

HƯỚNG DẪN THIẾT KẾ GIAO DIỆN ĐỘNG CHO MODULE SỬ DỤNG XML, XSL

Người viết: **Lê Văn Quân**

Email: levanquanna@yahoo.com, nguyendansoft@gmail.com

Mobile phone: 01689 227 100

(Đề nghị đọc kỹ hướng dẫn)

1. Mở đầu

Một số chức năng của website thường có yêu cầu biến đổi giao diện tại từng thời điểm nhất định, tại từng vị trí nhất định trên site. Trên một site có thể sử dụng 1 module tại nhiều vị trí, mỗi vị trí lại có thể yêu cầu cách trình bày giao diện khác nhau.

Xuất phát từ yêu cầu đó, module cần phải cho phép người dùng thay đổi được giao diện này mà không phải tiến hành sửa lại source code. Để thực hiện điều này thì ta phải sử dụng cách giao tiếp thông qua một file định nghĩa giao diện trung gian, có thể hiểu nó như là một cái khung dựng sẵn để lắp ghép dữ liệu vào.

Trong tài liệu này xin được trình bày phương pháp sử dụng chuẩn dữ liệu XML và file XSL.

Bài toán thực tế:

Như phần hướng dẫn tạo module cơ bản đã đề cập đến việc hiển thị danh mục, trong module đó thì ta đang sử dụng một GridView để hiển thị danh sách danh mục, trong phần hướng dẫn này sẽ trình bày cách hiển thị thay danh sách danh mục đó bằng XML và XSL.

2. Bắt đầu

a. Định nghĩa các phương thức cần thiết

Ta cần định nghĩa các phương thức cần thiết trong class **ListController**

- Khai báo sử dụng thêm một số thư viện sau:

```
using System.Xml;  
using System.Xml.Xsl;  
using System.IO;  
using System.Net;
```

- Định nghĩa các phương thức sau:

```
#region XML, XSL
```

```
//Phương thức đọc và phân tích nội dung file XSL
```

```

private XslCompiledTransform GetXSLContent(string ContentURL)
{
    XslCompiledTransform functionReturnValue = null;
    functionReturnValue = new XslCompiledTransform();
    WebRequest req = Globals.GetExternalRequest(ContentURL);
    WebResponse result = req.GetResponse();
    Stream ReceiveStream = result.GetResponseStream();
    XmlReader objXSLTransform = new XmlTextReader(result.GetResponseStream());
    functionReturnValue.Load(objXSLTransform, null, null);
    return functionReturnValue;
}

//Phương thức lấy đường dẫn đúng đắn
private string GetMappedPath(string localPath, int PortalId)
{
    string ret = DotNetNuke.Common.Globals.ApplicationMapPath + "/Portals/" +
PortalId.ToString() + "/" + localPath;
    ret = ret.Replace("//", "/");
    return ret;
}

//Phương thức chuyển đổi dữ liệu sang định dạng XML
private XmlDocument GetItemsXml()
{
    List<ListInfo> lstItems = GetItems();
    XmlDocument xmldoc = new XmlDocument();
    xmldoc = CBO.Serialize(lstItems);
    return xmldoc;
}

//Phương thức kết xuất dữ liệu thành định dạng xâu để hiển thị lên site theo
định dạng HTML
//XslPath: đường dẫn file XSL
public string GenerateDataToHTML(string XslPath, int PortalId)
{
    XmlDocument xmlDoc = GetItemsXml();
    XslCompiledTransform xslTran = new XslCompiledTransform();

    //Nếu đường dẫn file xsl rỗng thì lấy file xsl mặc định có sẵn của module
    if(XslPath=="")
    {
        XslPath = "/DesktopModules/BeginModule.List/XSL/List.xsl";
    }
    if (Globals.GetURLType(XslPath) != DotNetNuke.Entities.Tabs.TabType.Url)
    {
        xslTran.Load(GetMappedPath(XslPath, PortalId));
    }
    else
    {
        if (XslPath.IndexOf("DesktopModules/BeginModule.List/") >= 0)
            xslTran.Load(Globals.ApplicationMapPath + XslPath);
        else
            xslTran = GetXSLContent(XslPath);
    }

    StringWriter strw = new StringWriter();
    xslTran.Transform(xmlDoc, null, strw);
    string ret = strw.ToString();
    while (ret.IndexOf(" ") >= 0) ret = ret.Replace(" ", " ");
}

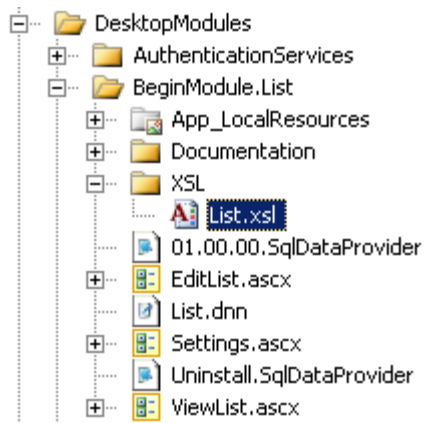
```

```

        return ret;
    }
#endregion

```

b. Thêm file XSL



Nội dung file XSL:

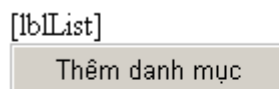
```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html"/>
<xsl:template match="/">
    <style>
        .Box{border:solid 1px blue;width:200px;}
        .List_Item{padding:10px;font-family:tahoma;font-size:12px;font-
weight:bold;color:red;}
    </style>
    <div class="Box">
        <xsl:apply-templates select="ArrayOfListInfo/ListInfo"/>
    </div>
</xsl:template>
<xsl:template match="ArrayOfListInfo/ListInfo">
<div class="List_Item">
    <xsl:value-of select="Name"/>
</div>
</xsl:template>
</xsl:stylesheet>

```

a. Thiết kế lại điều khiển hiển thị danh mục

- Giao diện điều khiển hiển thị danh mục như sau:



```

<asp:Label ID="lblList" runat="server" Text=""></asp:Label>
<br />
<asp:Button ID="btnAddNewList" runat="server" Text="Thêm danh mục"
    onclick="btnAddNewList_Click" />

```

- Xử lý hiển thị danh mục:

```

protected void Page_Load(System.Object sender, System.EventArgs e)
{
    try
    {
        if (!Page.IsPostBack)
        {
            string ListXslPath = "";
            if (Settings["ListXslPath"] != null)
                ListXslPath = Settings["ListXslPath"].ToString();
            lblList.Text = lstCtr.GenerateDataToHTML(ListXslPath, this.Portal
        }
    }
    catch (Exception exc) //Module failed to load
    {
        Exceptions.ProcessModuleLoadException(this, exc);
    }
}

```

Đề ý tại vùng khoanh tròn chính là cách lấy đường dẫn file XSL do người dùng thiết lập. Để cung cấp công cụ cho thiết đường dẫn này cho người dùng sử dụng ta cần định nghĩa một điều khiển sẽ trình bày phần dưới.

b. Định nghĩa điều khiển thiết lập đường dẫn XSL

Trong thư mục **DesktopModules/BeginModule.List** đã có sẵn một điều khiển **Settings.ascx**, nếu không tồn tại file này ta có thể tự tay thêm vào và đặt tên tùy ý.

Lưu ý: Tại đây không chỉ thiết lập đường dẫn cho file XSL mà còn có thể thiết lập bất kỳ cái gì mà muốn cho phép người dùng tự tay làm.

Thiết kế như sau:

THIẾT LẬP CHO MODULE

☒ Sử dụng file XSL mặc định

<input type="text"/>	<input type="text"/>
<input type="text"/>	
<input type="text"/>	
<input type="text"/>	<input type="text"/> <input data-bbox="1334 468 1498 480" type="button" value="Browse..."/>
<input type="text"/>	

☐ Track? ☐ Log?

```
<% Register Src="../../controls/URLControl.ascx" TagName="url" TagPrefix="dnn" %>
|<p>

<div><b>THIẾT LẬP CHO MODULE</b></div>

<br />

<asp:CheckBox ID="chkDefault" ResourceKey="chkDefault" runat="server" Checked="True"
    OnCheckedChanged="chkDefault_CheckedChanged" AutoPostBack="True" Text="Sử dụng file XSL mặc định" />

<br />

<dnn:url id="ctlXslPath" runat="server" FileFilter="xsl" required="False" showfiles="True"
    showlog="False" shownewwindow="False" showtabs="False" showtrack="False" showurls="false"
    Visible="False">

</dnn:url>

.</p>
```

- Xử lý code như sau:

```
partial class Settings : ModuleSettingsBase
{
    #region Base Method Implementations

    public override void LoadSettings()
    {
        try
        {
            if (Page.IsPostBack == false)
            {
                //File XSL
                bool xslDefault = true;
                if (Settings["xslDefault"] != null)
                    bool.TryParse(Settings["xslDefault"].ToString(), out
xslDefault);

                chkDefault.Checked = xslDefault;
                ctlXslPath.Visible = !xslDefault;
            }
        }
    }
}
```

```

        if (!xslDefault)
            if (Settings["ListXslPath"] != null)
                ctlXslPath.Url = Settings["ListXslPath"].ToString();
    }
}
catch (Exception exc) //Module failed to load
{
    Exceptions.ProcessModuleLoadException(this, exc);
}
}

public override void UpdateSettings()
{
    try
    {
        ModuleController objModules = new ModuleController();
        //File XSL
        objModules.UpdateModuleSetting(this.ModuleId, "xslDefault",
chkDefault.Checked.ToString());
        if (chkDefault.Checked)
            objModules.UpdateModuleSetting(this.ModuleId, "ListXslPath",
"/DesktopModules/BeginModule.List/XSL/List.xsl");
        else
            objModules.UpdateModuleSetting(this.ModuleId, "ListXslPath",
ctlXslPath.Url);

        //refresh cache
        SynchronizeModule();
    }
    catch (Exception exc) //Module failed to load
    {
        Exceptions.ProcessModuleLoadException(this, exc);
    }
}

#endregion

protected void chkDefault_CheckedChanged(object sender, EventArgs e)
{
    ctlXslPath.Visible = !chkDefault.Checked;
}
}

```

c. Định nghĩa điều khiển với hệ thống

- Vào **Host/Module Definitions**
- Chọn module **BeginModule.List**
- Tại Definition **BeginModule.List** thêm điều khiển như sau:

▼ **Edit Module Control**

Module: BeginModule.List

Definition: BeginModule.List

Key: Settings

Title:

Source: DesktopModules/BeginModule.List/Settings.ascx

Type: Admin

View Order:

Icon: <Not Specified>

Help URL:

Supports Partial Rendering? ☐

Update Cancel Delete

- Nhấn **Update** để cập nhật
- Đến đây coi như hoàn thành
- Giao diện mới như sau:

▼ **BeginModule.List**

Máy tính xách tay

Máy tính để bàn

Điện thoại di động

Thêm danh mục

d. Cách thay đổi giao diện thông qua file XSL

Bây giờ để thay đổi giao diện cho danh sách danh mục ta làm như sau:

- Viết một file XSL mới:

Ví dụ nội dung của một file XSL như sau:

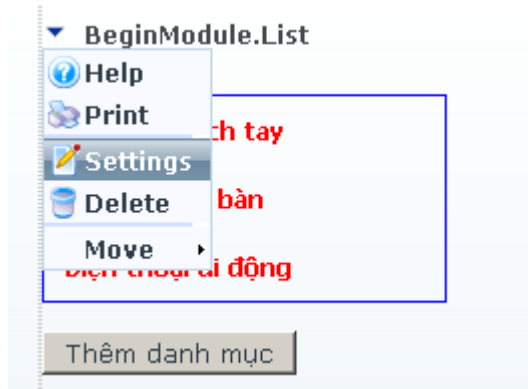
```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html"/>
  <xsl:template match="/">
    <style>
      .Box{border:solid 1px red;width:400px;font-family:tahoma;font-size:12px;font-
weight:bold;padding:10px;}
      .List_Item{padding:10px;font-family:tahoma;font-size:12px;font-
weight:bold;color:blue;}
    </style>
    <div class="Box">
      DANH SÁCH CÁC DANH MỤC HIỆN CÓ
      <div>
        <xsl:apply-templates select="ArrayOfListInfo/ListInfo"/>
      </div>
    </div>
  </xsl:template>
  <xsl:template match="ArrayOfListInfo/ListInfo">
    <div class="List_Item">
      - <xsl:value-of select="Name"/>
    </div>
  </xsl:template>
</xsl:stylesheet>

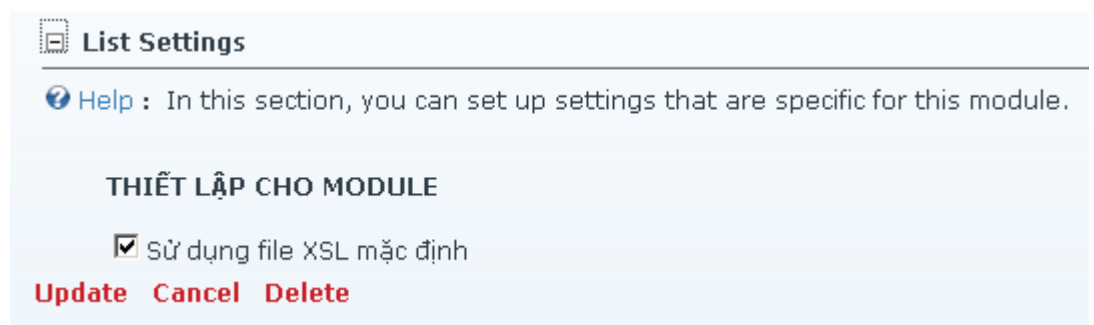
```

b. Thiết lập

- Vào thiết lập cho module:



- Ta nhìn thấy tại phía cuối cùng:



- Để chọn file XSL:

THIẾT LẬP CHO MODULE

☐ Sử dụng file XSL mặc định

File Location:
Root

File Name:
<None Specified>

Upload New File

- Nhấn vào nút **Upload New File**

THIẾT LẬP CHO MODULE

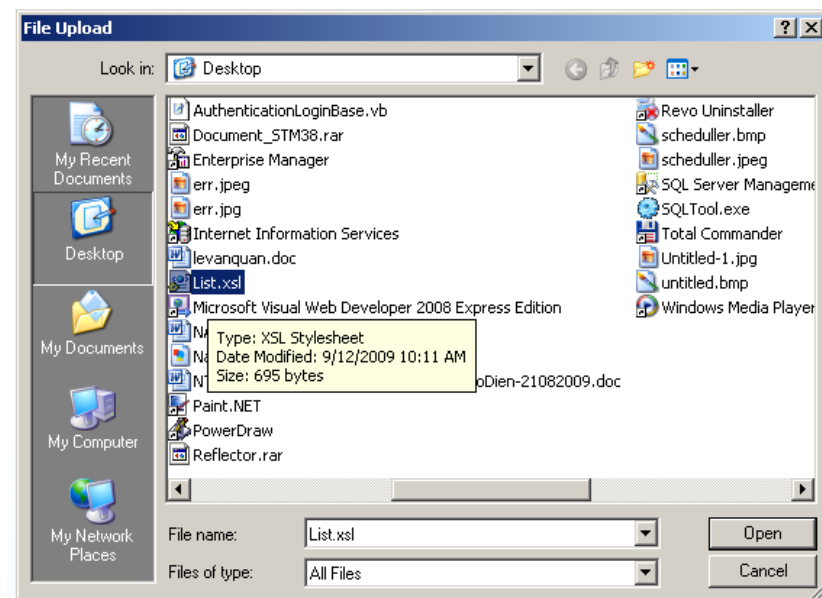
☐ Sử dụng file XSL mặc định

File Location:
Root

File Name:
 Browse...

Upload Selected File
Select An Existing File

- Chọn file XSL tùy ý để upload lên host(Chính là file ta định nghĩa ở trên):



☐ Sử dụng file XSL mặc định

File Location:

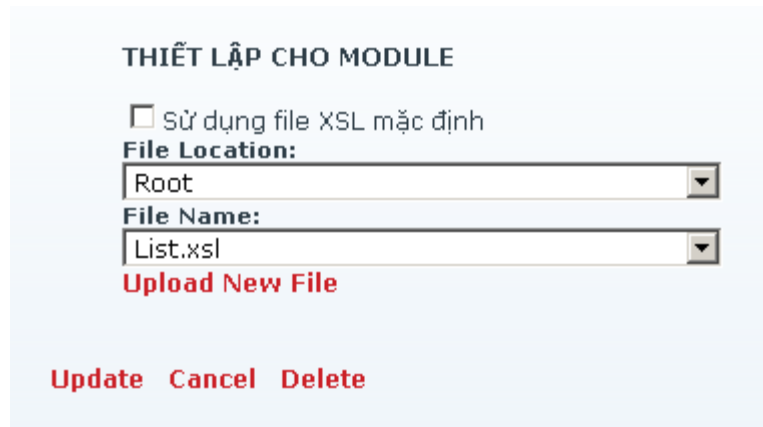
Root

File Name:

Upload Selected File

Select An Existing File

Sau khi nhấn **Upload Selected File** để tải lên ta có:



THIẾT LẬP CHO MODULE

☐ Sử dụng file XSL mặc định

File Location:
Root

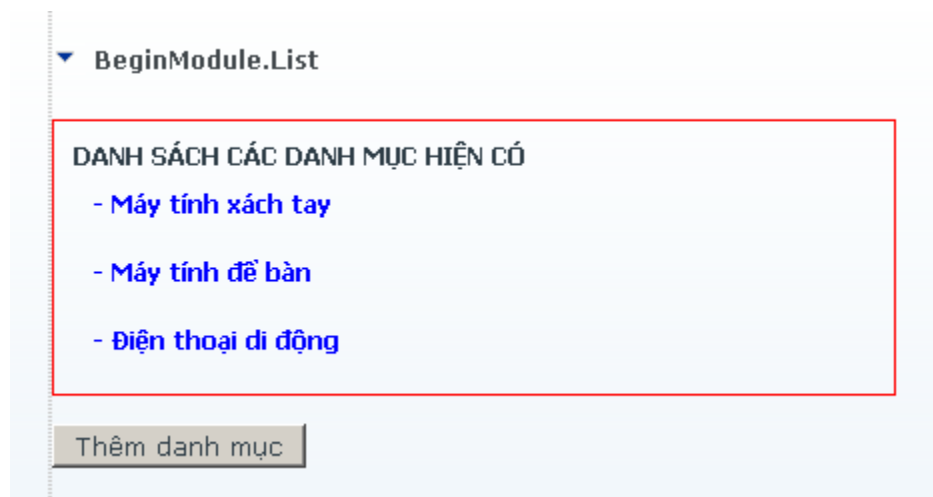
File Name:
List.xsl

Upload New File

Update Cancel Delete

- Nhấn **Update** để lưu lại

Kết quả như sau:



▼ **BeginModule.List**

DANH SÁCH CÁC DANH MỤC HIỆN CÓ

- Máy tính xách tay
- Máy tính để bàn
- Điện thoại di động

Thêm danh mục