**SITECORE SERVICES**

**Sitecore YAF Integration**

Version Document: 1.0  
Date: 01 March 2015  
Author: Damian Brooks

Content

[1. Preface 2](#_Toc413056607)

[1 Changes to Source Code 3](#_Toc413056608)

[1.1 YafUserProfile 3](#_Toc413056609)

[1.2 SitecoreMembershipUser and SitecoreDomainManager 3](#_Toc413056610)

[1.3 Sitecore Membership Helpers 3](#_Toc413056611)

[1.1 Security Settings 3](#_Toc413056612)

[2 How to build the project 4](#_Toc413056613)

[3 Setup YAF in Sitecore 5](#_Toc413056614)

[3.1 Prerequisites 5](#_Toc413056615)

[3.2 Files 5](#_Toc413056616)

[3.3 Web.config merges 6](#_Toc413056617)

[3.4 \App\_Config\ConnectionStrings.config 8](#_Toc413056618)

[4 Installing YAF database 9](#_Toc413056619)

[5 Applying YAF to a Sitecore layout 12](#_Toc413056620)

[6 Setting up YAF 14](#_Toc413056621)

[7 Notes 15](#_Toc413056622)

[7.1 The YAF Security provider is disabled 15](#_Toc413056623)

[7.2 YAF uses the Sitecore users from the current domain 15](#_Toc413056624)

[7.3 YAF needs the Anonymous user 15](#_Toc413056625)

[7.4 The YAF Guest user no longer exist 15](#_Toc413056626)

[7.5 The YAF Guests membership role cannot be used 15](#_Toc413056627)

[7.6 Provide your own login page 15](#_Toc413056628)

[7.7 Users can be managed from YAF … 15](#_Toc413056629)

[7.8 Roles can be managed from YAF 16](#_Toc413056630)

The information contained in this document represents the current view of Sitecore Corporation on the issues discussed as of the date of publication and is subject to change at any time without notice. This document and its contents are provided AS IS without warranty of any kind, and should not be interpreted as an offer or commitment on the part of Sitecore, and Sitecore cannot guarantee the accuracy of any information presented. SITECORE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

The descriptions of other companies’ products in this document, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Sitecore. Sitecore cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers.

All trademarks are the property of their respective companies

©2013 Sitecore Corporation. All rights reserved.

# Preface

YetAnotherForum (YAF) is an open source discussion forum written in .NET. YAF cannot run in the same application as Sitecore, unless the source code of YAF is changed.

Read more about YAF at:

<http://www.yetanotherforum.net/>

The document describes which changes have been made to the YAF source code, how the source code can be built, and how to set up Sitecore so YAF can be run from a Sitecore layout.

The package and all documentation is built around YAF version 2.2.0.

# Changes to Source Code

## YafUserProfile

The most important change is the change to the YafUserProfile, within the ***YAF.Utils*** assembly. If YAF must use the Sitecore security providers, YAF must inherit from the Sitecore.Security.UserProfile.

This change alone will allow YAF to use the Sitecore security providers, and the YAF providers are no longer used.

## SitecoreMembershipUser and SitecoreDomainManager

These classes are new. SitecoreMemberShipUser is a wrapper class for the ordinary MembershipUser which allows YAF to maintain Sitecore users without the domain prefix.

User names returned from Sitecore has the format <domain>\<user>, but YAF does not need the domain name. This is removed by SitecoreMemberShipUser.

SitecoreDomainManager is a helper class for managing the Sitecore domain. The class allows code to remove and add domain names to the user name, and to determine if the user is from an allowed domain. This is necessary to avoid having users from a different domain logging into YAF. If YAF runs in the extranet domain, only extranet users are allowed to log in.

## Sitecore Membership Helpers

Other changes in the source code are done to reference the SitecoreMembershipUser instead of the default MembershipUser. Here is an example:

// MembershipUser user = Membership.GetUser();  
SitecoreMembershipUser user = new SitecoreMembershipUser(Membership.GetUser());

Instead of instantiating a user class based on the MembershipUser, the SitecoreMembershipUser is instantiated, wrapping the user.

## Security Settings

By default YAF comes as a strongly signed assembly, which requires use of other strongly signed assemblies, which unfortunately Sitecore doesn’t provide. To get around this, each of the YAF projects need to be changed to remove signing. The code also needs to include some [SecurityCriticalAttribute] attributes on various classes and methods to ensure that they will work with the Sitecore assembly without breaking inheritance rules.

# How to build the project

Open the source code in Visual Studio 2013 using the yetanotherforum.net.sln.

The solution will open.

Projects YAF.Utils, YAF.Types YAF.Core, YAF.Controls, and YAF references the Sitecore.Kernel.DLL. You must make a reference yourself to a DLL, preferably of version 7.0.0.0 (which ships with Sitecore version 8). I used Sitecore 8.0 rev. 141212.

You can now build the solution.

# Setup YAF in Sitecore

## Prerequisites

You need to have a running installation of Sitecore 8.

## Files

Files are copied from the YAF source code (the source code) to the Sitecore installation (your website):

* Copy the **\YetAnotherForum.NET\App\_Code** to the **root** of your website.
* Copy the following files from the **\YetAnotherForum.NET\bin** folder to the **\bin\** folder of your website:
  + Intelligencia.UrlRewriter.dll
  + ServiceStack.Common.dll
  + ServiceStack.Interfaces.dll
  + ServiceStack.OrmLite.dll
  + ServiceStack.OrmLite.SqlServer.dll
  + ServiceStack.Text.dll
  + YAF.Data.MsSql.dll
  + YAF. dll
  + YAF.Classes.Config.dll
  + YAF.Classes.Data.dll
  + YAF.Controls.dll
  + YAF.Core.dll
  + YAF.Providers.dll
  + YAF.Types.dll
  + YAF.Utils.dll
  + *Also copy the \*.pdb files if you have compiled in debug mode.*
* YAF also relies on third party assemblies that need to be copied to your website bin folder;
  + Autofac.dll
  + OmuValueInjecter.dll
  + CookComputing.XmlRpcV2.dll
  + nStuff.UpdateControls.dll
  + FarsiLibrary.dll
* Copy **mail.config** and **UrlRewriter.config** from **\YetAnotherForum.NET\** to the **root** of your website.
* Copy the following folders and files from **\YetAnotherForum.NET\** to a folder in your website called **/yaf/**:
  + /Content
  + /controls
  + /Images
  + /install
  + /Languages
  + /pages
  + /resources
  + /Scripts
  + /themes
  + /uploads
  + advanced.ashx
  + default.aspx
  + error.aspx
  + forumlogo.jpg
  + popup.aspx
  + YafAjax.asmx
  + YajUploader.ashx
  + YafWebService.asmx

## Web.config merges

I have provided a sample Sitecore web.config at:

\YetAnotherForum.NET\webconfigs\sitecore.web.config

Use the web.config as a template and search for the word “YAF”. All YAF specific settings are marked with this word.

If you wish to apply the changes manually you may follow these directions:

Copy the following to the **/configuration/configSections**:

<section name="rewriter" type="Intelligencia.UrlRewriter.Configuration.RewriterConfigurationSectionHandler, Intelligencia.UrlRewriter" requirePermission="false"/>

Copy the following to after the end of the **/configuration/configSections**:

<rewriter configSource="URLRewriter.config" />

Copy the following to the **/configuration/appSettings** section:

<add key="YAF.AppRoot" value="/yaf/"/>

<add key="YAF.FileRoot" value="/yaf/"/>

<add key="yaf.forcescriptname" value=""/>

<add key="YAF.AllowLoginAndLogoff" value="false" />

<add key="YAF.BoardID" value="1" />

<add key="YAF.EnableURLRewriting" value="false" />

<add key="YAF.DatabaseObjectQualifier" value="yaf\_" />

<add key="YAF.DatabaseOwner" value="dbo" />

<add key="YAF.ProviderExceptionXML" value="ProviderExceptions.xml" />

<add key="YAF.UploadDir" value="/yaf/upload/" />

<add key="YAF.ProviderKeyType" value="System.Guid" />

<add key="YAF.BaseUrl" value="/yaf/" />

<add key="YAF.BaseUrlOverrideDomain" value="false" />

<add key="YAF.RadEditorSkin" value="Black" />

<add key="YAF.UseRadEditorToolsFile" value="true" />

<add key="YAF.RadEditorToolsFile" value="editors/radEditor/toolsFile.xml" />

<add key="YAF.ConfigPassword" value="yaf" />

Copy the following to the **/configuration/Sitecore/sites** section:

<site name="yaf" virtualFolder="/yaf" physicalFolder="/yaf" rootPath="/sitecore/content" startItem="/home" language="en" database="web" domain="extranet" allowDebug="true" cacheHtml="true" />

Insert the following just before the **/configuration/system.web** section starts:

<system.net>

<mailSettings>

<smtp configSource="mail.config"/>

</mailSettings>

</system.net>

Copy the following to the **/configuration/system.web/pages/controls** section:

<add tagPrefix="ajaxToolkit" namespace="AjaxControlToolkit" assembly="AjaxControlToolkit"/>

<add tagPrefix="YAF" namespace="YAF.Utils" assembly="YAF.Utils" />

<add tagPrefix="YAF" namespace="YAF.Classes.Data" assembly="YAF.Classes.Data" />

<add tagPrefix="YAF" namespace="YAF.Controls" assembly="YAF.Controls" />

<add tagPrefix="YAF" namespace="YAF.Controls.Statistics" assembly="YAF.Controls" />

<add tagPrefix="YAF" namespace="YAF.Classes" />

<add tagPrefix="YAF" namespace="YAF" assembly="YAF” />

<add tagPrefix="editor" namespace="YAF.Editor"/>

Insert the following just after the end of the **/configuration/system.web/pages/controls** section:

<namespaces>

<add namespace="YAF" />

<add namespace="YAF.Utils" />

<add namespace="YAF.Core" />

<add namespace="YAF.Controls" />

<add namespace="YAF.Classes.Data" />

</namespaces>

Change the **/configuration/system.web/profile** section. Remove the old section and insert the following:

<profile defaultProvider="sql" enabled="true" inherits="**YAF.Utils.YafUserProfile**">

<providers>

<clear />

<add name="sql" type="System.Web.Profile.SqlProfileProvider" connectionStringName="core" applicationName="sitecore" />

<add name="switcher" type="Sitecore.Security.SwitchingProfileProvider, Sitecore.Kernel" applicationName="sitecore" mappings="switchingProviders/profile" />

</providers>

<properties>

<clear />

<add type="System.String" name="SC\_UserData" />

</properties>

</profile>

Add the following to the **/configuration/system.web/compilation** section:

<add assembly="System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=B77A5C561934E089"/>

<add assembly="System.Design, Version=4.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A"/>

Edit the System.WebServer node to add thefollowing;

<system.webServer>

<modules>

<remove name="ScriptModule"/>

<remove name="YafTaskModule" />

<remove name="UrlRewriter" />

<add name="ScriptModule" preCondition="managedHandler" type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add name="YafTaskModule" type="YAF.Core.YafTaskModule, YAF.Core"/>

<add name="UrlRewriter" type="Intelligencia.UrlRewriter.RewriterHttpModule,Intelligencia.UrlRewriter" preCondition="managedHandler"/>

</modules>

<handlers>

<remove name="WebServiceHandlerFactory-Integrated"/>

<remove name="ScriptHandlerFactory"/>

<remove name="ScriptHandlerFactoryAppServices"/>

<remove name="ScriptResource"/>

<add name="ScriptHandlerFactory" verb="\*" path="\*.asmx" preCondition="integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add name="ScriptHandlerFactoryAppServices" verb="\*" path="\*\_AppService.axd" preCondition="integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

<add name="ScriptResource" preCondition="integratedMode" verb="GET,HEAD" path="ScriptResource.axd" type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

</handlers>

</system.webServer>

## \App\_Config\ConnectionStrings.config

Create a new empty database in order to hold the YAF data. Now add the YAF connection string (pointing to the database you just created) to your own **connectionstrings config** file:

<connectionStrings>

<!--

Sitecore connection strings.

All database connections for Sitecore are configured here.

-->

……

<add name="yafnet" connectionString="data source=LOCALHOST\SQLEXPRESS;initial catalog=YAFNET;user id=xxx;password=xxx" />

</connectionStrings>

# Installing YAF database

Ensure you have empty SQL database as described in section 3.4.

Then run http://<yourdomain>/yaf/ (you should be directed to the install screen). This will run a wizard to setup a new database for your forum.

First step requiring a decision is to add a connection string. Ensure you use the connection string you defined in section 3.4.

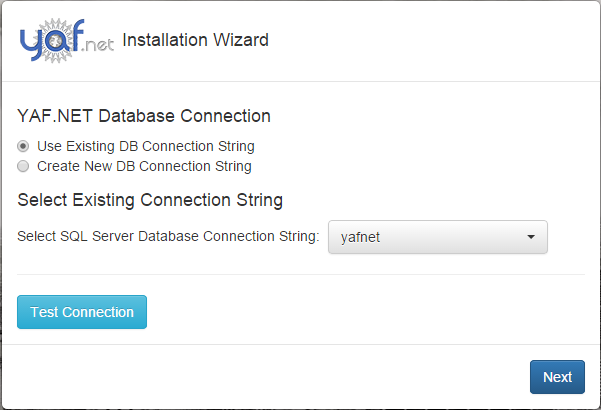


Figure 1 Install Step 1

The wizard should then take you through creating a new database. Once that is done, you need to create a new board for the forum.

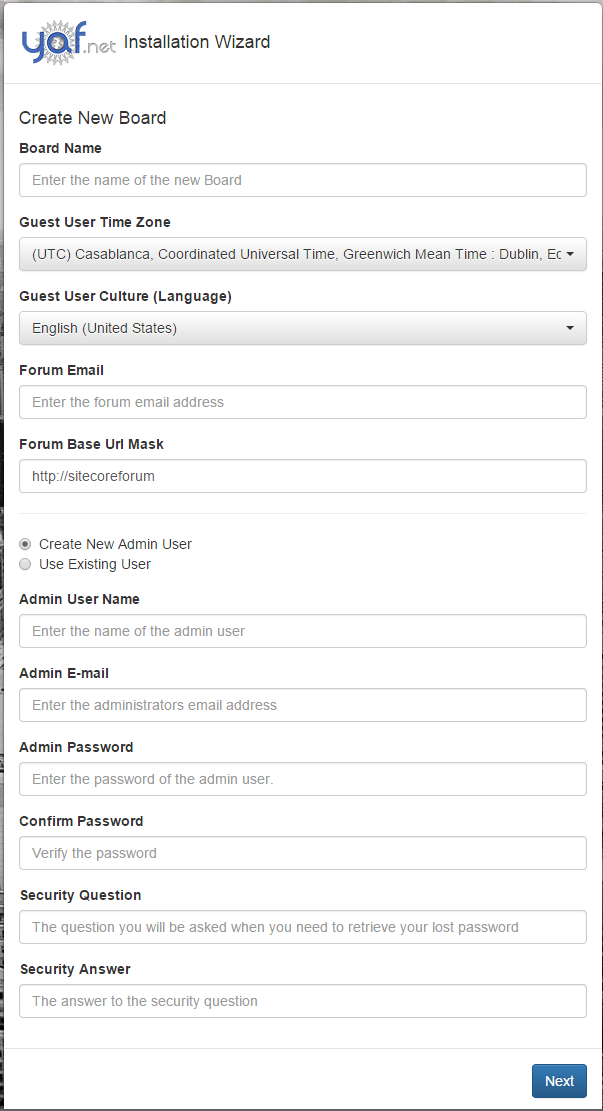


Figure 2 Install Step 2

If you choose to create a new Admin user, please note that the user is visible in Sitecore, but without a domain name. This is not a problem, as long as you supply a login mechanism for this user. This user cannot login using Sitecore extranet.

You can also use an existing user from Sitecore. Please make sure that the user is created in the “extranet” domain, and that the username is unique across all domains (do not use “admin” if you already have a admin user in the Sitecore domain). The user also needs to be a member of the

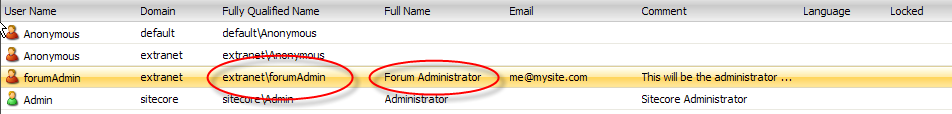


Figure 3 Forum administrator

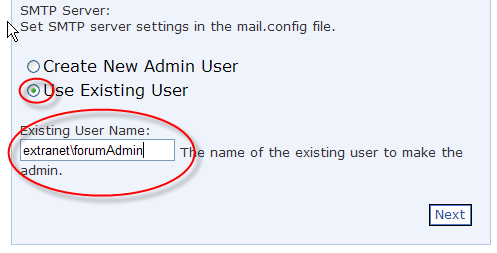


Figure 4 supplying an existing user as admininstrator

Now you can finish the installation.

If you have not added YAF to a Sitecore layout yet, you may use the following URL to access YAF:

http://<yourdomain>/yaf

You website may not have its own login screen yet. In that case you can access the YAF login screen at:

http://<yourdomain>/yaf/?g=login

Log in using domain and username:

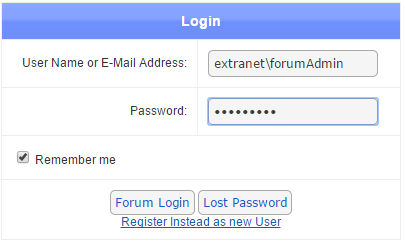


Figure Login using domain and username

Please note that the first time you log in as your exiting user, the domain name will be shown in the username. This is removed the second time you log in.

# Applying YAF to a Sitecore layout

You can apply YAF to any Sitecore layout, and put YAF on any page you wish.

In this example I have added YAF to the Sample Layout.aspx page:

**<%@ Register TagPrefix="YAF" Assembly="YAF" Namespace="YAF" %>**

**<head id="YafHead" runat="server">**

<title>Welcome to Sitecore</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<meta name="CODE\_LANGUAGE" content="C#" />

<meta name="vs\_defaultClientScript" content="JavaScript" />

<meta name="vs\_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5" />

<link href="/default.css" rel="stylesheet" />

</head>

<body>

<form id="mainform" method="post" runat="server">

<div id="MainPanel">

<sc:placeholder key="main" runat="server" />

<div id="CenterColumn">

<div id="content">

**<YAF:Forum runat="server" ID="forum"></YAF:Forum>**

</div>

</div>

</div>

</form>

</body>

</html>

Notice that the head tags is running server side. This is because YAF uses themes to apply style sheet.

Also note that YAF will change the title tag when running.

YAF will now run on pages using the Sample Layout.aspx page as layout:

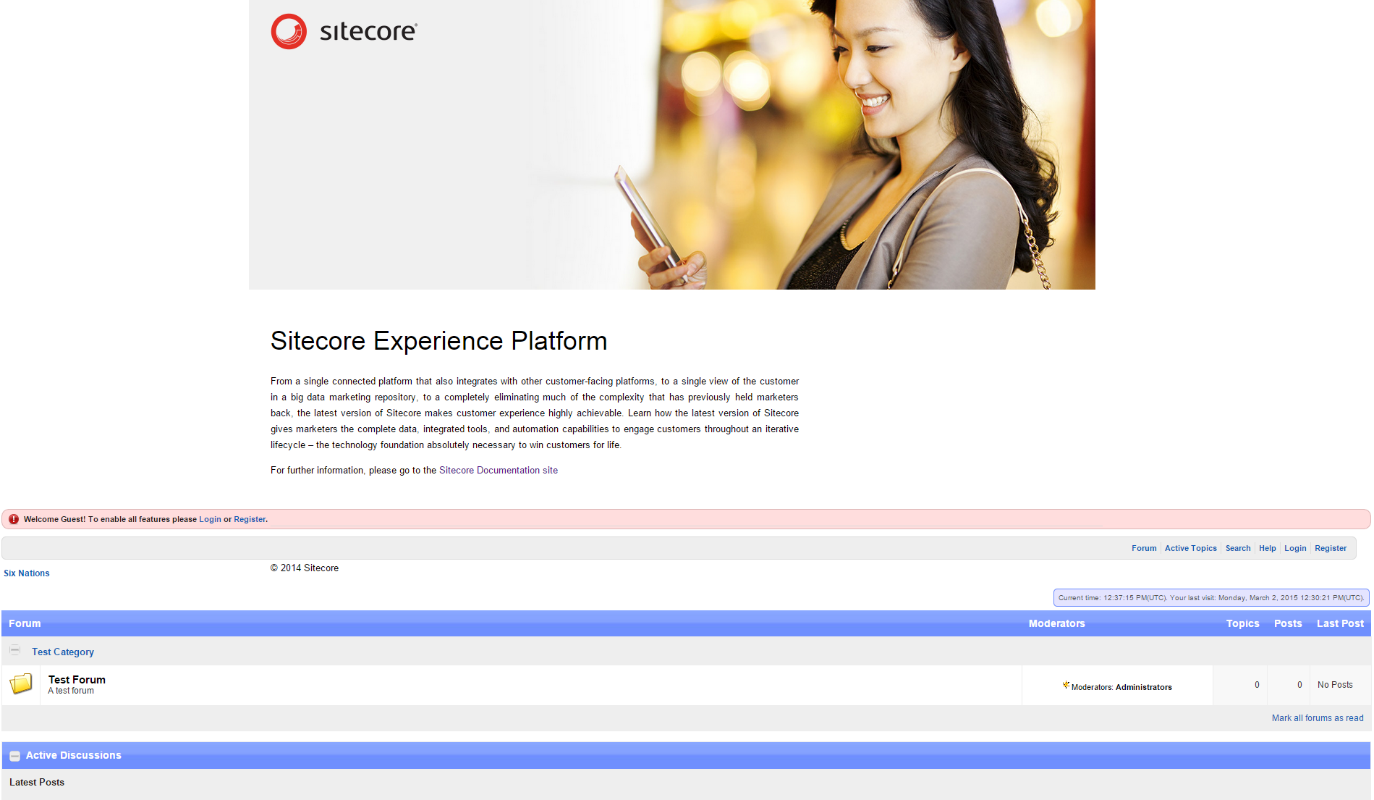


Figure YAF running in Sitecore

# Using Sitecore YAF Package

The solution includes a YAF.Sitecore project, which creates Sitecore items and integrations for use with YAF.

The project provides two parts; A Sublayout control to render a forum, and a customized field type for use with a Sitecore template, in order to allow selection of a specific forum to show in the control.

To enable the package, install the package file ‘Sitecore YAF Controls.zip’ contained in YAF.Sitecore->Package.

The optional package contains the following features:

* A control that allows you to control that YAF must display: The complete forum, a single group or a single forum.   
  See chapter 4, Controlling what YAF must display
* The YAF administrator interface embedded as a shell program.

The package contains:

* A Dropdown field type called "YAFDropList" that displays all groups/categories/forums from YAF.
* A template called "YAF/Forum" containing the field type.
* A sublayout called "YAF/YAF\_Forum" containing a YAF control that displays the group, category or forum that was selected using the YAFDropList on the YAF/Forum template.
* A layout called YAF/Admin that is used by the shell program to display the YAF forum admin inside Sitecore.

## Controlling what YAF must display

The optional Sitecore package allows you to control what YAF should display: The complete forum, a single group or a single forum. You may add YAF to any page in Sitecore you wish, and with the forum control you can decide that one page should display one forum, another page should display another forum. When a board, group or forum is chosen, that page will only display the selected:

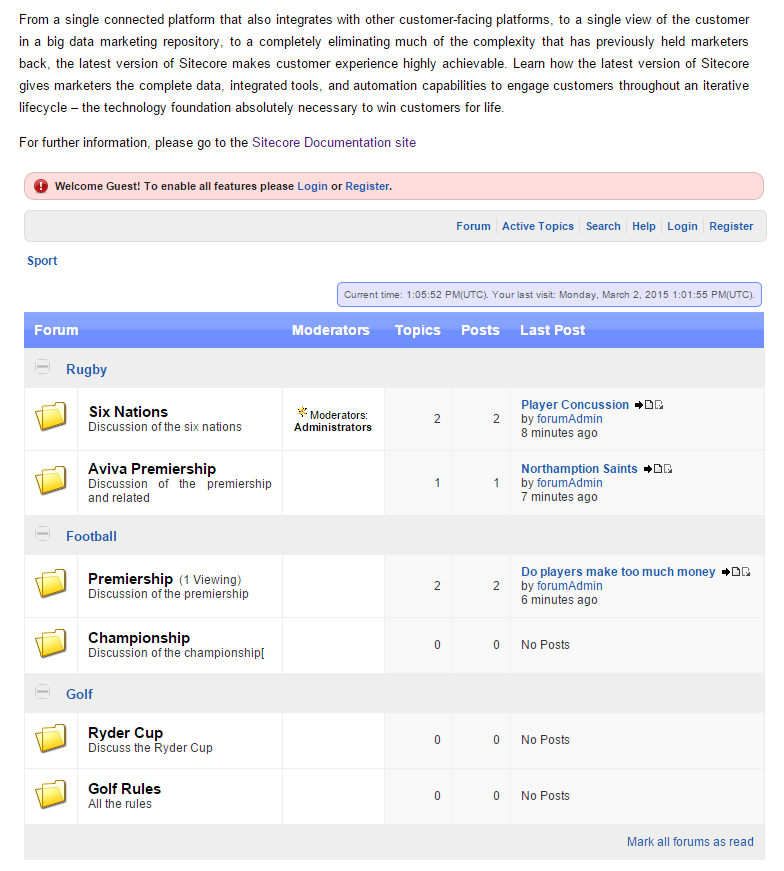


Figure YAF Displaying the complete forum

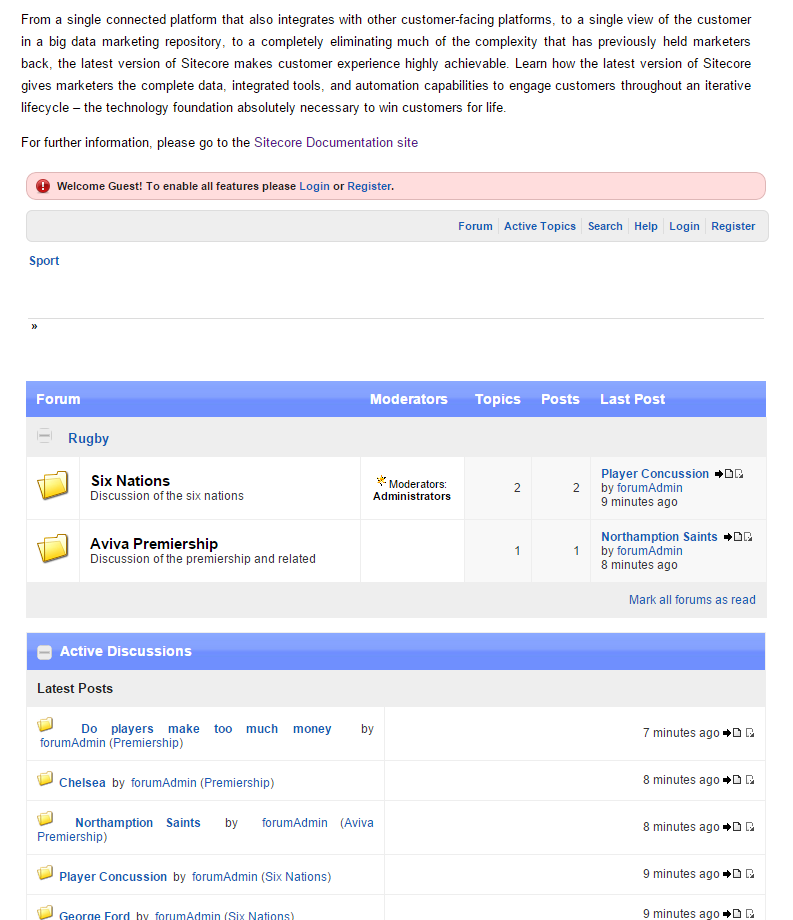


Figure YAF displaying one group (the "Rugby" group)

## Setting up the Package

First you must add the ”YAF/YAF\_Forum” sublayout to the pages that should display the forum:

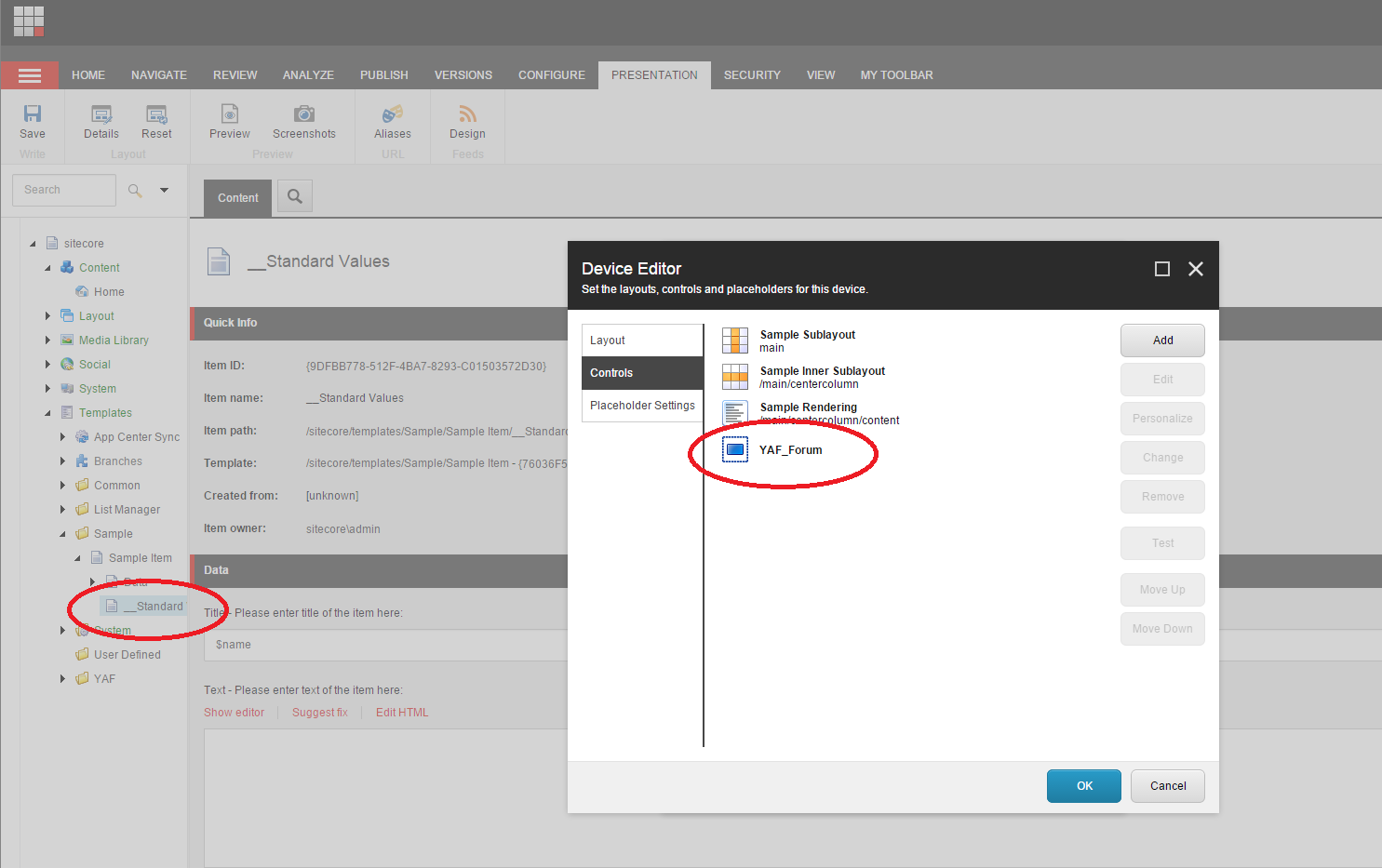


Figure Adding the sublayout to the page that displays the forum

Then you should let the pages inherit from the YAF/Forum template:

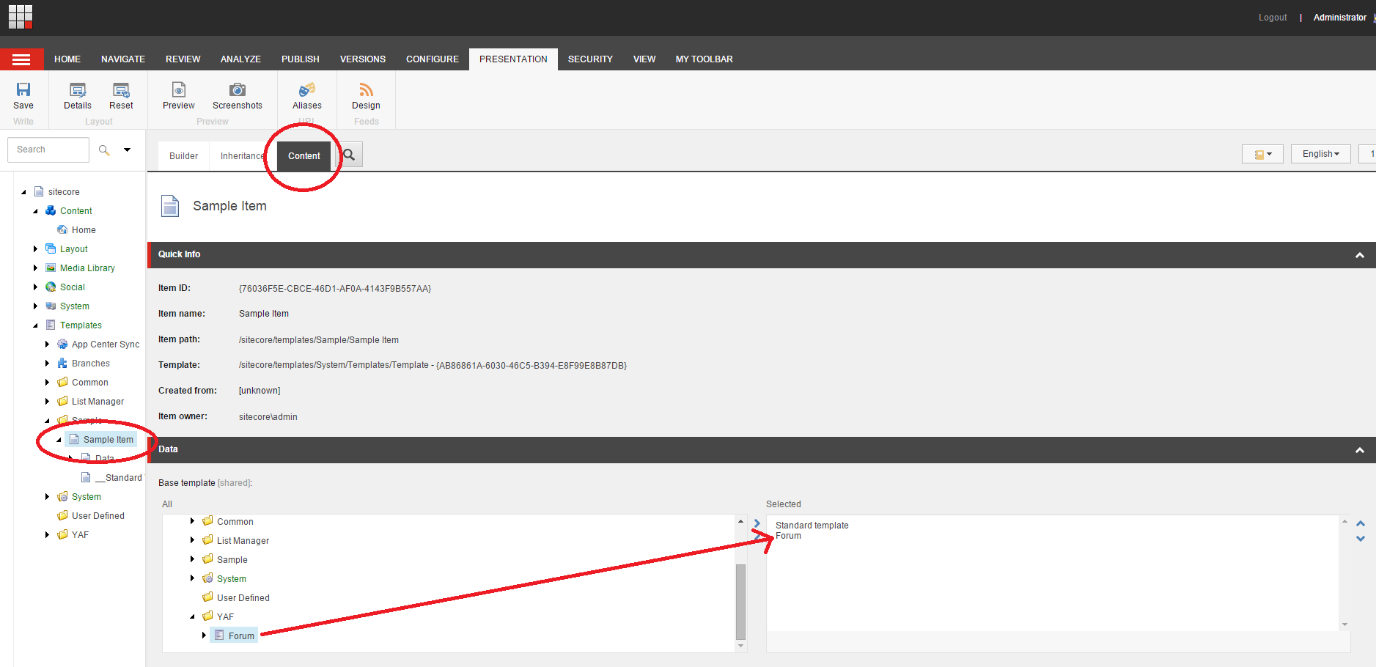


Figure Sitecore allows multiple inheritance. It's not problem to inherit from more than one template

You can now use the forum drop down on the page to select which board, group or forum that the page must display:

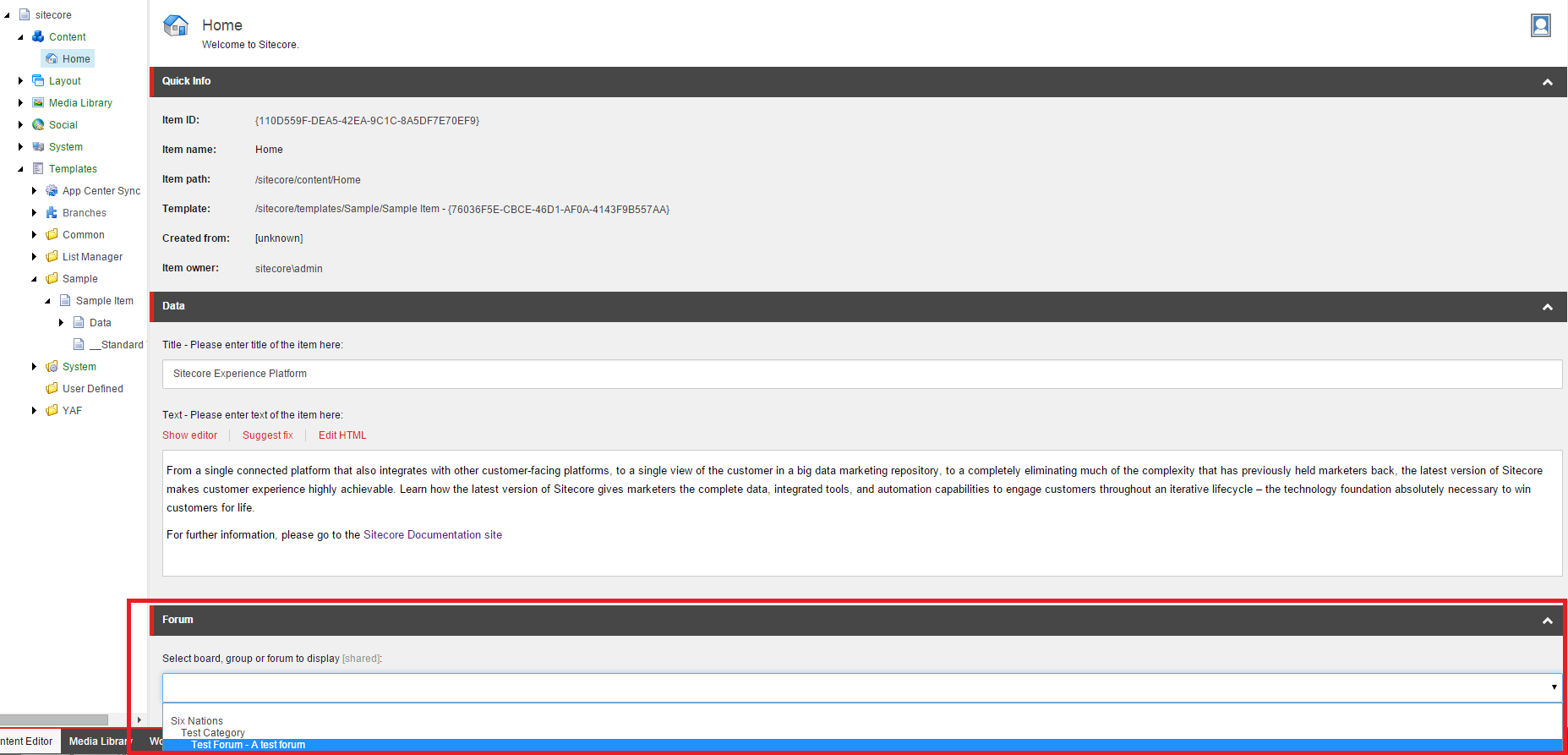


Figure Select board, group or forum to display

# Notes

Here are some key notes about the YAF integration, and what has happened:

## The YAF Security provider is disabled

YAF no longer use its own security providers. Instead it uses the Sitecore provider. YAF will read and write users and roles to the Sitecore security databases.

## YAF uses the Sitecore users from the current domain

YAF is modified to use the Sitecore users from the current domain. If your website uses the extranet domain, so will YAF.

YAF is programmed to avoid users from other domains than the current. If you to access YAF with a user from the Sitecore domain, and YAF are running in the extranet domain, YAF switches to the extranet/Anonymous user.

## YAF needs the Anonymous user

Do not remove the Anonymous user from the domain running YAF. I’m not sure that Sitecore will survive without the Anonymous user, but YAF certainly will not.

## The YAF Guest user no longer exist

It exists, but it is never used. The Guest user is a special YAF user that is used when there is no user logged in. Since Sitecore always provides a user (extranet/Anonymous when no one is logged in), YAF will never switch to the Guest mode. Your “guest” user is therefore the Anonymous user of the current domain.

YAF is programmed to treat the Anonymous user as a Guest user.

## The YAF Guests membership role cannot be used

The YAF Guests membership role is a special role that does not exist in the provider. Do not attempt to apply user to the Guests role as YAF will crash.

## Provide your own login page

YAF will never show a link to its own login page. The page is only visible when no one is logged in. As described in 9.4, Sitecore always provides a user, so no matter what you do, YAF believes that a user is logged in and therefore provides a logout link.

It is therefore necessary that you develop your own login screen.

## Users can be managed from YAF …

… But if you do so, users will be created using the Sitecore security providers with no domain name. Sitecore have no problem with this, but be aware that even if it’s possible to have 2 users with the same name in different domains, your YAF users will be overridden if the user created from YAF shares name with a user from the extranet domain.

## Roles can be managed from YAF

Since we use the same provider as Sitecore does, it means that all Sitecore roles are visible from YAF. And they can be edited. And deleted.