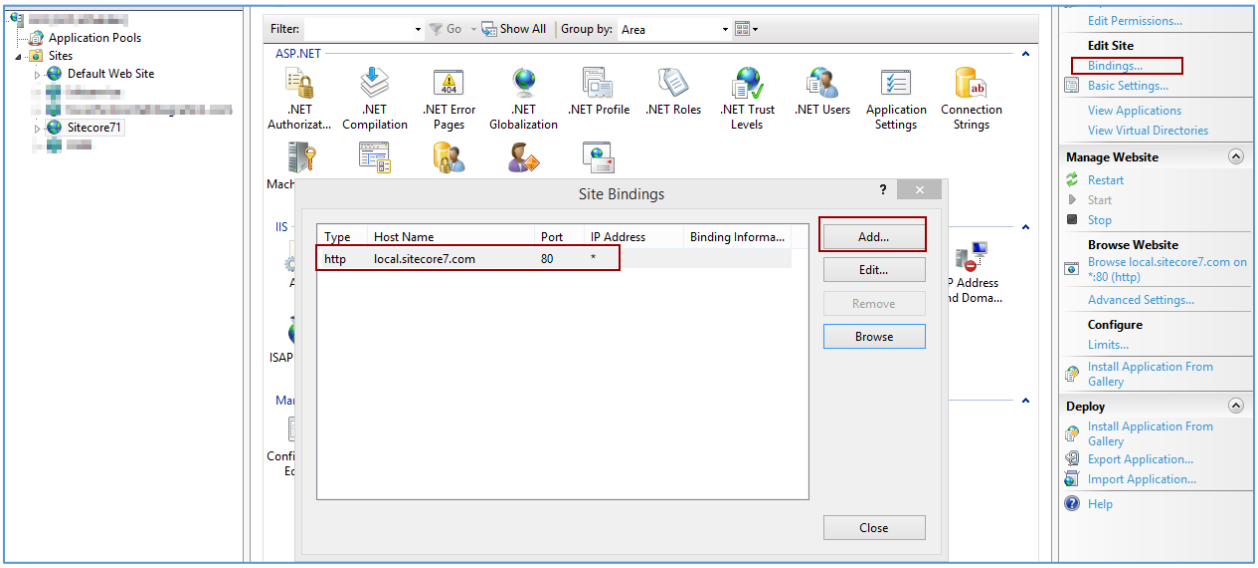


Now you may be wondering which one is good option. Well, let me help you on this. It is good to add new site in SiteDefinition.config file because it's always advisable not to touch default web.config file and override your custom settings in config files situated under App_Config folder provided by Sitecore.

After adding new site in config file, there are two more steps need to be performed out of Sitecore to access newly added site.

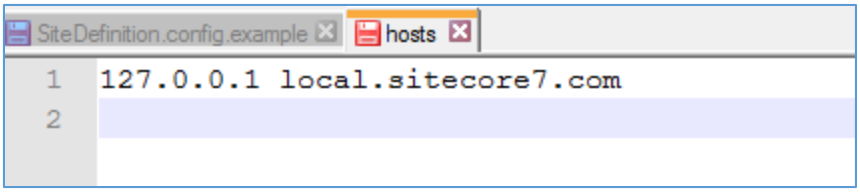
Step-2:

Add your site hostname in IIS binding



Step-3:

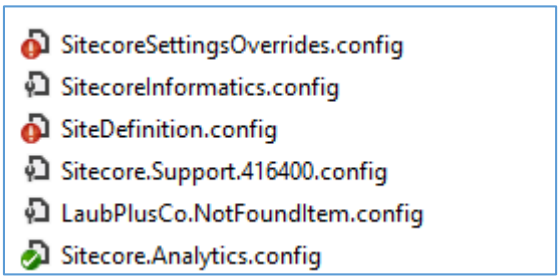
Add your site hostname in hosts file (under WINDOWS\System32\drivers\etc\ folder)



After completing these three steps your newly added Sitecore website is ready to access.

Where should you overrides default configuration in Sitecore?

Sitecore has a folder called **App_Config** which stores all the configuration files. Instead of modifying web.config file, it is always recommended to override the required setting by updating **SitecoreSettingsOverrides.config** file at /Website/App_Config/Include folder.



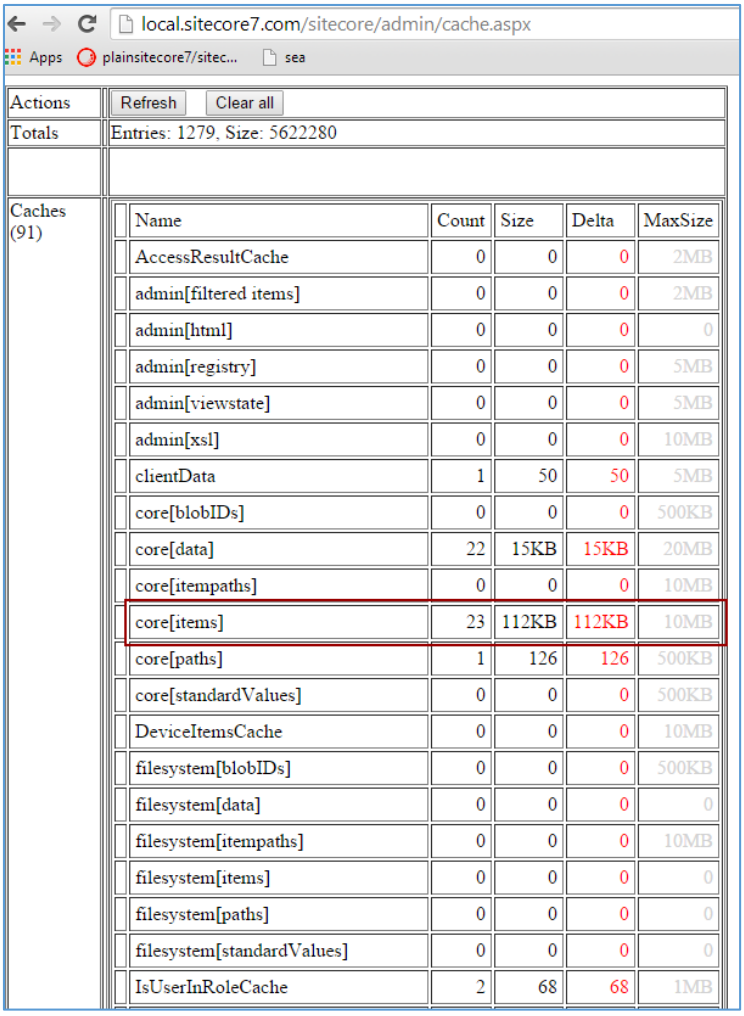
Which are different admin tools provided by Sitecore?

Sitecore gives some hidden features which every developer or administrator must be aware about. One of such features is Sitecore admin pages or you can say admin tools which are situated under \Website\sitecore\admin directory.

Let's understand these admin pages:

1) Cache

   The /Sitecore/admin/cache.aspx page displays details about caching setting configured for your Sitecore instance. You should be knowing that there are different types of cache such as prefetch cache, item cache, html cache managed by Sitecore internally to serve your page. You can find details about these caches under this page.



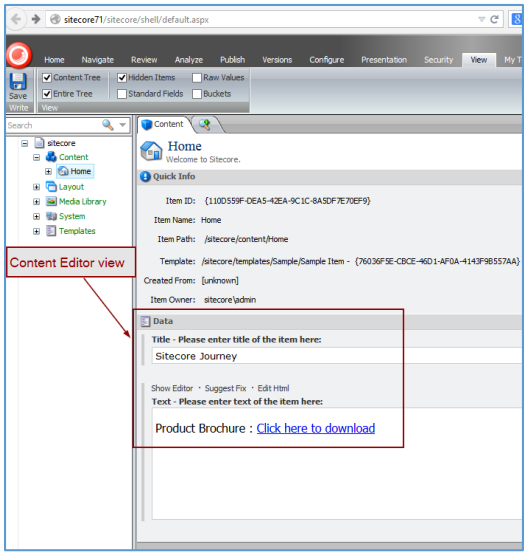
Actions	Refresh	Clear all
Totals	Entries: 1279, Size: 5622280	

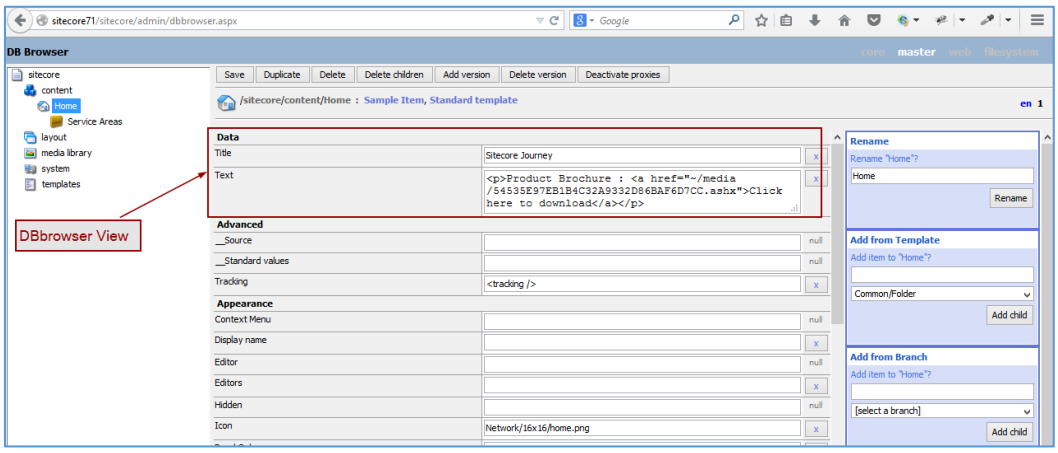
Caches (91)

Name	Count	Size	Delta	MaxSize
AccessResultCache	0	0	0	2MB
admin[filtered items]	0	0	0	2MB
admin[html]	0	0	0	0
admin[registry]	0	0	0	5MB
admin[viewstate]	0	0	0	5MB
admin[xsl]	0	0	0	10MB
clientData	1	50	50	5MB
core[blobIDs]	0	0	0	500KB
core[data]	22	15KB	15KB	20MB
core[itempaths]	0	0	0	10MB
core[items]	23	112KB	112KB	10MB
core[paths]	1	126	126	500KB
core[standardValues]	0	0	0	500KB
DeviceItemsCache	0	0	0	10MB
filesystem[blobIDs]	0	0	0	500KB
filesystem[data]	0	0	0	0
filesystem[itempaths]	0	0	0	10MB
filesystem[items]	0	0	0	0
filesystem[paths]	0	0	0	0
filesystem[standardValues]	0	0	0	0
IsUserInRoleCache	2	68	68	1MB

2) Data Browser

The /Sitecore/admin/dbbrowser.aspx page displays Sitecore database view. It is similar to content tree but it is lightweight interface.

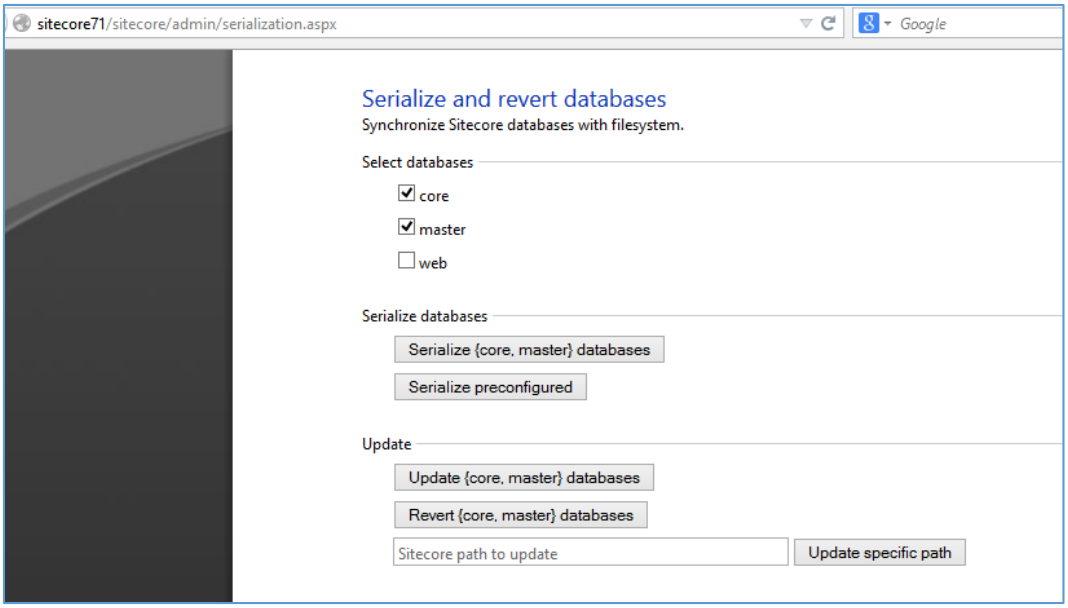




3) Serialization

The /Sitecore/admin/serialization.aspx page allows to serialize sitecore databases (e.g. core, master and web) into XML file system. This page can be very useful when you want to backup database content.

This is how the page actually looks like:

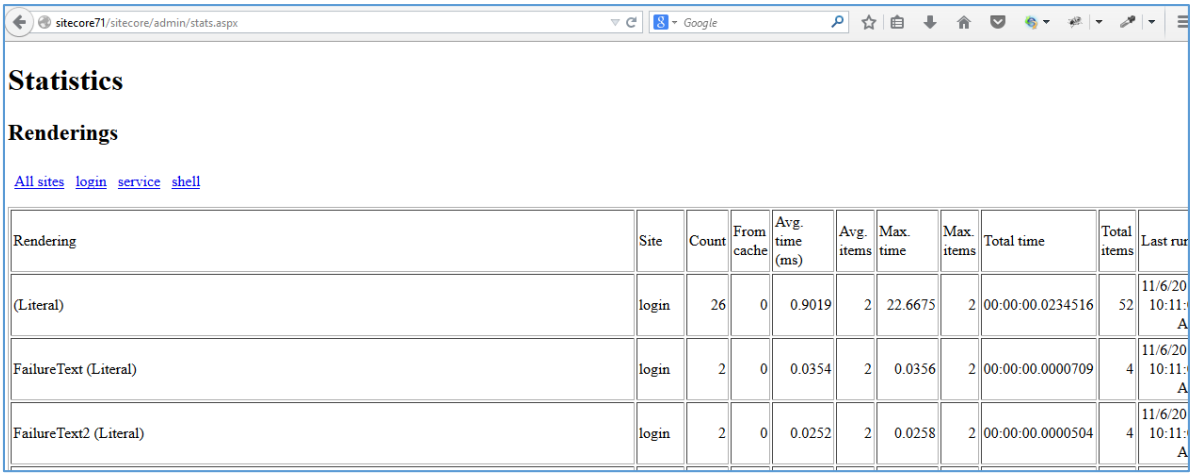


4) Show Config

The /Sitecore/admin/showconfig.aspx page shows final version of the sitecore configuration including all external configurations from App_Config folder. This page is very useful for debugging purpose to find if your web.config setting is overridden by any other external config settings from App_Config/Include folder.

5) Stats

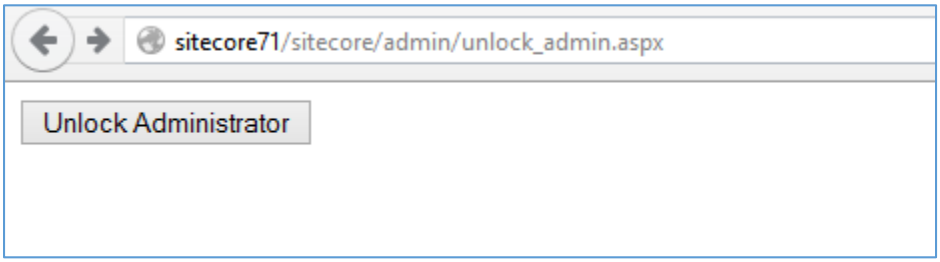
The /Sitecore/admin/stats.aspx page provides rendering statistics for presentation components of each site, it shows how much time components take to load. It also defines how many times components have been loaded from cache. This page is very useful for fine tuning caching settings.



Rendering	Site	Count	From cache	Avg. time (ms)	Avg. items	Max. time	Max. items	Total time	Total items	Last run
(Literal)	login	26	0	0.9019	2	22.6675	2	00:00:00.0234516	52	11/6/20 10:11: A
FailureText (Literal)	login	2	0	0.0354	2	0.0356	2	00:00:00.0000709	4	11/6/20 10:11: A
FailureText2 (Literal)	login	2	0	0.0252	2	0.0258	2	00:00:00.0000504	4	11/6/20 10:11: A

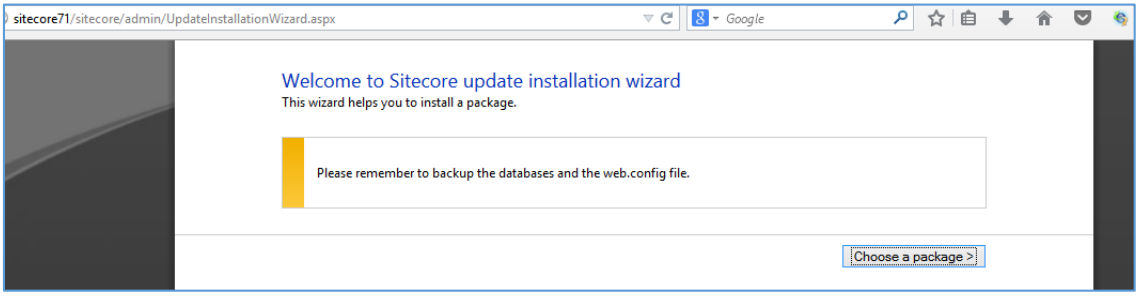
6) Unlock Admin

As the name implies /Sitecore/admin/unlock_admin.aspx page is used to unlock a locked admin account. After certain number of invalid login attempts, a user may be locked out. If this happens with admin user then this page will be very helpful to unlock admin account.



7) Update Installation Wizard

The /Sitecore/admin/UpdateInstallationWizard.aspx page helps to upgrade sitecore instance. It allows user to upload .update package and install on sitecore instance.



Which search mechanism is used by Sitecore default?

By default Sitecore uses an open source **Lucene** search provider. Both original API (Lucene.Net) and Sitecore API (Sitecore.Search) are accessible to developers. As Lucene lacks of some of advance features such as refinement/faceted navigation, these is an extension call SOLR which can also be used in Sitecore as a search provider. SOLR is not part of the standard Sitecore build.

Which are other search mechanism industry is using in Sitecore?

1) **SOLR**: It is popular, blazing fast open source enterprise search platform from the Apache Lucene project. Its major features include powerful full-text search, hit highlighting, facted search, document handling. Solr powers the search and navigation features of many of the world's largest internet sites.

2) **Coveo**: It offers a rich search interface for faceted search. It allows to customize all the search result fields without writing code or css. It provides highly advance search for personalized online experience in Sitecore websites.

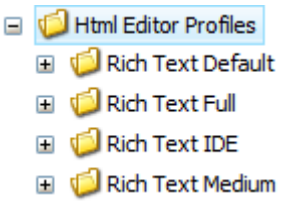
   Coveo recently (at Sitecore Symposium 2014) announced a free version for Sitecore – "Coveo for Sitecore – Free Edition" with some lmited features compare to paid Enterprise edition.

Which are different types of HTML Editor profiles available in Sitecore?

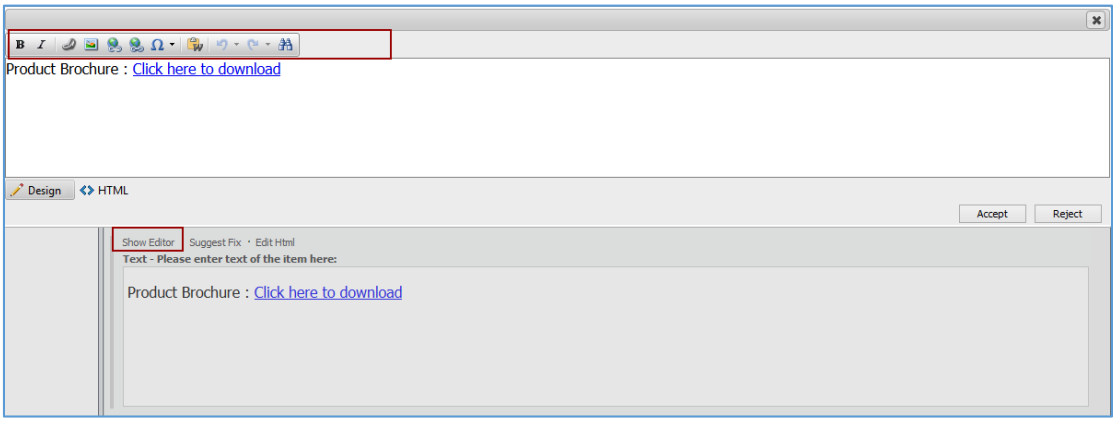
Sitecore provides four types of Html Editor profiles:

1) Rich Text Default
2) Rich Text Full
3) Rich Text IDE
4) Rich Text Medium

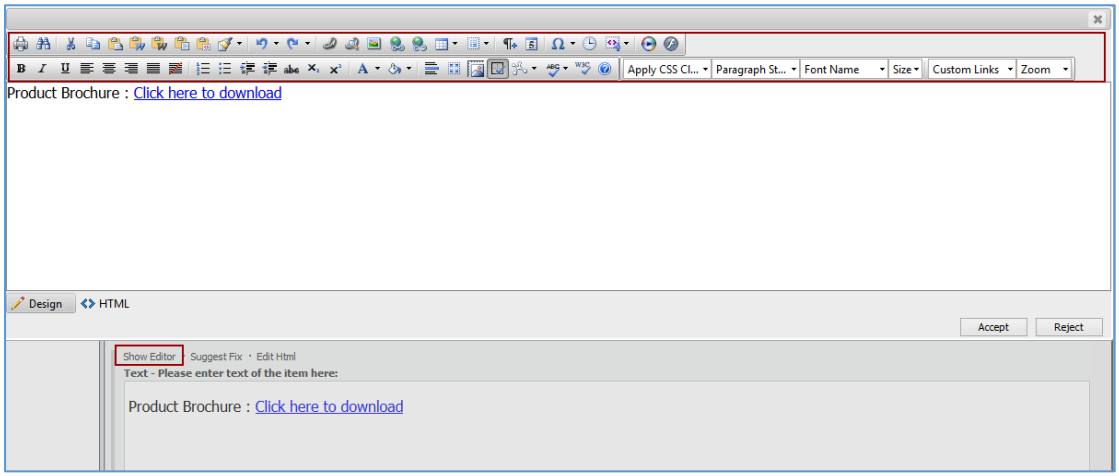
It can be found at /sitecore/system/Settings/Html Editor Profiles in core database.



Rich Text Default provides few editor options as shown in below figure.

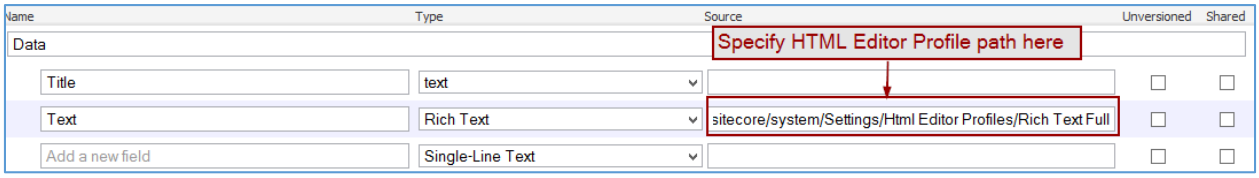


While **Rich Text Full** provides wide range of editor options as highlighted in below figure.

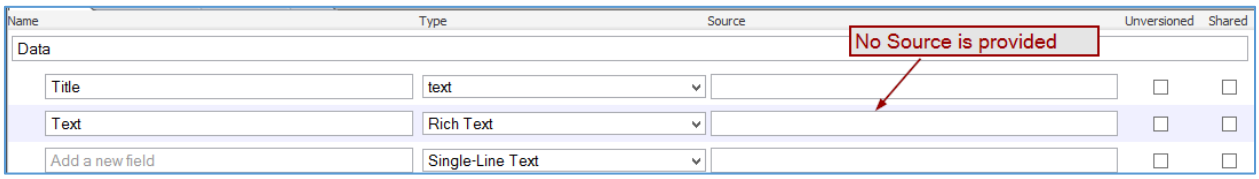


Where can you configure default HTML Editor profile in Sitecore?

HTML Editor profile for rich text field can be configured from source field as illustrated in below figure.



By default, Sitecore is configured to use **Rich Text Default** profile. That means if no Html Editor path is specified in source for Rich Text data type field, then it will use Rich Text Default profile.



This default setting can be updated from web.config file.

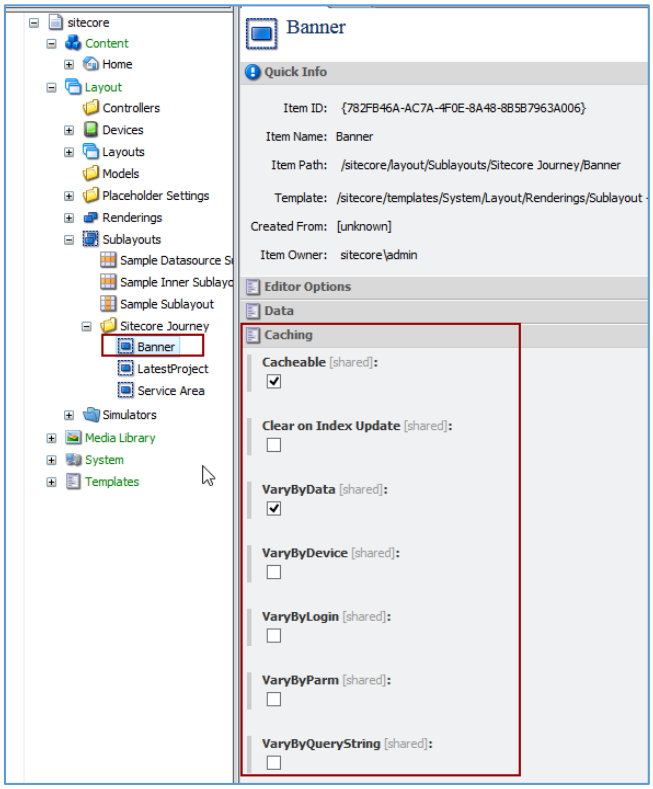
```
<!--  HTML EDITOR DEFAULT PROFILE
      Path to the default html editor profile.
      Default value: /sitecore/system/Settings/Html Editor Profiles/Rich Text Default
-->
<setting name="HtmlEditor.DefaultProfile" value="/sitecore/system/Settings/Html Editor Profiles/Rich Text Default"/>
```

From where can you configure caching of rendering controls?

There are broadly two ways to configure caching of rendering controls.

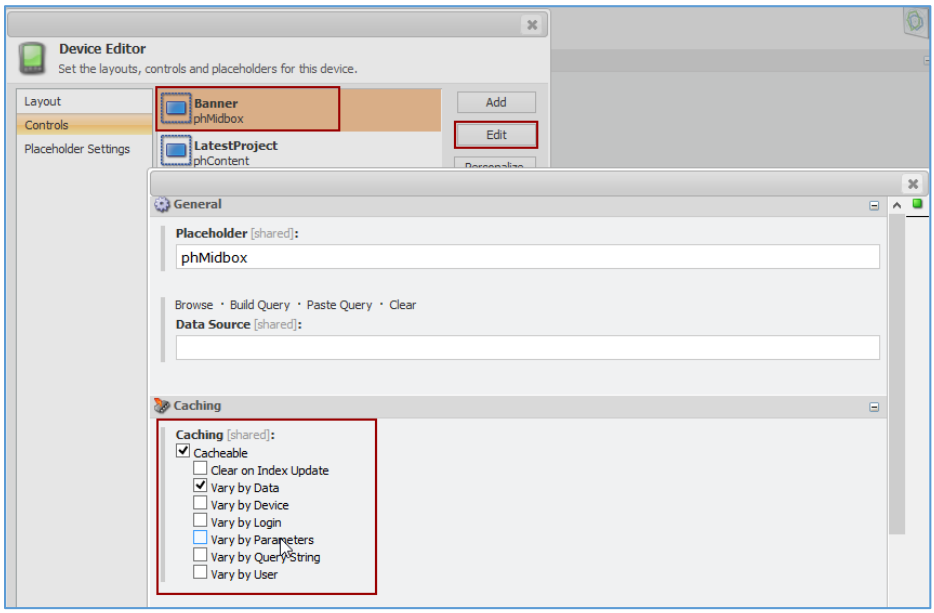
1) **Configure global level**

Each rendering control contains **Caching** section which can be used to set caching parameters. These caching parameters will be applied to all the pages though out the site wherever that rending control is used.



2) Configure item level

Unlike global level, if you want to set caching for particular page only then you can set it at item level as suggested in below figure.



How to get current item in Sitecore?

```
Item contextItem = Sitecore.Context.Item;
```

How to get home page item in Sitecore?

```
public Item GetHomeItem()
{
  return (Sitecore.Context.Database.Items[Sitecore.Context.Site.StartPath]);
 }

Item homeItem = GetHomeItem();
```

How to get string field value in Sitecore?

```
Item pageItem = Sitecore.Context.Item;

string strTitle = pageItem.Fields["Title"].value;
```

OR

```
string strTitle = pageItem["Title"];
```

I found and recommend second approach to get string value because it will not through any exception even though there is no "Title" field found in **pageItem.** While second approach will throw exception if "Title" field not found.

How to get other field value in Sitecore?

```
Item pageItem = Sitecore.Context.Item;

CheckboxField isMenuNavigation = pageItem.Fields["isMenu"];
```

How can you get all child items of item which has particular template?

```
  Item currentItem = Sitecore.Context.Item;

  List<Item> faqItems = currentItem.Children.Where(faq => faq.TemplateID.ToString() ==
"{76036F5E-CBCE-46D1-AF0A-4143F9B557AA}").ToList();
```

If you have image field in item and you want to get that image field URL in order to set on your image control, how can you achieve it using Sitecore API?

```
Sitecore.Data.Fields.ImageField careerAreaImageFieldItem = randCareerArea.Fields["Image"];

//Here is image is <asp:Image> control
image.ImageUrl = MediaManager.GetMediaUrl(careerAreaImageFieldItem.MediaItem);
```

How can you get reference of database in Sitecore?

Database db = Sitecore.context.database; // will return current database

OR

Database db = Sitecore.Configuration.Factory.GetDatabase("master"); //pass respective database name

How to retrieve a template by ID in Sitecore?

```
Sitecore.Data.Database db = Sitecore.Context.Database;
Sitecore.Data.Items.TemplateItem template = db. GetTemplate("{11111111-1111-1111-1111-11111111111}");
if (template!=null)
{
        //code to process template
}
```

How to retrieve a template by name in Sitecore?

```
Sitecore.Data.Database db = Sitecore.Context.Database;
Sitecore.Data.Items.TemplateItem template = db.Templates["path of template goes here"];
if (template!=null)
{
        //code to process template
}
```

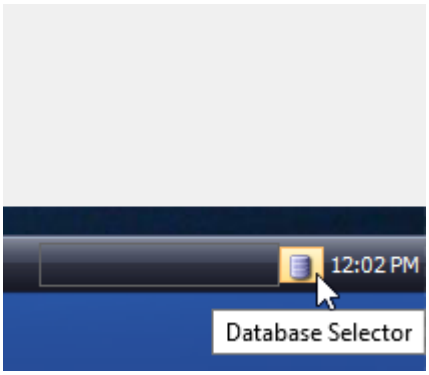
How to switch database from content editor?

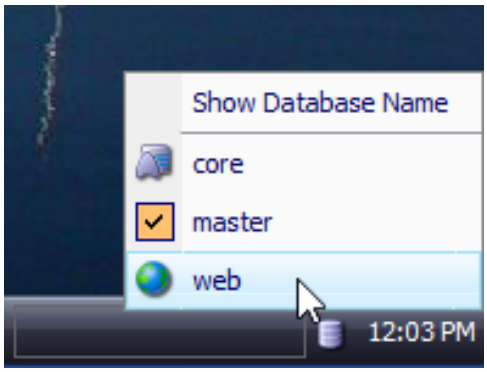
Many times you want to check whether item is published correctly or not, in such cases the obvious way is to check the page by accessing live URL. But sometimes you want to check live

version of sitecore item same as you see in stage environment (e.g. master database) for troubleshooting.

To see live copy of any item (e.g. web database), you need to login desktop mode and can switch database by clicking bottom right corner icon as suggested in below figure.

You can switch among any of Sitecore databases by clicking database selector icon.





How to retrieve the content database reference in Sitecore?

Sitecore.Data.Database db = Sitecore.Context.ContentDatabase;

Which is method provided in Sitecore to write output to HTML outstream?

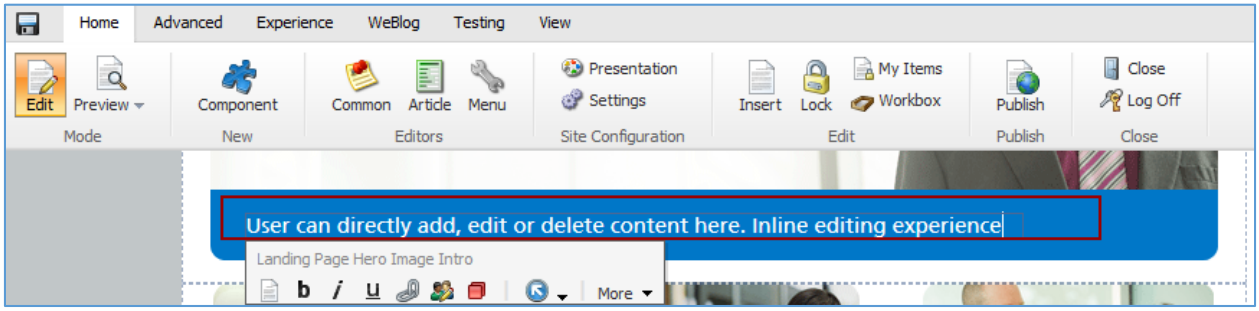
Sitecore.MainUtil.Out(string, Boolean);

How to retrieve current device that is getting used by user?

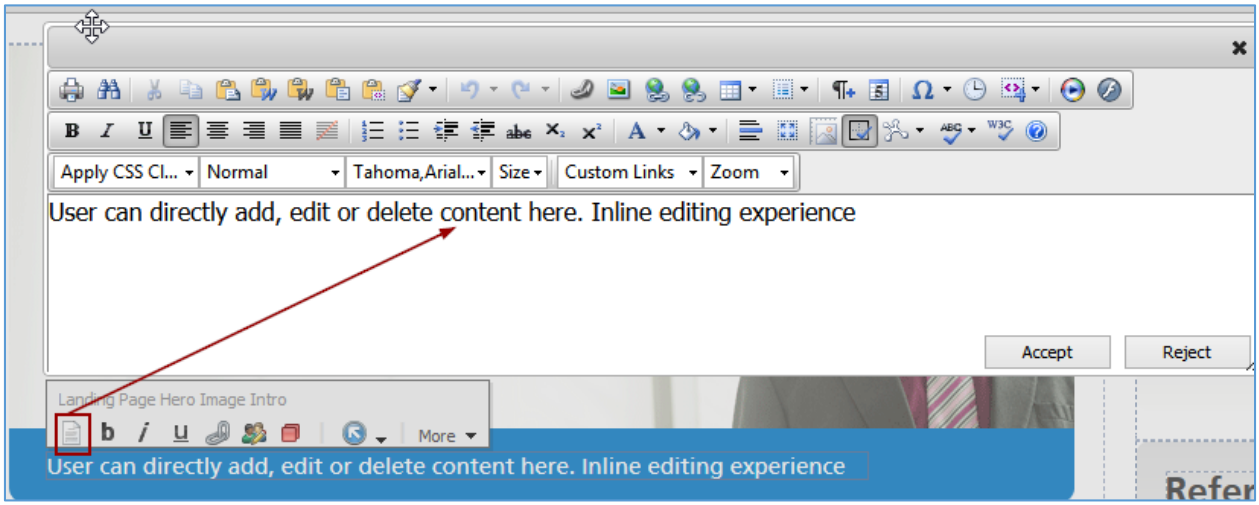
```
string rend = Sitecore.Context.Item.Fields["__renderings"].Value;
if (!string.IsNullOrEmpty(rend))
 {
        Sitecore.Layouts.LayoutDefinition layout =   LayoutDefinition.Parse(rend);

   Sitecore.Layouts.DeviceDefinition device =
layout.GetDevice(Sitecore.Context.Device.ID.ToString());

   Sitecore.MainUtil.Out((device.ID.ToString()));
 }
```

What is inline editing in Sitecore?

One of the eye catching feature of Sitecore is that it provides awesome inline page editing feature. Using inline editiing feature users can insert, edit or delete content from their end.



Moreover there is also option to open Rich Text Editor to update page data. Sitecore is using third party tool (i.e. Telerik control) to provide this Rich Text Editor (RTE).



How to display/render field value to support inline editing?

FieldRenderer.Render([Item],[FieldName]);

```
//lblTitle is <asp:Label> control
lblTitle.Text = FieldRenderer.Render(Sitecore.Context.Item, "Title");
```

How to access Sitecore page in edit, preview and normal mode using URL?

Using querystring like sc_mode = edit, preview and normal

How retrieve selected item from Sitecore Droplink field?

```
Sitecore.Data.Items.Item item = Sitecore.Context.Item;
Sitecore.Data.Fields.ReferenceField field = item.Fields["droplink"];
if (field != null && field.TargetItem != null)
 {
   Sitecore.Data.Items.Item target = field.TargetItem;
   Sitecore.MainUtil.Out(target.Name);
 }
```

How retrieve selected item from Sitecore Generic link field?

```
LinkField link = navigationMenu.Fields[navigationMenuField];
switch (link.LinkType.ToLower())
{
   case "internal":
         //Set link to internal sitecore page URL
         if (link.TargetItem != null)
         {
           string pageURL = LinkManager.GetItemUrl(link.TargetItem);
         }
         break;

   case "external":
   default:
         //Set link to external URL
         string pageURL = link.Url;
         break;
}
```

How retrieve selected item from Sitecore Multilist field?

```
Sitecore.Data.Items.Item item = Sitecore.Context.Item;
if (!String.IsNullOrEmpty(item["Products"]))
{
   MultilistField products = item.Fields["Products"];
   foreach (Item productItem in products.GetItems())
   {
```

```
        Sitecore.MainUtil.Out(productItem.Name);
    }
}
```

Suppose you want to check that which items are referring particular item and you want to retrieve list of these referrer items. How can you get it?

```
Sitecore.Data.Items.Item productItem =
Sitecore.Context.Database.GetItem("/sitecore/content/Home/Product1");
    Sitecore.Links.LinkDatabase linkDB = Globals.LinkDatabase;

    Sitecore.Links.ItemLink[] linkedItems = linkDB.GetReferrers(productItem);
        foreach (ItemLink link in linkedItems)
        {
            Sitecore.MainUtil.Out(link.GetSourceItem().Name);
        }
```

What is difference between item.Fields["Title"].HasValue and item.Fields["Title"].Value?

item.Fields["Title"].HasValue will check that whether title field has different value that standard value

item.Fields["Title"].Value will simply check whether title field contains any value or not.

Which are different types of validations in Sitecore?

- **Valid** = Green, everything is fine
- **Suggestion** = Bright Orange, hmm, take a look at this
- **Warning** = Orange, you should do something about this
- **Error** = Red, this is an error you know
- **CriticalError** = Red, user is warned before saving
- **FatalError** = Red, user cannot save item before validator is cleared

Which of the validations getting applied by Sitecore in Page Editor?

In the Page Editor, Sitecore applies only Critical and Fatal validation issues.

What is event queue and how does it work?

You may have found me saying many times that Sitecore is highly flexible and scalable CMS. The answer of this question is going to prove it one more time.

Event queue as introduced in Sitecore 6.3 version to communicate between CM and CD servers. It helps to trigger events from CM server to CD server.

For example:
When you publish an item in Sitecore, Sitecore will add a record in the EventQueue table of the corresponding database. There is a timer in Sitecore instance which will check EventQueue table periodically and will trigger any event if found.

Where are all events stored in event queue?

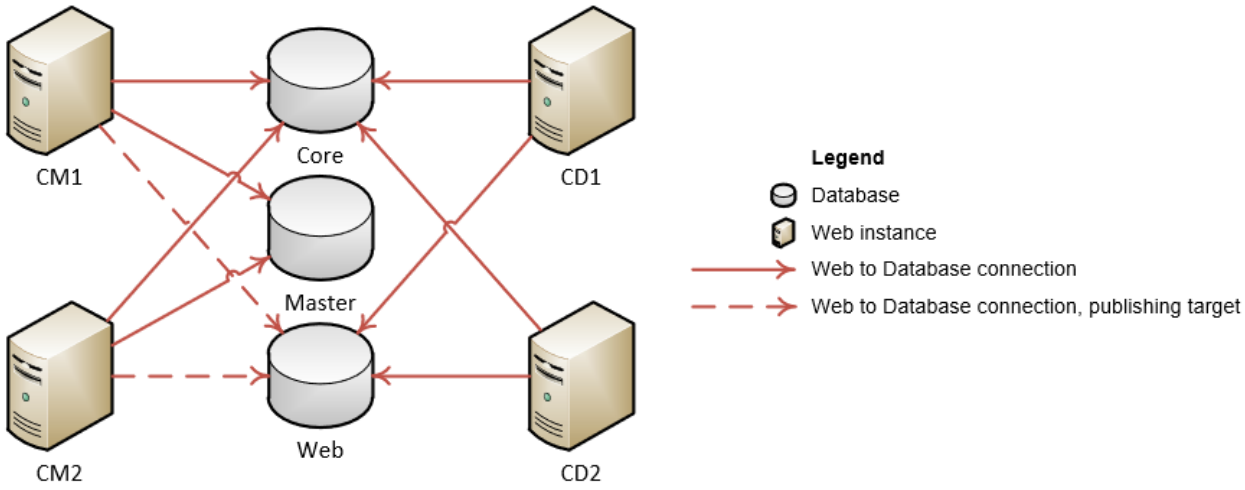
All events are stored under EventQueue table in Sitecore.

What is CM server and CD server?

CM Server stands for **C**ontent **M**anagement server which usually have **core** and **master** databases. It is **stage** server.

Content authors works on CM server to add/edit content. Whenever content author publishes one or more items from the CM servers, it gets published on CD server.

CD Server stands for Content Delivery server which usually have **core** and **web** databases. It is **live/production** server.



What is Sitecore DMS?

Today's marketers have wide range of complex needs. They not only want content management system but also robust **analytics tracking**, **personalization**, **multivariate(A/B) testing**.

Luckily Sitecore has tool like **D**igital **M**arketing **S**ystem (DMS) which provides all of these features under one umbrella.

First of all, how can you check whether DMS is installed in your Sitecore instance or not?

If you are not sure whether Sitecore DMS is configured/setup in your local Sitecore instance or not and want to confirm it, then you can do it by visiting your *ConnectionStrings.config* file. If you find any entry of *Sitecore_analytics* database then it means that DMS is configured in your environment.

So from where can you find Sitecore DMS setup?

If you find Sitecore DMS is not configured in your local instance, then you need to visit SDN portal - http://sdn.sitecore.net/Resources.aspx and download respective Sitecore DMS version.

The downloaded ZIP file will contain *Sitecore.Analytics.config* file and *Sitecore.Analytics* database.

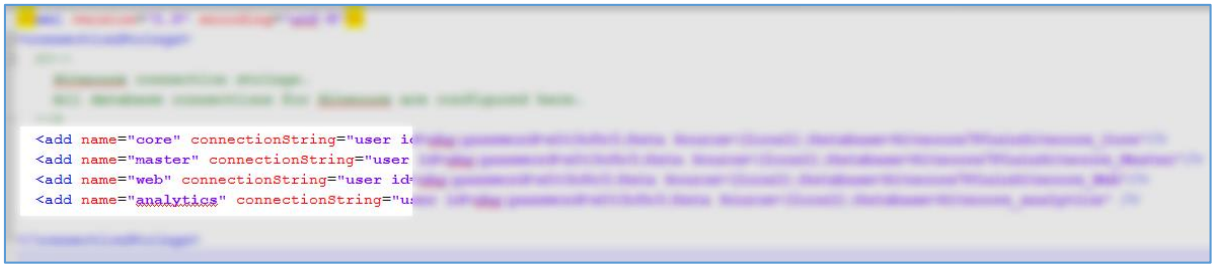
Name	Date modified	Type	Size
Sitecore.Analytics.config	8/15/2012 12:05 PM	XML Configuratio...	28 KB
Sitecore.Analytics.ExcludeRobots.config	3/16/2012 3:00 PM	XML Configuratio...	29 KB
Sitecore.Analytics.ldf	10/9/2012 6:19 PM	SQL Server Databa...	768 KB
Sitecore.Analytics.mdf	10/9/2012 6:19 PM	SQL Server Databa...	2,176 KB

Now how to configure Sitecore DMS after downloading?

Step-1#:

First, you need to add connection string for Sitecore_analytics database in *ConnectionStrings.config* under App_Config folder and attach Sitecore_analytics database in your SQL Server.

1. <add name="analytics" connectionString="user id=[usename];password=[pwd];Data Source=(local);Database=Sitecore_analytics" />



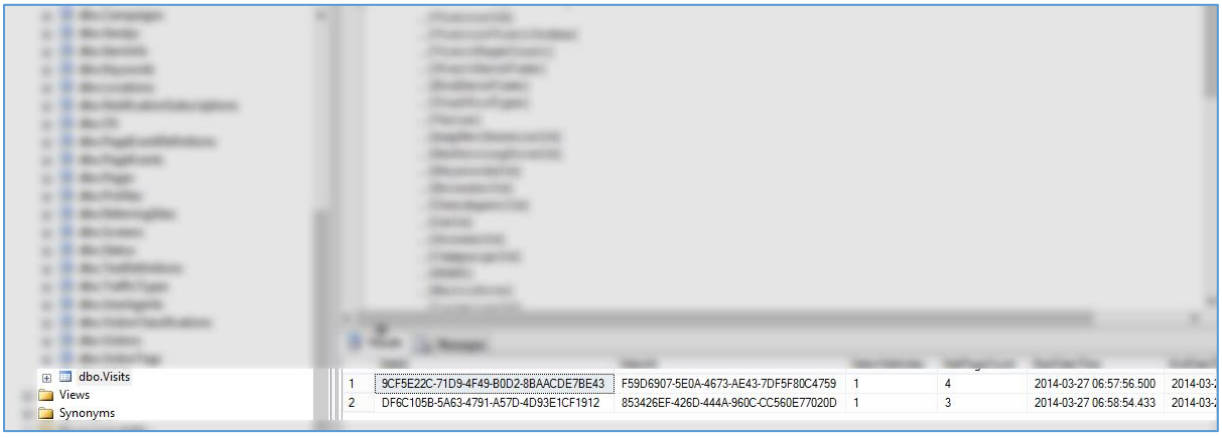
Step-2#:

Copy downloaded *Sitecore.Analytics.config* file to your App_Config/Include folder.

How to confirm Sitecore DMS is working now?

You may be wondering how can you check whether you have configure Sitecore DMS successfully or not. You may want to check whether Sitecore DMS is working and recording any data or not.

In order to check it, visit/preview couple of pages of your Sitecore instance where you have configured DMS and then go to Sitecore Analytics database. Check *Visits* table under Sitecore Analytics database, you should found records of your visits there.



This means your Sitecore DMS is working, enjoy!

But how can you start using Sitecore DMS as you are noob?

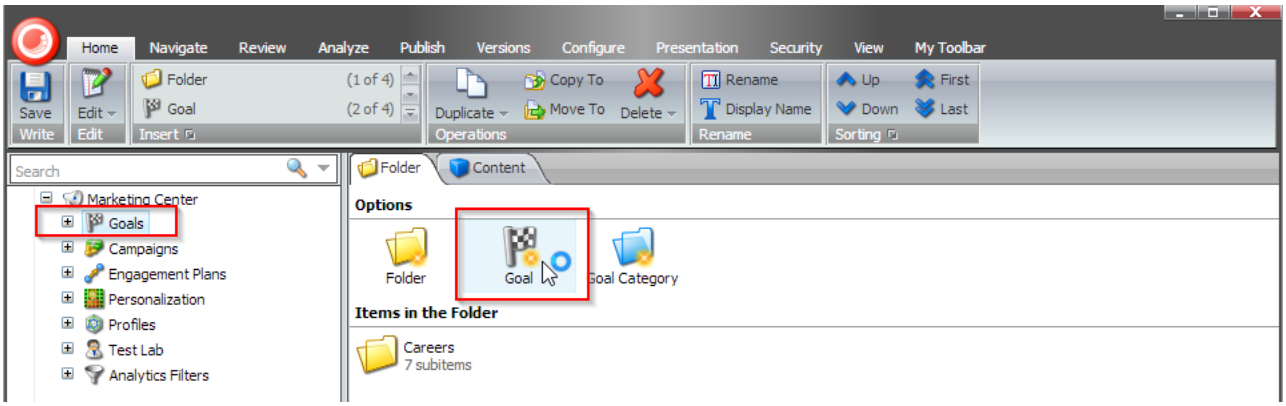
So far, you have only configured Sitecore DMS and confirmed whether it is up and running in your local enviroment.

Now Let's use it to get real advantage of DMS - for which it is made.

There is dedicated menu option given for DMS under Sitecore menu - **Marketing Center**

To work with DMS, you need to go to *Marketing Center*. Here you will find many options like setting **Goals, Campaigns, Engagement Plans, Personalization** and many more.

So let's start with *Goals*, add a new goal as suggested below.



Hey, someone of you may have question - what is goal? why should you create goal? So let's see why are we going to create goal?

Suppose that you want to track visits of all people who have downloaded *ProductManual* PDF from your website. This is called *Goal* in Sitecore DMS terminology, hence you need to create *Goal* under marketing center and assign it to respective PDF (ProductManual).

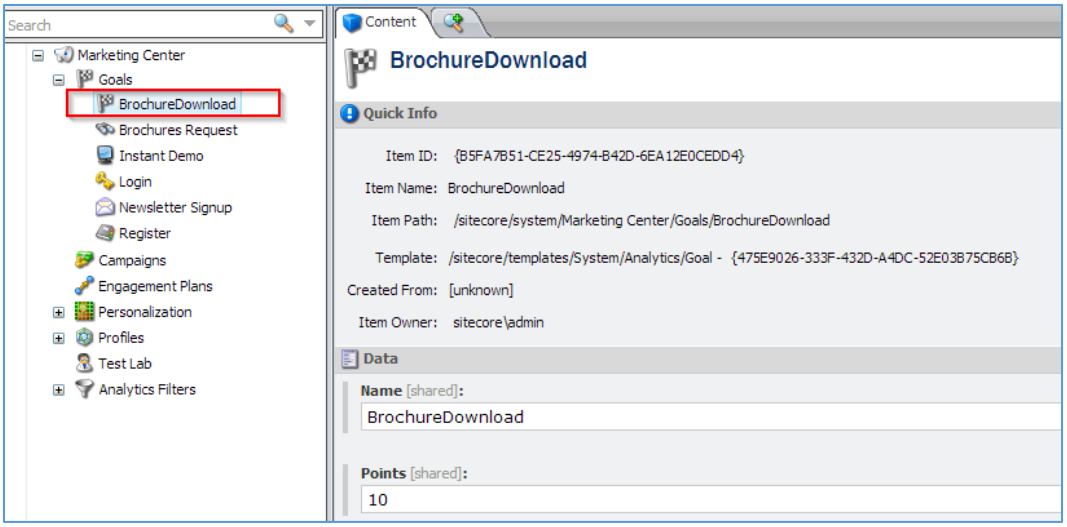
This process is also called as **CAT** in DMS:

Create goal

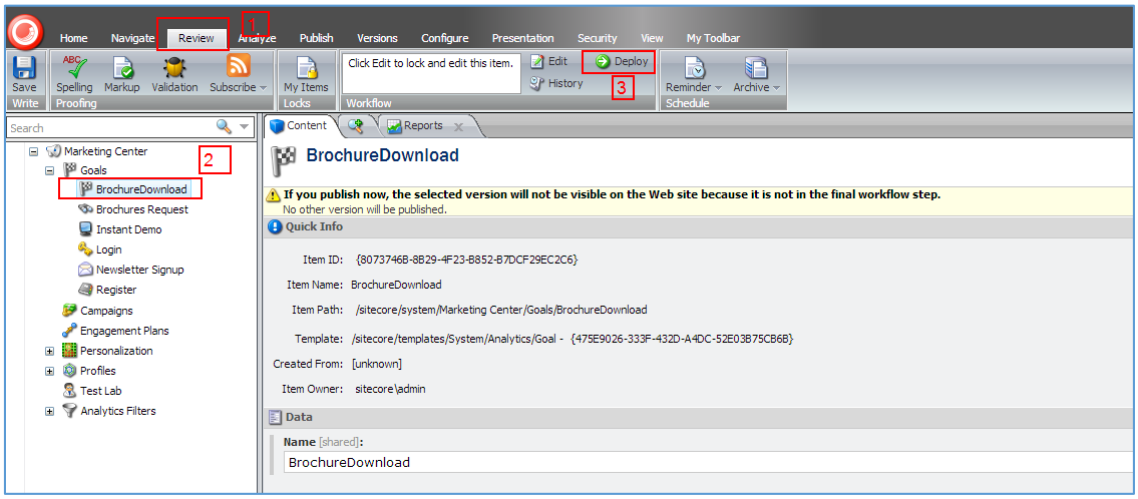
Assign goal

Track goal

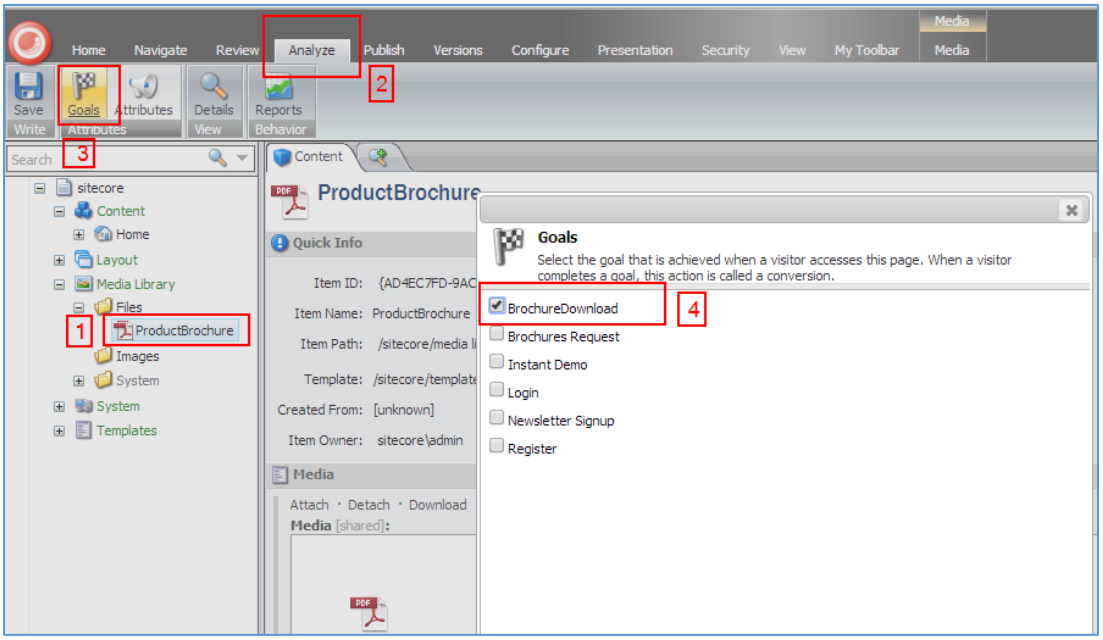
Hope you are now comfortable with terminology - goal. So let's proceed further and **create** goal, I have created goal named - *BrochureDownload*.



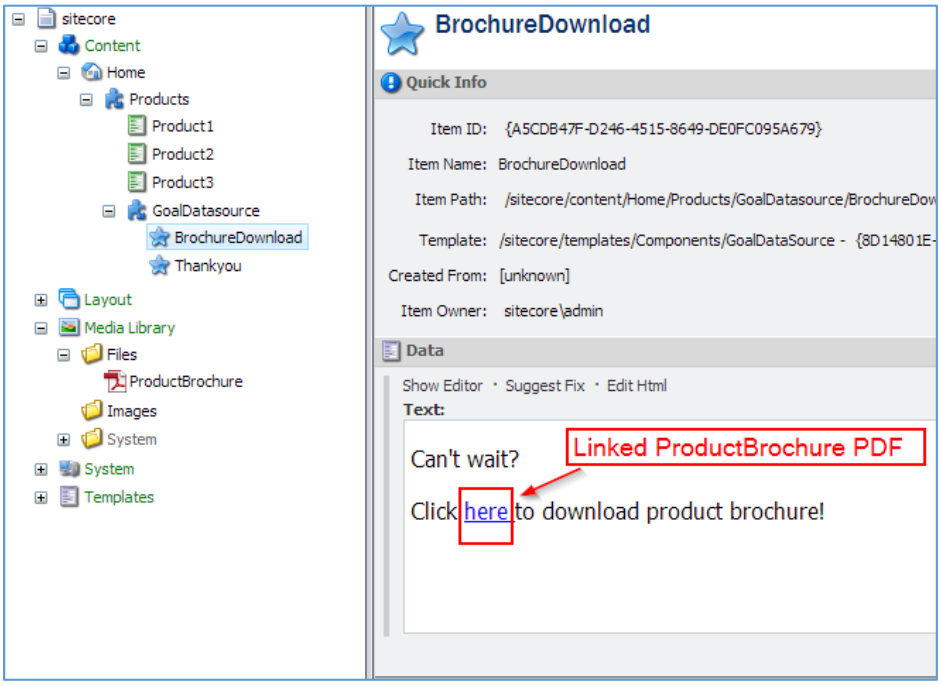
Please note that you must have to deploy goal in order to assign it.



If you forget to deploy goal then you will not be able to see your goal in goal list as mentioned in below screen capture. You need to now assign this goal to *ProductManual* PDF, use below mentioned steps to **assign** the goal.



That's it - your *ProductManual* PDF has goal now. So let's link this PDF to your product page and rest of all things will be managed by Sitecore DMS.

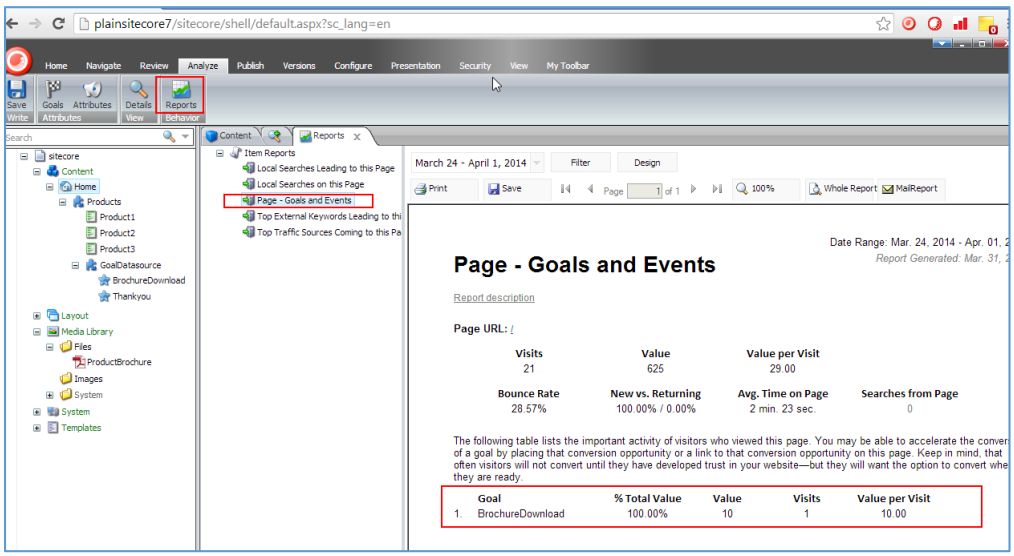


It's time to **track** the goal now. Preview your page where you have linked *ProductManual* PDF and click on link to download it.

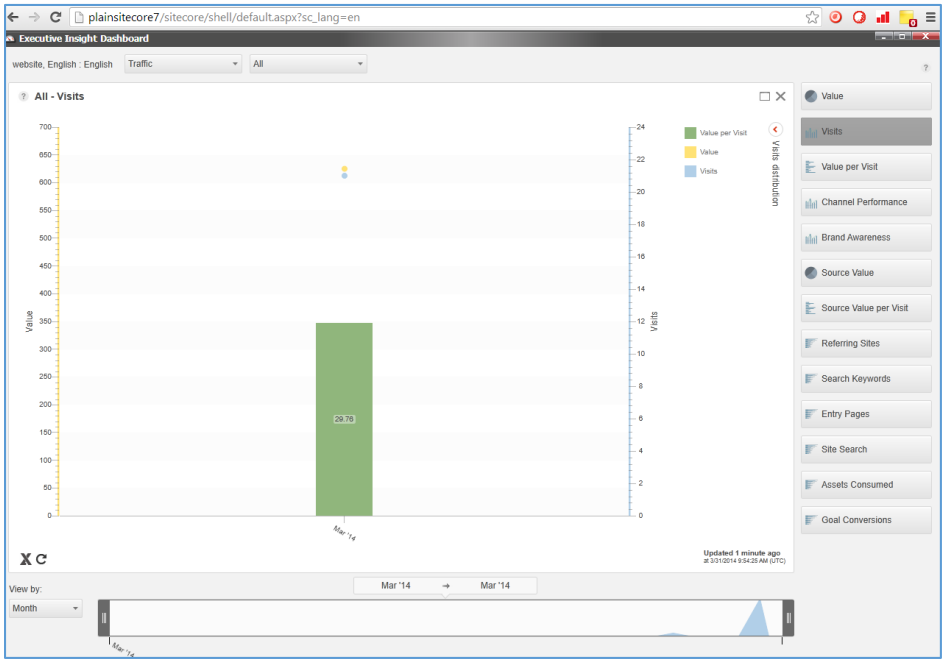
This will of-course download PDF and trigger the your *BrochureDownload* goal that you have set and assign to this PDF.

Again you may have question - how can you check whether my goal is triggered and achieved or not?

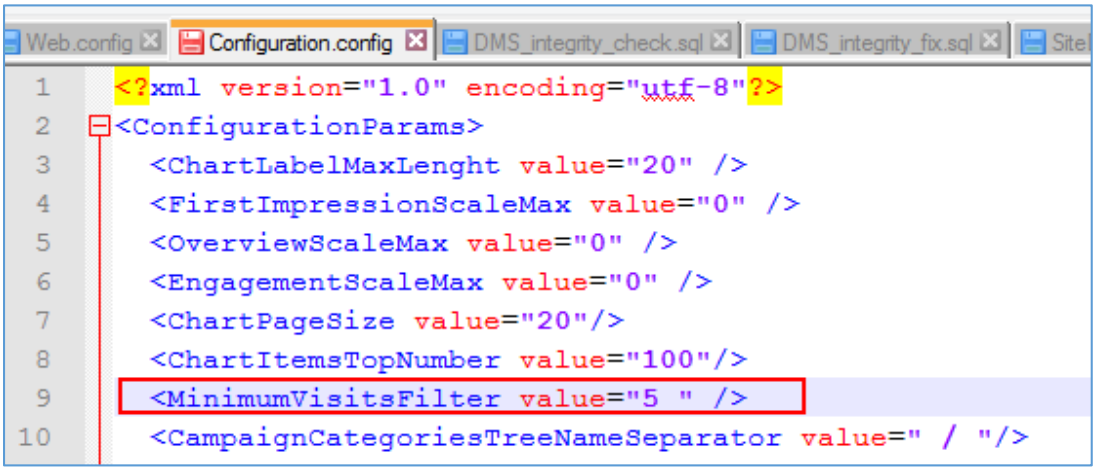
Yes, you have valid question. In order to confirm that, go to **Analyze** menu - select **Reports** option and then select **Page-Goals and Events** report type.



You can also check all such analytics report at **Executive Insight Dashboard** under Sitecore menu.
If you are **not able to see Goal records in reports** (as shown in above figure), please **make sure that you have published that goal.** Publishing goal will start showing data in Goals and Event report.



In case you are not able to view any visit data under Dashboard, that means you don't have enough visit data. To bypass this rule, you can change default sitecore configuration to pull whatever data you have. This can be achieved by changing **MinimumVisitsFilter** value from **50** to small value like **5** at: **\Website\sitecore\shell\Applications\Reports\Dashboard\Configuration.config**



So we have seen one way of triggering goal, fine? Actually there are three ways of triggering goal in Sitecore DMS. In this post, we are going to uncover remaining two ways to trigger goal.

Which are three ways to trigger goal in sitecore DMS?

1. Trigger goal on user action
2. Trigger goal using query string (will be covered in this post)
3. Trigger goal programmatically (will be covered in this post)

Now let's see how to trigger goal using query string?

Sometimes you need to trigger goal without requiring user action. In such case you can take advantage of Sitecore query string variable **sc_trk** like this: http://plainsitecore7/?sc_trk=[goal name]

For Example: http://plainsitecore7/?sc_trk=**BrochureDownload**

The next thing in your mind may come - how can I check whether this query-string triggered your goal or not? You can check it at below section: **How to check whether goal is triggered or not?**

Third way - how to trigger goal programmatically?

Before we proceed to see implementation of programmatically triggering goal, let's see in which scenario you would need this?

Suppose you want to trigger a goal on button click, and you cannot assign goal to button from Sitecore content editor. In such case, you can go with third option - trigger goal programmatically.

Scenario:

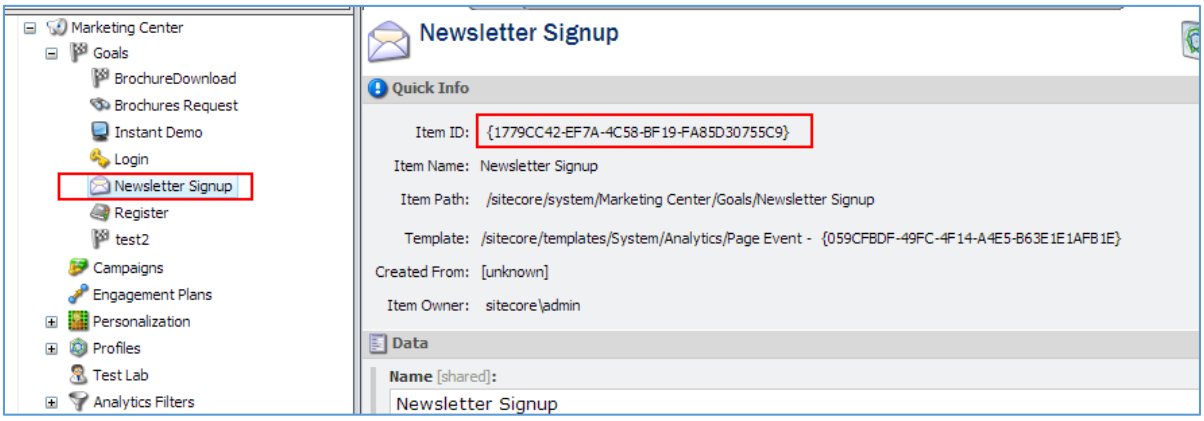
Your site has feature to subscribe for product newsletter, and you need to trigger goal whenever any user subscribed for this newsletter.

Implementation:

Sitecore.Analytics API provides built-in methods to trigger your goal. Here is code that you can hook into button_click event.

```
protected void btnSubscribe_Click(object sender, EventArgs e)
    {
        if (Sitecore.Analytics.Tracker.IsActive && Sitecore.Analytics.Tracker.CurrentPage != null)
        {
            Sitecore.Data.Items.Item GoaltoTrigger = Sitecore.Context.Database.GetItem("{1779CC42-EF7A-4C58-BF19-FA85D30755C9}");
            if (GoaltoTrigger != null)
            {
                Sitecore.Analytics.Data.Items.PageEventItem registerthegoal = new Sitecore.Analytics.Data.Items.PageEventItem(GoaltoTrigger);
                Sitecore.Analytics.Data.DataAccess.DataSets.VisitorDataSet.PageEventsRow eventData = Sitecore.Analytics.Tracker.CurrentPage.Register(registerthegoal);
                eventData.Data = GoaltoTrigger["Newsletter subscription"];
                Sitecore.Analytics.Tracker.Submit();
            }
        }
    }
```

In above code snippet, Item ID-{1779CC42-EF7A-4C58-BF19-FA85D30755C9} is ID of your goal that you want to trigger on button click.

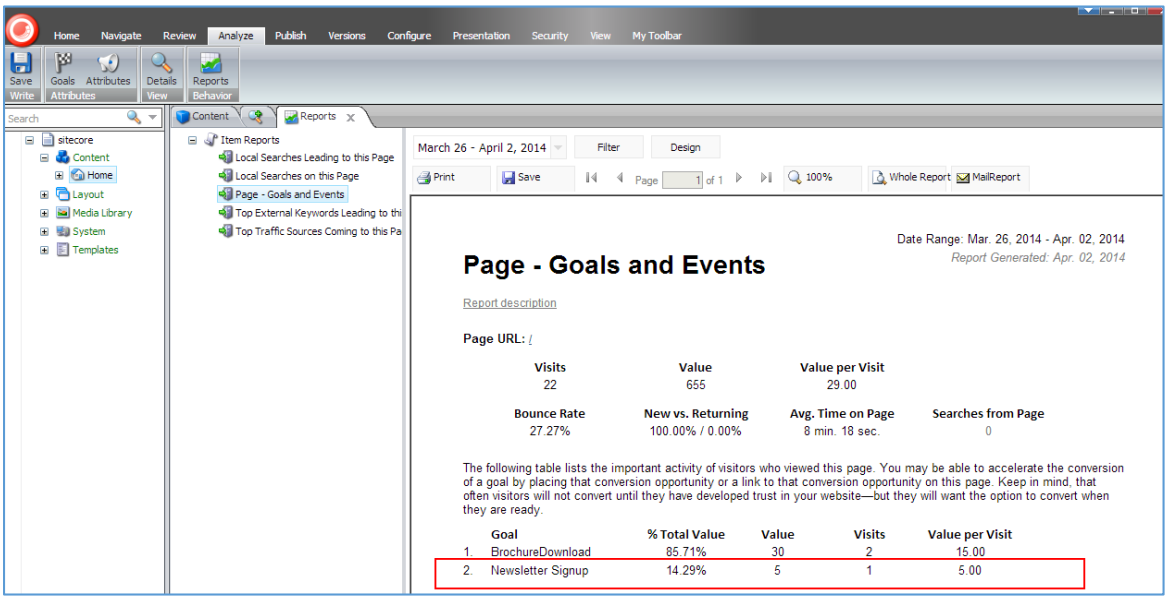


That's it!
Click on "Subscribe" button and it will trigger your "Newsletter Signup" goal. The next question in your mind may come - how can I check whether goal is triggered or not?

How to check whether goal is triggered or not?

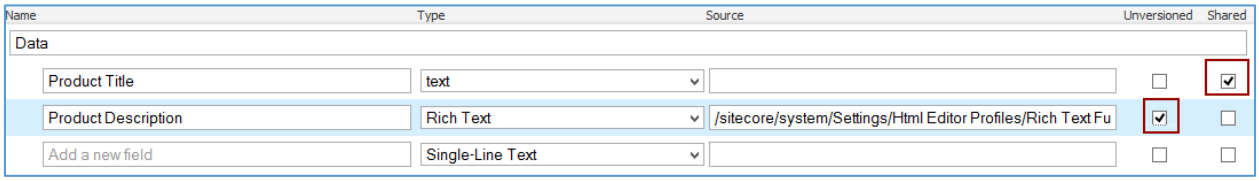
If you have this question in your mind, then at this time I hope you know answer of it also. Previously we have seen how to check report of "Goal and Events". If not able to recall, no problem - let's see report of this new goal - "Newsletter Signup"

Select the page on which you have added "Newsletter Subscribe" button -> go to "Analyze" menu -> click on "Reports" -> select "Page - Goals and Events" report. You will see report having "Newsletter Signup" goal there.



What is shared and unversioned fields?

In multilingual websites, sometimes you want to have same field value across the multiple languages. In such situation, you can take advantage of shared fields. Each field in Sitecore has a checkbox call shared as illustrated in below figure.



Shared: When this checkbox is selected, the field has same value for every version of item for all created languages.

Unversioned: When this checkbox is selected, the field has the same value for every version of item within single language. But it may have different values between languages.

Which are new features in Sitecore 7?

Sitecore 7 has wide range of new features, below are few of them.

- Improved search interface

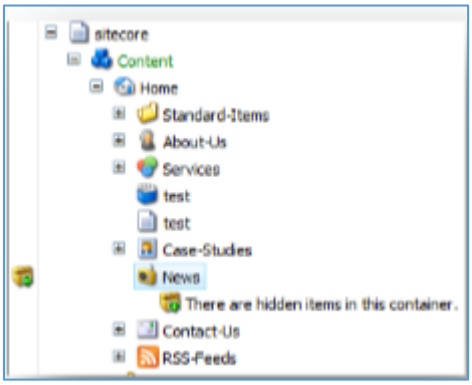
- Content faceting
- Flexible search results
- Solr search
- Introduces **item bucket** to remove restriction of 100 child items

What is item bucket and what is advantage of it?

Sitecore 7 has introduced new feature/concept call item bucket.

An item bucket is repository in the content tree that can store other content items. The difference between an item bucket and regular container in the content tree is that an item bucket can store a theoretically unlimited amount of items without displaying them in content tree.

If the items in bucket are hidden, a small notification tells you that there are hidden items in the container as shown in below figure.



How to configure image quality when you resize image using querystring parameter like w (width) and as (allow stretch)?

There is setting in web.config like:
<setting name="Media.Resizing.Quality" value="95" />

How to reset engagement profile value in Sitecore DMS?

It can be reset by using below Sitecore API:
Sitecore.Analytcis.Tracker.CurrentVisit.GetOrCreateProfile("Name of the Profile").RollBackScore("Name of the Profile Key",0);

What is the best way to play with querystring in Sitecore?

Very unknown feature provided by Sitecore out-of-the-box is Sitecore.Text.UrlString().
You can easily extract or add parameters in your querystring using this method.

Let's see an example:
Do you want the value of the "b" querystring?

```
var url = new Sitecore.Text.UrlString("http://www.someurl.com?a=1&b=2");
var b = url.Parameters["b"]; //2
```

Do you need to add a "c" querystring?

```
var url = new Sitecore.Text.UrlString("http://www.someurl.com?a=1&b=2");
url.Add("c", "3");
```

Want to get rid of the "a" querystring?

```
var url = new Sitecore.Text.UrlString("http://www.someurl.com?a=1&b=2");
url.Remove("a");
```

How to change height, width and other parameters of image dynamically in Sitecore?

w = Image width
h = Image height
as = Allow stretch.
Set as=1 to allow image to be larger than original.
Set as=0 to disallow image to be upscaled, filling any space with a color (default is black).

bc = The background color of the fill when as=0. bc=000000 is black and bc=ffffff is white.
sc = scale image. Use decimal numbers. sc=1 is the original size. sc=1.5 is 150% above normal size.

When should you use Sitecore Clean Up Database feature?

It's recommended to clean up Sitecore databases at regular interval. **Clean Up Databases** allows you to select the database(s) to clean with following options:

1. Remove items those have parents, but the parents are not in the item tree.
2. Remove invalid language data.
3. Remove fields for non-existing items.
4. Remove orphaned items.
5. Remove unused blob records.
6. Rebuild the Descendants table (which stores parent/child relationships).
7. Clear all caches.

How to add a new CSS class in rich text editor dropdown?

There are two steps involved in order to add a new CSS class in RTE.

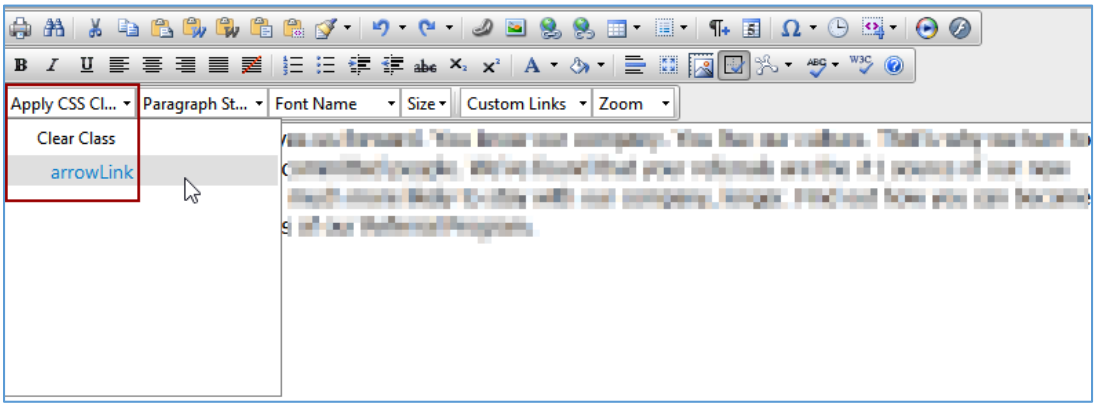
1) Add <class> element in **Website\sitecore\shell\Controls\Rich Text Editor\ToolsFile.xml** file

```
<classes>
<class name="arrowLink" value=".arrowLink" />
</classes>
```

2) Now add the definition of that CSS class in **Website\default.css** file.

```
.arrowLink {
    font-size: 14px;
    color: #0077c8;
    padding-left: 10px;
    position: relative;
}
```

You can see newly added **arrowlink** css class in rich text editor dropdown.



How to hide an item in Sitecore?

1) Manually from content editor:

2) Programmatically:

item.Appearance.Hidden = true;

How to see hidden items in Sitecore?

Which of the following Sitecore DLLs cannot be decompiled?

1) Sitecore.Kernel.dll
2) Sitecore.Client.dll
3) Sitecore.nexus.dll

Answer: **3**

How to report issue to Sitecore team?

While working on Sitecore, many times developer come across an issue or a bug and would like to discuss the same with Sitecore team. In such situation, developer can get in touch with Sitecore support team via support portal.

http://support.sitecore.net/

Please note that you must be Sitecore certified developer having necessary credentials in order to access Sitecore support portal.

Where to find third party tools or modules about Sitecore?

There are wide range of awesome tools and modules contributed by Sitecore community available at Sitecore Marketplace. Every Sitecore developer must be aware about it.

https://marketplace.sitecore.net/

From where can you get more help about Sitecore?

Whenever you stuck up while working on Sitecore and need help, you can reach out to Sitecore community via Sitecore Developer Network (SDN) forum to discuss your requirement.

http://sdn.sitecore.net/Forum.aspx

Besides that you can also check at Sitecore knowledgebase for known issues and how-to's.

https://kb.sitecore.net/

By Author of Sitecore Journey Blog
http://sitecorejourney.nileshthakkar.com