ENSIM PRO - LINUX

URL configuration guide

ensim®

CORPORATE HEADQUARTERS

ENSIM CORPORATION
3945 Freedom Circle, Suite 1100
Santa Clara, California 95054
(408) 496-3700

www.ensim.com

Contents

Introduction

Using this feature, you can now add a hyperlink with a URL in the Menu option in the left-navigation bar. For example, you can add a new link **My Link**, to the existing list which contains items like **Home**, **Mailboxes/Users** etc.

This feature also allows you to add a link as a sub-item to an existing item. For example, you can add a sub-item **My Sub-Link** to an already existing item **Reports**.

This document provides you instructions for adding an item on the site administrator's navigation bar.

In this section:

Adding a static entry in the XMLfile

This section provides instruction for creating a new item with a URL of the format **http://<siteadmin>/<filename>**, where the **<sitename>** is dynamic value, and the **<filename>** is a static value.

▼ To add a static entry in the XML file:

1 Locate and open file **/usr/lib/ensim/frontend/static/xml/siteadmin.navbar.xml** using an editor.

2 The navigation bar has two sections,

    a The section containing the main items such as Home, Site Information, etc.

    b The other section contains sub-items of the main item such as File Manager, Log Manager, etc.

• If the new item has to be added as a main item section, then add the following code in the **siteadmin.navbar.xml** file:

```
<item>
  <display>item_name</display>
  <service>service_name</service>
```

This node should be added only if this node is related to any service enabled/disabled for the sites. If for a particular site the   service is disabled, this node will not be visible on the navigation bar.

```
<link>/file_name</link>
<icon>image_name</icon>
</item>
```

- If the new item has to be added as a sub item of the main item, then add the following code in the **siteadmin.navbar.xml** file:

```
<item>
  <display>Email</display>
  <icon>/static/images/navbar_icon_email.gif</icon>
  <subitem>
    <display>item_name</display>
  <service>service_name</service>
```

This node should be added only if this node is related to any service enabled/disabled for the sites. If for a particular site the   service is disabled, this node will not be visible on the navigation bar.

```
<link>/file_name</link>
<icon>image_name</icon>
```

This node is required only if you have applied the classic skin.

```
  </subitem>
</item>
Here,
```

item_name - The name of the item that would be displayed in the control panel.

service_name - This option should be added only if this node is related to any service enabled/disabled for the sites. If the service is disabled for a particular site, this option will not be visible on the navigation bar.

file_name - The file you want to access from your site. This is a static value. Ensure that the file_name is preceded by a /. For example `/outstat.html`.

image_name - The location of the image (gif) file. This option is required only if you have applied the **classic** skin.

3. Save and close the file.

Applying appropriate hyperlinks to the URL

▼ **To apply the appropriate hyperlinks to the URL:**

1   Locate and open the file **/usr/lib/ensim/frontend/skins/cocoon/xsl/siteadmin.navbar.xsl** using an editor application.

2   On the top of the XSL, after the `<xsl:output>` tag, add the following:

```
<func:function name="func:getLink">
<xsl:param name="link_text"/>
```

```
<xsl:choose>
 <xsl:when test="string(starts-with($link_text,'/'))='true'">
  <xsl:choose>
   <xsl:when test="string(starts-with($link_text,'/siteadmin'))='true'">
    <func:result select="$link_text"/>
   </xsl:when>
   <xsl:otherwise>
    <xsl:variable name='domain'
select="string(/navbar/site/properties/siteinfo/domain)"/>
    <xsl:variable name='link' select="concat('http://', $domain,
$link_text)"/>
    <func:result select="$link"/>
   </xsl:otherwise>
  </xsl:choose>
 </xsl:when>
 <xsl:otherwise>
  <func:result select="$link_text"/>
 </xsl:otherwise>
</xsl:choose>
</func:function>
```

This function accepts the value of link node from the **siteadmin.navbar.xml** file. It then tests this value (present in variable 'link_text'), to see if it starts with the string **/siteadmin**. If it does, then it understands that the link points to python's byte codes and no manipulations are required. Hence it just returns the same value.

On the other hand, if the value in **link_text** does not start with the string **/siteadmin**, then it takes the value of **sitename** from **/navbar/site/properties/siteinfo/domain**, which is the location of the dynamic xml file, and creates the hyperlink URL by concatenating the strings **http:**, **sitename** and **link_text** to get file URL of the format **http://<sitename>/<filename>**.

- All the above function has been defined in all places before `<a>` tag as described below:

```
<p xmlns="http://www.w3.org/1999/xhtml" class="navbar-item">
    <xsl:variable name="link" select="./link/text()"/>
    <a href="{$link}" target="main">
      <xsl:value-of select="epl:_(string(./display/text()))"/>
    </a>
</p>
```

The code should be modified to look like this:

```
<p xmlns="http://www.w3.org/1999/xhtml" class="navbar-item">
    <xsl:variable name="link_text" select="./link/text()"/>
    <xsl:variable name="link" select="func:getLink($link_text)"/>
    <a href="{$link}" target="main">
      <xsl:value-of select="epl:_(string(./display/text()))"/>
    </a>
</p>
```

- By default the newly added item is displayed in a new window. The modified code for this function is:

```
<p xmlns="http://www.w3.org/1999/xhtml" class="navbar-item">

    <xsl:variable name="link_text" select="./link/text()"/>

    <xsl:variable name="link" select="func:getLink($link_text)"/>

    <xsl:choose>

      <xsl:when test="$link_text=$link">

        <a href="{$link}" target="main">

            <xsl:value-of select="epl:_(string(./display/text()))"/>

        </a>

      </xsl:when>

      <xsl:otherwise>

        <a href="{$link}" target="_blank">

            <xsl:value-of select="epl:_(string(./display/text()))"/>

        </a>

      </xsl:otherwise>

    </xsl:choose>

</p>
```

  This code checks if the value returned by the `getLink()` function is same as the value of link node passed to the function. If this value is different, then the target attribute of `<a>` tag should have value `_blank` instead of `main`.

3. Save and close the file.

Viewing your changes

▼ **To view and verify your changes on the browser:**

1 Open the browser.

2 Clear browser's cache.

3 Log in to the Ensim Pro control panel as the service administrator.

4 Auto-log in to any site.

5 Verify that the changes have been made.

Related documentation

For a list of related documentation, see the release notes. Documentation is also available in the section of the Ensim Support site, http://support.ensim.com. All customers receive passwords that allow access to this site. If you do not have a password, contact your organization's liaison to Ensim.

Support and feedback

For Ensim online support or feedback, use the following links:

- http://support.ensim.com to create an Ensim Support account and access other documentation
- https://onlinesupport.ensim.com to log in to Ensim Support using an existing account
- http://onlinesupport.ensim.com/kb_search.asp?product=lwp to search the knowledge base
- http://www.ensim.com/about/feedback.asp to provide feedback to Ensim

> **Note:** All customers receive passwords that allow access to the Ensim Support site. If you do not have a password, contact your organization's liaison to Ensim.

ENSIM PRO - LINUX

ENSIM CORPORATION
3945 Freedom Circle, Suite 1100
Santa Clara, California 95054
www.ensim.com