Hosting with .NET

Version 1.0

Document History

Author/Editor	Date	Reason for Change	Version
Andrey Karpik	01/03/2014	Draft	0.90
Andrey Karpik	03/24/2014	Changes based on the playbook review	1.0

Table of Contents

.NET hosting Overview

.NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. Programs written for .NET Framework execute in a software environment, known as the Common Language Runtime, an application virtual machine that provides services such as security, memory management, and exception handling. It also provides data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications,.NET Framework, is intended to be used by most new applications created for the Windows platform. Microsoft also produces an integrated development environment largely for .NET software called Visual Studio.

.NET hosting provided by CrownPeak means that CrownPeak is offering his clients web space on the Amazon EC2 cloud environment and is giving them the option of using ASP.NET for enhancing the out of the box CMS functionality to add server-side features for the web applications.

.NET hosting Benefits

ASP.NET dramatically reduces the amount of code needed to build server-side code for the Web applications (a framework rich with features and capabilities).

ASP.NET allows creation-rich applications with flexible business logic and rules.

Different best practices that can be easily integrated to the CMS like ASP.NET master pages that allow you to create a consistent layout for the pages on the sites. A single master page defines the look and feel and standard behavior that you want for all of the pages (or a group of pages) in your application. You can then create individual content pages that contain the content you want to display. When users request the content pages, they merge with the master page to produce output that combines the layout of the master page with the content from the content page.

Master pages can be used to manage and deploy global content. For information about master page configuration see Integrating .NET Framework in CrownPeak.

User controls enable you to re-use content and code. This is useful for widget implementations.

Built-in Windows/Forms authentication and application settings to ensure safe and secure applications through the Microsoft membership module.

ASP.NET framework is complemented by rich tools in Visual Studio integrated development environment.

The source code and HTML tied together, so that ASP.NET pages are easy to maintain and write.

All processes are carefully controlled and managed by ASP.NET, so that if the process is dead, the new process can be created in its place, which helps to keep your application constantly available to handle requests.

ASP.NET code is purely server-side technology that runs on the server before being sent to the browser. This is useful in case some content must be hidden from the client-side based on some user/database information.

The framework allows creation of custom search based solutions using .Net consuming XML content from the CMS, providing up to date search results or faceted navigation on the site.

Note: CrownPeak search is limited to four crawls a day.

Server-side .NET allows database interactions, the framework provides a lot of predefined classes that simplify database calls and integration.

.Net/IIS provides flexible rewrite module that allows easy rewrites and redirects management.

Configuring .NET hosting

CrownPeak .NET hosting can be configured as part of the CMS creation process.

Create a ticket for IT and send it to *help@crownpeak.com* providing the following information:

- Instance Name: Name of the CMS instance
- Hosting:.NET for both Stage and Live
- .NET version: Default .NET version currently is 4.0 (you can request the latest version or specific version based on the Client's IT requirements)
- Set index.aspx page as default page for the site

IT creates the IIS Web application and sets up the .NET web application when the approval process is complete.

The CMS configuration properties will be adjusted to allow publishing to the newly created application folder via sFTP.

The Assets' filenames should be configured to use .aspx extension for all pages

Usually for the .NET hosted sites we also recommend setting up the default file extension = aspx in the CMS filenames settings.

To request .NET version upgrade on the web server, send a request to *help@crownpeak.com.*

Integrating .NET Framework in CrownPeak

CrownPeak provides optional developers training as a part of Enablement Services.

Masterpage template can be create in the CMS to leverage this powerful .NET feature

Developers can use Visual Studio integrated development environment to create server-side application code. CrownPeak best practice is to compile the server-side code to a DLL file(s) and publish it out from the CMS to the /bin/folder on the web hosted site.

The server-side code can also be included inline to the templates output files if needed.

Capability	.NET Web Forms
Presentation	Web pages deployed as files with an .Aspx extension.
Server-side Logic	• Code behind developed as supporting files with the same name as the pages they support with a .cs extension (for C#), and are deployed in a complied DLL. • DLL is a preferred way of deploying the server-side logic. It allows to use Visual Studio and local environment to create/debug/test the server-side logic prior to upload to the server, and is much easier from the CMS support perspective. • DLLs can be published as binary asset or as part of the Asset Wrapper upload.

Shared Global Content (Header/Footer)	Masterpages can be deployed containing all global content (header, footer), and deployed with a .master extension. The masterpages are referenced in the .aspx pages.
	Recommended master page location is in the root of the site or in dedicated/masterpage/folder.
	All other content pages must have a reference to the master page, for example:
	`if (context.IsPublishing)`
	`{`
	`Out.WriteLine(String.Format("<$@ Page Title=\"{0}\" Language=\"C#\" MasterPageFile=\"~{1}\" AutoEventWireup=\"true\" CodeBehind=\"index.aspx.cs\" Inherits=\"MySiteNamespace.MyClass\" $>", asset["html_title"], master.GetLink()));`
	`}`
	See *http://connect.crownpeak.com/documents/best-practices/dotnetbestpracticesmasterpageset* for details of masterpages implementation

Includes	User controls provide include capabilities, and are files deployed as pages with an .ascx exntesion, and referenced by any .aspx page.

Capability	.NET Web Forms

In-line code	C# code can be also inserted in-line in any aspx page.
	For example:
	`<script runat="server" language="C#">`
	`public static Control FindControlRecursive(Control Root, string Id)`
	`{`
	`if (Root.ID == Id)`
	`return Root;`
	`foreach (Control Ctl in Root.Controls)`
	`{`
	`Control FoundCtl = FindControlRecursive(Ctl, Id);`
	`if (FoundCtl != null)`
	`return FoundCtl;`
	`}`
	`return null;`
	`}`
	`</script>`

Output Template	Web Pages containing html, css, js, and any server-side C# code are implemented in output templates.
Input Template	Content placeholders are exposed as editable input form fields.
Folder structure	In order to simplify the CMS support, we recommend the following folder structure: /Site Root Folder/ /_Global Assets/ /bin/ (configure publishing to put the folder content to the /bin/ folder on the web server) Site.dll (individual DLL assets) web.config (should be published to the root on the web server)
Deployment	Files are deployed with the content baked-in with a .aspx extension.
Shared Global Content (Header/Footer)	Navigation Wrappers are configured to publish as master pages with a .master extension.
Includes	Widgets and other assets can be configured to be published as user controls with a .ascx extension, and referenced by other templates using name.aspx and url.aspx.

Capability	.NET Web Forms
Server-side code	Compiled and deployed as dll.

Preview in the CMS	CMS is not able to run the compiled server-side code. The template should have some output code wrapped to *context.IsPublishing* CMS API to duplicate some server-side features in the preview:
	`if (context.IsPublishing)`
	`{`
	`Out.WriteLine("any server side controls/user controls or references goes here");`
	`}`
	`else`
	`{`
	`Out.WriteLine("HTML for the CMS preview ");`
	`}`
	Another option is to use *preview.aspx* template file to produce the preview HTML

Notes

More information can be found in *http://connect.crownpeak.com/documents/best-practices/*

Playbook Notes

http://www.microsoft.com/net

Masterpages implementation - *http://connect.crownpeak.com/documents/best-practices/dotnetbestpracticesmasterpageset*