

---

# WEBppliance™ 3.0 for Windows® .NET Guide

August 02, 2002

This guide covers the procedure for enabling and using Microsoft® .NET on WEBppliance™ 3.0.1 for Windows® for Appliance and Site Administrators.

**Important:** To enable the .NET Framework on WEBppliance 3.0 for Windows, you must apply the WEBppliance 3.0.1 patch release. For more information about installing the 3.0.1 patch, see *Ensim WEBppliance 3.0.1 for Windows Patch Release Notes*.

Microsoft defines .NET as a set of Microsoft software technologies for connecting information, people, systems, and devices. It enables an unprecedented level of software integration through the use of XML Web services: small, discrete, building-block applications that connect to each other—as well as to other, larger applications—via the Internet.

The Microsoft .NET Framework is an environment for building, deploying, and running XML Web services and other applications. It consists of three main parts: the common language runtime, the Framework classes, and ASP.NET.

**Important:** It is recommended that .NET be used only in dedicated hosting.

The WEBppliance 3.0.1 for Windows Patch installs the .NET Framework (version 1.0.3705) necessary to run .NET applications, however, this is disabled by default. For information about enabling .NET on WEBppliance 3.0.1, see [Enabling .NET on WEBppliance 3.0.1 for Windows](#).

It is recommended to use virtual directories to deploy .NET applications. Virtual directories can be created through the Site Administrator interface of WEBppliance 3.0.1. For more information about creating virtual directories, see [Creating and using Virtual Directories](#).

## **WEBppliance 3.0.1 for Windows and .NET**

The .NET concept encompasses a wide range of software technologies. WEBppliance 3.0.1 installs only the Framework necessary to run .NET applications on WEBppliance sites. Note the following points about .NET and WEBppliance 3.0.1 for Windows.

- By default, the .NET Framework is disabled on WEBppliance. For information about enabling .NET on WEBppliance 3.0.1, see [Enabling .NET on WEBppliance 3.0.1 for Windows](#).
- WEBppliance does not provide a interface to enable or disable .NET for sites. This has to be done using the command line utility provided by Ensim or Microsoft.
- .NET is not treated as a service and, hence, will not be listed in the Administrator GUI.
- Site Administrators have to create virtual directories using the option given in the WEBppliance interface and upload the necessary files or folders using a FTP client, the WEBppliance File Manager, or Microsoft FrontPage®. For information about creating and using virtual directories through WEBppliance, see [Creating and using Virtual Directories](#).
- Any .NET application cannot be deployed on WEBppliance. The application must be properly configured before deployment. This limitation is generic to .NET and not specific to WEBppliance. For more information, see the subsection [For the Site Administrator](#) under the section [Using .NET with WEBppliance 3.0.1](#).
- Existing ASP pages will continue to work and not be affected in any way by the installation of the .NET Framework.
- There is no shortcut to convert ASP pages to ASP.NET and just changing the .asp extension to .aspx will not upgrade existing ASP applications to ASP.NET. For more information, see <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/convertasptoaspnet.asp>.

## Enabling .NET on WEBppliance 3.0.1 for Windows

**Important:** To enable the .NET Framework on WEBppliance 3.0 for Windows, you must apply the WEBppliance 3.0.1 patch release. For more information about installing the 3.0.1 patch, see *Ensim WEBppliance 3.0.1 for Windows Patch Release Notes*.

The WEBppliance 3.0.1 Patch installs the .NET Framework, however, it is disabled by default. You can enable the .NET Framework in one of the two ways as follows:

### Procedure 1 – Using the Ensim utility

- 1 Connect to the WEBppliance server through the console or terminal services and log on as Administrator.
- 2 At the command prompt, enter the following commands.

```
cd %OCW_HOME%  
cscript enabledotnet.vbs ALL
```

This command enables .NET for all existing sites and new sites created using WEBppliance. To enable .NET for only a single site, enter the following command.

```
cscript enabledotnet.vbs <site_name>
```

where <site\_name> is the name of the site for which .NET is to be enabled.

**Note:** Typing the above command without any parameters displays the command usage.

To disable .NET for all sites, use the following command.

```
csript disabledotnet.vbs ALL
```

To disable .NET for a single site, use the following command.

```
csript disabledotnet.vbs <site_name>
```

where, <site\_name> is the name of the site.

## Procedure 2 – Using Microsoft utilities

- 1 Connect to the WEBppliance server through the console or terminal services and log on as Administrator.
- 2 At the command prompt, change to the directory `C:\WINNT\Microsoft.NET\Framework\v1.0.3705` (assuming WEBppliance is installed in the default location), and enter the following commands.

```
aspnet_regiis.exe -s w3svc/<n>/ROOT
```

This command enables .NET mappings for the Web site whose ID is <n>.

```
aspnet_regiis.exe -c
```

This command copies the client side script files to all existing sites.

To see more information about this utility, print the usage for the command, by typing the following:

```
aspnet_regiis.exe
```

To remove .NET mappings from the Web site whose ID is <n>, enter the following command.

```
aspnet_regiis.exe -k w3svc/<n>/ROOT
```

**Note:** The ID for a Web site created by WEBppliance is located under the key `HKLM\software\Ensim\WEBppliance\domains\<site_name>\services\w3svc\id` in the Windows registry, where <site\_name> is the name of the site.

**Important:** For sites created after the WEBppliance 3.0.1 Patch installation and deploying .NET applications, the following error may be displayed for certain applications, for example, a Windows Form-based application.

```
Unable to find script library '  
aspnet_client/system_web/1_3705_0/  
webuivalidation.js'
```

If this error occurs, run the following command.

```
aspnet_regiis.exe -c
```

This command will have to be executed for each new site, where the above error occurs.

## Using .NET with WEBpliance 3.0.1

### For the Appliance Administrator

The Appliance Administrator does not have to perform any specific tasks to deploy .NET applications, other than enabling the .NET Framework on WEBpliance. However, if necessary, the Appliance Administrator can do the following tasks by logging on to the WEBpliance server as Administrator, either through the console or using terminal services.

- Editing properties of a virtual directory using Internet Services Manager.
- Enabling 'tracing'. Tracing is disabled by default in the file machine.config for security reasons. To enable tracing, edit the file **machine.config** and locate **trace.axd**. The http handler for trace.axd is changed by WEBpliance. Use the original configuration that has been commented out by WEBpliance and available just above this line.
- Performing advanced administrative tasks such as adding classes to the global assembly cache (GAC). If two .NET applications need to share common class files, they must be available in the GAC. This procedure is technically very involved; for more information, visit <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndotnet/html/faq111700.asp>.

## **For the Site Administrator**

Site Administrators can download sample .NET applications from the Internet. However, keep in mind that no application downloaded from the Internet will run as-is on a site hosted on WEBpliance. While developing .NET applications for sites hosted on WEBpliance, the Site Administrator needs to be aware of the following:

- When you develop .NET applications on a local computer using Microsoft or other tools, some folders and virtual directories are created on the computer automatically. These tools also create a configuration file, usually, web.config. Thus, any generic .NET application can be broadly classified as consisting of the following:
  - The file web.config – This is an optional file and need not be deployed unless you want to change the configuration.
  - IIS Virtual Directories – You can create these using the new option provided in the WEBpliance Site Administrator interface.
  - Physical files and folders – You can upload this to the WEBpliance server, using a FTP client, WEBpliance File Manager, or Microsoft FrontPage.
- You can test the working of a .NET application by deploying a simple “Hello World” application. However, you may have to modify sample applications downloaded from the Internet, since these may contain references to the local development environment, such as the server or host name, rather than the environment where the application is to be deployed.
- Do not use localhost during development if your site is hosted in a shared hosting environment. In this case, the localhost will map to the primary IP of the server, which points to the default Web site.
- IIS virtual directories are not an absolute must to host .NET applications, these can be hosted in physical directories as well. However, this is not recommended, because this does not allow complete isolation between two applications hosted on the same site. For example, you cannot have two separate config (web.config) files for these two applications. Hence, we recommend you use virtual directories to host .NET applications.

## Creating and using Virtual Directories

A virtual directory is a directory that is not contained in `\inetpub\wwwroot`, but appears to client browsers as though it were. A virtual directory points to a directory in the file system of the site.

For example, to publish content stored in the directory `\salesdata\customers`, create a virtual directory, say `cust`, and set its home directory to `\salesdata\customers`. The content in the `\salesdata\customers` directory can then be accessed by client browsers using the URL `http://<your-site>/cust`, where `<your-site>` is the name of a site.

To create a virtual directory follow these steps.

- 1 Log on to WEBpliance as the Site Administrator.
- 2 Click **Services** in the left navigation bar.
- 3 In the Service List, locate IIS Web Server, and click the **Configure** icon  in the **Actions** column.
- 4 In the Site Web Publishing Service Manager page, click the **Virtual Directory** option in the menu.

The page displays two forms. The top form displays the current path and list of virtual directories under it. The second form is Add Virtual Directory, where you specify the name of the virtual directory and its home directory.

- 5 In the **Name** field, enter the name of the virtual directory. This directory will be created under the current directory.
- 6 In the **Home Directory** field, enter the name of the physical directory, with the complete path, to whom this virtual directory points, for example, `\salesdata\customers`.

Physical directories can be created using the WEBppliance Windows File Manager in the Site Administrator interface. To access the File Manager, click **Files** in the left navigation bar.

**Note:** You can modify the home directory of an existing virtual directory by clicking on the **Configure** icon in **Actions** column of the virtual directory list.