SiteNOTE

Designer's guide

Version 3.2.XXXX (8/7/2009)

Table of contents

1 Introduction

Welcome to SiteNOTE user's guide, a simple and powerful solution to maintain web site content. With SiteNOTE, it is possible to manage, update or create the content of a web site, while preserving a consistent look and feel throughout the pages. Maintenance and update tasks are dedicated to content experts, who will directly control the web site quality and lifecycle.

The reader should have a working knowledge of SiteNOTE as content management system (see user guide) as well as to the use of CSS and HTML.

1.1 Sample site used in this document

This documentation is using an empty sample site containing only 2 empty pages (containing no content except the shared (empty) default ones). The default page is a page group that can be used as a splash page and the entry "en-us" page is the homepage of the site.

2 Administration interfaces

2.1 Login screen

To access the administration version of the site, it is necessary to login in its "offline" version. Doing this is achieved by opening the "admin.aspx" page in any folder of the site which prompts the login screen as displayed in the picture below.

2.2 Content management toolbar

When accessing the site administration, after entering an authorized username and password in the login page (admin.aspx), the web site is displayed in the browser. The top of the window shows the toolbar, which allows to perform administration tasks.

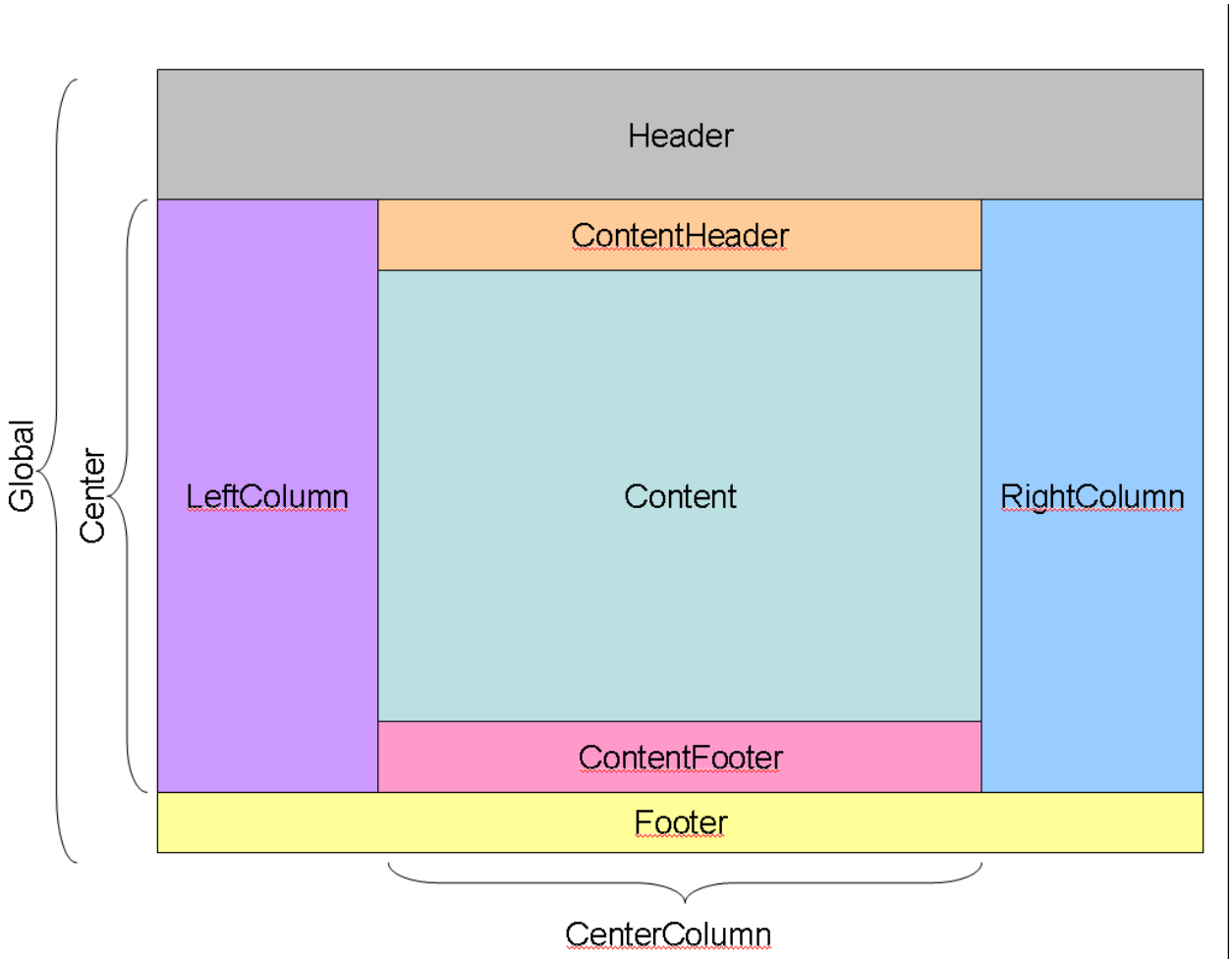
2.3 Administration toolbar

Clicking on the "website administration" icon of the main SiteNOTE toolbar opens the administration. On top of that screen a toolbar is displayed that contains the following elements: Users administration, Groups administration, Templates Design, All Pages View and Status page (monitoring). Templates Design is reserved to developers and is the main purpose of this document.

3 Designing the look of a SiteNOTE website

3.1 General structure of the site

A SiteNOTE site's layout is based on a general template as pictured below. This template is divided in blocks. It is not mandatory to use all blocks, some of them may remain empty or hidden, but the division as provided allows to apply mostly any design.



Each "block" in the structure is associated to a CSS identifier listed here:

- #Global - #Header - #Center - #LeftColumn - #CenterColumn - #ContentHeader - #Content - #ContentFooter - #RightColumn - #Footer

The HTML "body" tag contains a "#BodyPage" identifier and several classes :

- ***Level_"level of the page"***: Page level starts at 0 for the "default" page and raises by 1 for the subpages (level of 'default'=0, level of 'en-us'=1, subpages of 'en-us'=2, etc...). For example, the page 'en-us' will have the "Level_1" class.
- ***Name of the page's template***: When creating a page, it is possible to choose the template to use (Page or Popup). In the case of the page 'en-us' which is of type = 'Page', the class will be "Page".
- ***Page name***: In the case of the 'en-us' page whose type is 'Page', the class will be "en-us".

In short, the 'en-us" page's body properties will be: id="BodyPage" class="level_1 Page en-us"

Using the CSS classes related to the "body" tag, it is possible to obtain easily a different design according to the pages and their level in the site hierarchy, displaying or hiding one or more blocks (ex: the homepage is often different tha then other pages of the site)

For Example :

- level_1 #LeftColumn { display:hidden; }
- .level_2 #LeftColumn { display:block; }

or another example

- .page1 #LeftColumn { display:hidden; }
- .page2 #LeftColumn { display:block; }

3.2 Templates design tool

3.2.1 Accessing the template(s)

It is possible to define several different templates for the same site. This gives the advantage to be able to realize and test a new design while keeping the existing one running.

Clicking on the **Templates Design** (

) menu item allows to access the page displaying the list of designs that can be applied to the site.

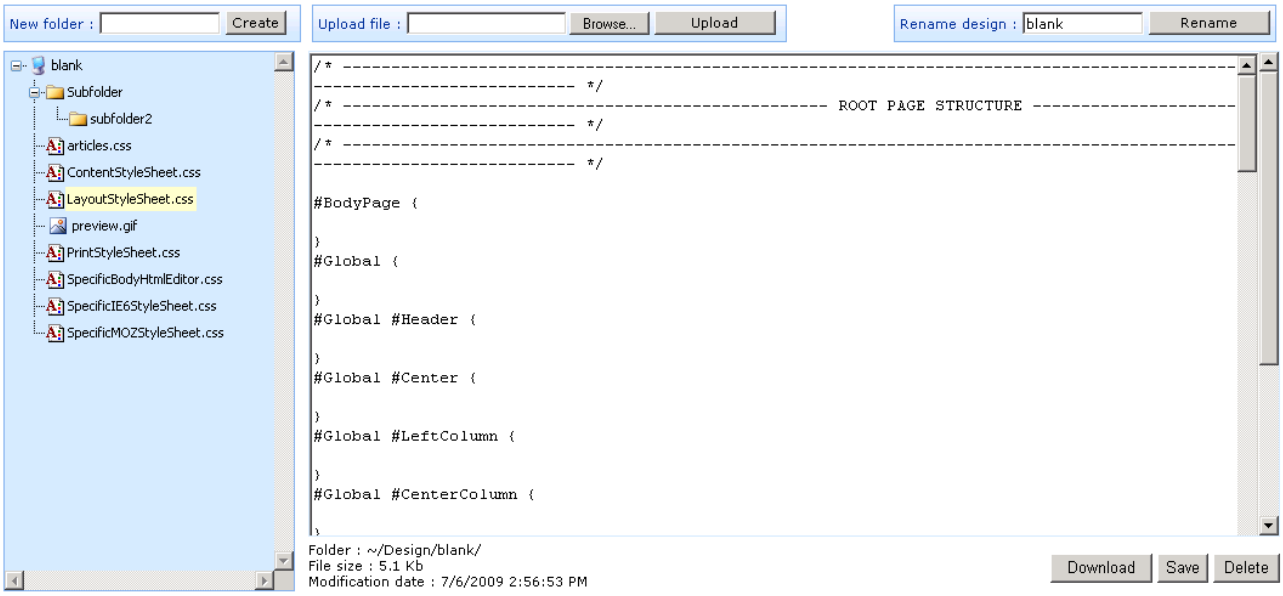
A click on the image allows to display it in a bigger size popup window.

- ***Preview***: applies temporarily the corresponding design. The user is then able to browse the site previewing the selected design. A design toolbar will appear under the content edition toolbar. This design toolbar allows switching the design to another one using a dropdown list that contains the different designs, as well as 3 buttons that allow to definitely apply the previewed design, or to cancel the preview or to edit the previewed design.
- ***Apply***: applies definitely the corresponding design to the site.
- ***Edit***: allows accessing the edition page of the corresponding design.
- ***Copy***: allows making a copy of the corresponding design.

- ***Delete***: allows deleting the corresponding design.

3.2.2 Editing the design of a template

A click on the *Edit* allows to access the edition page of the corresponding design.



The screen is composed of the following blocks:

- ***New Folder***: this block allows to create a folder in the root folder of the design or under the selected folder.
- ***Upload file***: this block allows to add or replace a file located in the root folder of the design or under the selected folder.
- ***Rename design***: this block allows to rename the design (useful after a copy-paste of a design).
- ***File tree***: this block allows to browse the root and sub folders of the design.
- ***Edition bloc***: block in which the selected file is displayed and from which some operations can be performed.

Related to the *Edition bloc*, when a file is selected, several manipulations can be done depending on the file type. A CSS file can be edited (modified and saved), downloaded and deleted. Files of other types can be downloaded and deleted. Information related to the file are displayed below the block (folder in which it is located, size in kb, last modification date)

N.B: Creating and deleting a folder is also possible. Notice that deletion will be achieved in 2 steps: first step corresponds to the deletion of folder's files but the folder remain accessible; second step is deletion of the folder which occurs only when the site is restarted.

3.3 Site Layout management

While the template design defines the general look of the site as well as the look of the component that may be included in it, the Layout design tool allows to add/delete items on a page (or section of the site ...) in each block. These items can be related to content, navigation or search or be specifically developped for the site.

3.3.1 Layout design tool

Clicking on the **Edit Layout** () icon of the SiteNOTE toolbar allows adding, deleting or modifying properties of several content types.

Clicking on the **Add content** () icon of the SiteNOTE toolbar allows accessing the catalog of the several authorized content types.

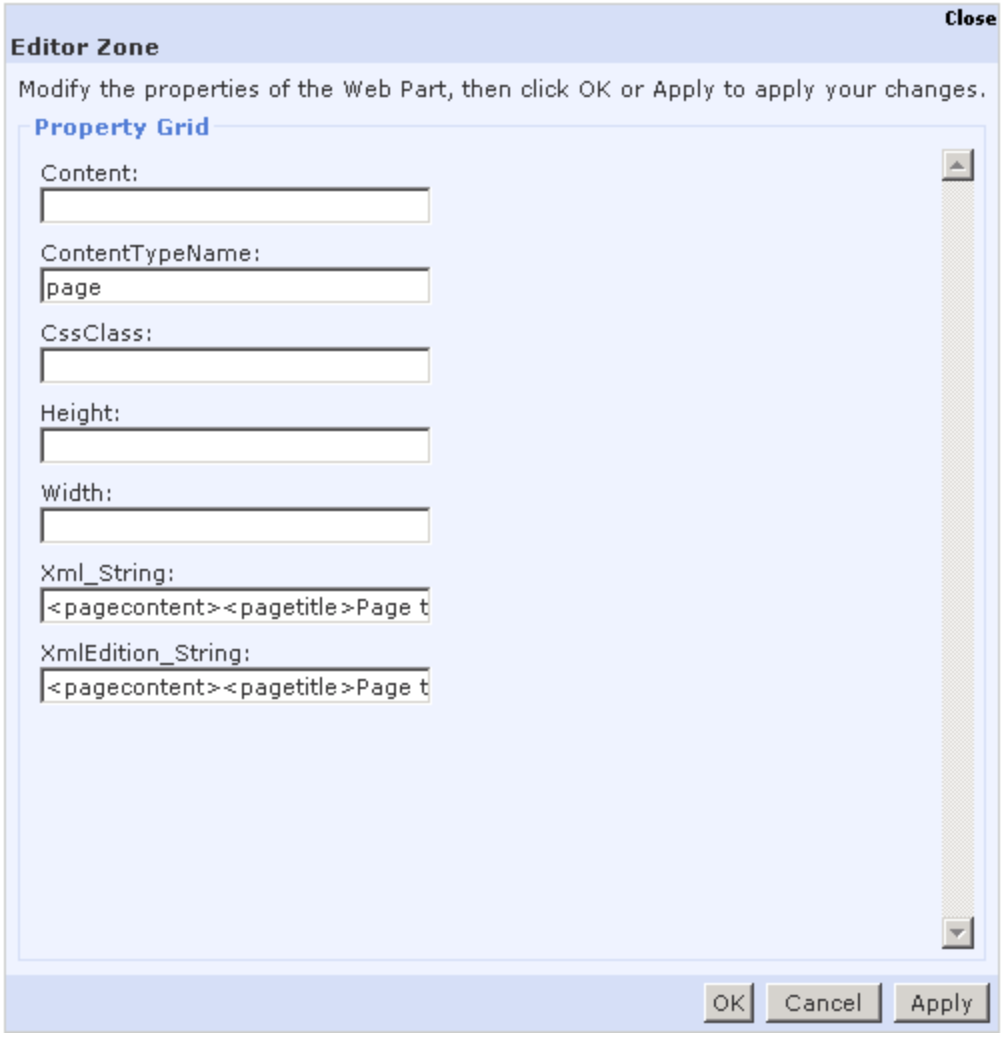
Adding content consist in checking the needed content item(s) and selecting the block in which this(these) content item(s) have to be inserted. Once a content item added and the catalog closed, this content item can always be moved by drag-dropping from a block to another (the mouse button can be clicked up once a blue line appears in the block in which the content item should be inserted).

Editing a content item properties can be done by clicking on the small white arrow at the right of the content item name. A sub-menu appears containing the following actions:

- ***Delete***: Appears only for content items that are not default ones. Allows to delete the content item.
- ***Close***: Allows also to remove the content item.
- ***Edit***: Allows to open a box where it is possible to edit the content item properties (see picture).

When saving a modification of a page's layout, the page is listed in the publish screen of SiteNOTE. The modification will then only be visible online after approval/publishing of the page.

It is naturally possible to return to a previous version via the history of the page.



3.3.2 Particular Items

The use of some particular elements requires a specific care, especially related to the reason of usage and the pursued aim.

3.3.2.1 Free Style Content

The "Free Style Content" item is linked to a HTML content editor that works via a WYSIWYG interface or via a HTML coding one. It is very adviced to use this content only for special cases that do not fit inside the limits of the common graphical layout. (ex: using a specific color, inserting a table, creating an IFrame ...)

3.3.2.2 XmlEditor VS Section Structured Content

The "Section Structured Content" item is equivalent to the "Section" object of an XmlEditor content item but it is single and simplified (by defaul, it contains none of the CSS classes and does not have any external and internal margins). Its main advantage is that it can be redefined independantly of a normal section (via the XSD schema to add some specific objects and/or via the XSLT for the layout design).

3.3.2.3 Shared Content

A "Shared Content" allows that a page inherits some content from another page. Typically, the "Header" block is defined once on the default page (ex: adding a "Section Structured Content" item in which a banner image is inserted in edit mode

+ adding a menu) and all other pages inherit of this block. Modifications are then done only once in the "Header" block of the "default" page and not in all pages of the site that display this block.

Once the content item(s) placed in the "Header" block of the "default" page, editing of those is performed either via a click on the "Edit" button of the SiteNOTE toolbar displayed on the "default" page, or via a click on the shared content when editing a page that inherits of this contet (this one appears gryed), which will trigger the redirection to the default" page in edit mode.

By default, all blocks (see 3.1 General structure of the site) except the "Content" blok (that cntains an XmlEditor content item) contain a shared content.

3.3.3 Menus

Menus usage is generally as such:

- Add menu in a top-level page.
- Modify parameters of the menu properties (see 3.4.2.1 Menu)
- Inherit the menu in sub-pages.
-

For example, for nthe www.SiteNOTE.net website, menus are defined as such:

Header menu

- Menu is located on "home" page
- Parameters are defined as such : DataSourceID : SiteMapDataSource1 - StartingNodeOffset : 1 - AddParentNode : true
- Sub-pages inherit of the "parent" menu
- CSS classes are defined as such : see 3.4.2.1 Menu, CSS Example of horizontal menu having several levels, in the Header block.

Left column Menu

- Menu is located in each page below the "Home" page (Features, FAQ,... that are "page groups")
- Parameters are defined as such : DataSourceID : SiteMapDataSource2 - StartingNodeOffset : 1 - AddParentNode : false
- Sub-pages inherit of the "parent" menu
- CSS classes are defined as such : see 3.4.2.1 Menu, CSS Example of verticla menu having 1 level, in the LeftColumn block

Notice that there are several ways to obtain the same result (examples : a menu in each page, left column menu on a page of a higher level and raising the "StartingNodeOfsset" parameter,...)

3.3.4 Specific custom control

3.3.4.1 Developping a specific custom control

Development of a specific DOT NET control for a SiteNOT website may be necessary to realize business specific functions/applications that are not part of a basic SiteNOT installation. This consists in developping an ASCX control or a CS class.

Difference is :

ASCX

- This control is made of the 2 files. One ***.ascx*** file that contains the code executed client side (HTML, JS, ...) and one ***.ascx.cs*** file that contains code executed server-side (C#).
- This control may be at the root of the site, ot in any sub-folder. The best way is to store all controls in a folder that's dedicated for this usage (ex: /Controls). Take care that the folder's name cannot be a reserved name (ex:App_Code) !

CS

- This control is made of a single ***.cs*** file that contains code executed server-side (C#). Layout design must then be also realized using this code.
- This control must be located in the "/App_Code" folder at the root of the site.

In both cases, if a modification is made to the control, the code will automatically be recompiled and the modification will be active immediately.

Some specific constraints should be taken into account while developping such control.

- Use control-specific CSS classes so that you do not accidentally redefines a SiteNOTE base CSS class. For example, use the control's name as prefix for the classes.
- At the minimum, add the following attributes to your control's eventual editable properties [WebBrowsable, Personalizable], so that they can be edited via SiteNOTE's "Edit Layout" tool.
- Use relative paths ("/default/" or "../default/").

3.3.4.2 Allowing usage of a specific custom control in SiteNOTE

If the developped control is an ASCX one :

- Add the "register" tag below the first line of the "RootMasterPage.master" file locate in the "/Templates" folder.(ex: Register Src="~/Controls/MyControl.ascx" TagName="MyControl" TagPrefix="uc1")
- Add the control (defined via the "register" tag) in the authorized content types (tag SiteNoteContents:SiteNoteAllowedContents) located at the bottom of the "RootMasterPage.master" file (ex: uc1:MyControl Title="MyControl" ID="MyControl" runat="server")

If the developped control is a CS Class :

- Add the "register" tag below the first line of the "RootMasterPage.master" file locate in the "/Templates" folder.(ex: Register TagPrefix="cc1" Namespace="MyControls")
- Add the control (defined via the "register" tag) in the authorized content types (tag SiteNoteContents:SiteNoteAllowedContents) located at the bottom of the "RootMasterPage.master" file (cc1:MyControlclassName Title="MyControlclassName" ID="MyControlclassName" runat="server")

3.4 CSS classes

3.4.1 Content types and related CSS classes

3.4.1.1 Structured content (XmlEditor)

- ***Section***: .section - .sectionColored - .sectionIndent - sectionMargin
- ***Page title***: h1
- ***Section title***: h2 - .coloredSectionTitle
- ***Section subtitle***: h3
- ***Paragraph***: .paragraph - .paragraphSpacing - .paragraphIndent
- ***List of paragraph***: .unorderedList - .unorderedListIndent - .orderedList - .orderedListIndent - .paragraphList - .paragraphListSpacing - .paragraphListIndent - .listHyphen - .listArrow - .listSquare - .listCircle
- ***List of objects***: .unorderedList - .unorderedListIndent - .orderedList - .orderedListIndent - .listItem - .listItemIndent - .listHyphen - .listArrow - .listSquare - .listCircle
- ***Link 'More'***: .moreLink
- ***Link to top of page***: .topOfPageLink
- ***Picture***: .noborder - .imageborder - .alignright - .aligncenter - .alignleft - .floatright - .floatleft
- ***File***: .linkfile
- ***Flash***: .noborder - .flashborder - .vfsalignright - .vfsaligncenter - .vfsalignleft - .vfsfloatright - .vfsfloatleft
- ***Video***: .noborder - .videoborder - .vfsalignright - .vfsaligncenter - .vfsalignleft - .vfsfloatright - .vfsfloatleft
- ***Video***: .noborder - .slideshowborder - .vfsalignright - .vfsaligncenter - .vfsalignleft - .vfsfloatright - .vfsfloatleft
- ***Horizontal line***: .horizontalline - .horizontallineColored
- ***Bloc***: .bloc - .blocColored - .blocIndent - .blocMargin - .blocBorderTop - .blocBorderLeft - .blocBorderRight - .blocBorderBottom - .alignright - .aligncenter - .alignleft - .floatright - .floatleft
- ***Tabbed sections***: .SiteNOTE_Tab - .SiteNOTE_SectionsContainer - .SiteNOTE_TabbedSections - .SiteNOTE_TabsContainer - .SiteNOTE_ImageTab - .SiteNOTE_ImgTabTag - .noborder - .imageborder

3.4.1.2 Form

.formContent

- ***Form***: .sectionForm - .sectionFormColored - .sectionFormMargin
- ***Sub section***: .subSectionForm - .subSectionFormColored - .subSectionFormMargin

- ***Others elements***: .formlabel - .tableform - .formtext - .tdformlabel - .formvalue - .tdformvalue - .forminput - .ddlRecipient - .mandatory - .mandatoryLabel

3.4.1.3 Events

Classe(s): .eventContent - .eventContainer - .dateEvent - .eventTitle - .eventText

3.4.1.4 Gallery

Classe(s): .galleryContent

3.4.1.5 FAQ

Classe(s): .faqContent - .FaqLabelQuestion - .FaqLabelAnswer

3.4.1.6 Links list (Links)

Classe(s): #linksBloc - .linksList - .linktext - .linkimg

3.4.1.7 Paragraphs list (Paragraphs)

Classe(s): #paragraphsBloc - .paragraphsList

3.4.1.8 Section Structured content

Classe(s): idem section object from structured content (XmlEditor)

3.4.1.9 Shared content

A shared content allows to inherit a part of the content from another page. The useful parameters are the following:

- ***Shared Bloc ID***: allows to define the bloc that will be inherited of (Header, LeftColumn,... - See "General structure of the site". The block will always have the 'SiteNoteBlocContent_' prefix. ex: SiteNoteBlocContent_Header).
- ***ContentId***: allows to define the content inside the block that's specified in 'shared Bloc ID" that will be inherited of (if not specified, all of the content items in the specified 'Shared Bloc ID' bloc will be inherited of).
- ***Shared page url***: allows to define the page on which the content to inherit of is located. The possible values are:
  - ***parent***: the content to inherit of is located in the parent page (the parent page's content can also be a shared content).
  - ***root***: the content to inherit of is located in the first page of the site (the default page).
  - ***languageEntry***: the content to inherit of is located in the language entry page (in this case, the 'en-us' page).
  - ***firstSibling***: the content to inherit of is located in the first reachable page on the same level in the site hierarchy.
  - ***Internal relative path***: the content to inherit of is located in a page of that site that can be accessed via the entered url (ex: default/en-us/mypage.off.aspx)

3.4.2 Navigation items and related CSS classes

3.4.2.1 Menu

The useful parameters are as follows:

- ***CSSClass***: CSS class name (ex: myMenu). By default, class = ".siteNote-Menu".
- ***DataSourceID***: the possible values are:
  - ***SiteMapDataSource1***: returns all menu elements from the first page of the site, under the 'default' page (StartingNodeOffset 0 = page 'en-us').
  - ***SiteMapDataSource2***: returns all menu elements from the current page (StartingNodeOffset 0 = page courante).

- ***StartingNodeOffset***: defines the level from which menu items are sent back.
- ***AddParentNode***: defines if the parent element is displayed (relative to the StartingNodeOffset)

Menus are built as lists of "ul" elements. The useful CSS classes are the following:

- ***.SiteNote-Menu***: This class (or the CSSclass defined in the menu properties) is applied on a div element that contains the list.
- ***#Current***: identifier applied on the selected node.
- ***#Parent***: identifier applied to all parents of the selected node.
- ***#Child***: identifier applied to all children of the selected node.
- ***.first***: Class applied to the first element of a each level.
- ***.isParent***: if the 'AddParentNode' property is checked, this class will be applied only to this parent element.
- ***.separator***: class applied on an empty element, inserted between each menu element to allow adding a separator between them (only if the menu adapter is 'MenuAdapterSimpleSeparator').
- Each element contains a class that is prefixed by 'pn_' followed by the page name (ex : pn_en-us) as well as a class prefixed by 'pos' followed by the level of the page in the menu and the page position in this same level (ex:pos11).

  For example, our menu contains the following page structure:

  - ***en-us (displayed thanks to the AddParentNode property)***
    - ***page1***
      - ***subpage1***
      - ***subpage2***

  1. 'en-us' page is selected

     - ***en-us***: li ID="Current" class="isParent first pn_en-us pos01"
     - ***page1***: li ID="Child" class="hasChildren first pn_page1 pos11"
     - ***subpage1***: li ID="Child" class="first pn_subpage1 pos21"
     - ***subpage2***: li ID="Child" class="pn_subpage2 pos22"

  2. 'page1' page is selected

     - ***en-us***: li ID="Parent" class="isParent first pn_en-us pos01"
     - ***page1***: li ID="Current" class="hasChildren first pn_page1 pos11"
     - ***subpage1***: li ID="Child" class="first pn_subpage1 pos21"
     - ***subpage2***: li ID="Child" class="pn_subpage2 pos22"

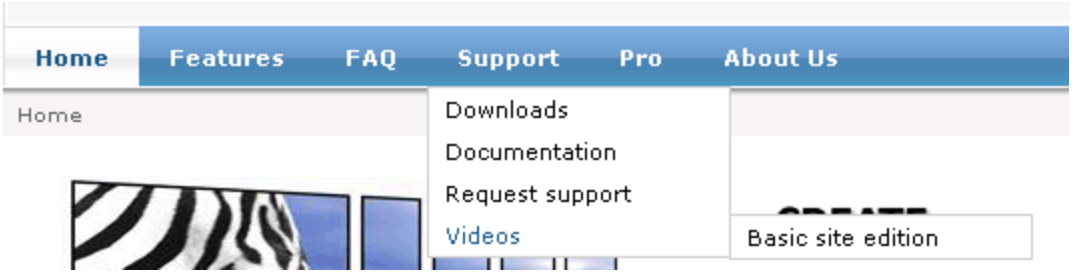
  3. 'subpage1' page is selected

     - ***en-us***: li ID="Parent" class="isParent first pn_en-us pos01"
     - ***page1***: li ID="Parent" class="hasChildren first pn_page1 pos11"
     - ***subpage1***: li ID="Current" class="first pn_subpage1 pos21"
     - ***subpage2***: li class="pn_subpage2 pos22"

  4. 'subpage2' page is selected

     - ***en-us***: li ID="Parent" class="isParent first pn_en-us pos01"
     - ***page1***: li ID="Parent" class="hasChildren first pn_page1 pos11"
     - ***subpage1***: li class="first pn_subpage1 pos21"
     - ***subpage2***: li ID="Current" class="pn_subpage2 pos22"

CSS example of an horizontal menu having several level, and located in the header block



#Global #Header .siteNote-Menu ul li#Current a { color: #19598d; background: #fff url ("Global Header Menu hover bkg.gif") repeat-x center center; } #Global #Header .siteNote-Menu ul li#Parent a { color:

#19598d; background: #fff url("Global_Header_Menu_hover_bkg.gif") repeat-x center center; } #Global #Header .siteNote-Menu ul li#Parent.isParent a { background:Transparent; color:#fff; } #Global #Header .siteNote-Menu { position: absolute; top: 136px; left: 10px; z-index: 100; } /* FIRST LEVEL */ #Global #Header .siteNote-Menu ul { list-style: none; margin: 0; padding: 0; float: left; line-height: 30px; } #Global #Header .siteNote-Menu ul li { position: relative; float: left; width: auto; } #Global #Header .siteNote-Menu ul li a, #Global #Header .siteNote-Menu ul li a:visited { font-weight: bold; color: #fff; height: 30px; padding: 0 15px; display: block; float: left; text-decoration: none; } #Global #Header .siteNote-Menu ul li a:hover { color: #19598d; background: #fff url("Global_Header_Menu_hover_bkg.gif") repeat-x center center; } /* SECOND LEVEL */ #Global #Header .siteNote-Menu ul li ul { line-height: 15px; list-style: none; margin: 0; padding: 0; width: 150px; float: left; position: absolute; top: 30px; left: 0; z-index: 500; border: 1px solid #ccc; display: none; } #Global #Header .siteNote-Menu ul li ul li a, #Global #Header .siteNote-Menu ul li ul li a:visited { font-weight: normal; color: #000 !important; background: #fff !important; display: block; float: none; width: 134px; text-decoration: none; padding: 3px 8px; height: auto; } #Global #Header .siteNote-Menu ul li ul li a:hover { color: #19598d !important; background: #fff; } /* THIRD LEVEL AND NEXT */ #Global #Header .siteNote-Menu ul li ul li ul { position: absolute; top: 0; left: 100%; } #Global #Header div.siteNote-Menu ul ul, #Global #Header div.siteNote-Menu ul li:hover ul ul, #Global #Header div.siteNote-Menu ul ul li:hover ul ul { display: none; } #Global #Header div.siteNote-Menu ul li:hover ul, #Global #Header div.siteNote-Menu ul ul li:hover ul, #Global #Header div.siteNote-Menu ul ul ul li:hover ul { display: block; }

CSS Example of a vertical menu having 1 level and located in the LeftColumn block

#Global #LeftColumn .siteNote-Menu { } #Global #LeftColumn .siteNote-Menu ul { list-style: none; } #Global #LeftColumn .siteNote-Menu ul li { list-style: none; border-right: 1px solid #ccc; border-bottom: 1px solid #ccc; } #Global #LeftColumn .siteNote-Menu ul li#Current { font-weight:bold; } #Global #LeftColumn .siteNote-Menu ul ul { display:none; }

3.4.2.2 Sitemap

The useful parameters are the following:

- ***CSSClass***: name of the CSS class (ex: myClasse). By default, class = ".sitemap".

In this case, the defined CSS class is the default one: "sitemap". If another class is defined in the "CSSClass" property, "SiteMap" will have to be replaced by the name of the class defined in the property (ex: CSSClass=myclasse => myclasse-node). The related CSS classes are the following:

- ***sitemap-rootnode***
- ***sitemap-node***
- ***sitemap-hovernode***
- ***sitemap-selectednode***
- ***sitemap-leafnode***

3.4.2.3 Sitemap path (SiteMapPath)

The useful parameters are the following:

- ***CSSClass***: name of the CSS class (ex: myClass). By default, class is ".SiteMapPath".
- ***PathSeparator***: characters separating elements (">" by default).
- ***StartingNodeOffset***: determines the level from which the elements are displayed (0 by default, level 0 = "default" page).

In this case, the defined CSS class is the default one: "sitemapPath". If another class is defined in the "CSSClass" property, "SiteMapPath" will have to be replaced by the name of the class defined in the property (ex: CSSClass=myclasse => myclasse-node). The related CSS classes are the following:

- ***SiteMapPath***

- ***SiteMapPath-node***
- ***SiteMapPath-rootnode***
- ***SiteMapPath-currentnode***
- ***SiteMapPath-pathseparator***

3.4.3 Search elements and related CSS classes

3.4.3.1 SearchInput

The useful parameters are the following:

- ***CSSClass***: name of the CSS class (ex: myClass). By default, class is ".searchInput".
- ***DisplaySuggest***: if the property is checked, when the user types a word in the search box, some words will be suggested depending on what the user has typed in.
- ***DisplaySuggestPageCount***: if the property is checked, besides each suggested word, the number of pages that contain this word is displayed.
- ***MaxSuggestCount***: maximum number of suggested words.
- ***ImageName***: name of the image that is displayed if the search button's type is Button or ImageButton. The image must be located at the root of the "Design" folder of the site ("SearchButton.gif" by default).
- ***OutputPage***: relative path of the page that contains the results display control (current page by default, ex: default/en-us/results.off.aspx).
- ***SearchButtonType***: type of button used to submit search (Button, ImageButton or LinkButton).
- ***Text***: text of the link used to submit search (if the button's type is "LinkButton").

In this case, CSS class defined by default is "searchInput". If another class is defined in the "CSSClass", "searchInput" has to be replaced by the name of the class that is defined in the property (ex: CSSClass = myClass => myClass_Button). The related CSS classes are the following:

- ***searchInput***
- ***searchInput_Button***
- ***searchInput_Textbox***
- ***searchInput_TypeList***

3.4.3.2 SearchOutput

The useful parameters are the following:

- ***AllowPaging***: Defines if paging is allowed (checked by default)
- ***PageSize***: number of results per page (20 by default)
- ***CSSClass***: name of the CSS class (ex:myClass). By default, class = "searchOutput".
- ***PageLinkCount***: number of links displayed in the paging (10 by default).
- ***PageLinkSeparator***: separator character between paging links.
- ***PageLinkCSSClass***: CSS class applied to the link in the paging. By default, class is ".searchLinkCSSClass".
- ***HighlightCSSClass***: CSS class applied on the "mouse-over" link in the paging. By default, class is ".searchHighlightCSSClass".
- ***CurrentPageLinkCSSClass***: CSS class applied on the current link in the paging. By default, class = ".searchCurrentPageLinkCSSClass".
- ***TextCSSClass***: CSS class applied on the result's text resume. By default, class = ".searchTextCSSClass".
- ***TitleCSSClass***: CSS class applied on the result's title. By default, class = ".searchTitleCSSClass".
- ***UrlCSSClass***: CSS class applied on the page url containing the result. By default, class = ".searchUrlCSSClass".
- ***NoDocumentationFoundFormat***: text's format when there is no result.
- ***HeaderClass***: CSS class applied on the paging in the header. By default, class = ".searchOutputHeader".
- ***HeaderFormat***: header's paging format.
- ***FooterClass***: CSS class applied on paging in the footer.
- ***FooterFormat***: footer's paging format.
- ***FirstPageText***: text of the link to the first results pages in the paging ("First" by default).
- ***PrevPageText***: text of the link to the previous results page in the paging ("Prev" by default).
- ***LastPageText***: text of the link to the last results page in the paging ("Last" by default).
- ***NextPageText***: text of the link to the next results page in the paging ("Next" by default).

3.4.4 Specific elements and related CSS classes

3.4.4.1 Login

The useful parameters are the following:

- ***AlignMode***: align mode of the different elements. Possible values are the following:
  - ***vertical***: default mode. elements are one below the other.
  - ***horizontal***: elements are one aside the other.
  - ***mixed***: login and password elements are one above the other. The button is beside them.

- ***LoginClass***: CSS class applied to the login control (as a table). The default class is ".SiteNoteLogin".

  In this case, the default CSS class is "SiteNoteLogin". If another class is defined in the "CSSClass" property, "SiteNoteLogin" has to be replaced by the name of the class defined in the property (ex: CSSClass = myClass => myClass_txtLogin). Related CSS classes are the following:

  - ***SiteNoteLogin***
  - ***SiteNoteLogin_TdLabel***
  - ***SiteNoteLogin_TdTextBox***
  - ***SiteNoteLogin_TrLogin***
  - ***SiteNoteLogin_TextBoxLogin***
  - ***SiteNoteLogin_TrPassword***
  - ***SiteNoteLogin_TextBoxPassword***
  - ***SiteNoteLogin_TrButton***
  - ***SiteNoteLogin_TdButton***

- ***ImageUrl***: relative path to the image of the login button (SiteNOTE login box image by default)
- ***InAdminMode***: allows to define if, once logged, the user has access to the SiteNOTE administration or if he has access to a secured part of the site (SiteNOTE access checked by default)
- ***LoginText***: text of the label of the login ("Login" by default).
- ***PasswordText***: text of the label of the password ("Password" by default).
- ***LogRedirect***: relative path of the page containing the form that allows to view the logged user profile.
- ***RedirectUrl***: relative path of the page where the user will be redirected to after clicking on the login button.
- ***RedirectUrlIfAuthenticate***: relative path of the page where the user will be redirected to if he opens the page containing the login control while he's already logged-on.
- ***PasswordReminder***: If checked, a link to request a password reminder is displayed with the login control ("forgot password ?" by default).
- ***PasswordReminderEmailSubject***: subject of the email sent to the user who requests a password reminder.
- ***PasswordReminderEmailDescription***: content of the email sent to the user who requests a password reminder.
- ***PasswordReminderEmailSign***: signature of the email sent to the user who requests a password reminder.

3.4.4.2 Switch Mode

The useful parameters are the following:

- ***SwitchModeClass***: CSS class applied to the control. By default, the class is ".SiteNOTE_switchModeControlClass".
- ***SwitchLinkText***: text of the link.

End of document